

计算机网络原理

第6章：应用层

<https://internet.hactcm.edu.cn>

河南中医药大学信息技术学院互联网技术教学团队
河南中医药大学医疗健康信息工程技术研究所

2024.2

OSI 的七层协议体系结构



TCP/IP 的四层协议体系结构



五层协议的体系结构



本章教学计划

- ✓ 域名系统 DNS
- ✓ 文件传送协议 FTP...
- ✓ 远程终端协议 TELNET...
- ✓ 万维网 WWW
- ✓ 电子邮件 SMTP...

应用协议

- ✓ 动态主机配置协议 DHCP
- ✓ 简单网络管理协议 SNMP
- ✓ 应用进程跨越网络的通信

网络管理

本章教学计划

- 本章讨论通信服务是如何提供给应用进程来使用的。
 - 也就是说，讨论各种应用进程通过什么样的应用层协议来使用网络所提供的通信服务。
- 不同的网络应用的应用进程之间，需要有不同的通信规则。
 - 因此在运输层协议之上，需要有应用层协议（Application Layer Protocol）。
- 应用层的具体内容就是精确定义这些通信规则。

本章教学计划

- 应用层协议需要定义的内容有：
 - 应用进程交换的报文类型，如请求报文和响应报文。
 - 各种报文类型的语法，如报文中的各个字段及其详细描述。
 - 字段的语义，即包含在字段中的信息的含义。
 - 进程何时、如何发送报文，以及对报文进行响应的规则。

- 应用层的许多协议都是基于**客户-服务器**方式。
 - 客户是服务请求方。
 - 服务器是服务提供方。

1. 域名系统 DNS

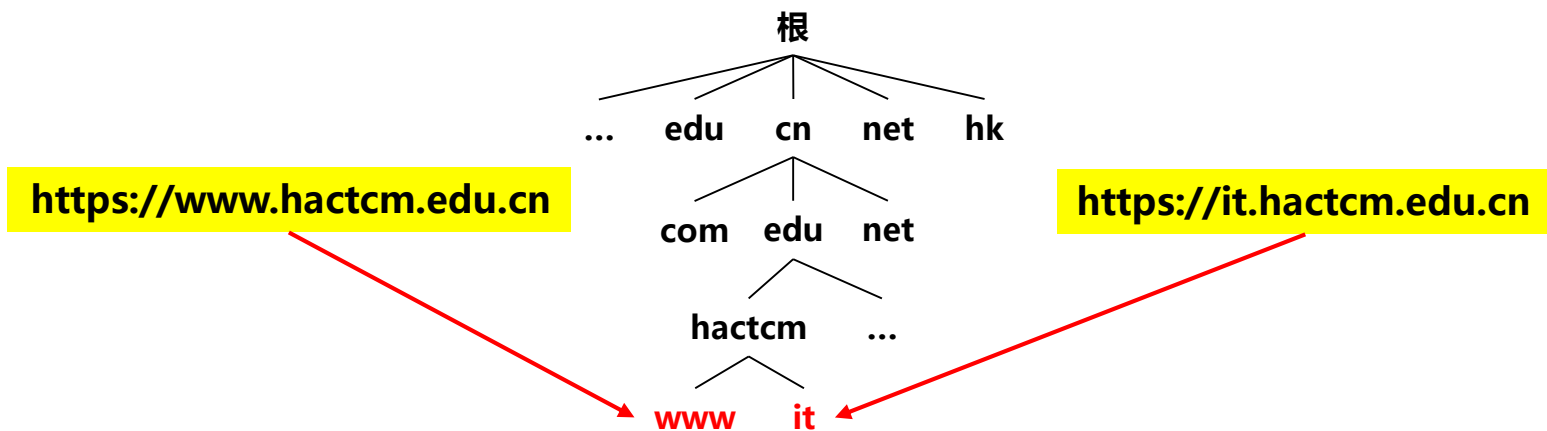
1.1 域名系统概述

- 域名系统 DNS (Domain Name System)
 - 互联网使用的命名系统。
 - 用来把人们使用的机器名字 (域名) 转换为 IP 地址。
 - 为互联网的各种网络应用提供了核心服务。
 - 应用层软件经常直接使用域名系统 DNS (Domain Name System), 但计算机的用户只是间接而不是直接使用域名系统。

1. 域名系统 DNS

1.1 域名系统概述

- 域名系统 DNS (Domain Name System)
 - 域名采用层次树状结构的命名方法：www.myschool.edu.cn。
 - DNS 是一个联机分布式数据库系统，采用客户服务器方式。
 - 域名到 IP 地址的解析是由若干个域名服务器程序共同完成。
 - 域名服务器程序在专设的结点上运行，运行该程序的机器称为域名服务器。



1. 域名系统 DNS

1.1 域名系统概述

- 为什么要使用域名系统 DNS 呢？
 - 其根本原因在于：IP地址不足。
 - 不便记忆：
 - 十进制的 IPv4 地址仍然比较长，远没有以字符串命名的DNS名称好记。
 - 不便地址变更：每次更改服务器地址都要更改IP地址的话，是很难做到的。
 - 云计算时代下的业务发布需要经常变更。
 - 高负载下的 CDN 分发。
 - 不安全：
 - IP 地址一旦对外公布的话，很容易受到攻击。

1. 域名系统 DNS

1.1 域名系统概述

- 名字系统的实现方式有哪些？
 - 在 ARPANET 时代，使用 hosts 文件，列出所有主机名字和相应的IP地址。
 - 随着计算机数量的增多，采用域名系统 DNS。
- 域名系统要解决的两个关键问题？
 - 性能：抛弃整个互联网使用一个域名服务器的集中式解决方案，采用分布式的域名系统 DNS。（RFC 1034, 1035）
 - 冲突：采用层次树状结构的命名方法，确保不存在相同的域名，杜绝了名字冲突。

1. 域名系统 DNS

1.1 域名系统概述

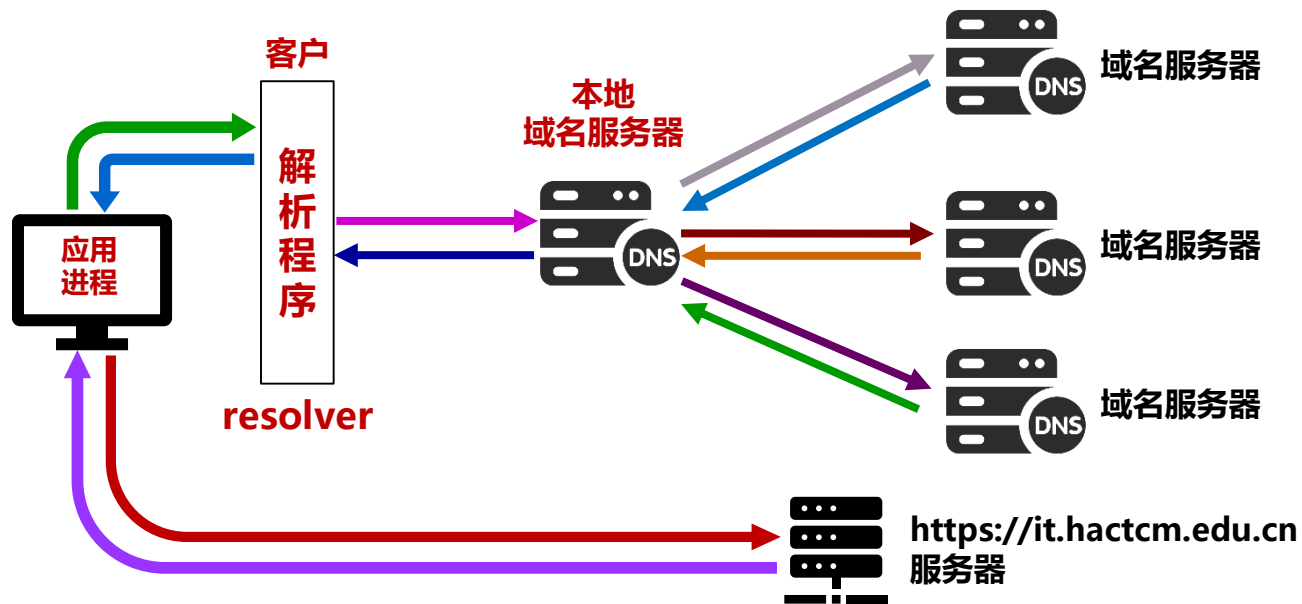
□ 域名到 IP 地址的解析过程：

- 当某一个应用进程需要把主机名解析为IP地址时，该应用进程就调用解析程序（resolver）。
- 作为 DNS 的一个客户，把待解析的域名放在 DNS 请求报文中，以 UDP 用户数据报方式发给本地域名服务器。
- 本地域名服务器在查找域名后，把对应的 IP 地址放回到回答报文中返回。
- 应用进程获得目的主机的 IP 地址后即可进行通信。
- 如果本地域名服务器不能够回答该请求，则本地域名服务器就暂时成为 DNS 中的一个客户机，向其他域名服务器发出查询请求。

1. 域名系统 DNS

1.1 域名系统概述

DNS 使用 UDP 通信



1. 域名系统 DNS

1.2 互联网的域名结构

- 互联网采用层次树状结构的命名方法。
- 任何一个连接在互联网的主机或路由器，都有一个唯一的层次结构的名字，即域名 (domain name)。
- 域 (domain):
 - 是名字空间中一个可被管理的划分。
 - 可以划分为子域，而子域还可继续划分为子域的子域，这样就形成了顶级域、二级域、三级域，等等。

... 三级域名. 二级域名. 顶级域名

1. 域名系统 DNS

1.2 互联网的域名结构

- 域名结构：层次结构。
 - 由标号 (label) 序列组成，各标号之间用点 (.) 隔开，各标号分别代表不同级别域名。
 - 域名中的标号都由英文字母和数字组成。
 - 每一个标号不超过63个字符。
 - 标号不区分大小写。
 - 多个标号组成的完整域名总共不超过255个字符。
- DNS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含义。
- 各级域名有其上一级的域名管理机构管理，而最高的顶级域名则有 ICANN 进行管理。

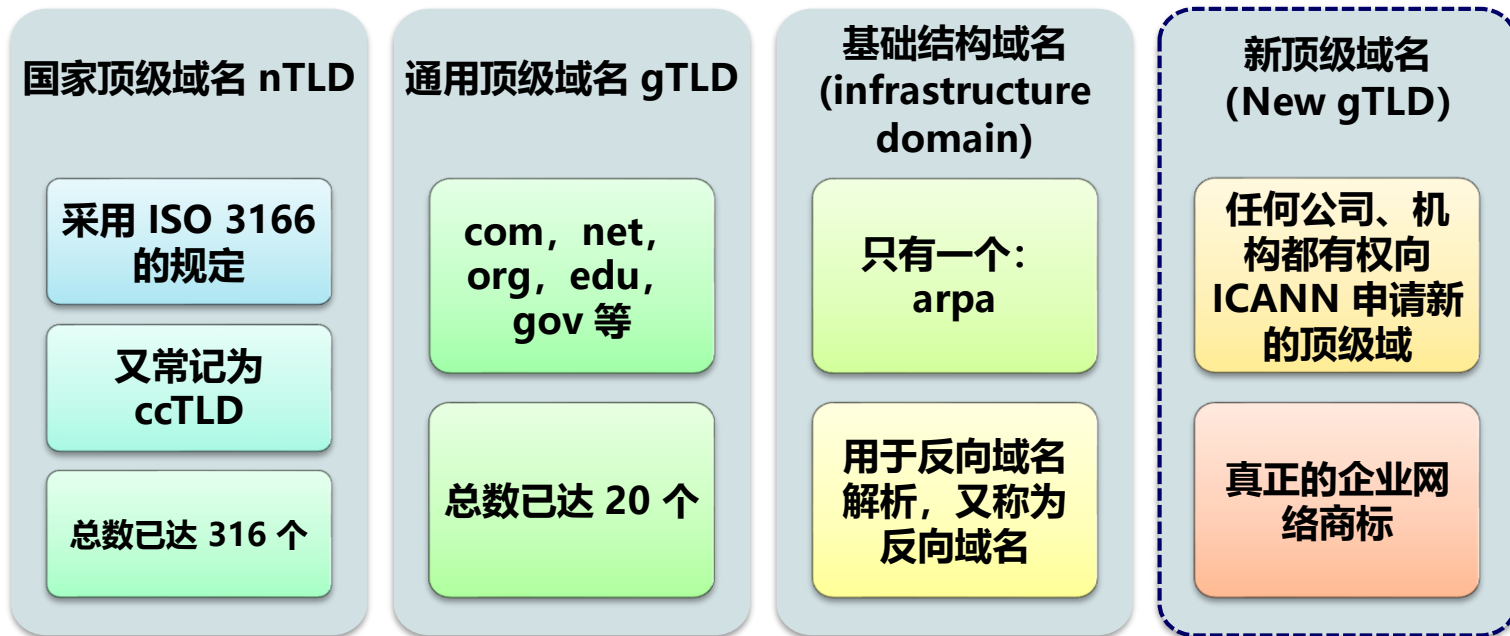
1. 域名系统 DNS

1.2 互联网的域名结构

- 域名只是个逻辑概念，并不代表计算机所在的物理地点。
 - 变长的域名和使用有助记忆的字符串，是为了便于人来使用。
 - IP 地址是定长的 32 位二进制数字，非常便于机器进行处理。
- 域名中的“点”和点分十进制 IP 地址中的“点”并无一一对应的关系。
 - 点分十进制 IP 地址中一定是包含三个“点”，但每一个域名中“点”数目则不一定是三个。

1. 域名系统 DNS

1.2 互联网的域名结构



在国家顶级域名下注册的二级域名均由该国家自行确定。
我国把二级域名划分为“**类别域名**”和“**行政区域名**”两大类。

1. 域名系统 DNS

1.2 互联网的域名结构

- 顶级域名 TLD (Top Level Domain)
 - 国家顶级域名 nTLD: 如: .cn 表示中国, .us 表示美国, .uk 表示英国, 等等。
 - 通用顶级域名 gTLD: 最早的顶级域名是:
 - .com (公司和企业)
 - .net (网络服务机构)
 - .org (非赢利性组织)
 - .int (国际组织)
 - .edu (美国的教育机构)
 - .gov (美国的政府部门)
 - .mil (美国的军事部门)
 - 基础结构域名 (infrastructure domain): 这种顶级域名只有一个, 即 arpa, 用于反向域名解析, 因此又称为反向域名。

1. 域名系统 DNS

1.2 互联网的域名结构

□ 通用顶级域名 gTLD (陆续增加的13个通用顶级域名)

- .aero (航空运输企业)
- .aisa (亚太地区)
- .biz (公司和企业)
- .cat (加泰隆人的语言和文化团体)
- .coop (合作团体)
- .info (各种情况)
- .jobs (人力资源管理者)
- .mobi (移动产品与服务的用户和提供者)
- .museum (博物馆)
- .name (个人)
- .pro (有证书的专业人员)
- .tel (Telnic股份有限公司)
- .travel (旅游业)




ICANN76 Look Back

Missed ICANN76 and want to catch up on the sessions you didn't attend? Visit the ICANN76 meeting site to view transcripts, watch video recordings, listen to audio recordings, and more.

[Visit the ICANN76 Website](#)



Latest ICANN News



<https://icann.org/ceosearch>

ICANN Begins Search for Next President and CEO

The ICANN Board of Directors has officially begun the process to search for and select ICANN's next President and Chief



Blog: Recap of the March Board Workshop and ICANN76

During the public Board Meeting at ICANN76, the ICANN Board adopted 98 recommendations contained in the New Generic Top-Level



ICANN 2023 NomCom Extends Deadline to 30 March

The ICANN 2023 Nominating Committee (NomCom) announced today that it has extended the deadline for ICANN leadership



ICANN Board to Begin Preparations for New gTLDs

During its meeting at the ICANN76 Community Forum, the ICANN Board of Directors adopted 98 recommendations contained

A note about our privacy policies and terms of service:

We have updated our privacy policies and certain website terms of service to provide greater transparency, promote simplification, and align with recent changes in privacy laws applicable to us. [Learn more.](#)

This site uses cookies to deliver an efficient user experience and to help us see how the site is used. [Learn more.](#)

https://www.iana.org/domains/root/db

Root Zone Database

The Root Zone Database represents the delegation details of top-level domains, including gTLDs such as .com, and country-code TLDs such as .uk. As the manager of the DNS root zone, we are responsible for coordinating these delegations in accordance with our [policies and procedures](#).

Much of this data is also available via the WHOIS protocol at whois.iana.org.

- Domain Names
- Overview
- Root Zone Management**
- Overview
- Root Database**
- Hint and Zone Files
- Change Requests
- Instructions & Guides
- Root Servers
- .INT Registry
- .ARPA Registry
- IDN Practices Repository
- Root Key Signing Key (DNSSEC)
- Reserved Domains

DOMAIN	TYPE	TLD MANAGER
.aaa	generic	American Automobile Association, Inc.
.aarp	generic	AARP
.abarth	generic	Fiat Chrysler Automobiles N.V.
.abb	generic	ABB Ltd
.abbott	generic	Abbott Laboratories, Inc.
.abbvie	generic	AbbVie Inc.
.abc	generic	Disney Enterprises, Inc.
.able	generic	Able Inc.
.abogado	generic	Registry Services, LLC
.abudhabi	generic	Abu Dhabi Systems and Information Centre
.ac	country-code	Internet Computer Bureau Limited
.academy	generic	Binky Moon, LLC
.accenture	generic	Accenture plc
.accountant	generic	dot Accountant Limited
.accountants	generic	Binky Moon, LLC
.aco	generic	ACO Severin Ahlmann GmbH & Co. KG
.active	generic	Not assigned
.actor	generic	Dog Beach, LLC
.ad	country-code	Andorra Telecom
.adac	generic	Not assigned
.ads	generic	Charleston Road Registry Inc.
.adult	generic	ICM Registry AD LLC
.ae	country-code	Telecommunications and Digital Government Regulatory Authority (TDRA)
.aeg	generic	Aktiebolaget Electrolux
.aero	sponsored	Societe Internationale de Telecommunications Aeronautique (SITA INC USA)
.aetna	generic	Aetna Life Insurance Company
.af	country-code	Ministry of Communications and IT
.afamilycompany	generic	Not assigned

.aip	generic	VeriSign Sarl
.aiq	generic	RISE VICTORY LIMITED
.aiz	generic	Sina Corporation
.aj	test	Not assigned
.ajl	generic	Temasek Holdings (Private) Limited
.ajman	generic	Amazon Registry Services, Inc.
.ajp	generic	Public Interest Registry
.ak	generic	VeriSign Sarl
.akn	generic	Amazon Registry Services, Inc.
.akp	generic	Amazon Registry Services, Inc.
.akq	generic	SAMSUNG SDS CO., LTD
.aks	country-code	Singapore Network Information Centre (SGNIC) Pte Ltd
.akt	generic	Internet DotTrademark Organisation Limited
.akz	generic	Binky Moon, LLC
.al	generic	Zodiac Aquarius Limited
.ala	generic	The Foundation for Network Initiatives "The Smart Internet"
.ald	country-code	Macedonian Academic Research Network Skopje
.ale	test	Not assigned
.alf	country-code	EURid vzw
.alg	generic	Amazon Registry Services, Inc.
.alq	generic	Guangzhou YU Wei Information and Technology Co.,Ltd
.alr	generic	Not assigned
.als	generic	Amazon Registry Services, Inc.
.alt	generic	VeriSign Sarl
.alw	generic	TLD REGISTRY LIMITED
.alx	generic	CITIC Group Corporation
.aly	country-code	China Internet Network Information Center (CNNIC)
.alz	country-code	China Internet Network Information Center (CNNIC)
.am	generic	Binky Moon, LLC
.ama	generic	Charleston Road Registry Inc.
.amb	country-code	National Internet Exchange of India

Delegation Record for 微博

(Generic top-level domain)

Sponsoring Organisation

Sina Corporation
20F Ideal Haza No. 59 Bei Si Huan Xi Road
Haidian District
Beijing 100080
China

Administrative Contact

Director, Legal Dept.
Sina Corporation
20F Ideal Haza No. 59 Bei Si Huan Xi Road
Haidian District
Beijing 100080
China
Email: register@staff.sina.com.cn
Voice: +86-10-82628888
Fax: +86-10-82607527

Technical Contact

Senior Director, DNS Infrastructure Group
Affilias
300 Welsh Road
Building 3, Suite 105
Horsesham PA 19044
United States of America (the)
Email: tld-tech-poc@affilias.info
Voice: +1 215.706.5700
Fax: +1 215.706.5701

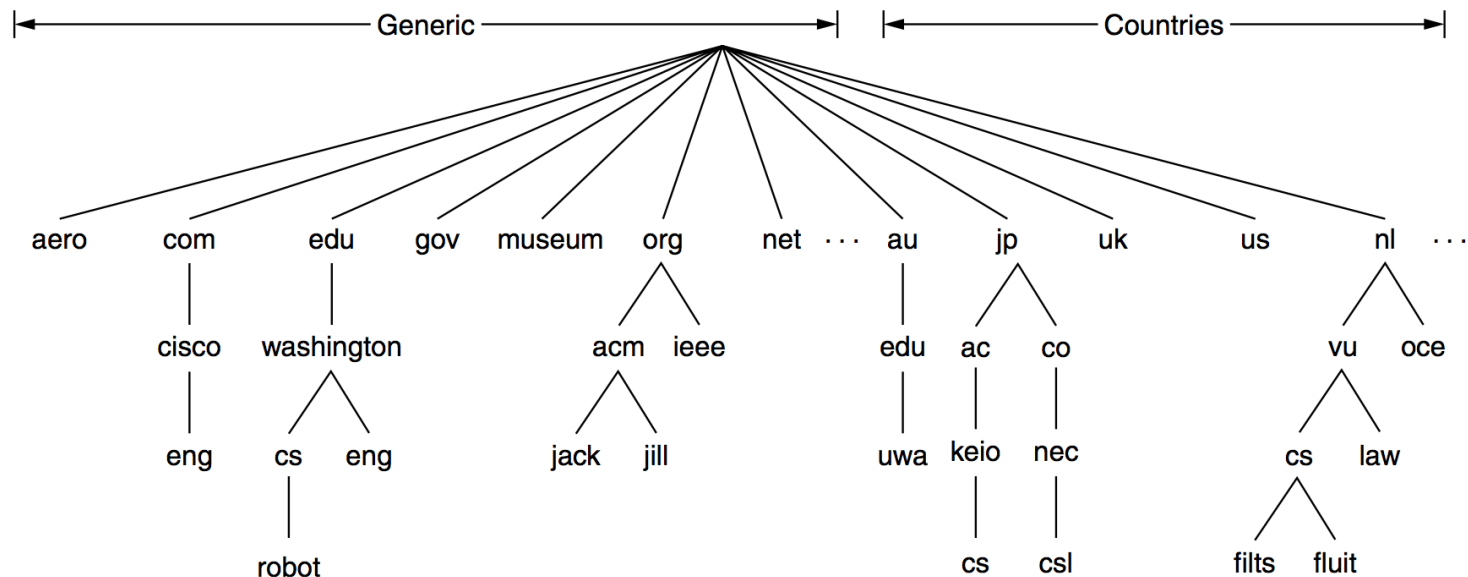
Name Servers

HOST NAME	IP ADDRESSES
a0.nic.xn--9kr700a	65.22.112.1 2401:8840:6e:0:0:0:1
a2.nic.xn--9kr700a	65.22.115.1 2401:8840:71:0:0:0:1
b0.nic.xn--9kr700a	65.22.113.1

1. 域名系统 DNS

1.2 互联网的域名结构

□ 互联网的域名空间



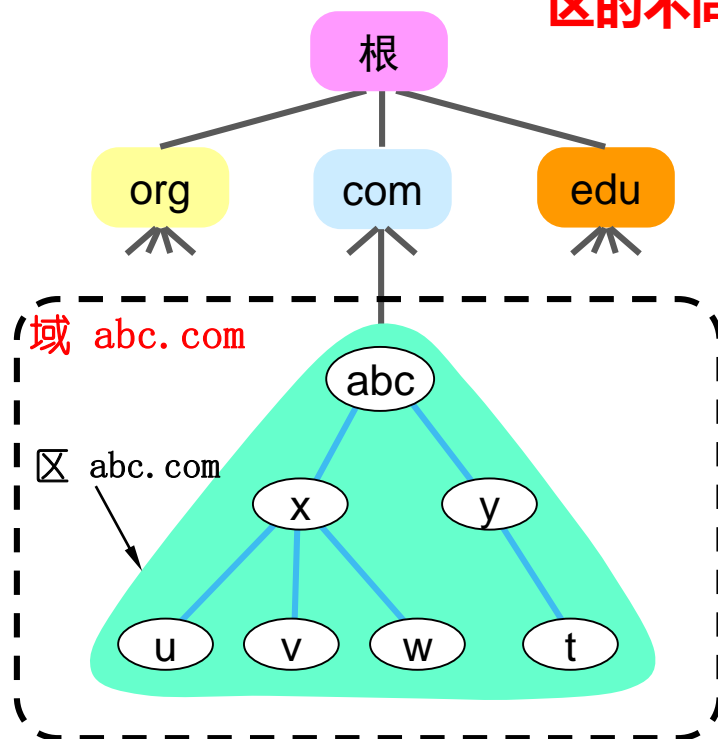
1. 域名系统 DNS

- 域名系统由分布在各地的域名服务器（DNS 服务器）实现。
- 一个服务器所负责管辖的（或有权限的）范围叫做区（zone）。
 - 各单位根据具体情况来划分自己管辖范围的区。
 - 但在一个区中的所有节点必须是能够连通的。
 - 每一个区设置相应的权限域名服务器，用来保存该区中的所有主机的域名到 IP 地址的映射。
- DNS 服务器的管辖范围不是以“域”为单位，而是以“区”为单位。

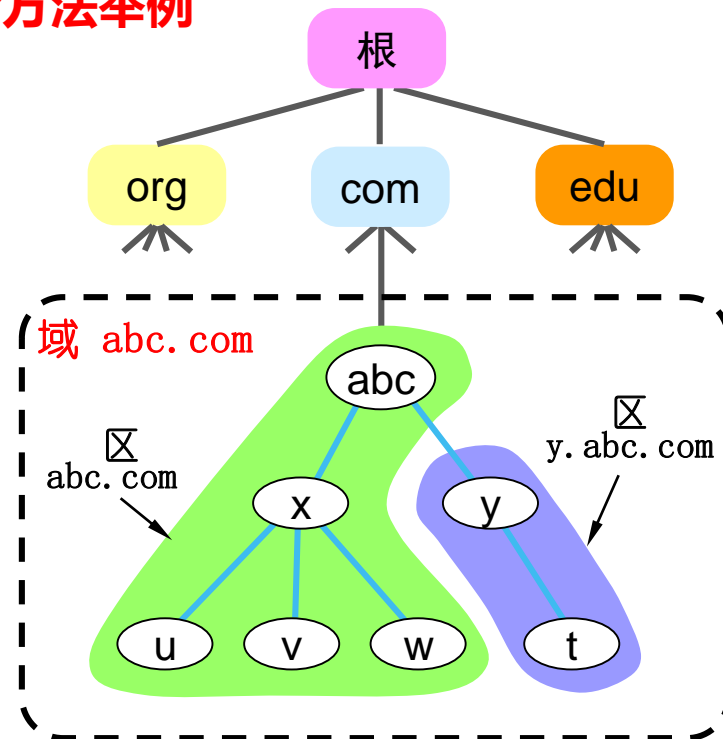
1. 域名系统 DNS

1.3 域名服务器

区的不同划分方法举例



(a) ☒ = 域

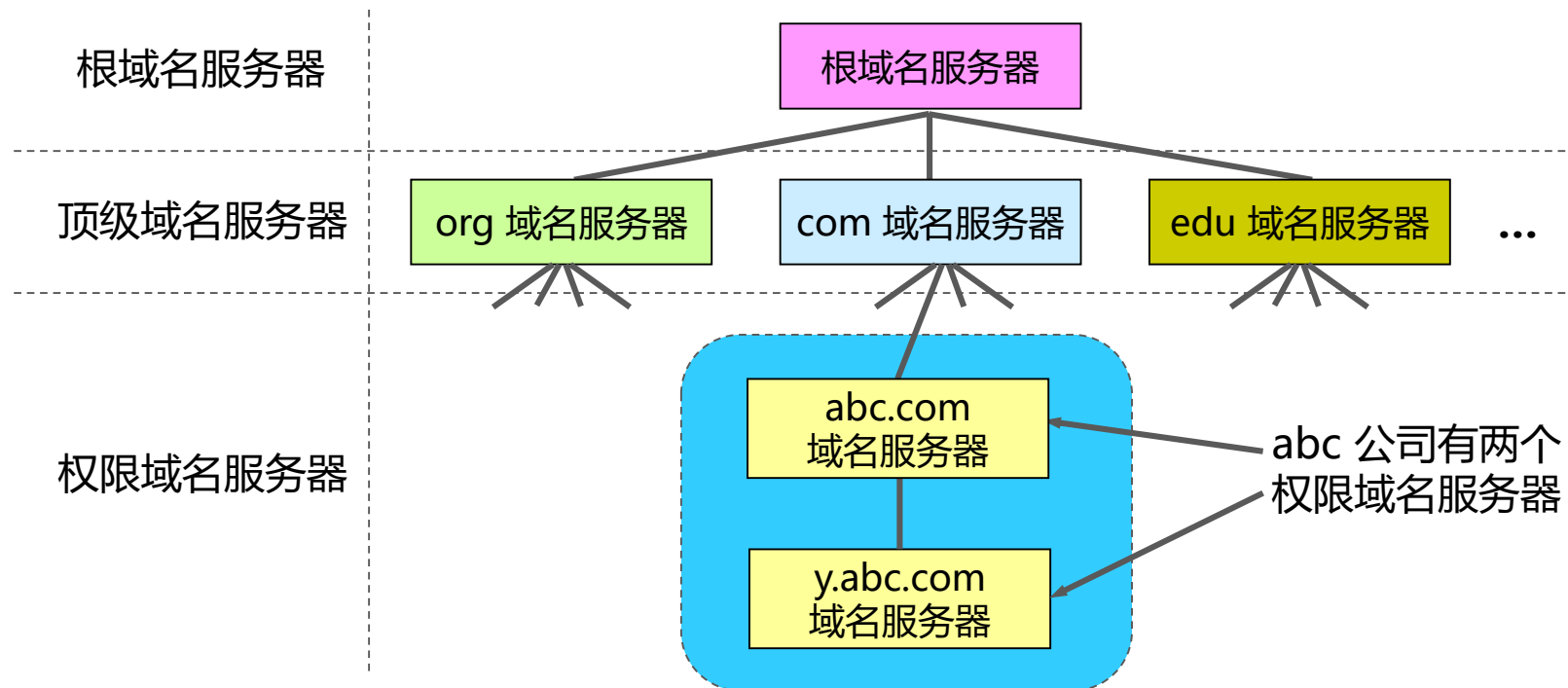


(b) ☒ < 域

1. 域名系统 DNS

1.3 域名服务器

树状结构的 DNS 域名服务器



1. 域名系统 DNS

1.3 域名服务器

1

1. 根域名服务器
2. 顶级域名服务器 (TLD服务器)
3. 权限域名服务器

2

4. 本地域名服务器 (Local Name Server)

1. 域名系统 DNS

□ 根域名服务器

- 最高层次，最为重要。
- 所有根域名服务器都知道所有的顶级域名服务器的域名和 IP 地址。
- 不管是哪一个本地域名服务器，若要对互联网上任何一个域名进行解析，只要自己无法解析，就首先求助于根域名服务器。
- 若所有的根域名服务器都瘫痪了，整个互联网中的 DNS 系统就无法工作了。

□ 注意：

- 根域名服务器并不直接把域名转换成 IP 地址（根域名服务器也没有存放这种信息），而是告诉本地域名服务器下一步应当找哪一个顶级域名服务器进行查询。

1. 域名系统 DNS

1.3 域名服务器

□ 根域名服务器

- 根域名服务器共有 13 套装置，构成 13 组根域名服务器。
- 根域名服务器共只有 13 个不同 IP 地址的域名，但并非仅由 13 台机器所组成。

域名	IP 地址	运营商
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, Information Sciences Institute
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

根域名服务器位置分布

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

- 根域名服务器分布在全世界。
- 为了提供更可靠的服务，在每一个地点的根域名服务器往往由多台机器组成。
- 根域名服务器采用任播 (anycast) 技术，当 DNS 客户向某个根域名服务器发出查询报文时，路由器能找到离这个 DNS 客户最近的一个根域名服务器。
- 截至 2024年2月17日，全球共有 1733 个根域名服务器在运行，其中在我国的共有 61 个。

News and publications [show all](#)

2022-08-02 [Statement on adding ZONEMD to the root zone](#)

2021-03-30 [Statement on DNS Encryption](#)

2019-08-14 [Threat Mitigation For the Root Server System](#)

Meeting agendas [show all](#)

2022-03-23 [IETF 113/Vienna/Virtual \(PDF\)](#)

2021-11-09 [IETF 112/Virtual \(PDF\)](#)

2021-07-26 [IETF 111/Virtual \(PDF\)](#)



As of 09/08/2022 5:58 a.m., the root server system consists of 1546 instances operated by the 12 independent root server operators.

The 13 root name servers are operated by 12 independent organisations.

You can find more information about each of these organisations by visiting their homepage as found in the 'Operator' field below.

Technical questions about the Root Server System as a whole can be directed to the [Ask RSSAC e-mail address](#).

Visualisations produced from RSSAC002 data submitted by the root server operators can be viewed at [rssac002.root-servers.org](#)

Root Servers

[Archives](#)

A B C D E F G H I J K L M

Operator Verisign, Inc.

[Homepage](#) [Statistics](#) [Contact Email](#) [RSSAC](#)

1. 域名系统 DNS

- 顶级域名服务器（TLD服务器）
 - 顶级域名服务器负责管理在该顶级域名服务器注册的所有二级域名。
 - 当收到 DNS 查询请求时，就给出相应的回答（可能是最后的结果，也可能是下一步应当找的域名服务器的 IP 地址）。

1. 域名系统 DNS

□ 权限域名服务器

- 权限域名服务器负责一个区的域名服务器。
- 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。

1. 域名系统 DNS

1.3 域名服务器

□ 本地域名服务器

- **非常重要。**
- 当一个主机发出 DNS 查询请求时，这个查询请求报文就发送给本地域名服务器。
- 每一个互联网服务提供者 ISP，或一个大学，都可拥有一个本地域名服务器。
- 当所要查询的主机也属于同一个本地 ISP 时，该本地域名服务器立即就能将所查询的主机名转换为它的 IP 地址，而不需要再去询问其他的域名服务器。
- 本地域名服务器有时也称为**默认域名服务器**。

1. 域名系统 DNS

1.3 域名服务器

□ 提高域名服务器的可靠性的方法

- DNS 域名服务器都把数据复制到几个域名服务器来保存，其中的一个是**主域名服务器**，其他的是**辅助域名服务器**。
- 当主域名服务器出故障时，辅助域名服务器可以保证 DNS 的查询工作不会中断。
- 主域名服务器**定期**把数据复制到辅助域名服务器中，而**更改数据只能在主域名服务器中进行**，保证了数据的一致性。

1. 域名系统 DNS

1.4 域名的解析过程

- 主机向本地域名服务器的查询一般都是采用**递归查询**。
 - 如主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。
- 本地域名服务器向根域名服务器的查询通常是采用**迭代查询**。
 - 当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的 IP 地址，要么告诉本地域名服务器：“你下一步应当向哪一个域名服务器进行查询”。
 - 然后让本地域名服务器进行后续的查询。

1. 域名系统 DNS

1.4 域名的解析过程

递归查询

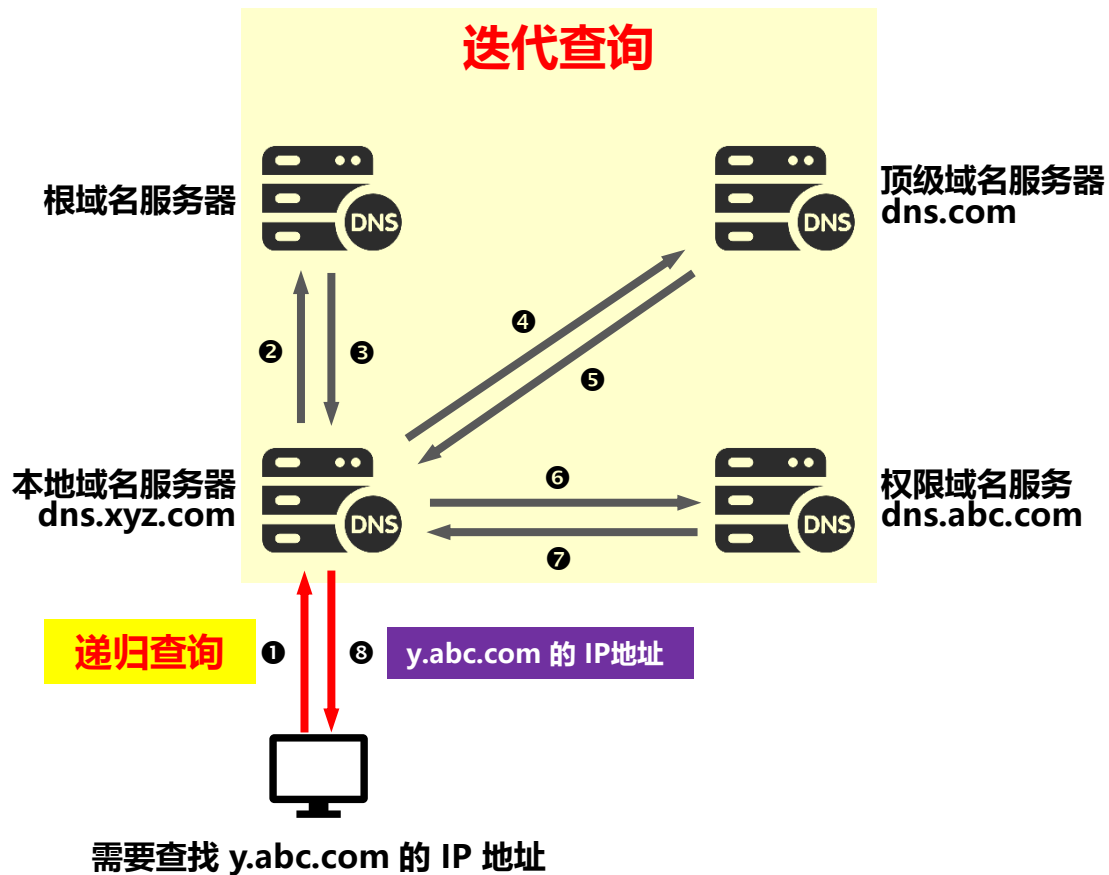
- 通常，主机向本地域名服务器查询时使用。
- 若不知道，就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。

迭代查询

- 本地域名服务器向根域名服务器查询时使用。
- 要么给出所要查询的 IP 地址，要么告诉下一个要查询的域名服务器的 IP 地址。
- 本地域名服务器继续后续查询。

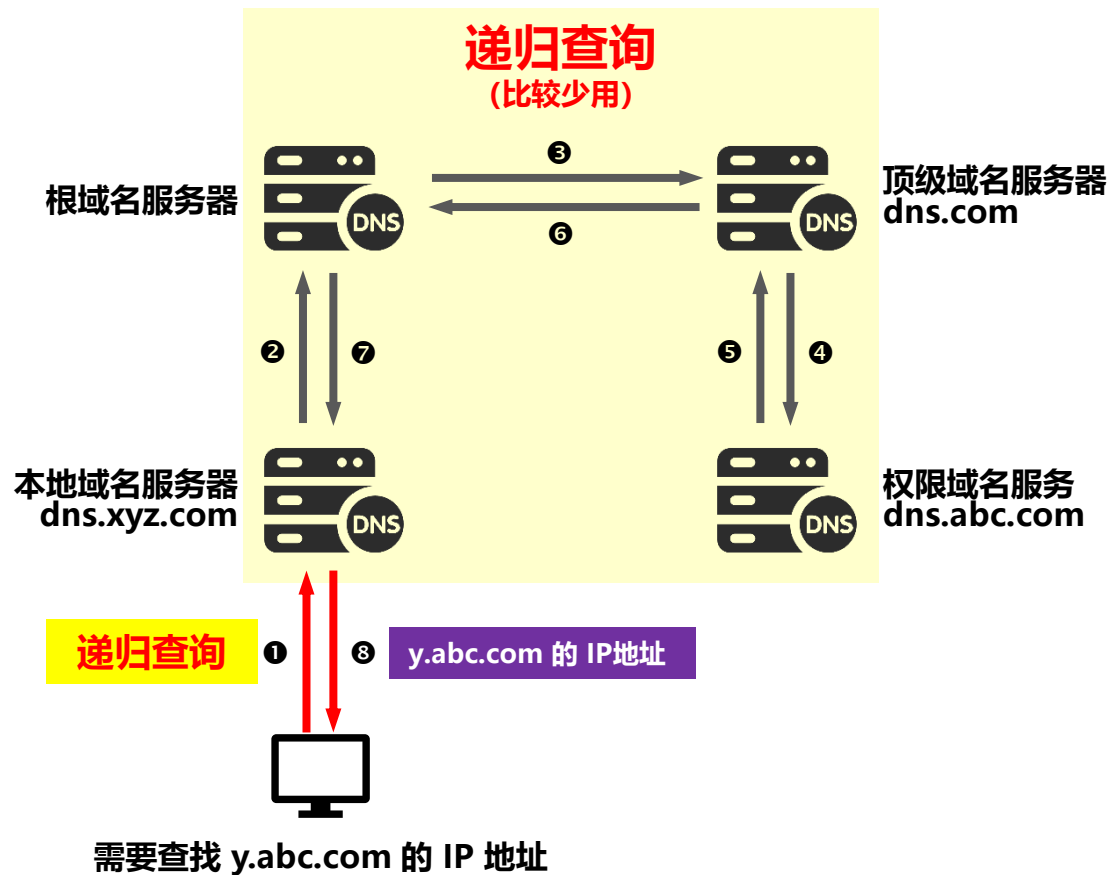
1. 域名系统 DNS

1.4 域名的解析过程

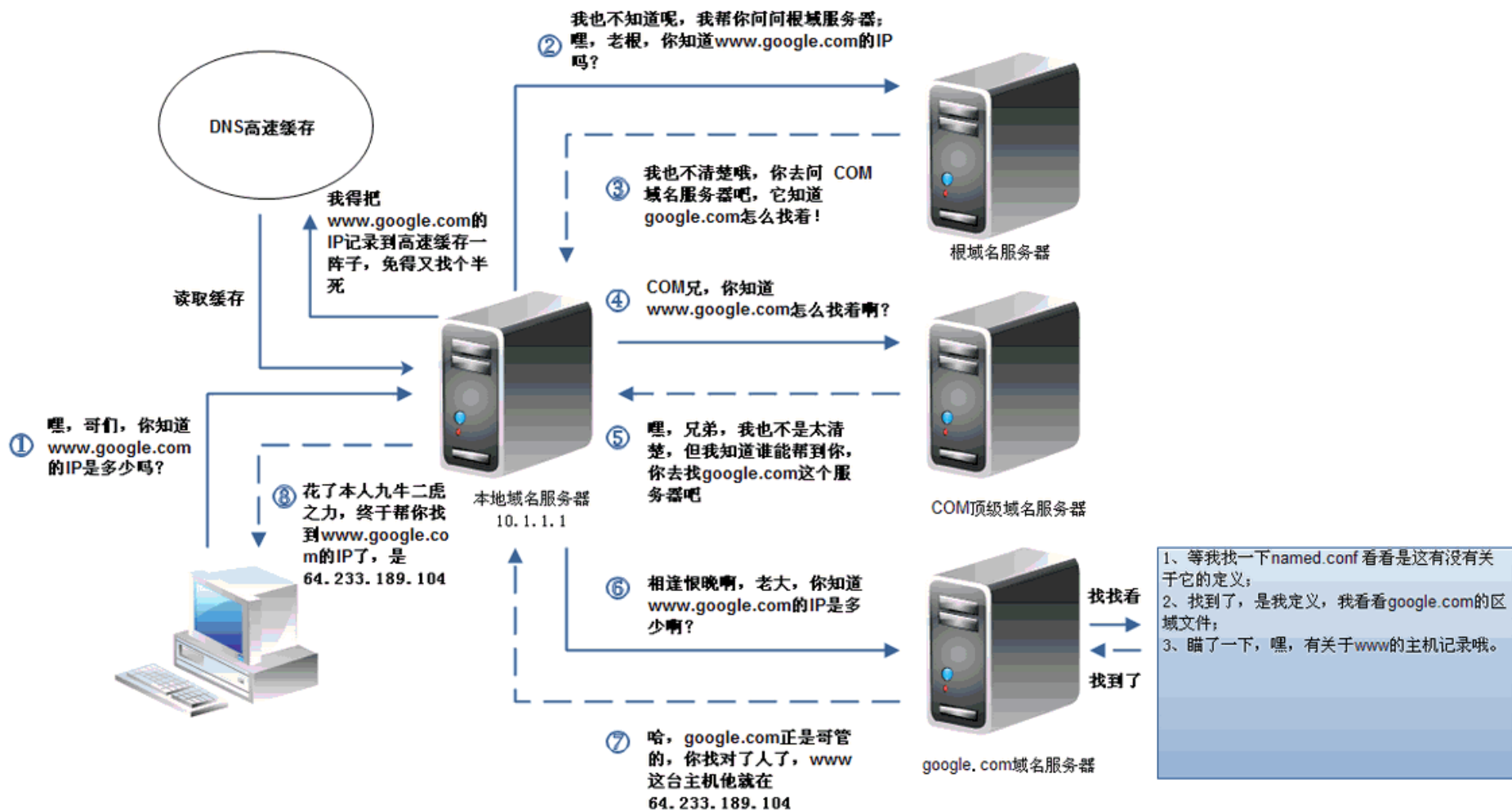


1. 域名系统 DNS

1.4 域名的解析过程



DNS域名解析过程



递归查询

迭代查询

1. 域名系统 DNS

1.4 域名的解析过程

The image shows a Wireshark capture of network traffic on the Ethernet interface (port 53). The capture shows four packets related to a DNS query for www.edu.cn. The first packet is a query from 8.8.8.8 to 10.10.3.202. The second packet is a response from 8.8.8.8 to 10.10.3.202, containing the IP address 202.112.0.36 and the authoritative name server 202.205.109.203. The third packet is another query from 10.10.3.202 to 8.8.8.8 for the AAAA record. The fourth packet is the response from 8.8.8.8 to 10.10.3.202, containing the AAAA record 2001:da8:ff3a:c8fd:400::.

[Response In: 2]
 Transaction ID: 0x0006
 Flags: 0x0100 Standard query
 0... .. = Response: Message is a query
 .000 0... .. = Opcode: Standard query (0)
0. = Truncated: Message is not truncated
1. = Recursion desired: Do query recursively
0. = Z: reserved (0)
0. = Non-authenticated data: Unacceptable
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0

```

0000  00000000 00100100 10101100 00001110 11111001 10011000 00000000 01010000  .$. ....P
0008  01010110 10110111 00000010 11000100 00001000 00000000 01000101 00000000  V. ....E
0010  00000000 00111000 00100010 00100100 00000000 00000000 10000000 00010001  .8"$...
0018  00000000 00000000 00001010 00001010 00000011 11001010 00001000 00001000  .....
0020  00001000 00001000 11101010 11111011 00000000 00110101 00000000 00100100  ....5.$
0028  00011110 00011001 00000000 00000110 00000001 00000000 00000000 00000001  .....
0030  00000000 00000000 00000000 00000000 00000000 00000000 00000011 01110111  .....w
0038  01110111 01110111 00000011 01100101 01100100 01110101 00000010 01100011  ww.edu.c
0040  01101110 00000000 00000000 00000001 00000000 00000001  n.....
  
```

主机在需要解析域名时，仅通过递归查询方式向本地域名服务器发出查询请求，余下的查询工作全部交由本地域名服务器完成。

1. 域名系统 DNS

1.5 域名的高速缓存

- 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。
 - 高速缓存可大大减轻根域名服务器的负荷，使互联网上的 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，而减少此时间值可提高域名转换的准确性。

1. 域名系统 DNS

1.5 域名的高速缓存

- 不仅在本地域名服务器中需要高速缓存，在主机中也会保存域名解析的缓存。
- 主机在启动时从本地域名服务器下载名字和地址的全部数据库，维护存放自己最近使用的域名的高速缓存，并且只在从缓存中找不到名字时才使用域名服务器。
- 由于域名改动不是很频繁，因此高速缓存能够极大的提高域名查询的效率，提升网络访问的速度。

1. 域名系统 DNS

1.5 域名的高速缓存

- 管理Windows操作系统中的域名高速缓存。
 - ipconfig /displaydns 查看域名高速缓存记录
 - ipconfig /flushdns 清除域名高速缓存记录

- 管理ubuntu操作系统中的域名高速缓存。
 - /etc/init.d/dns-clean start 清除dns缓存记录

1. 域名系统 DNS

1.6 如何选用本地域名服务器

- 域名系统 DNS 在网络通信中的作用非常重要，确保 DNS 查询的可用性和正确性，是进行网络故障判断的常用手段。
- 域名解析的测试工具：
 - 操作系统内置工具：nslookup、dig
 - DNS测试软件：DNS Benchmark、Google namebench
 - 在线DNS测试服务：
 - <http://tools.pingdom.com>
 - <http://ce.cloud.360.cn>
 - <http://tool.chinaz.com/dns>
 - <http://www.17ce.com/site/dns.html>

```
C:\WINDOWS\system32\cmd.exe - nslookup
Microsoft Windows [版本 10.0.19044.1586]
(c) Microsoft Corporation。保留所有权利。
```

```
C:\Users\ruanx>nslookup
默认服务器: XiaoQiang
Address: 172.19.100.1
```

```
> 1.1.1.1
服务器: XiaoQiang
Address: 172.19.100.1
```

```
名称: one.one.one.one
Address: 1.1.1.1
```

```
> www.baidu.com
服务器: XiaoQiang
Address: 172.19.100.1
```

```
非权威应答:
名称: www.a.shifen.com
Addresses: 220.181.38.150
           220.181.38.149
Aliases: www.baidu.com
```

```
> www.hactcm.edu.cn
服务器: XiaoQiang
Address: 172.19.100.1
```

```
非权威应答:
名称: www.hactcm.edu.cn
Addresses: 2001:250:480a:80:80::11
           211.69.32.50
```

```
> network.xg.hactcm.edu.cn
服务器: XiaoQiang
Address: 172.19.100.1
```

```
非权威应答:
名称: network.xg.hactcm.edu.cn
Address: 211.69.33.161
```

```

administrator@ubuntuserver-teach-ruan: ~
administrator@ubuntuserver-teach-ruan:~$ dig

;<<>> DiG 9.9.5-11ubuntu1.3-Ubuntu <<>>
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 56571
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 23

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
; .                IN      NS

;; ANSWER SECTION:
. 14317 IN      NS      m.root-servers.net.
. 14317 IN      NS      f.root-servers.net.
. 14317 IN      NS      h.root-servers.net.
. 14317 IN      NS      k.root-servers.net.
. 14317 IN      NS      l.root-servers.net.
. 14317 IN      NS      c.root-servers.net.
. 14317 IN      NS      a.root-servers.net.
. 14317 IN      NS      j.root-servers.net.
. 14317 IN      NS      i.root-servers.net.
. 14317 IN      NS      g.root-servers.net.
. 14317 IN      NS      d.root-servers.net.
. 14317 IN      NS      e.root-servers.net.
. 14317 IN      NS      b.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net. 345431 IN      A       198.41.0.4
a.root-servers.net. 345431 IN      AAAA    2001:503:ba3e::2:30
b.root-servers.net. 345431 IN      A       192.228.79.201
c.root-servers.net. 345431 IN      A       192.33.4.12
d.root-servers.net. 345431 IN      A       199.7.91.13
d.root-servers.net. 345431 IN      AAAA    2001:500:2d::d
e.root-servers.net. 345431 IN      A       192.203.230.10
f.root-servers.net. 345431 IN      A       192.5.5.241
f.root-servers.net. 345431 IN      AAAA    2001:500:2f::f
g.root-servers.net. 345431 IN      A       192.112.36.4
h.root-servers.net. 345431 IN      A       128.63.2.53
h.root-servers.net. 345431 IN      AAAA    2001:500:1::803f:235
i.root-servers.net. 345431 IN      A       192.36.148.17
i.root-servers.net. 345431 IN      AAAA    2001:7fe::53
j.root-servers.net. 345431 IN      A       192.58.128.30
j.root-servers.net. 345431 IN      AAAA    2001:503:c27::2:30
k.root-servers.net. 345431 IN      A       193.0.14.129
k.root-servers.net. 345431 IN      AAAA    2001:7fd::1
l.root-servers.net. 345431 IN      A       199.7.83.42
l.root-servers.net. 345431 IN      AAAA    2001:500:3::42
m.root-servers.net. 345431 IN      A       202.12.27.33
m.root-servers.net. 345431 IN      AAAA    2001:dc3::35

;; Query time: 6 msec
;; SERVER: 211.69.32.8#53(211.69.32.8)
;; WHEN: Thu Jun 02 00:09:38 CST 2016
;; MSG SIZE rcvd: 699

administrator@ubuntuserver-teach-ruan:~$

```

```

administrator@ubuntuserver-teach-ruan: ~
administrator@ubuntuserver-teach-ruan:~$ dig www.qq.com

;<<>> DiG 9.9.5-11ubuntu1.3-Ubuntu <<>> www.qq.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 6546
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 11

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
; www.qq.com.     IN      A

;; ANSWER SECTION:
www.qq.com.      299    IN      A       115.25.209.39

;; AUTHORITY SECTION:
qq.com.          31940 IN      NS      ns2.qq.com.
qq.com.          31940 IN      NS      ns1.qq.com.
qq.com.          31940 IN      NS      ns3.qq.com.
qq.com.          31940 IN      NS      ns4.qq.com.

;; ADDITIONAL SECTION:
ns1.qq.com.      3539  IN      A       101.226.68.138
ns1.qq.com.      3539  IN      A       14.17.19.139
ns2.qq.com.      3539  IN      A       101.227.169.106
ns2.qq.com.      3539  IN      A       125.39.202.108
ns3.qq.com.      597   IN      A       182.140.177.149
ns3.qq.com.      597   IN      A       182.140.167.157
ns4.qq.com.      3539  IN      A       125.39.247.247
ns4.qq.com.      3539  IN      A       184.105.206.124
ns4.qq.com.      3539  IN      A       203.205.144.156
ns4.qq.com.      3539  IN      A       123.151.178.115

;; Query time: 222 msec
;; SERVER: 211.69.32.8#53(211.69.32.8)
;; WHEN: Thu Jun 02 00:11:10 CST 2016
;; MSG SIZE rcvd: 287

administrator@ubuntuserver-teach-ruan:~$

```

NEW RELEASE adds nameservers 1.1.1.1, 1.0.0.1 and 9.9.9.9

A unique, comprehensive, accurate & free Windows (and Linux/Wine) utility to determine the exact performance of local and remote DNS nameservers . . .

"You can't optimize it until you can measure it"
 Now you **CAN** measure it!

[Click here or on the image above to download this 159 KBvte program.](#)

File stats for: **DNSBench** [Download now](#) [Jump to freeware page](#)

Last Updated: Apr 21, 2019 at 00:00 Downloads/day: 2,078 Current Rank: 1
 Size: 159k (1,063.85 days ago) Total downloads: 7,525,908 Historical Rank: 2

Although GRC's DNS Benchmark is packed with features to satisfy the needs of the most demanding Internet gurus (and this benchmark offers features designed to enable serious DNS performance investigation), the box below demonstrates that it is also extremely easy for casual and first-time users to run:

How to Run the DNS Benchmark

After downloading and starting the utility (there's nothing to install), it's as easy as . . . 1 . . . 2 . . . 3

Name	Owner	Status	Response Time
172.19.100.1	Local Network Nameserver		
198.153.192.1	ULTRADNS, US		
156.154.71.1	ULTRADNS, US		
156.154.71.22	ULTRADNS, US		
156.154.71.25	ULTRADNS, US		
156.154.70.1	ULTRADNS, US		
156.154.70.22	ULTRADNS, US		
156.154.70.25	ULTRADNS, US		
198.153.194.1	ULTRADNS, US		
8.8.4.4	GOOGLE, US		
129.250.35.251	NTT-LTD-2914, US		
129.250.35.250	NTT-LTD-2914, US		
8.8.8.8	GOOGLE, US		
24.113.32.30	AS-WAVE-1, US		
208.67.220.123	OPENDNS, US		
204.97.212.10	SPRINTLINK, US		
204.117.214.10	SPRINTLINK, US		
208.67.220.220	OPENDNS, US		
208.67.222.123	OPENDNS, US		
208.67.222.220	OPENDNS, US		
4.2.2.1	LEVEL3, US		
4.2.2.5	LEVEL3, US		
4.2.2.4	LEVEL3, US		
9.9.9.9	QUADS-AS-1, CH		
199.2.252.10	SPRINTLINK, US		
204.194.232.200	302-DIRECT-MEDIA-ASN, US		
208.67.222.222	OPENDNS, US		
1.0.0.1	CLOUDFLARENET, US		
4.2.2.6	LEVEL3, US		
204.194.234.200	302-DIRECT-MEDIA-ASN, US		
4.2.2.2	LEVEL3, US		
4.2.2.3	LEVEL3, US		
24.113.32.29	AS-WAVE-1, US		
1.1.1.1	CLOUDFLARENET, US		
64.81.45.2	ATI-CBEYOND, US		
64.81.79.2	MEGAPATHS-US		
64.81.111.2	MEGAPATHS-US		
64.81.127.2	ATI-CBEYOND, US		
64.81.159.2	ATI-CBEYOND, US		
66.92.64.2	MEGAPATHS-US		
66.92.159.2	RBS11696, US		
66.92.224.2	ATI-CBEYOND, US		
66.93.87.2	ATI-CBEYOND, US		
68.1.18.25	ASN-CXA-ALL-CCI-22773-RDC, US		
68.1.18.30	ASN-CXA-ALL-CCI-22773-RDC, US		
68.2.16.25	ASN-CXA-ALL-CCI-22773-RDC, US		
68.2.16.25	ASN-CXA-ALL-CCI-22773-RDC, US		
68.2.16.30	ASN-CXA-ALL-CCI-22773-RDC, US		
68.4.16.25	ASN-CXA-ALL-CCI-22773-RDC, US		
68.4.16.30	ASN-CXA-ALL-CCI-22773-RDC, US		

Name	Min	Avg	Max	Std.Dev	Reliab%
172.19.100.1					
198.153.192.1					
156.154.71.1					
156.154.71.22					
156.154.70.1					
156.154.70.22					
156.154.70.25					
198.153.194.1					
8.8.4.4					
129.250.35.251					
129.250.35.250					
8.8.8.8					
24.113.32.30					
208.67.220.123					
204.97.212.10					
204.117.214.10					
208.67.220.220					
208.67.222.123					
208.67.222.220					
4.2.2.1					
4.2.2.5					
4.2.2.4					
9.9.9.9					
199.2.252.10					
204.194.232.200					
208.67.222.222					
1.0.0.1					
4.2.2.6					
204.194.234.200					
4.2.2.2					
4.2.2.3					
24.113.32.29					
1.1.1.1					
64.81.45.2					
64.81.79.2					
64.81.111.2					
64.81.127.2					
64.81.159.2					
66.92.64.2					
66.92.159.2					
66.92.224.2					
66.93.87.2					
68.1.18.25					
68.1.18.30					
68.2.16.25					
68.2.16.25					
68.2.16.30					
68.4.16.25					
68.4.16.30					

2. 文件传送协议

- 网络环境中的一项基本应用就是将文件从一台计算机中复制到另一台可能相距很远的计算机中。
- 网络环境下复制文件的复杂性：
 - 计算机存储数据的格式不同。
 - 文件的目录结构和文件命名的规定不同。
 - 对于相同的文件存取功能，操作系统使用的命令不同。
 - 访问控制方法不同。

2. 文件传送协议

□ 文件共享协议

■ 文件传送协议：FTP，TFTP 等。

□ 复制整个文件。对文件副本进行访问。

- 若要存取一个文件，就必须先获得一个本地文件副本。
- 若要修改文件，只能对文件副本进行修改，然后再将修改后的文件副本传回到原节点。

■ 联机访问 (on-line access) 协议：NFS 等。

- 允许同时对一个文件进行存取。
- 远地共享文件访问，如同对本地文件的访问一样。
- 透明存取，不需要对该应用程序作明显的改动。
- 由操作系统负责。

2. 文件传送协议

- 文件传送协议 FTP (File Transfer Protocol) 是互联网上使用得最广泛的文件传送协议。
 - 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，因而适合于在异构网络中任意计算机之间传送文件。
 - 是文件共享协议的一个大类。
 - 是有 RFC 959 定义的，且很早就成为了互联网的正式标准。

2. 文件传送协议

□ FTP 特点

- 只提供文件传送的一些基本服务，它使用 TCP 可靠的运输服务。
- 主要功能是减少或消除在不同操作系统下处理文件的不兼容性。
- 使用客户/服务器方式。
 - 一个 FTP 服务器进程可同时为多个客户进程提供服务。
 - FTP 的服务器进程由两大部分组成：
 - 一个主进程，负责接受新的请求；
 - 另外有若干个从属进程，负责处理单个请求。

2. 文件传送协议

□ FTP 的主进程工作步骤

1. 打开熟知端口（端口号为 21），使客户进程能够连接上。
2. 等待客户进程发出连接请求。
3. 启动从属进程来处理客户进程发来的请求。
 - 从属进程对客户进程的请求处理完毕后即终止。
 - 从属进程在运行期间根据需要还可能创建其他一些子进程。
4. 回到等待状态，继续接受其他客户进程发来的请求。
 - 主进程与从属进程的处理是并发地进行。

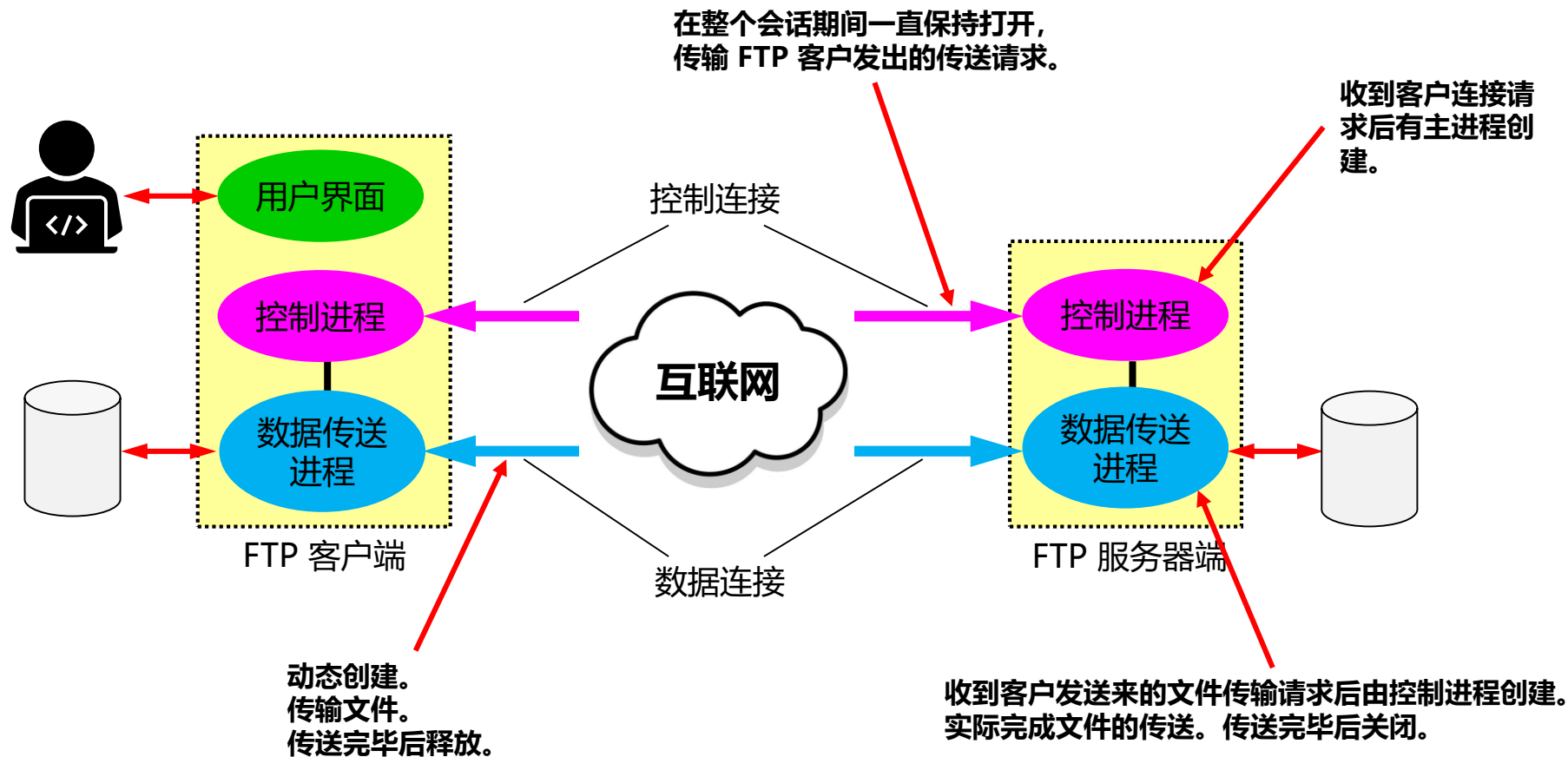
2. 文件传送协议

2.2 FTP 的基本工作原理

- FTP 在通信中保持两个连接：
 - **控制连接**在整个会话期间一直保持打开，FTP 客户发出的传送请求通过控制连接发送给服务器端的控制进程，但控制连接不用来传送文件。
 - 实际用于传输文件的是“**数据连接**”。
 - 服务器端的控制进程在接收到 FTP 客户发送来的文件传输请求后就创建“数据传送进程”和“数据连接”，用来连接客户端和服务器的数据传送进程。
 - 数据传送进程实际完成文件的传送，在传送完毕后关闭“数据传送连接”并结束运行。

2. 文件传送协议

2.2 FTP 的基本工作原理



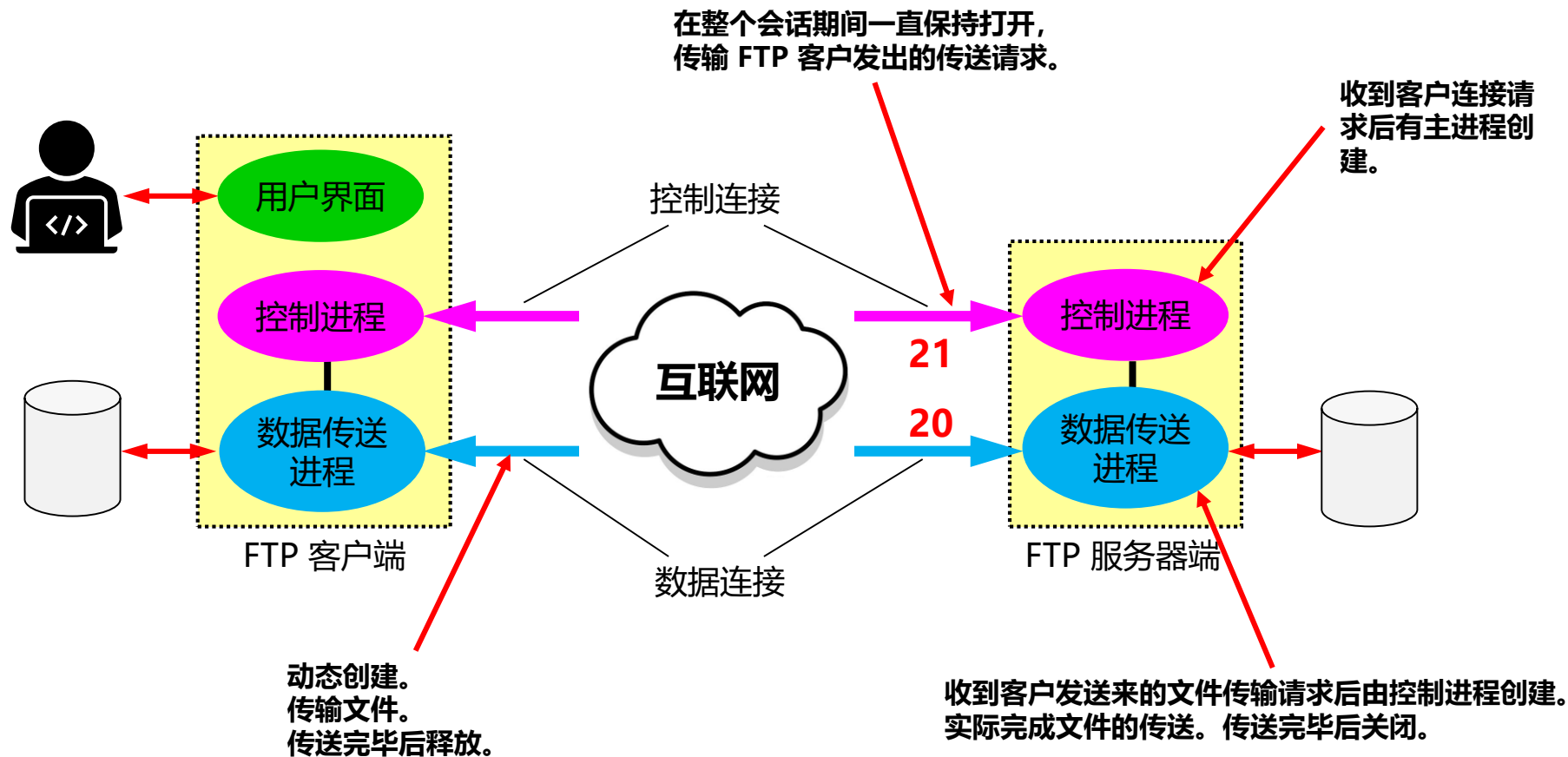
2. 文件传送协议

2.2 FTP 的基本工作原理

- FTP 在通信中使用两个端口号：
 - 当客户进程向服务器进程发出建立连接请求时，要寻找**连接服务器进程的熟知端口(21)**，同时还要告诉服务器进程自己的另一个端口号码，用于建立数据传送连接。
 - 接着，服务器进程用自己**传送数据的熟知端口(20)**与客户进程所提供的端口号码建立数据传送连接。
 - 由于 FTP 使用了两个不同的端口号，所以数据连接与控制连接不会发生混乱。
- 使用两个端口号的优势：
 - 使协议更加简单和更容易实现。
 - 在传输文件时还可以利用控制连接。

2. 文件传送协议

2.2 FTP 的基本工作原理



2. 文件传送协议

2.3 简单文件传送协议 TFTP

- TFTP 是一个很小且易于实现的文件传送协议。
- TFTP 使用客户服务器方式和使用 UDP 数据报，因此 TFTP 需要有自己的差错改正措施。
- TFTP 只支持文件传输而不支持交互。
- TFTP 没有一个庞大的命令集，没有列目录的功能，也不能对用户进行身份鉴别。

2. 文件传送协议

2.3 简单文件传送协议 TFTP

- TFTP 的主要特点是：
 - 每次传送的数据 PDU 中有 512 字节的数据，但最后一次可不足 512 字节。
 - 数据 PDU 也称为文件块 (block) ，每个块按序编号，从1开始。
 - 支持 ASCII 码或二进制传送。
 - 可对文件进行读或写。
 - 使用很简单的首部。

2. 文件传送协议

2.3 简单文件传送协议 TFTP

- TFTP 的工作很像停止等待协议：
 - 发送完一个文件块后就等待对方的确认，确认时应指明所确认的块编号。
 - 发完数据后在规定时间内收不到确认就要重发数据 PDU。
 - 发送确认 PDU 的一方若在规定时间内收不到下一个文件块，也要重发确认 PDU。这样就保证文件的传送不致因某一个数据报的丢失而告失败。

2. 文件传送协议

2.3 简单文件传送协议 TFTP

- TFTP 的工作很像停止等待协议：
 - 在一开始工作时。
 - TFTP 客户进程发送一个读请求 PDU 或写请求 PDU 给 TFTP 服务器进程，其熟知端口号码为 69。
 - TFTP 服务器进程要选择一个新的端口和 TFTP 客户进程进行通信。
 - 若文件长度恰好为 512 字节的整数倍，则在文件传送完毕后，还必须在最后发送一个只含首部而无数据的数据 PDU。
 - 若文件长度不是 512 字节的整数倍，则最后传送数据 PDU 的数据字段一定不满 512 字节，这正好可作为文件结束的标志。

2. 文件传送协议

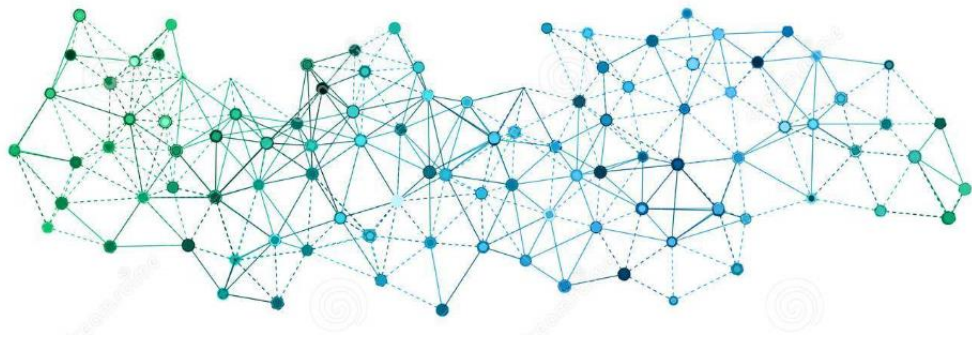
2.4 FTP 与 TFTP 的对比

- TFTP 是 FTP 的简化版本，在确切地知道想要得到的文件名及准确位置情况下，才可有选择地使用 TFTP。
- TFTP 是一个非常易用的、快捷的程序。
 - TFTP 不提供目录浏览的功能，只能完成文件的发送和接收操作。
 - TFTP 发送比 FTP 更小的数据块，同时没有 FTP 所需要的传送确认，因而是不可靠的。

2. 文件传送协议

2.4 FTP 与 TFTP 的对比

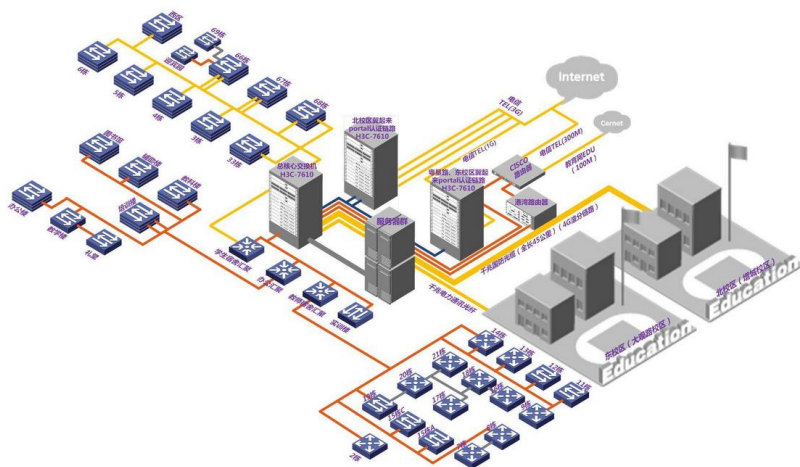
- 在用途上：
 - FTP 是完整、面向会话、常规用途文件传输协议。
 - TFTP 用作特殊目的的文件传输协议。
- 在交互性上：
 - FTP 允许交互通信。
 - TFTP 仅允许单向传输文件。
- 在认证上：
 - FTP 提供身份验证。
 - TFTP 不支持身份验证。
- 在运输层上：
 - FTP 使用已知 TCP 端口号 21 作为控制连接，TCP 端口号 20 作为数据连接。
 - TFTP 使用 UDP 端口号 69 作为文件传输活动。



3. 远程终端协议 TELNET

3.1 客观问题

- 随着网络规模的扩大，网络应用的丰富，园区网部署的网络设备（例如路由器、交换机、无线设备等）越来越多，对设备的可靠运行要求也越来越高。
- 如何进行设备管理与维护？



3. 远程终端协议 TELNET

3.1 客观问题

远程
管理



集中
控制

3. 远程终端协议 TELNET

3.2 TELNET 的基本概念

- TELNET 是简单的远程终端协议。
- TELNET 是互联网正式标准。
 - 该标准定义文档为【RFC 854】【RFC 5198】。
 - <https://www.rfc-editor.org/info/rfc854>。
- TELNET 协议是 TCP/IP 协议族的一员，是 Internet 远程登录服务的标准协议。
 - 用 TELNET 可通过 TCP 连接登录到远程主机上。
 - 用 TELNET 访问远程主机使用主机名或 IP 地址。
 - 远程主机使用 TCP 23 端口进行通信。

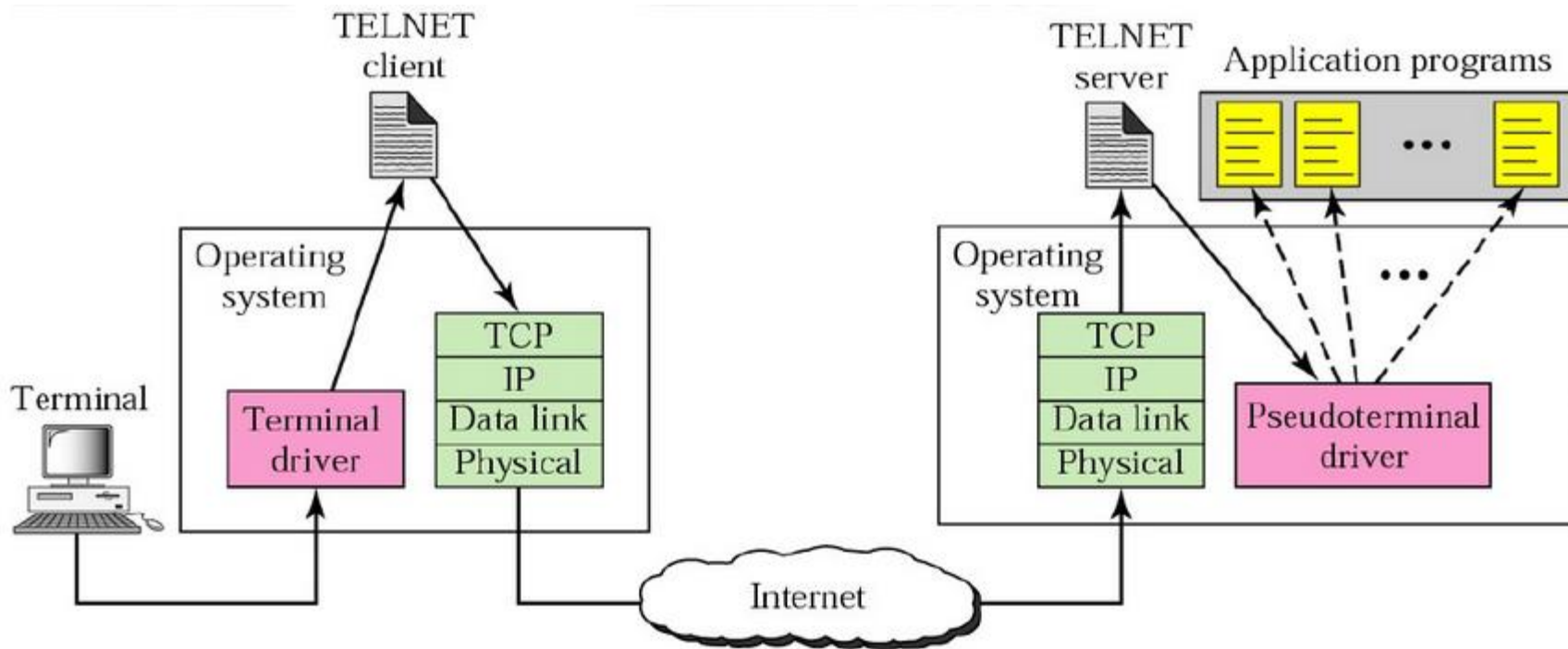
3. 远程终端协议 TELNET

3.3 TELNET 的实现方式

- TELNET 使用客户/服务器 方式。
- TELNET 实现需要三个组成部分。
 - TELNET 客户端软件：
 - 安装在本地主机，处理用户输入/输出。
 - TELNET 服务端软件：
 - 安装在远程受控设备。
 - TELNET 协议接口程序：
 - 存在 TELNET 客户端和服务端软件中，实现 TELNET 协议，并维护协议状态。

3. 远程终端协议 TELNET

3.3 TELNET 的实现方式



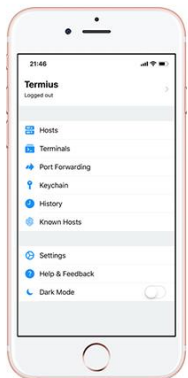
3. 远程终端协议 TELNET

3.3 TELNET 的实现方式

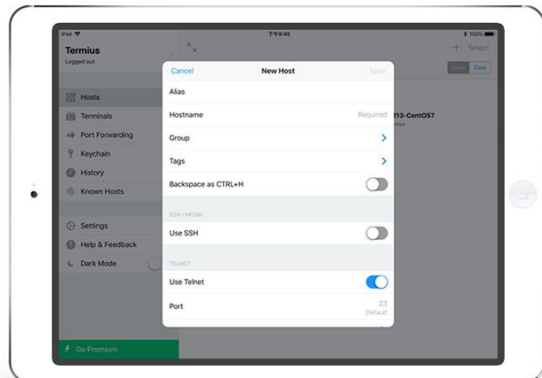
- TELNET 客户端软件要做两件事：
 - 读取用户在键盘上键入的字符，通过 TCP 连接发送到远程服务器上。
 - 读取从 TCP 连接上收到的字符，显示在用户的终端上。



JuiceSSH
For Android



Terminus
For iPhone



Terminus
For iPad



Xshell
For Windows

3. 远程终端协议 TELNET

3.4 TELNET 的使用方法

- TELNET 将用户的击键传到远程受控设备，同时将远程受控设备的输出返回到用户屏幕。
- TELNET 服务是透明的，使用 TELNET 进行远程控制就仿佛自己的键盘和显示器是直接连在远地主机上。

3. 远程终端协议 TELNET

3.5 TELNET 的特点

跨平台
支持

传送
击键

明文
传输

低成本
实现

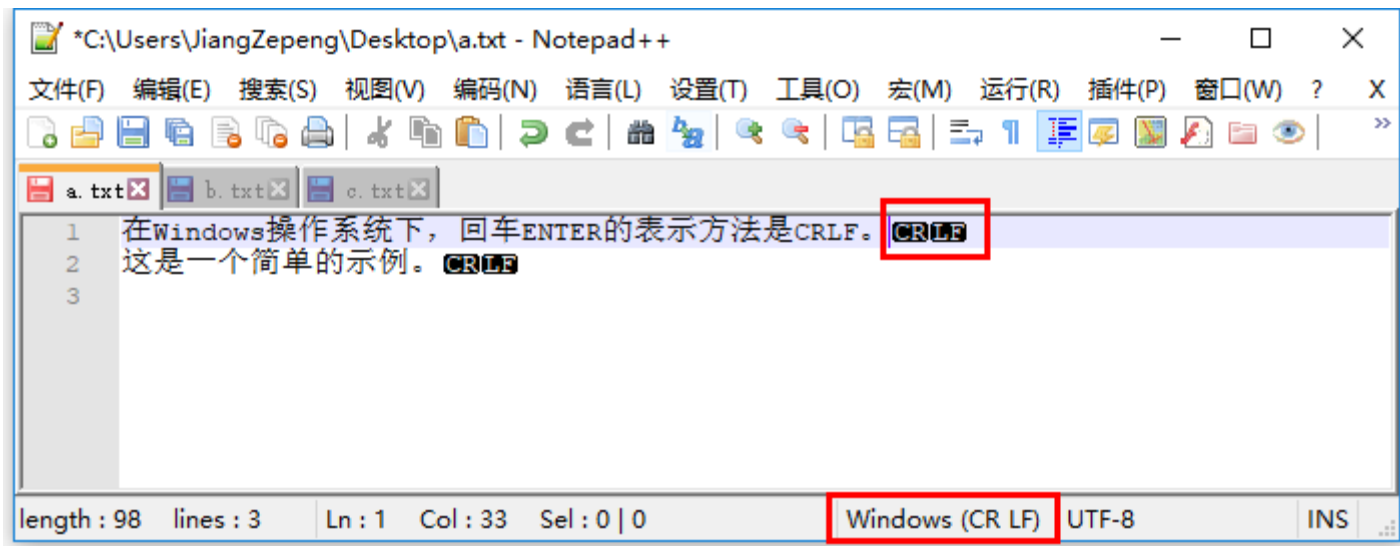


讨论：为什么 TELNET 不用加密的方式通信？



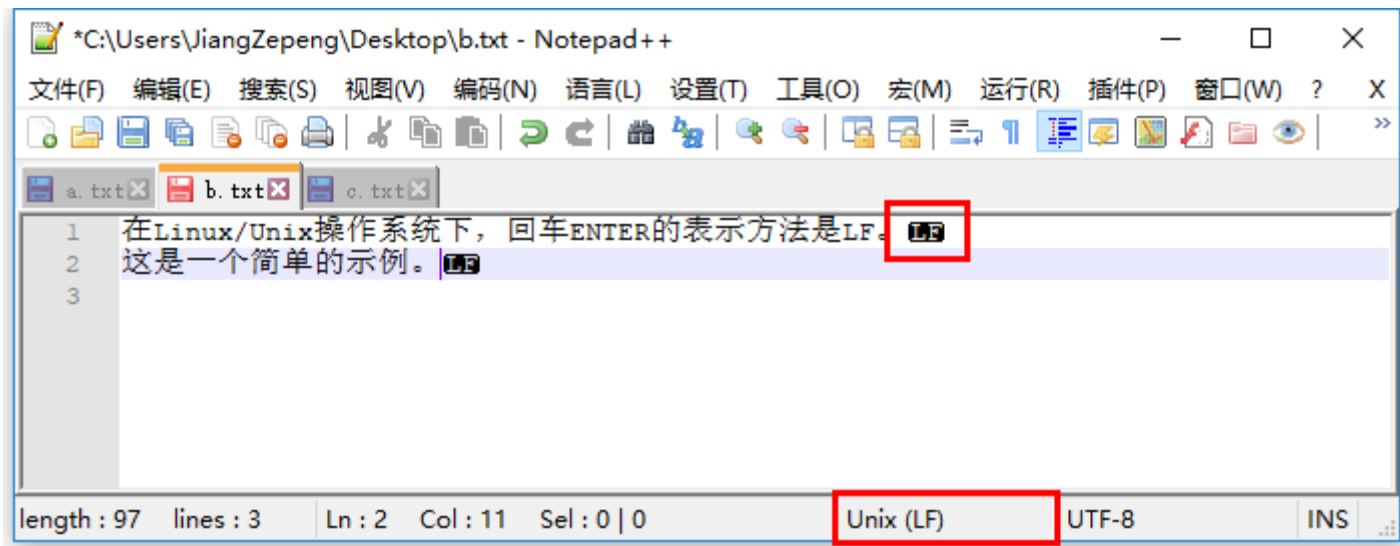
工程应用以最优为原则

3. 远程终端协议 TELNET



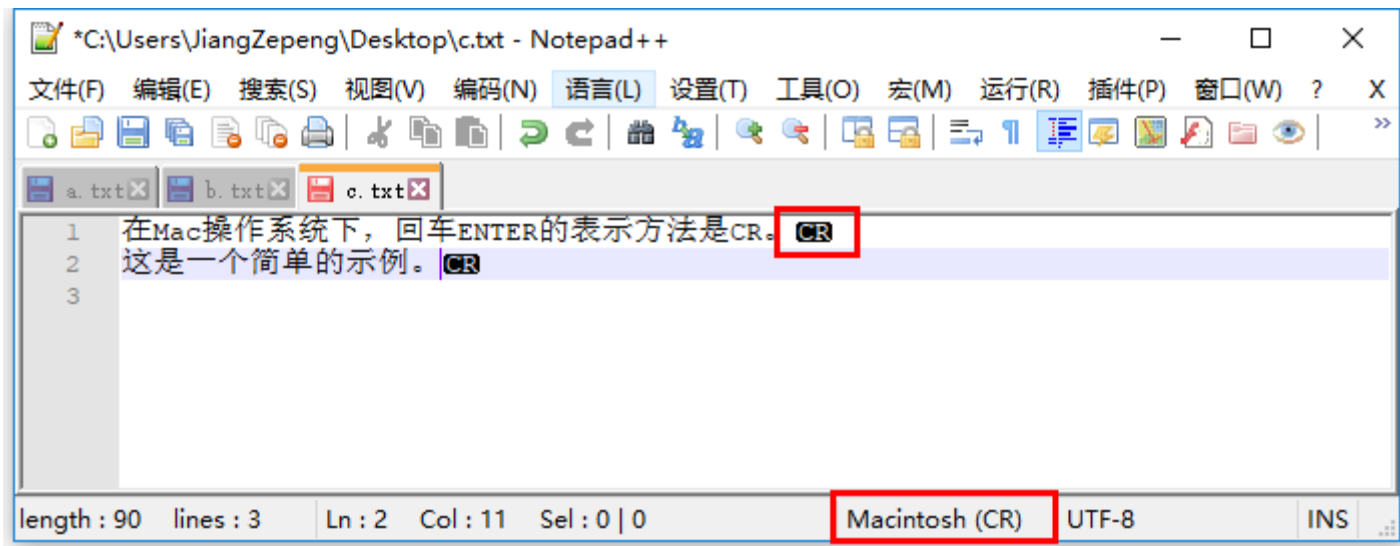
3. 远程终端协议 TELNET

3.6 NVT



3. 远程终端协议 TELNET

3.6 NVT





讨论：

本地主机是 Windows 操作系统，
远程受控设备是 Linux 操作系统，
通过 TELNET 进行远程管理时进行回车操作，
远程受控主机能够识别么？



3. 远程终端协议 TELNET

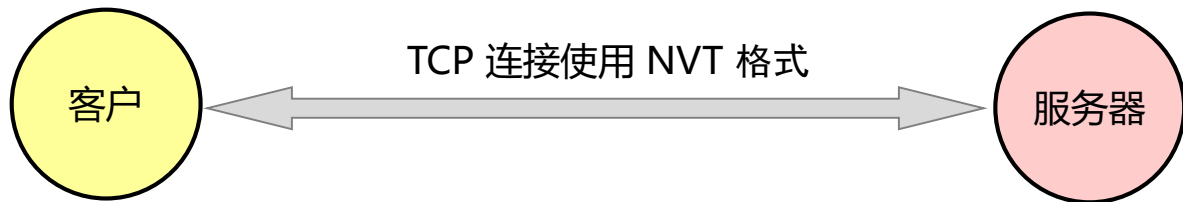
- 为了使多个操作系统间的 TELNET 交互操作成为可能，就必须详细了解异构计算机和操作系统。
 - 比如，一些操作系统需要每行文本用 ASCII 回车控制符 (CR) 结束，另一些系统则需要使用 ASCII 换行符 (LF)，还有一些系统需要用两个字符的序列回车-换行 (CR-LF)；
 - 比如，大多数操作系统为用户提供了一个中断程序运行的快捷键，但这个快捷键在各个系统中有可能不同（一些系统使用CTRL+C，而另一些系统使用ESCAPE）。如果不考虑系统间的异构性，那么在本地产出的字符或命令，传送到远地并被远地系统解释后很可能会不准确或者出现错误。

3. 远程终端协议 TELNET

- 为了适应异构环境，TELNET 协议定义了数据和命令在 Internet 上的传输方式，此定义被称作网络虚拟终端 NVT (Net Virtual Terminal) 。
- TELNET 使用网络虚拟终端 NVT 格式。
 - 客户软件把用户的击键和命令转换成 NVT 格式，并送交服务器。
 - 服务器软件把收到的数据和命令，从 NVT 格式转换成远地系统所需的格式。
 - 向用户返回数据时，服务器把远地系统的格式转换为 NVT 格式，本地客户再从 NVT 格式转换到本地系统所需的格式。

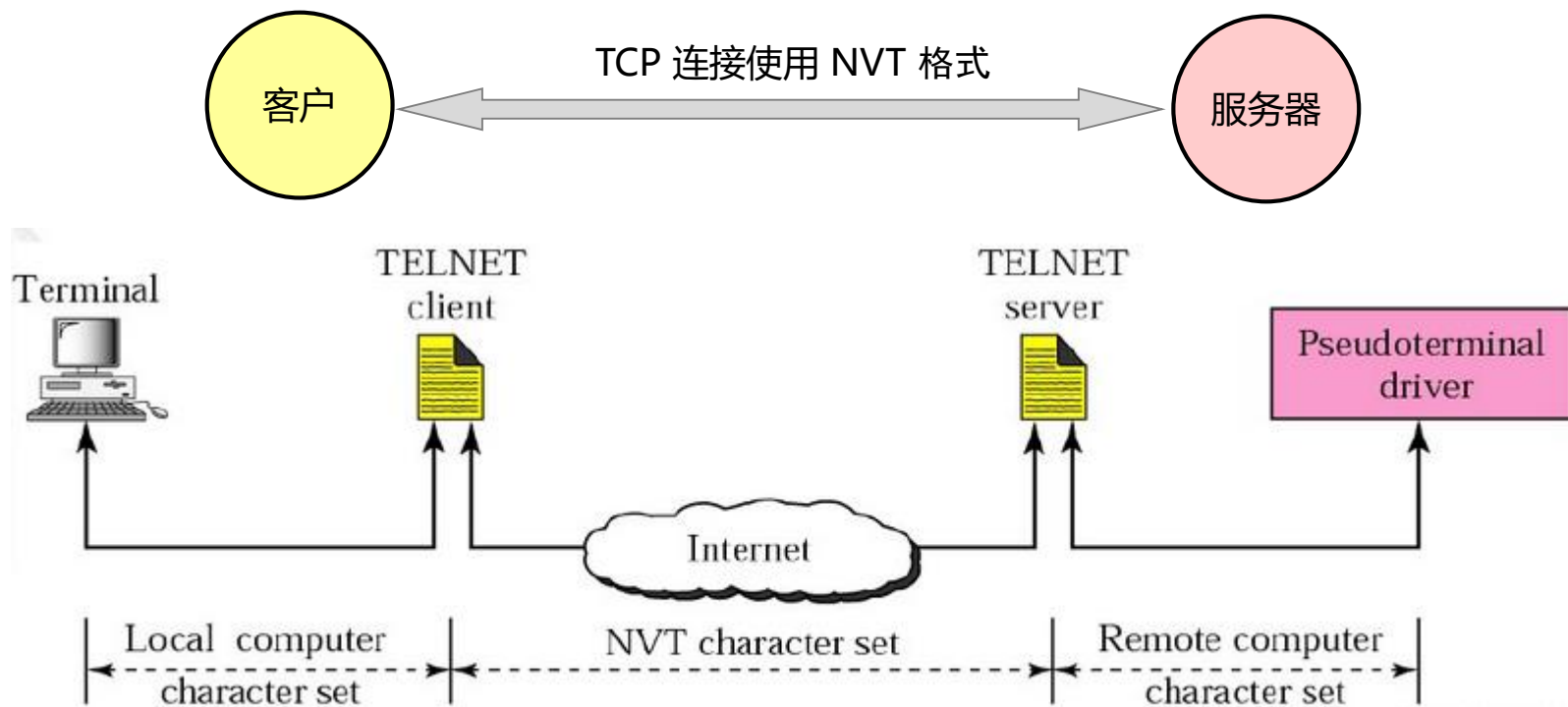
3. 远程终端协议 TELNET

- TELNET 使用网络虚拟终端 NVT 格式。



3. 远程终端协议 TELNET

- TELNET 使用网络虚拟终端 NVT 格式。





讨论：还有其他远程管理方式么？

3. 远程终端协议 TELNET

3.7 扩展：远程管理的更多实现方式



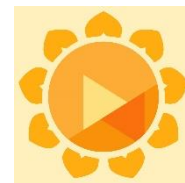
RDP
Windows



VNC
Linux、UNIX、Windows、Mac



SSH
Linux、UNIX、Mac



Agent
Linux、UNIX、Windows、Mac

3. 远程终端协议 TELNET

3.8 还有两个问题

- 操作协商：
 - 基本的 NVT 设备所具有特性十分有限，只能接收和显示 7 位的 ASCII 码，为此 TELNET 协议定义了一组协议用于扩展基本 NVT 的功能，使 NVT 达到用户终端所具有的功能。
 - TELNET 协议规定在扩展 NVT 功能时采用协商的机制，只有通信双方通过协商后达成一致的的特性才能使用。
 - 有哪些扩展？如何进行协商？

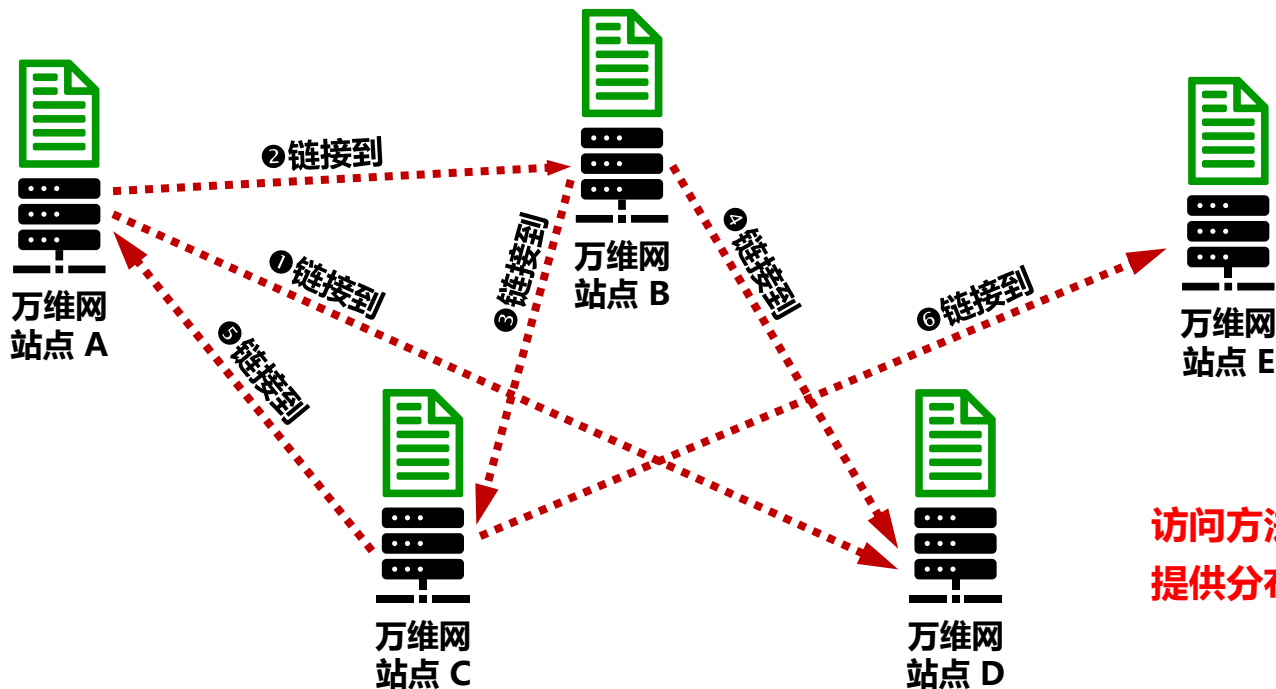
- TELNET 报文结构：
 - 发送与接收报文的结构如何？各字段的含义与取值？

4. 万维网 WWW

- 万维网 WWW (World Wide Web) 并非某种特殊的计算机网络。
- 万维网是一个大规模的、联机式的信息储藏所。
- 万维网用链接的方法能非常方便地从互联网上的一个站点访问另一个站点，从而主动地按需获取丰富的信息。
- 这种访问方式称为“**链接**”。

4. 万维网 WWW

4.1 万维网概述



访问方法：链接。
提供分布式服务。

万维网 WWW (World Wide Web) 是一个大规模的、联机式的信息储藏所，并非某种特殊的计算机网络。

4. 万维网 WWW

- 万维网是分布式超媒体 (hypermedia) 系统。
 - 是超文本 (hypertext) 系统的扩充。
 - 超文本：由多个信息源链接成。是万维网的基础。
 - 超媒体与超文本的区别：文档内容不同。
 - 超文本文档仅包含文本信息。
 - 超媒体文档还包含其他信息，如图形、图像、声音、动画，甚至活动视频图像等。
 - 分布式系统
 - 信息分布在整个互联网上。
 - 每台主机上的文档都独立进行管理。

4. 万维网 WWW

- 万维网的工作方式
 - 以**客户/服务器**方式工作。
 - 客户端程序：浏览器。
 - 服务器程序：在万维网文档所驻留的主机上运行。
 - 这个计算机也称为万维网服务器。
 - 客户程序向服务器程序发出请求，服务器程序向客户程序送回客户所要的万维网文档。
 - 在一个客户端程序主窗口上显示出的万维网文档称为页面 (page)。

4. 万维网 WWW

- 万维网必须解决的问题：
 - 怎样标志分布在整个互联网上的万维网文档？
 - 使用统一资源定位符 URL (Uniform Resource Locator) 。
 - 使每一个文档在整个互联网的范围内具有唯一的标识符 URL。
 - 用什么协议来实现万维网上的各种链接？
 - 使用超文本传送协议 HTTP (HyperText Transfer Protocol)。
 - HTTP 是一个应用层协议，使用 TCP 连接进行可靠的传送。

4. 万维网 WWW

- 万维网必须解决的问题：
 - 怎样使不同作者创作的不同风格的万维网文档都能在互联网上的各种主机上显示出来，同时使用户清楚地知道在什么地方存在着链接？
 - 使用超文本标记语言 HTML (HyperText Markup Language) 。
 - 怎样使用户能够很方便地找到所需的信息？
 - 使用各种的搜索工具（即搜索引擎）。
 - 使用推荐系统。

4. 万维网 WWW

4.2 统一资源定位符 URL

- 统一资源定位符 URL：
 - 是对互联网上资源的位置和访问方法的一种简洁表示。
 - 给资源的位置提供一种抽象的识别方法，并用这种方法给资源定位。
 - 实际上就是在互联网上的资源的地址。
 - 显然，互联网上的所有资源，都有一个唯一确定的 URL。
 - 资源：指在互联网上可以被访问的任何对象，包括文件目录、文件、文档、图像、声音等，以及与互联网相连的任何形式的数据。
 - URL 相当于一个文件名在网络范围的扩展。
 - 因此，URL 是与互联网相连的机器上的任何可访问对象的一个指针。

4. 万维网 WWW

4.2 统一资源定位符 URL

- URL 的格式：
 - 由以**冒号 (:)** 隔开的两大部分组成，对字符大写或小写没有要求。
 - 一般形式：

<协议>://<主机>:<端口>/<路径>

ftp —— 文件传送协议 FTP

http —— 超文本传送协议 HTTP

News —— USENET 新闻

4. 万维网 WWW

4.2 统一资源定位符 URL

- URL 的格式：
 - 由以**冒号 (:)** 隔开的两大部分组成，对字符大写或小写没有要求。
 - 一般形式：

<协议>://<主机>:<端口>/<路径>

<主机>
是存放资源的主机
在互联网中的域名

4. 万维网 WWW

4.2 统一资源定位符 URL

□ URL 的格式:

- 由以**冒号 (:)** 隔开的两大部分组成, 对字符大写或小写没有要求。
- 一般形式:

<协议>://<主机>:<端口>/<路径>

端口: 省略时使用默认端口号。

路径: 资源所在目录位置。区分大小写。

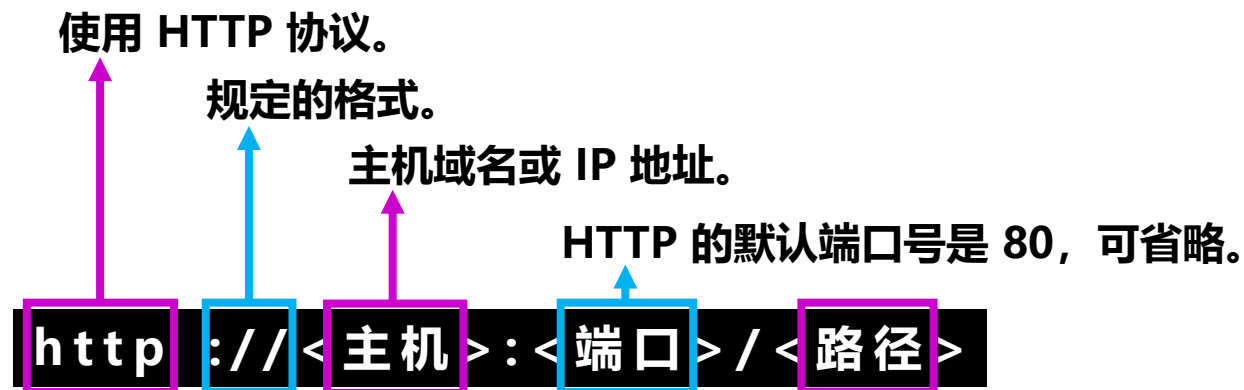
省略时使用所定义的默认路径。

后面可能还有一些选项。

4. 万维网 WWW

4.2 统一资源定位符 URL

□ 使用 HTTP 的 URL



省略时指到互联网上的某个主页 (home page)。更复杂一些的路径是指向层次结构的从属页面。

<https://internet.hactcm.edu.cn/network/sjyssp/index.html>

主机域名

路径名

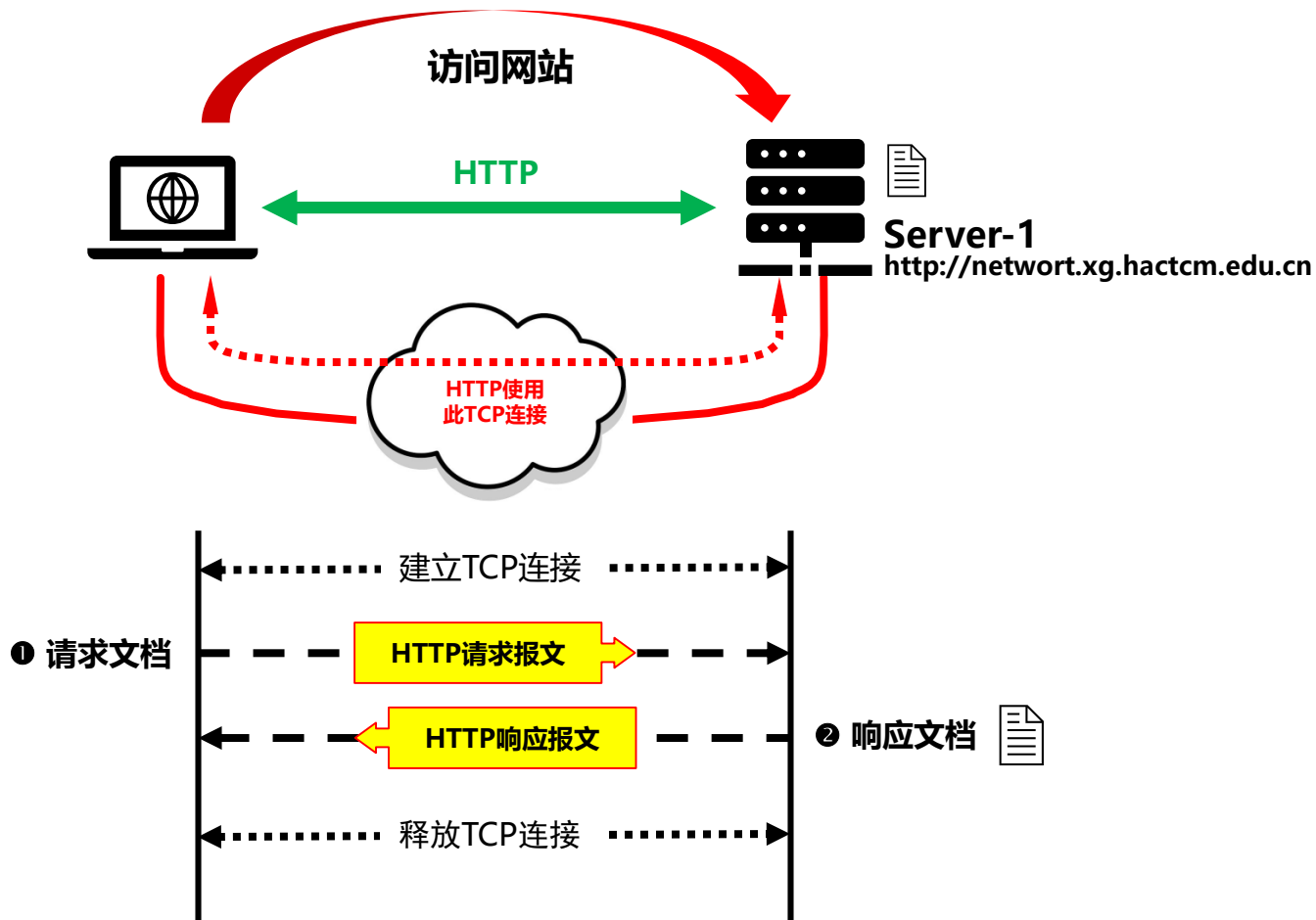
4. 万维网 WWW

4.3 超文本传送协议 HTTP

- HTTP 是面向事务的 (transaction-oriented) 应用层协议。
- 使用 TCP 连接进行可靠的传送。
- 定义了浏览器与万维网服务器通信的格式和规则。
- 是万维网上能够可靠地交换文件的重要基础。
 - 包括文本、声音、图像等各种多媒体文件。
- HTTP 不仅传送完成超文本跳转所必需的信息，而且也传送任何可从互联网上得到的信息，如文本、超文本、声音和图像等。

4. 万维网 WWW

4.3 超文本传送协议 HTTP

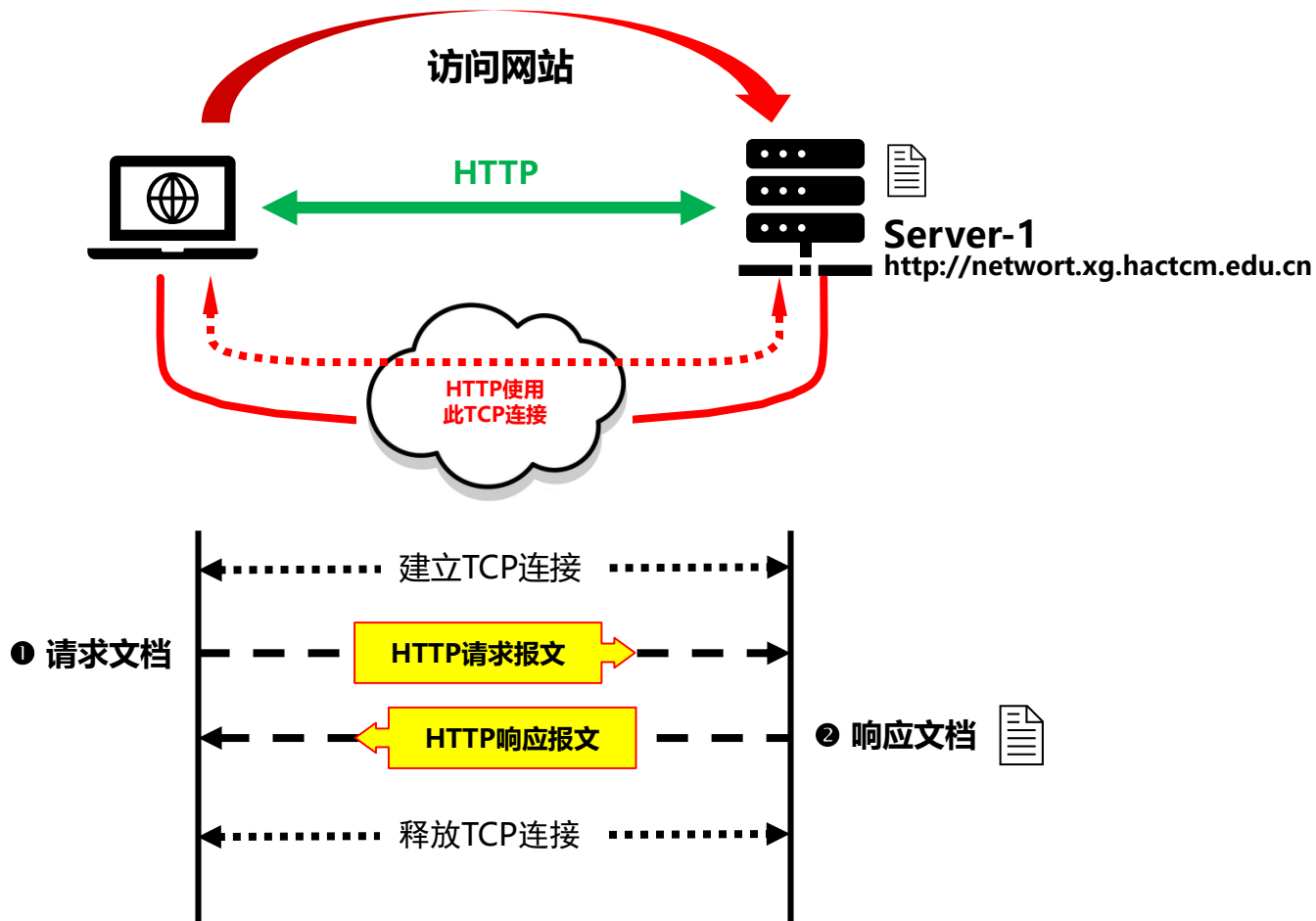


4. 万维网 WWW

4.3 超文本传送协议 HTTP

HTTP 规定:

- ✓ 在 HTTP 客户与 HTTP 服务器之间的每次交互，都由一个 ASCII 码串构成的请求和一个类似的通用互联网扩充，即“类 MIME (MIME-like)”的响应组成。
- ✓ HTTP 报文通常都使用 TCP 连接传送。



4. 万维网 WWW

4.3 超文本传送协议 HTTP

□ HTTP 的操作过程：

- ① 浏览器分析超链指向页面的 URL。
- ② 浏览器向 DNS 请求解析 internet.hactcm.edu.cn 的 IP 地址。
- ③ 域名系统 DNS 解析 IP 地址。
- ④ 浏览器与服务器建立 TCP 连接。
- ⑤ 浏览器发出取文件命令：GET /*/index.htm。
- ⑥ 服务器给出响应，把网页文件发给浏览器。
- ⑦ TCP 连接释放。
- ⑧ 浏览器显示网站内容。

4. 万维网 WWW

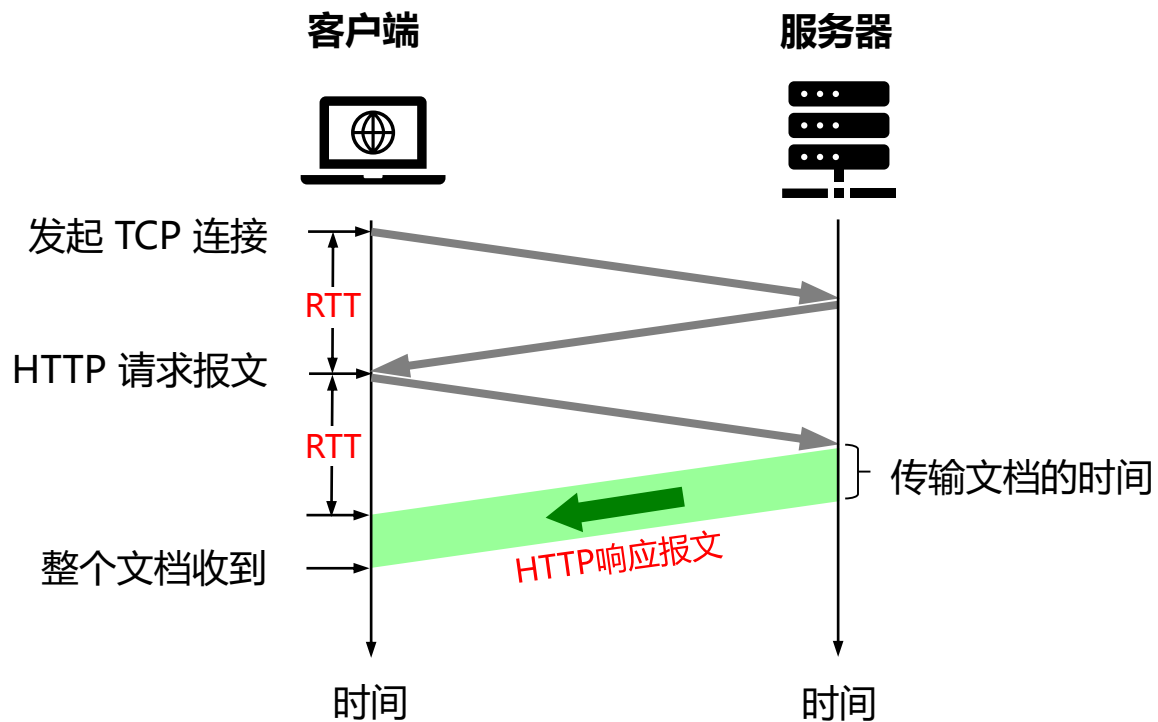
4.3 超文本传送协议 HTTP

- HTTP 的主要特点：
 - HTTP 使用了面向连接的 TCP 作为运输层协议，保证了数据的可靠传输。
 - HTTP 协议本身也是无连接的。
 - HTTP 是无状态的 (stateless)，简化了服务器的设计，使服务器更容易支持大量并发的 HTTP 请求。

4. 万维网 WWW

4.3 超文本传送协议 HTTP

- 请求一个万维网文档所需的时间:



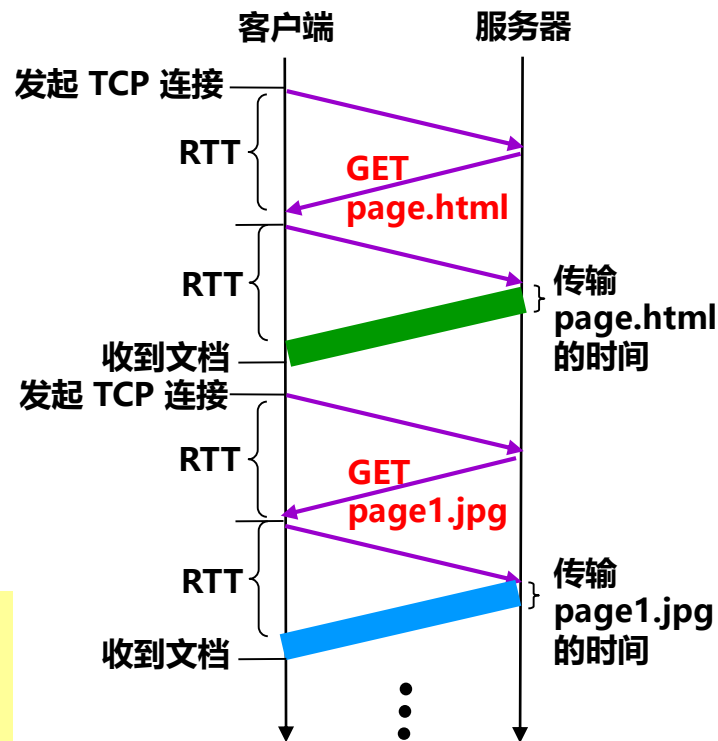
4. 万维网 WWW

4.3 超文本传送协议 HTTP

- 协议 HTTP/1.0 的主要缺点
 - 每请求一个文档就要有两倍 RTT 的开销。
 - 客户和服务器每一次建立新的 TCP 连接都要分配缓存和变量。
 - 这种非持续连接使服务器的负担很重。

假设一个主页
page.html 上有 10
个链接的图片：
page1.jpg,
page2.jpg,
page3.jpg,
...

所需的时间 \geq
22 RTT
+ 11 个文档的传输时间



4. 万维网 WWW

4.3 超文本传送协议 HTTP

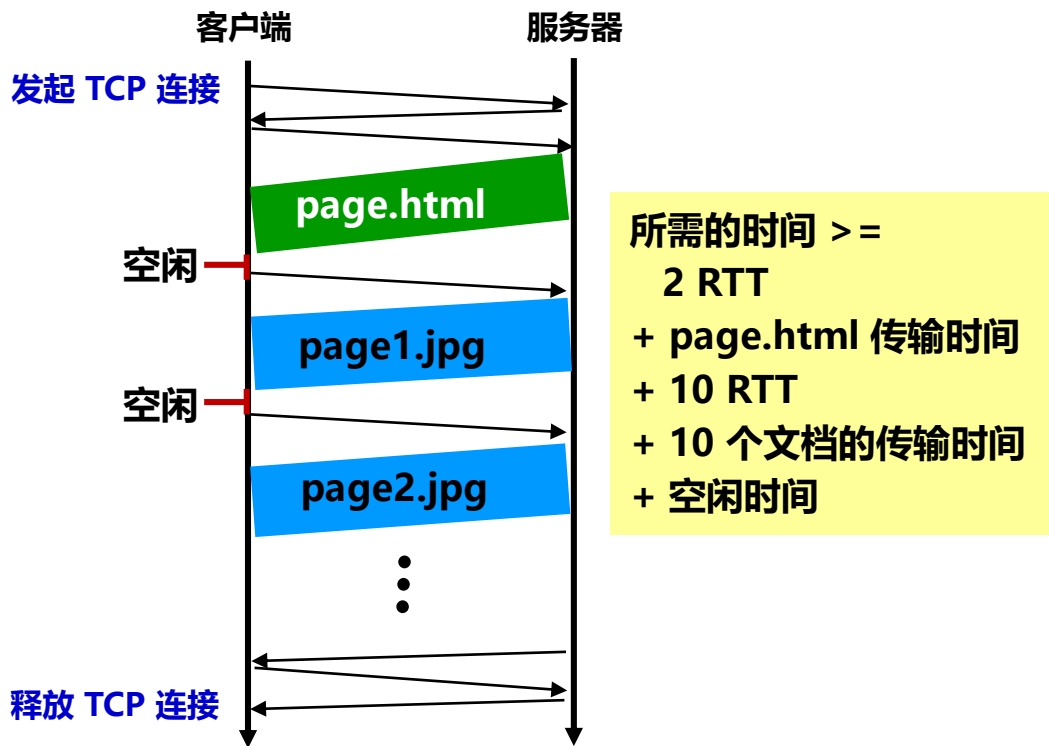
- 协议 HTTP/1.1 使用持续连接
 - 持续连接 (persistent connection) :
 - 服务器在发送响应后仍然在一段时间内保持这条连接 (不释放), 使同一个客户 (浏览器) 和该服务器可以继续在这条连接上传送后续的 HTTP 请求报文和响应报文。
 - 只要文档都在同一个服务器上, 就可以继续使用该 TCP 连接。
 - 两种工作方式:
 - 非流水线方式 (without pipelining)
 - 流水线方式 (with pipelining)。

4. 万维网 WWW

4.3 超文本传送协议 HTTP

持续连接：非流水线方式

- 客户在收到前一个响应之后才能发出下一个请求。
- 缺点：TCP 连接空闲状态。

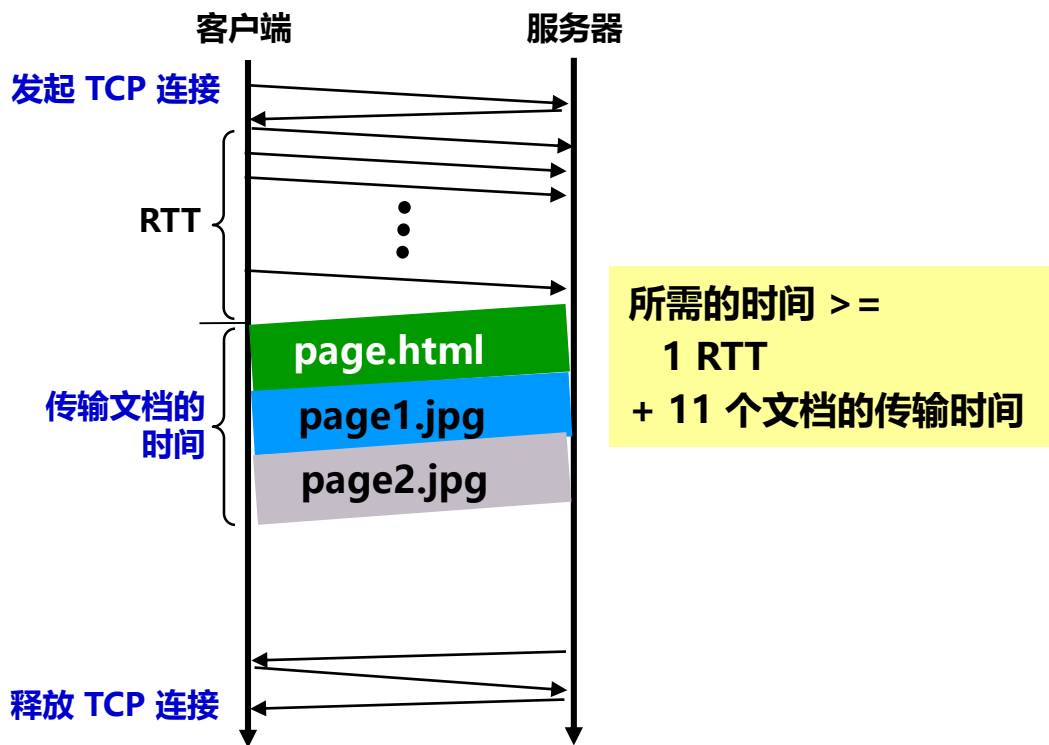


4. 万维网 WWW

4.3 超文本传送协议 HTTP

- 客户在收到响应报文之前就能够接着发送新的请求报文。
- 连续的多个请求报文到达服务器后，服务器就可连续发回响应报文。
- 下载效率提高。

持续连接：流水线方式



4. 万维网 WWW

4.3 超文本传送协议 HTTP

□ 协议 HTTP/2

- 是协议 HTTP/1.1 的升级版本。
- 服务器可以并行发回响应（使用同一个 TCP 连接）。
- 允许客户复用 TCP 连接进行多个请求。
- 把所有的报文都划分为许多较小的二进制编码的帧，并采用了新的压缩算法，不发送重复的首部字段，大大减小了首部的开销，提高了传输效率。
- 向后兼容。

4. 万维网 WWW

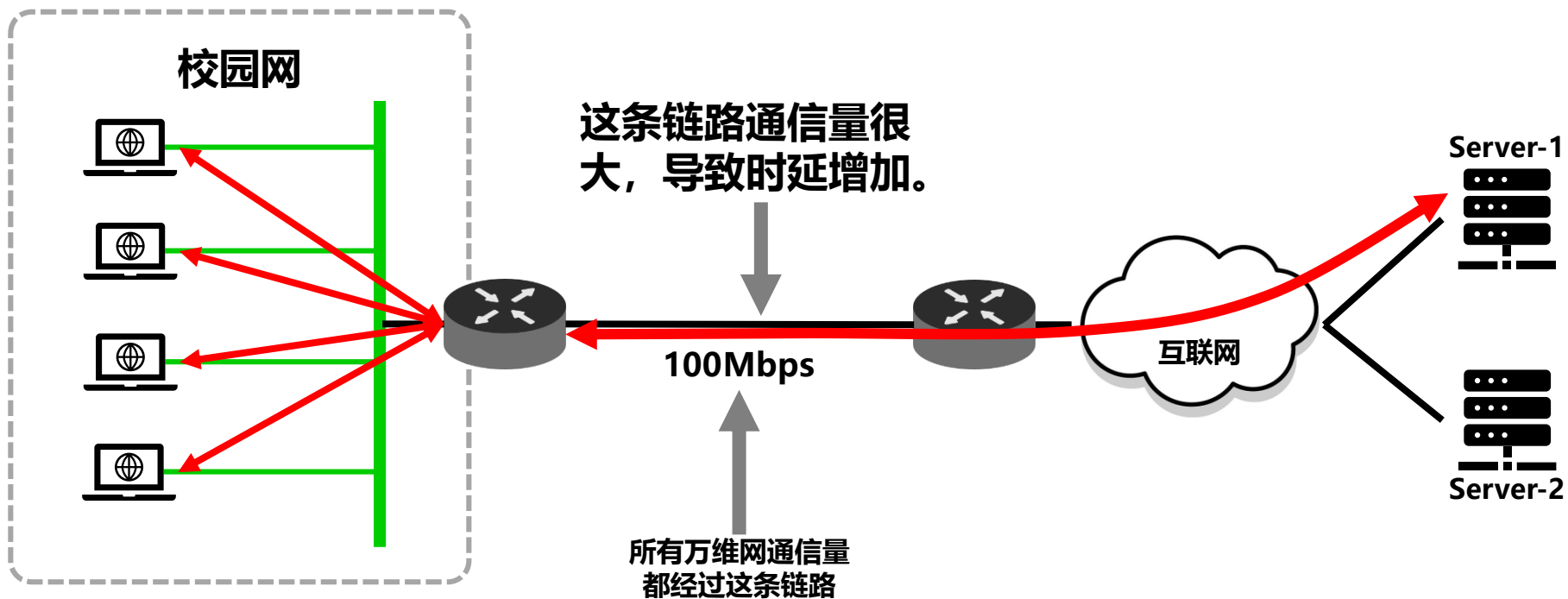
4.4 代理服务器

- 代理服务器 (proxy server) :
 - 代理服务器 (proxy server) 又称为万维网高速缓存 (Web cache), 它代表浏览器发出 HTTP 请求。
 - 使用高速缓存可减少访问互联网服务器的时延。
 - 万维网高速缓存把最近的一些请求和响应暂存在本地磁盘中。
 - 当与暂时存放的请求相同的新请求到达时, 万维网高速缓存就把暂存的响应发送出去, 而不需要按 URL 的地址再去互联网访问该资源。

4. 万维网 WWW

4.4 代理服务器

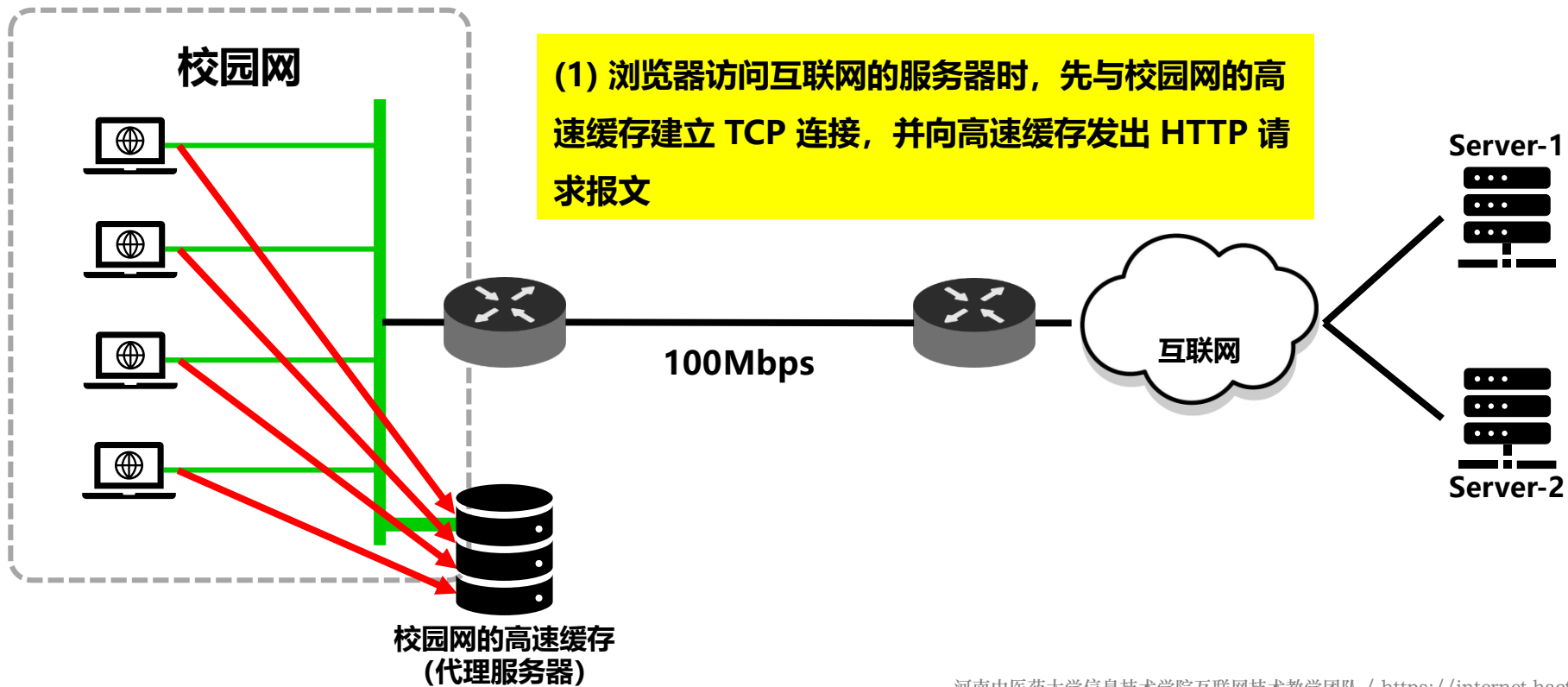
不使用高速缓存的情况



4. 万维网 WWW

4.4 代理服务器

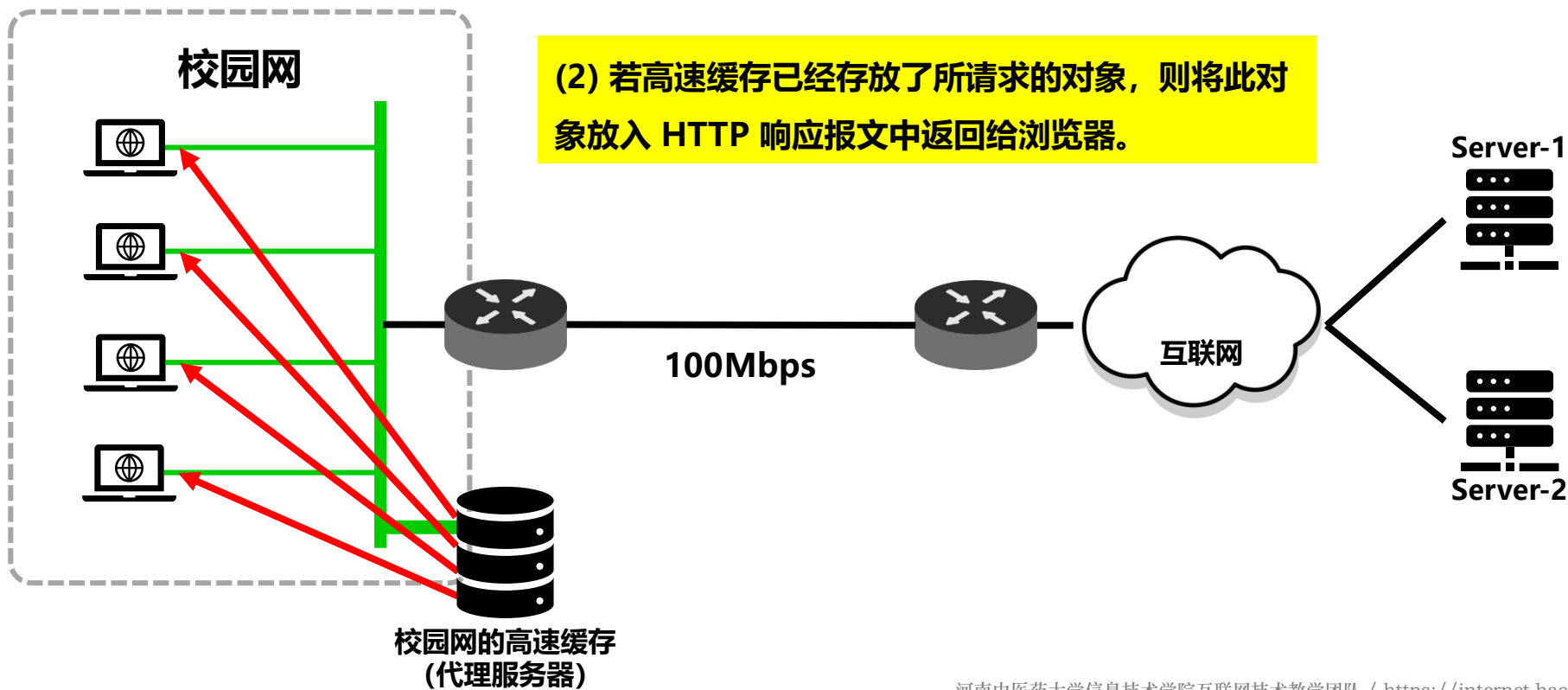
使用高速缓存的情况



4. 万维网 WWW

4.4 代理服务器

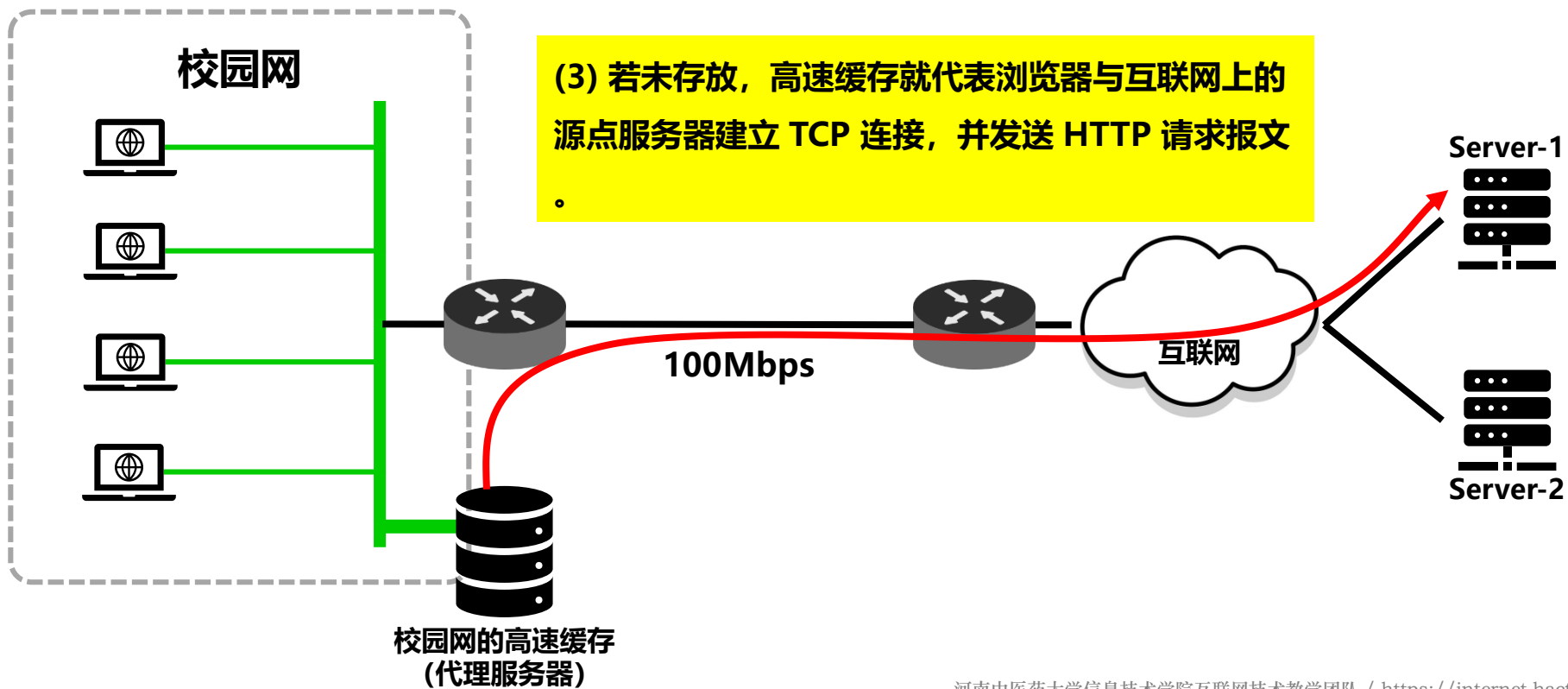
使用高速缓存的情况



4. 万维网 WWW

4.4 代理服务器

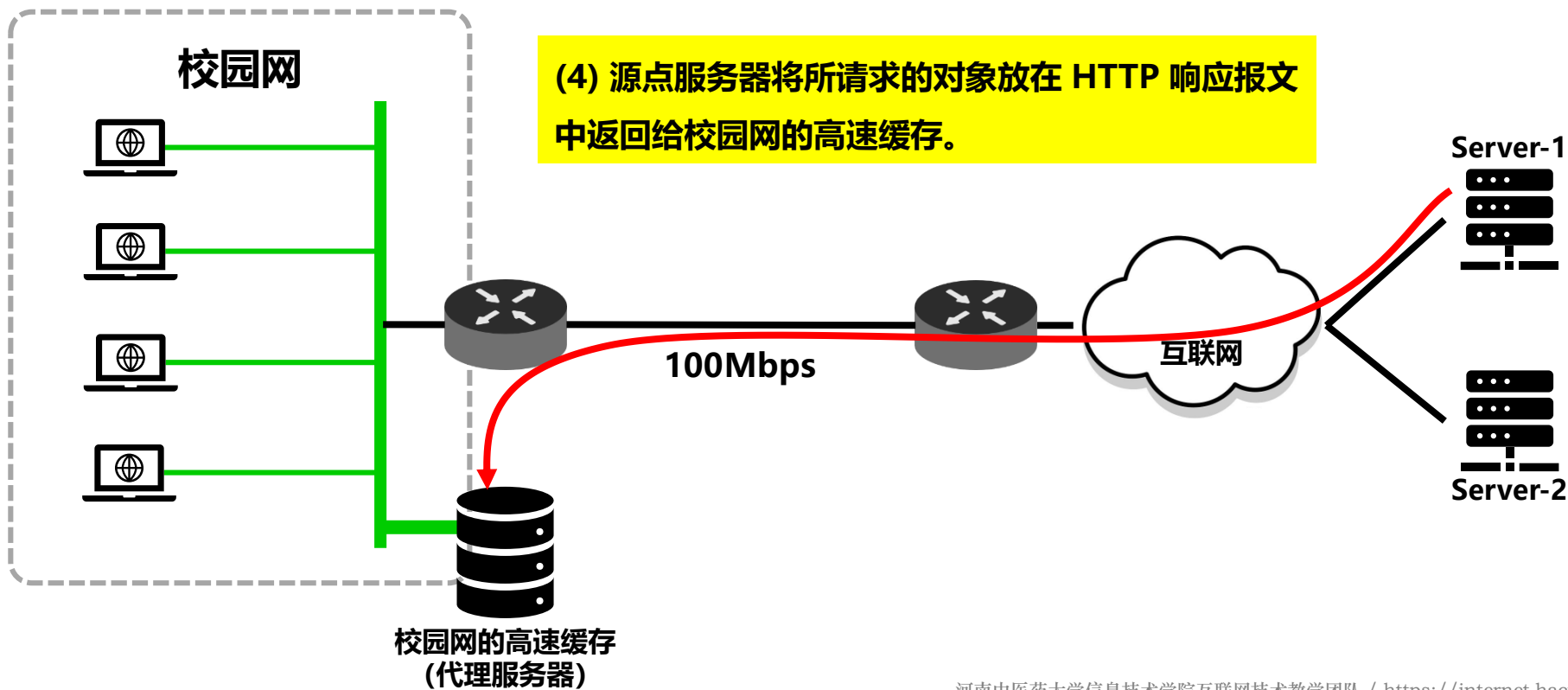
使用高速缓存的情况



4. 万维网 WWW

4.4 代理服务器

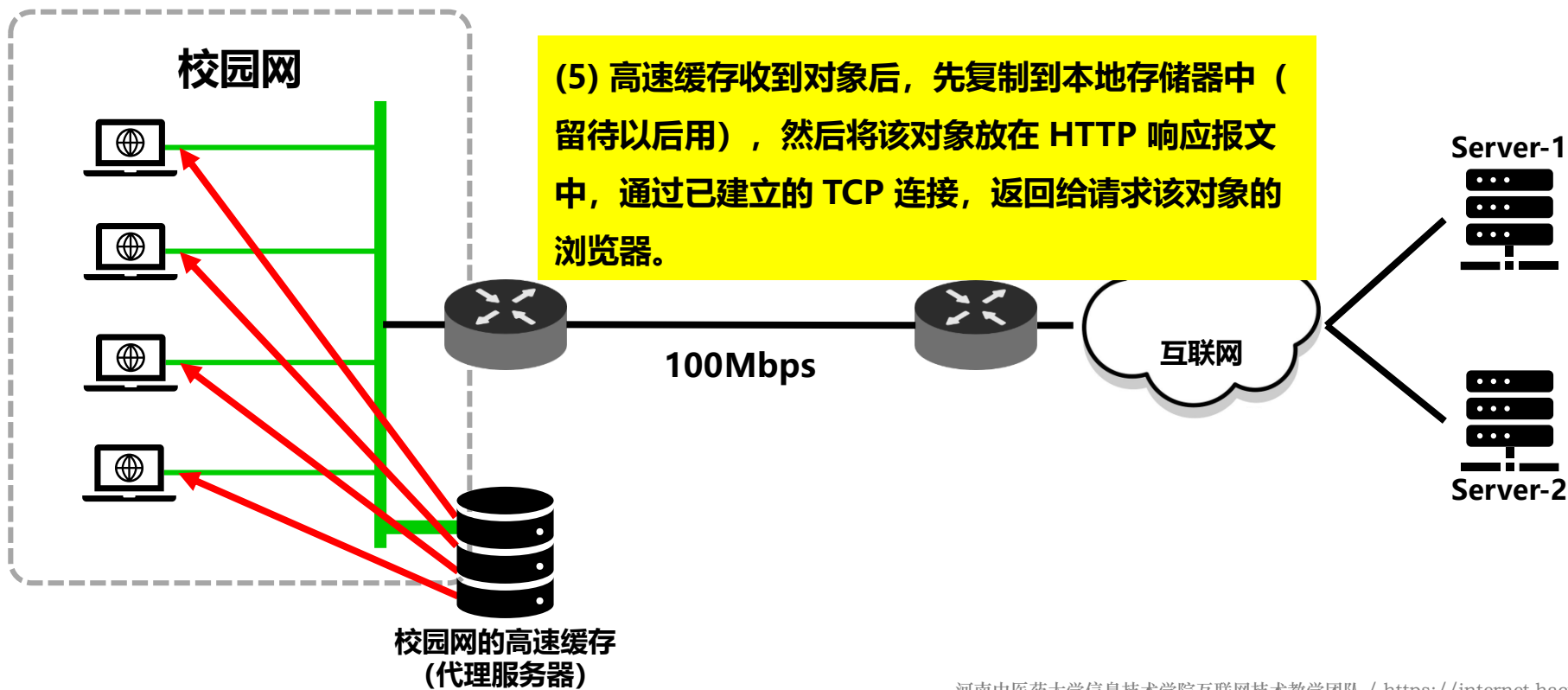
使用高速缓存的情况



4. 万维网 WWW

4.4 代理服务器

使用高速缓存的情况



4. 万维网 WWW

4.4 代理服务器

□ 代理服务器 (proxy server) :

- 浏览器访问互联网的服务器时, 要先与校园网的高速缓存建立 TCP 连接, 并向高速缓存发出 HTTP 请求报文
- 若高速缓存已经存放了所请求的对象, 则将此对象放入 HTTP 响应报文中返回给浏览器。
- 否则, 高速缓存就代表发出请求的用户浏览器, 与互联网上的源点服务器建立 TCP 连接, 并发送 HTTP 请求报文。
- 源点服务器将所请求的对象放在 HTTP 响应报文中返回给校园网的高速缓存。
- 高速缓存收到此对象后, 先复制在其本地存储器中 (为今后使用), 然后再将该对象放在 HTTP 响应报文中, 通过已建立的 TCP 连接, 返回给请求该对象的浏览器。

4. 万维网 WWW

4.5 HTTP 的报文结构

□ HTTP 的报文结构

■ 两类报文：

- 请求报文：从客户向服务器的请求。
- 响应报文：从服务器到客户的回答。
- 由于 HTTP 是面向正文的 (text-oriented)，因此报文中每一个字段的值都是一些 ASCII 码串，每个字段的长度都是不确定的。

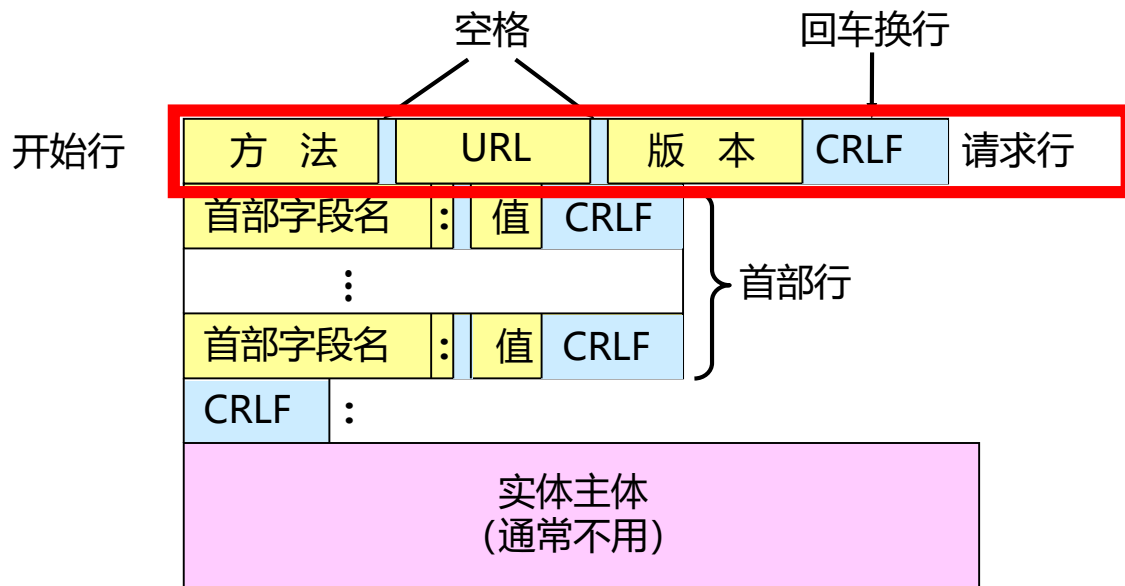
■ 三个组成部分：

- 开始行：用于区分是请求报文还是响应报文。
- 首部行：说明浏览器、服务器或报文主体的一些信息。可以有多行，也可以不使用。
- 实体主体：请求报文中一般不用，响应报文中也可能没有该字段。

4. 万维网 WWW

4.5 HTTP 的报文结构

- HTTP 的报文结构（请求报文）
 - 在请求报文中，开始行是请求行。

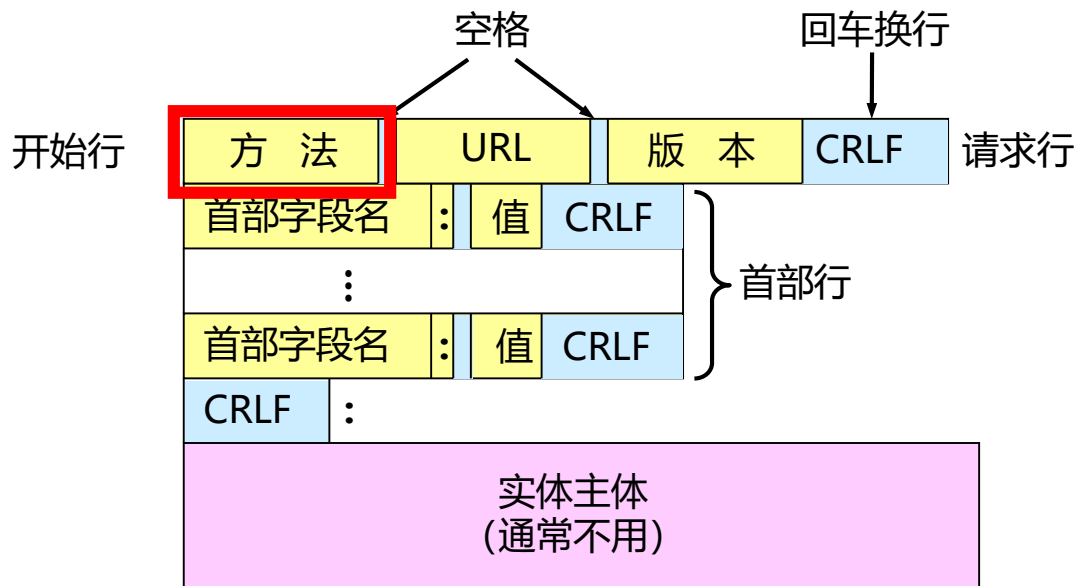


4. 万维网 WWW

4.5 HTTP 的报文结构

□ HTTP 的报文结构（请求报文）

- 方法：对所请求的对象进行的操作，实际上就是一些命令。
- 请求报文的类型是由它所采用的方法决定的。



4. 万维网 WWW

4.5 HTTP 的报文结构

□ HTTP 的报文结构（请求报文）

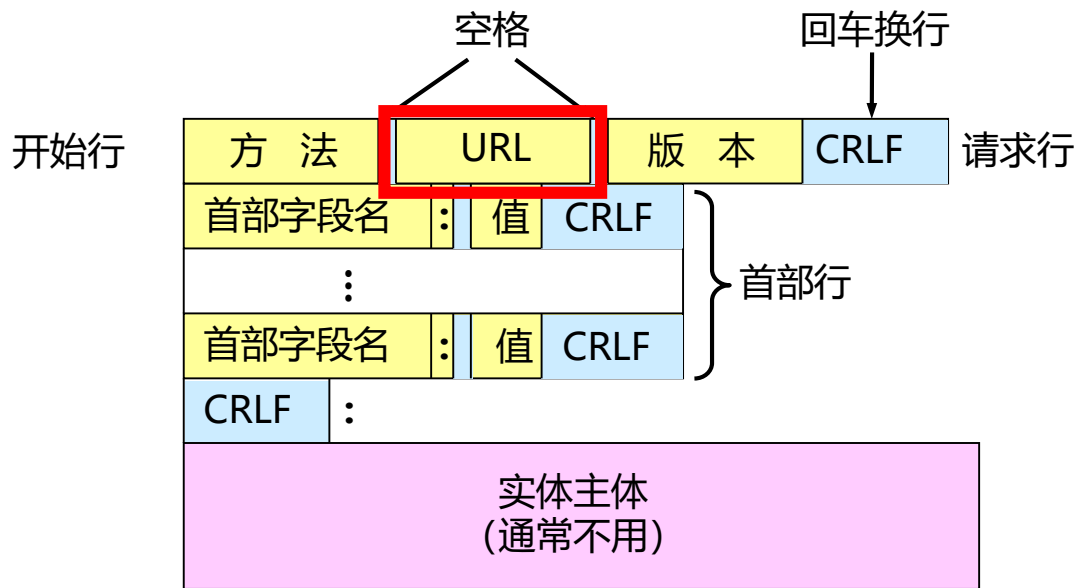
方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL 所标志的信息
HEAD	请求读取由 URL 所标志的信息的首部
POST	给服务器添加信息（例如，注释）
PUT	在指明的 URL 下存储一个文档
DELETE	删除指明的 URL 所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

- <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

4. 万维网 WWW

4.5 HTTP 的报文结构

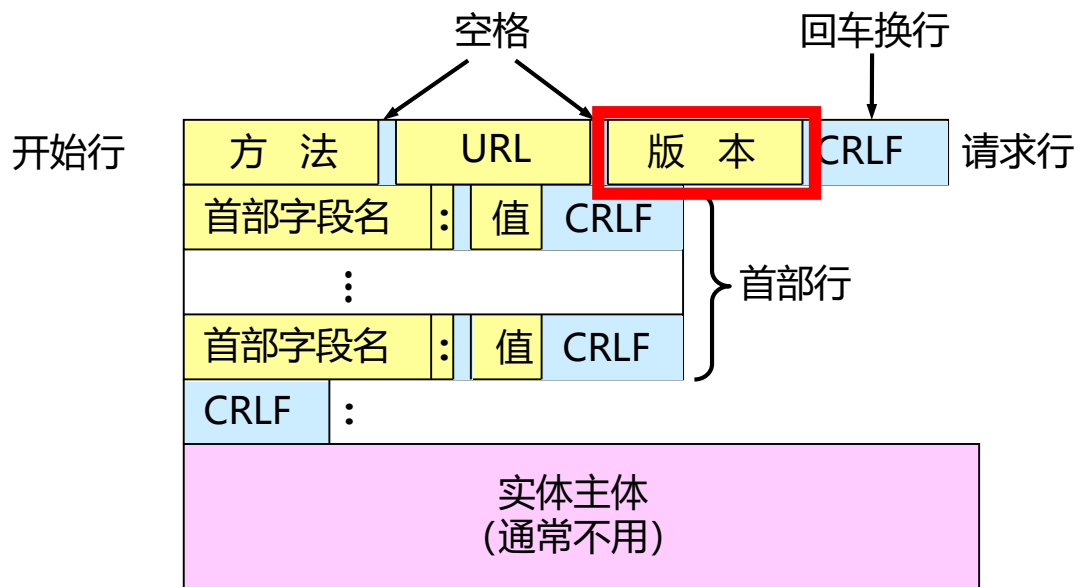
- HTTP 的报文结构（请求报文）
 - URL：所请求的资源的 URL。



4. 万维网 WWW

4.5 HTTP 的报文结构

- HTTP 的报文结构（请求报文）
 - 版本：HTTP 的版本。



状态	方...	域名	文件	发起者	类...	传输	大...
304	GET	inter...	/network/	docume...	ht...	已缓存	1...
200	GET	inter...	jquery.min.js	script	js	已缓存	0...
200	GET	inter...	itbase.js	script	js	已缓存	0...
200	GET	inter...	qswcmVisitUS.js	script	js	已缓存	0...
200	GET	inter...	qswcmSSOJS.js	script	js	已缓存	0...
200	GET	inter...	bootstrap.min.js	script	js	已缓存	0...
200	GET	inter...	ie10-viewport-bug-workarou	script	js	已缓存	6...
200	GET	inter...	rightSlider.js	script	js	已缓存	0...
200	GET	internet...	logo.png	img	png	已缓存	3...
⊗	GET	⊗ tongj...	matomo.js	qswcm...		NS_ERR...	0...
⊗	GET	⊗ sjk.x...	itbase.css	jquery...		NS_ERR...	0...
⊗	GET	⊗ sjk.x...	i_logodl.png	jquery...		NS_ERR...	0...
⊗	GET	⊗ sjk.x...	i_close.png	jquery...		NS_ERR...	0...
⊗	GET	⊗ sjk.x...	i_user.png	jquery...		NS_ERR...	0...
⊗	GET	⊗ sjk.x...	i_pwd.png	jquery...		NS_ERR...	0...
⊗	GET	⊗ sjk.x...	i_code.png	jquery...		NS_ERR...	0...
⊗	GET	⊗ admi...	GetValidateCode?time=1547	jquery...		NS_ERR...	0...
⊗	GET	⊗ admi...	GetLoginUser?callback=json	itbase.js...		NS_ERR...	0...
200	GET	inter...	UpdateVisitCount?callback=	jquery...	js...	308 字节	3...
200	GET	inter...	favicon.ico	Favicon...	ht...	已缓存	3...

消息头 Cookie 请求 响应 缓存 耗时 安全性

过滤消息头 拦截 重发

GET https://internet.hactcm.edu.cn/network/

状态 304 Not Modified ?

版本 HTTP/1.1

传输 3.99 kB (大小 17.26 kB)

Referrer 策略 strict-origin-when-cross-origin

响应头 (234 字节) 原始

- Accept-Ranges: bytes
- Connection: keep-alive
- Content-Security-Policy: upgrade-insecure-requests
- Date: Mon, 03 Apr 2023 01:13:21 GMT
- Etag: "9f9d879be55fd91:0"
- SS: s
- X-Frame-Options: SAMEORIGIN

请求头 (632 字节) 原始

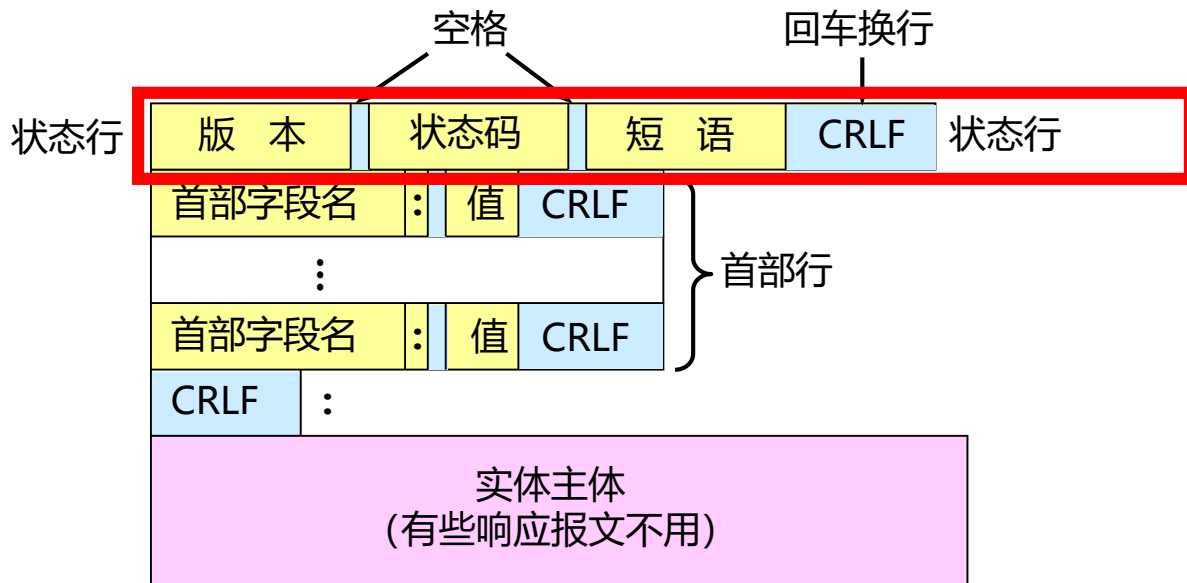
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
- Connection: keep-alive
- DNT: 1
- Host: internet.hactcm.edu.cn
- If-Modified-Since: Sun, 26 Mar 2023 13:19:34 GMT
- If-None-Match: W/"9f9d879be55fd91:0"
- Referer: https://internet.hactcm.edu.cn/
- Sec-Fetch-Dest: document
- Sec-Fetch-Mode: navigate
- Sec-Fetch-Site: same-origin
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0

4. 万维网 WWW

4.5 HTTP 的报文结构

□ HTTP 的报文结构（响应报文）

- 在响应报文中，开始行是状态行。
- 版本：HTTP 的版本。状态码：服务器操作完成的状态。短语：解释状态码。



4. 万维网 WWW

4.5 HTTP 的报文结构

□ HTTP 的报文结构（响应报文）

■ 状态码都是三位数字。

- 1xx 表示通知信息的，如请求收到了或正在进行处理。
- 2xx 表示成功，如接受或知道了。
- 3xx 表示重定向，表示要完成请求还必须采取进一步的行动。
- 4xx 表示客户的差错，如请求中有错误的语法或不能完成。
- 5xx 表示服务器的差错，如服务器失效无法完成请求。

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.1.1>


```

Status-Code =
  "100" ; Section 10.1.1: Continue
  | "101" ; Section 10.1.2: Switching Protocols
  | "200" ; Section 10.2.1: OK
  | "201" ; Section 10.2.2: Created
  | "202" ; Section 10.2.3: Accepted
  | "203" ; Section 10.2.4: Non-Authoritative Information
  | "204" ; Section 10.2.5: No Content
  | "205" ; Section 10.2.6: Reset Content
  | "206" ; Section 10.2.7: Partial Content
  | "300" ; Section 10.3.1: Multiple Choices
  | "301" ; Section 10.3.2: Moved Permanently
  | "302" ; Section 10.3.3: Found
  | "303" ; Section 10.3.4: See Other
  | "304" ; Section 10.3.5: Not Modified
  | "305" ; Section 10.3.6: Use Proxy
  | "307" ; Section 10.3.8: Temporary Redirect
  | "400" ; Section 10.4.1: Bad Request
  | "401" ; Section 10.4.2: Unauthorized
  | "402" ; Section 10.4.3: Payment Required
  | "403" ; Section 10.4.4: Forbidden
  | "404" ; Section 10.4.5: Not Found
  | "405" ; Section 10.4.6: Method Not Allowed
  | "406" ; Section 10.4.7: Not Acceptable

  | "407" ; Section 10.4.8: Proxy Authentication Required
  | "408" ; Section 10.4.9: Request Time-out
  | "409" ; Section 10.4.10: Conflict
  | "410" ; Section 10.4.11: Gone
  | "411" ; Section 10.4.12: Length Required
  | "412" ; Section 10.4.13: Precondition Failed
  | "413" ; Section 10.4.14: Request Entity Too Large
  | "414" ; Section 10.4.15: Request-URI Too Large
  | "415" ; Section 10.4.16: Unsupported Media Type
  | "416" ; Section 10.4.17: Requested range not satisfiable
  | "417" ; Section 10.4.18: Expectation Failed
  | "500" ; Section 10.5.1: Internal Server Error
  | "501" ; Section 10.5.2: Not Implemented
  | "502" ; Section 10.5.3: Bad Gateway
  | "503" ; Section 10.5.4: Service Unavailable
  | "504" ; Section 10.5.5: Gateway Time-out
  | "505" ; Section 10.5.6: HTTP Version not supported
  | extension-code

```

extension-code = 3DIGIT

Reason-Phrase = *<TEXT, excluding CR, LF>

状态	方...	域名	文件	发起者	类...	传输	大...
304	GET	inter...	/network/	docume...	ht...	已缓存	1...
200	GET	inter...	jquery.min.js	script	js	已缓存	0...
200	GET	inter...	itbase.js	script	js	已缓存	0...
200	GET	inter...	qswcmVisitUS.js	script	js	已缓存	0...
200	GET	inter...	qswcmSSOJS.js	script	js	已缓存	0...
200	GET	inter...	bootstrap.min.js	script	js	已缓存	0...
200	GET	inter...	ie10-viewport-bug-workarou	script	js	已缓存	6...
200	GET	inter...	rightSlider.js	script	js	已缓存	0...
200	GET	internet...	logo.png	img	png	已缓存	3...
⊗	GET	! tongj...	matomo.js	qswcm...		NS_ERR...	0...
⊗	GET	! sjk.x...	itbase.css	jquery...		NS_ERR...	0...
⊗	GET	! sjk.x...	i_logodl.png	jquery...		NS_ERR...	0...
⊗	GET	! sjk.x...	i_close.png	jquery...		NS_ERR...	0...
⊗	GET	! sjk.x...	i_user.png	jquery...		NS_ERR...	0...
⊗	GET	! sjk.x...	i_pwd.png	jquery...		NS_ERR...	0...
⊗	GET	! sjk.x...	i_code.png	jquery...		NS_ERR...	0...
⊗	GET	! admi...	GetValidateCode?time=1547	jquery...		NS_ERR...	0...
⊗	GET	! admi...	GetLoginUser?callback=json	itbase.js...		NS_ERR...	0...
200	GET	inter...	UpdateVisitCount?callback=	jquery...	js...	308 字节	3...
200	GET	inter...	favicon.ico	Favicon...	ht...	已缓存	3...

消息头 Cookie 请求 响应 缓存 耗时 安全性

拦截 重发

GET https://internet.hactcm.edu.cn/network/

状态 304 Not Modified

版本 HTTP/1.1

传输 3.99 kB (大小 17.26 kB)

Referer 策略 strict-origin-when-cross-origin

▼ 响应头 (234 字节) 原始

- Accept-Ranges: bytes
- Connection: keep-alive
- Content-Security-Policy: upgrade-insecure-requests
- Date: Mon, 03 Apr 2023 01:13:21 GMT
- Etag: "9f9d879be55fd91:0"
- SS: s
- X-Frame-Options: SAMEORIGIN
- XX: p

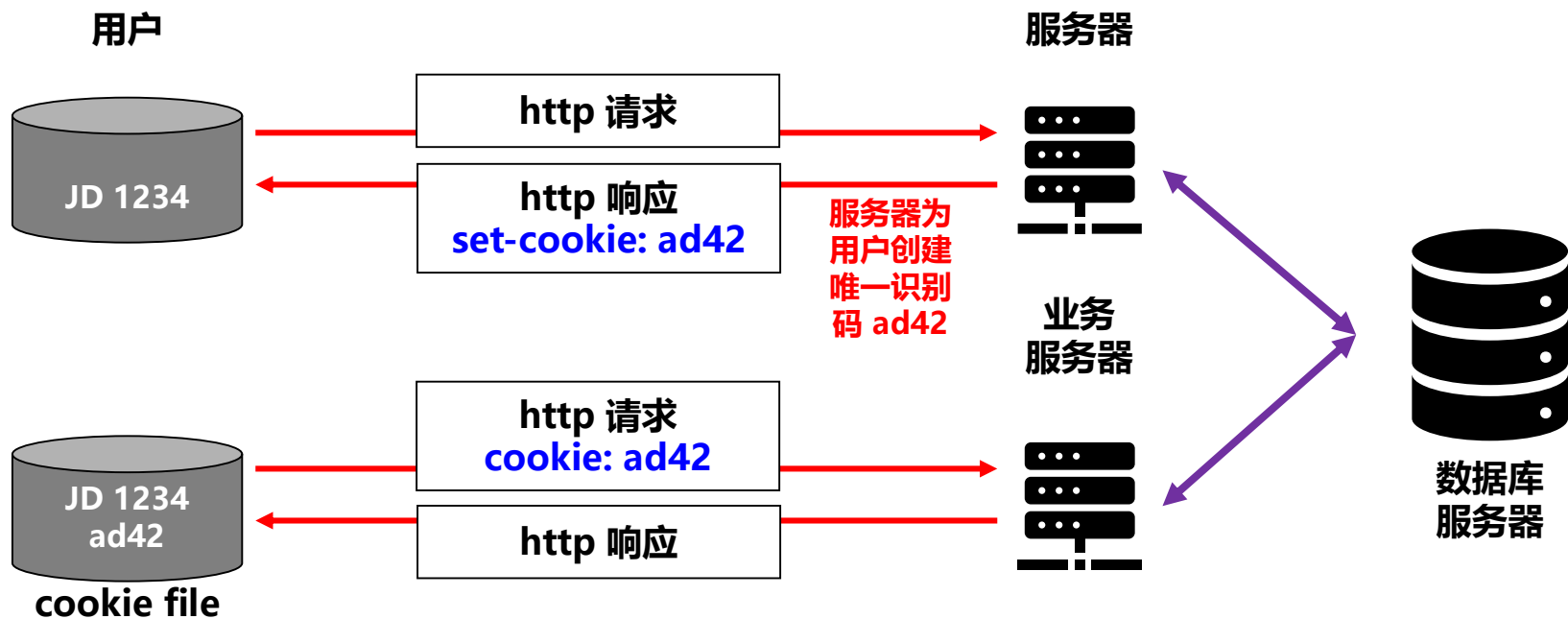
请求头 (632 字节) 原始

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
- Connection: keep-alive
- DNT: 1
- Host: internet.hactcm.edu.cn
- If-Modified-Since: Sun, 26 Mar 2023 13:19:34 GMT
- If-None-Match: W/"9f9d879be55fd91:0"
- Referer: https://internet.hactcm.edu.cn/
- Sec-Fetch-Dest: document
- Sec-Fetch-Mode: navigate
- Sec-Fetch-Site: same-origin
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0

4. 万维网 WWW

4.6 在服务器上存放用户的信息

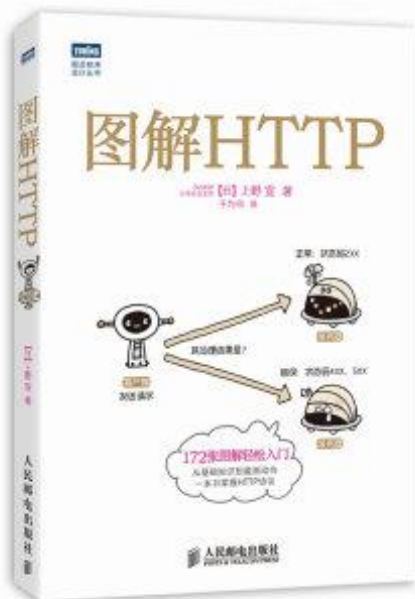
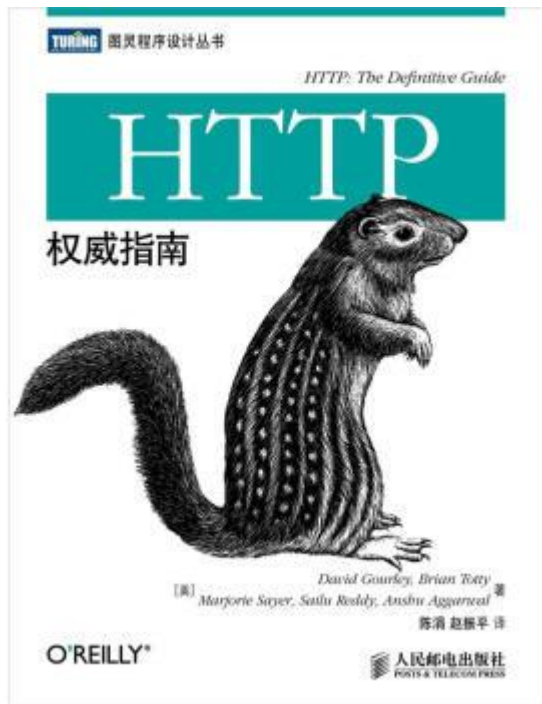
- 万维网使用 Cookie 跟踪在 HTTP 服务器和客户之间传递的状态信息。



Network tab showing request log with columns: 状态, 方法, 域名, 文件, 发起者, 类型, 传输, 大小. Includes various GET requests to sso.jd.com and other domains.

Details panel for a selected request. Includes: 消息头 (Cookie, 请求, 响应, 耗时, 栈跟踪, 安全性), 过滤器, 状态 (200 OK), 版本 (HTTP/1.1), 传输 (896 字节), Referrer, 响应头 (870 字节), 请求头 (11.105 KB), and Accept headers.

4. 万维网 WWW



5. 电子邮件

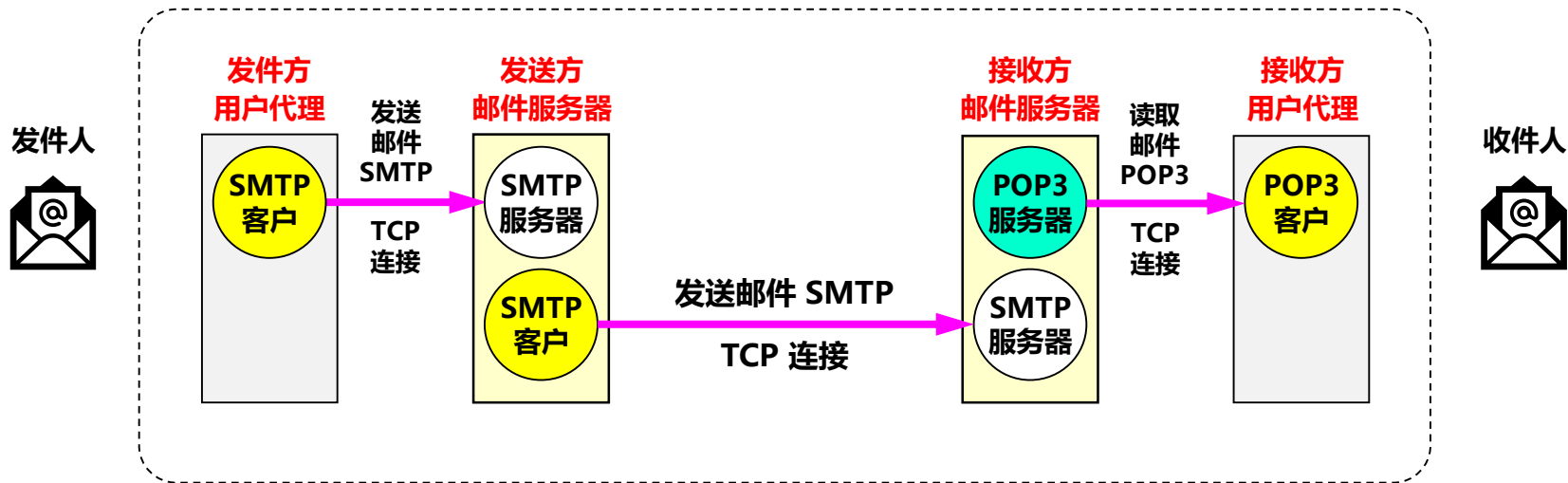
- 电子邮件 (e-mail): 指使用电子设备交换的邮件及其方法。
- 优点:
 - 使用方便, 传递迅速, 费用低廉。
 - 可以传送多种类型的信息 (包括: 文字信息, 声音和图像等) 。
- 重要标准:
 - 简单邮件发送协议: SMTP
 - 互联网文本报文格式
 - 通用互联网邮件扩充 MIME
 - 邮件读取协议: POP3 和 IMAP

5. 电子邮件

5.1 电子邮件概述

电子邮件系统的组成：三个主要构件

用户代理，邮件服务器，以及邮件发送和读取协议。



5. 电子邮件

5.1 电子邮件概述

- 用户代理 UA (User Agent):
 - 用户代理 UA 就是用户与电子邮件系统的接口，是电子邮件客户端软件。
 - 用户代理的功能：撰写、显示、处理和通信。
- 邮件服务器 (Mail Server)
 - 又被称为邮件传输代理。
 - 功能：发送和接收邮件，同时还要向发信人报告邮件传送的情况。
 - 按照客户服务器方式工作。

5. 电子邮件

5.1 电子邮件概述

- 邮件发送和读取协议
 - 邮件发送和读取使用不同的协议。
 - 简单邮件发送协议 SMTP：用于在用户代理向邮件服务器 或 邮件服务器之间发送邮件。
 - 邮局协议 POP3：用于用户代理从邮件服务器读取邮件。

- 补充说明：
 - 邮件服务器必须能够同时充当客户和服务。
 - SMTP 和 POP3（或 IMAP）都使用 TCP 连接可靠地传送邮件。

5. 电子邮件

5.1 电子邮件概述

- 发送和接收电子邮件的几个重要步骤：
 - 发件人调用用户代理撰写和编辑要发送的邮件。
 - 发件人的用户代理把邮件用 SMTP 协议发给发送方邮件服务器。
 - SMTP 服务器把邮件临时存放在邮件缓存队列中，等待发送。
 - 发送方邮件服务器的 SMTP 客户与接收方邮件服务器的 SMTP 服务器建立 TCP 连接，然后就把邮件缓存队列中的邮件依次发送出去。
 - 运行在接收方邮件服务器中的 SMTP 服务器进程收到邮件后，把邮件放入收件人的用户邮箱中，等待收件人进行读取。
 - 收件人在打算收信时，就运行用户代理，使用 POP3 或 IMAP 协议读取邮件。

5. 电子邮件

5.1 电子邮件概述

- 电子邮件的组成：
 - 电子邮件由信封 (envelope) 和内容 (content) 两部分组成。
 - 电子邮件的传输程序根据邮件信封上的信息来传送邮件。
 - 用户在从自己的邮箱中读取邮件时才能见到邮件的内容。

5. 电子邮件

5.1 电子邮件概述

□ 电子邮件地址的格式：

- TCP/IP 体系的电子邮件系统规定电子邮件地址的格式如下：

收件人邮箱名@邮箱所在主机的域名

- 符号“@”读作“at”，表示“在”的意思。

5. 电子邮件

5.2 简单邮件传送协议 SMTP

- SMTP 规定了在两个相互通信的 SMTP 进程之间交换信息的方法。
- SMTP 使用客户服务器方式。
 - 负责发送邮件的 SMTP 进程就是 SMTP 客户。
 - 负责接收邮件的 SMTP 进程就是 SMTP 服务器。
- SMTP 是一个基于文本的（即 ASCII 码）的协议。
- SMTP 客户与服务器之间采用命令-响应方式进行交互。
- SMTP 基于 TCP 实现客户与服务器的通信。

5. 电子邮件

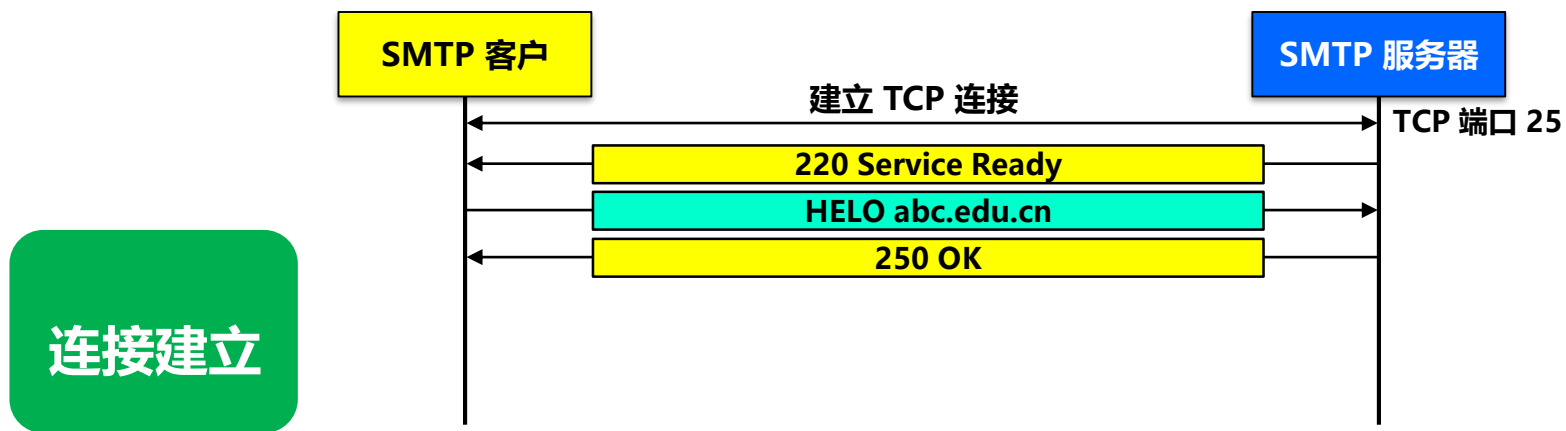
5.2 简单邮件传送协议 SMTP

- SMTP 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。

5. 电子邮件

5.2 简单邮件传送协议 SMTP

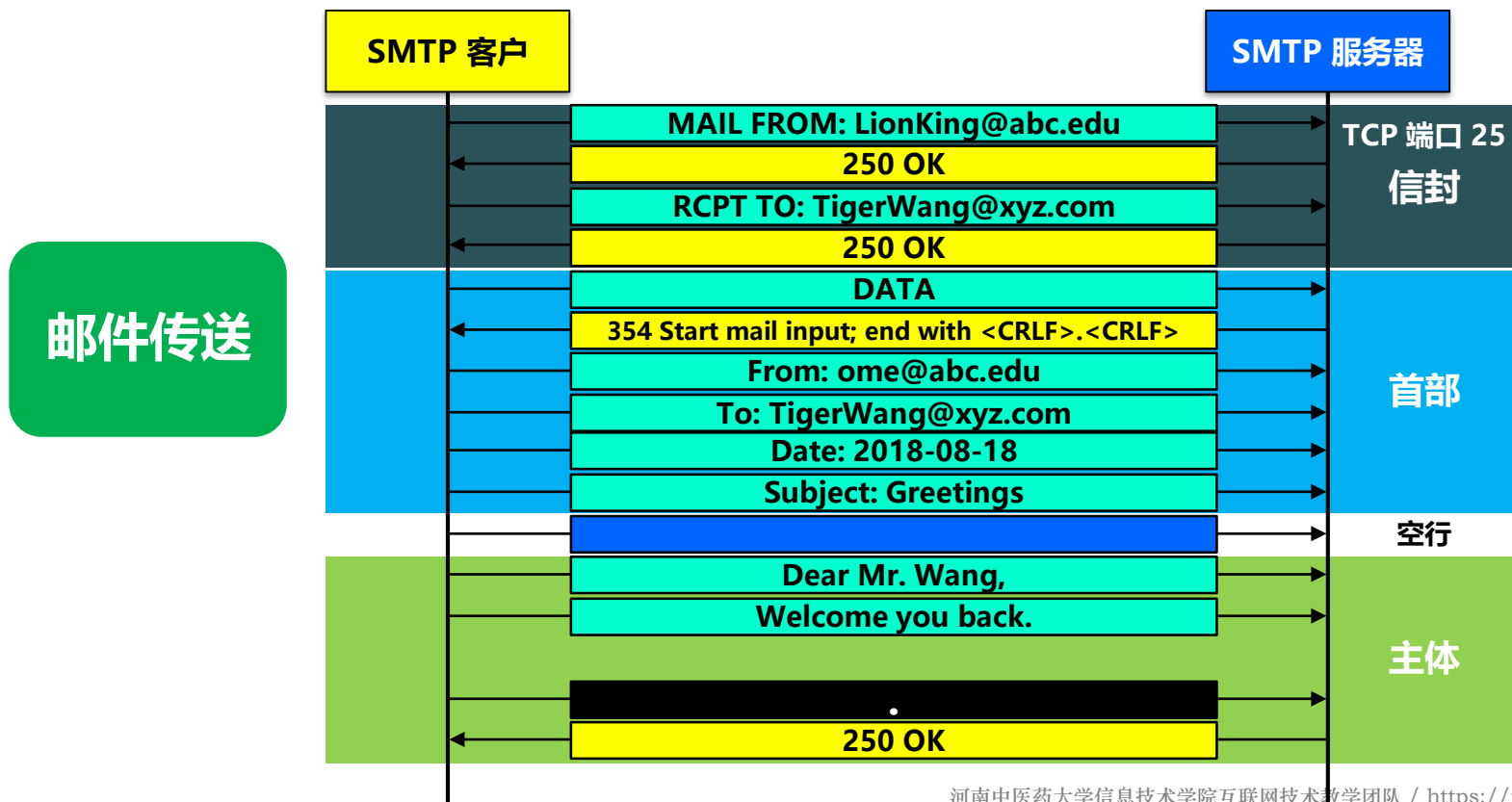
- SMTP 通信的三个阶段：



5. 电子邮件

5.2 简单邮件传送协议 SMTP

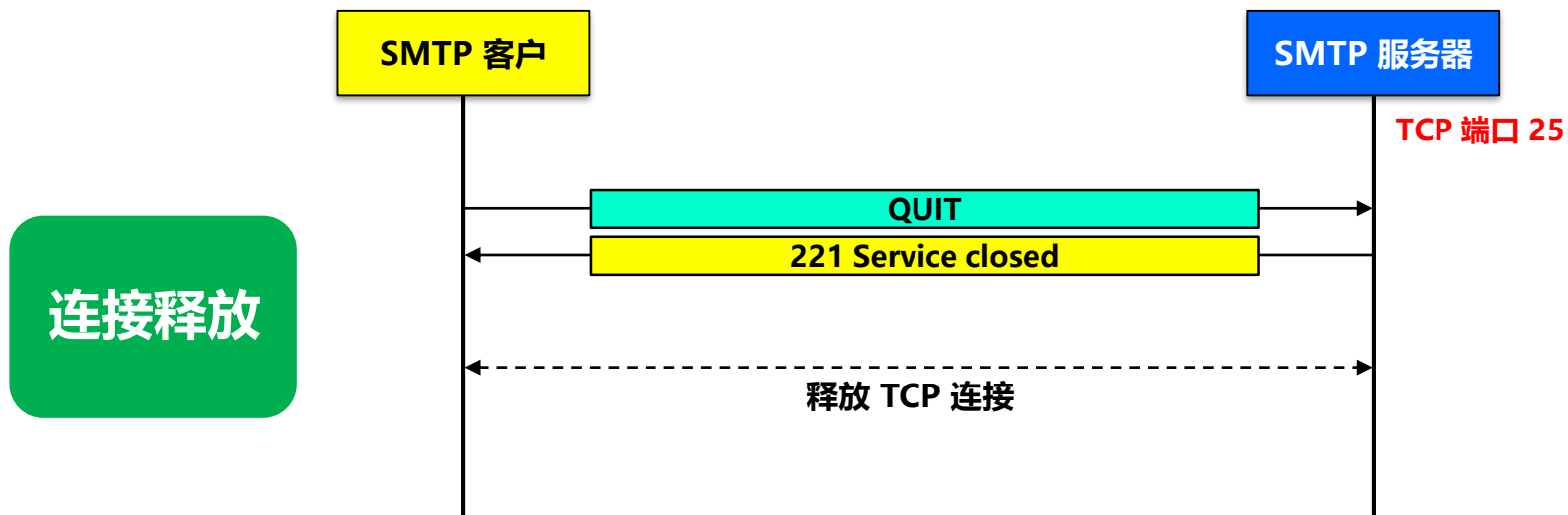
- SMTP 通信的三个阶段：



5. 电子邮件

5.2 简单邮件传送协议 SMTP

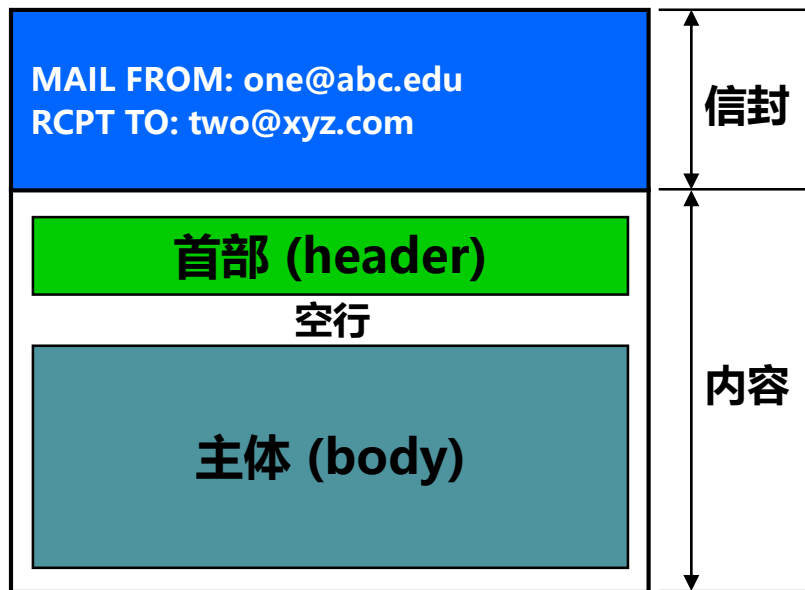
- SMTP 通信的三个阶段：



5. 电子邮件

5.3 电子邮件的信息格式

- 一个电子邮件分为**信封**和**内容**两大部分。
- RFC 5322 只规定了邮件内容中的首部 (header) 格式, 而对邮件的主体 (body) 部分则让用户自由撰写。
 - 用户写好首部
 - 邮件系统将自动地将信封所需的信息提取出来并写在信封上。用户不需要填写电子邮件信封上的信息。
 - 邮件内容首部
 - 包括一些关键字, 后面加上冒号。
 - 最重要的关键字是: To 和 Subject。



5. 电子邮件

5.3 电子邮件的信息格式

- 邮件内容的首部：
 - To: 后面填入一个或多个收件人的电子邮件地址。
 - 用户只需打开地址簿，点击收件人名字，收件人的电子邮件地址就会自动地填入到合适的位置上。
 - Subject: 是邮件的主题。
 - 它反映了邮件的主要内容，便于用户查找邮件。
 - Cc: 表示应给某某人发送一个邮件副本。
 - From 和 Date: 表示发信人的电子邮件地址和发信日期。
 - Reply-To: 是对方回信所用的地址。

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

- 两个常用的邮件读取协议：



- POP3：邮局协议 (Post Office Protocol) 第3个版本
- IMAP：网际报文存取协议 (Internet Message Access Protocol)

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

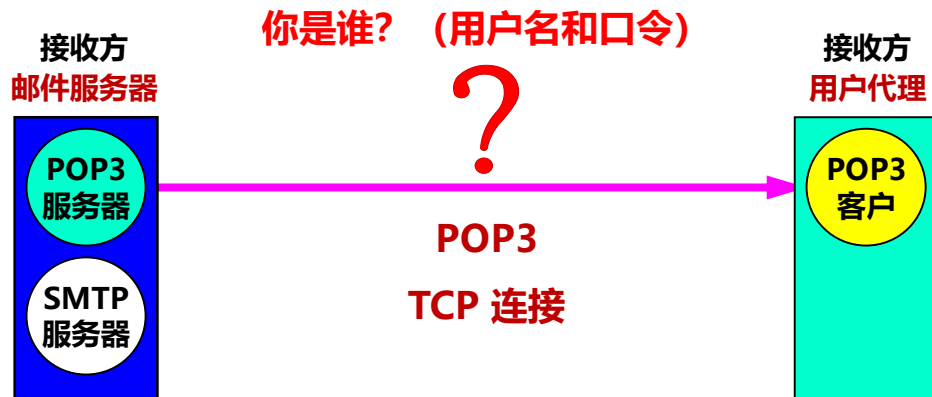
□ POP3:

- 邮局协议 POP 是一个非常简单、但功能有限的邮件读取协议，现在使用的是它的第三个版本 POP3。
- POP 使用客户/服务器的工作方式。
- 在接收邮件的用户端必须运行 POP 客户程序，而在用户所连接的 ISP 的邮件服务器中则运行 POP 服务器程序。

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

□ POP3:



- POP3 支持用户鉴别。
- POP3 服务器删除被用户读取了的邮件。

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

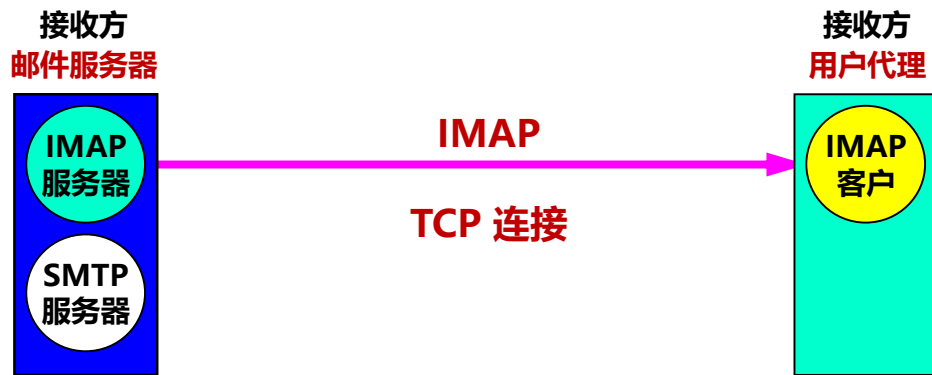
□ IMAP:

- IMAP 是按客户服务器方式工作，现在较新的是版本4，即IMAP4。
- 用户端可以操作 ISP 的邮件服务器的邮箱，就像在本地操纵一样。
- IMAP 是一个联机协议。
 - 当用户端 IMAP 客户程序打开 IMAP 服务器的邮箱时，用户就可看到邮件的首部。
 - 若用户需要打开某个邮件，则该邮件才传到用户的计算机上。

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

□ IMAP:



- IMAP 使用客户-服务器方式。
- IMAP 基于 TCP 实现客户与服务器的通信。
- IMAP是一个联机协议。

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

□ IMAP:

■ IMAP 的特点:

- 连接后只下载邮件首部（部分下载）。
- 用户直接在 IMAP 服务器上创建和管理文件夹。
- 用户可以搜索邮件内容。
- 用户可以在不同的地方使用不同的计算机随时上网阅读和处理自己的邮件。
- 允许收信人只读取邮件中的某一个部分。

■ IMAP 的缺点:

- 要想查阅邮件，必须先联网。

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

IMAP 与 POP3 比较

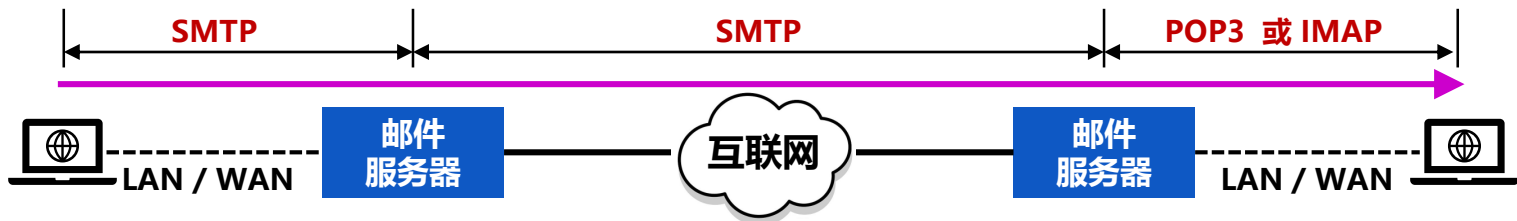
操作位置	操作内容	IMAP	POP3
收件箱	阅读、标记、移动、删除邮件等	客户端与邮箱更新同步	仅在客户端内
发件箱	保存到已发送	客户端与邮箱更新同步	仅在客户端内
创建文件夹	新建自定义的文件夹	客户端与邮箱更新同步	仅在客户端内
草稿	保存草稿	客户端与邮箱更新同步	仅在客户端内
垃圾文件夹	接收并移入垃圾文件夹的邮件	支持	不支持
广告邮件	接收并移入广告邮件夹的邮	支持	不支持

5. 电子邮件

5.4 邮件读取协议 POP3 和 IMAP

□ POP、IMAP 和 SMTP:

- 邮件读取协议 POP 或 IMAP 与邮件传送协议 SMTP 完全不同。
- 发信人的用户代理向源邮件服务器发送邮件，以及源邮件服务器向目的邮件服务器发送邮件，都是使用 SMTP 协议。
- POP 协议或 IMAP 协议则是用户从目的邮件服务器上读取邮件所使用的协议。



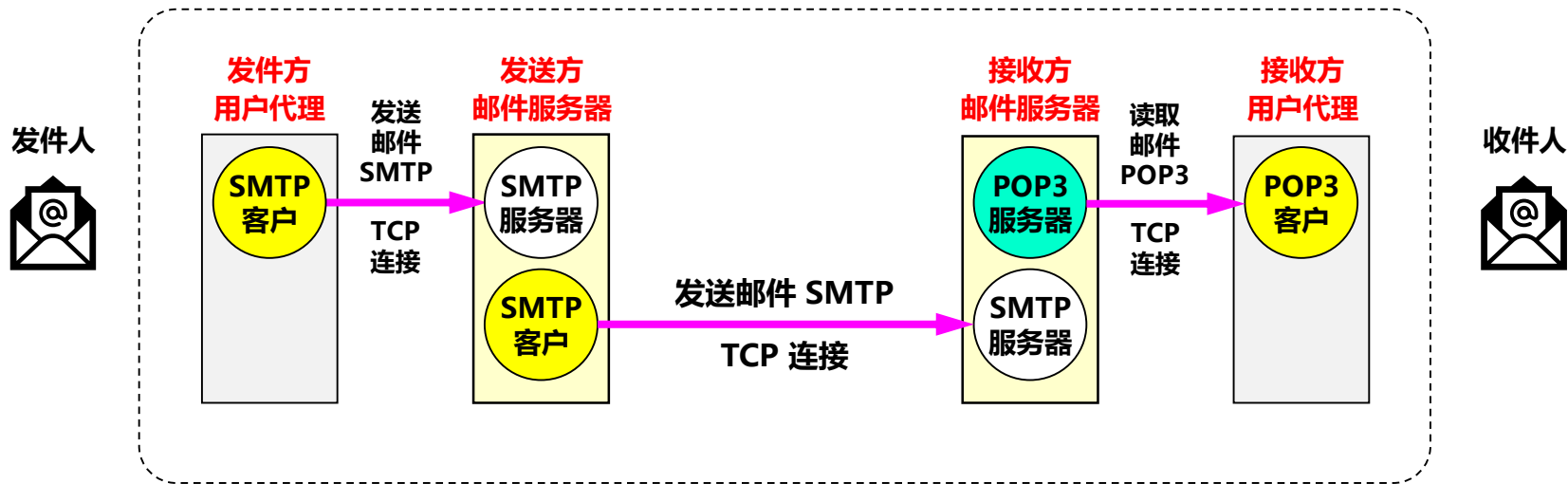
5. 电子邮件

5.5 基于万维网的电子邮件

- 用户代理 (UA) 的缺点：
 - 必须在计算机中安装用户代理软件。
 - 收发邮件不方便。
- 万维网电子邮件优点：
 - 不需要在计算机中再安装用户代理软件。
 - 计算机能联网，就能非常方便地收发电子邮件。
 - 界面非常友好。

5. 电子邮件

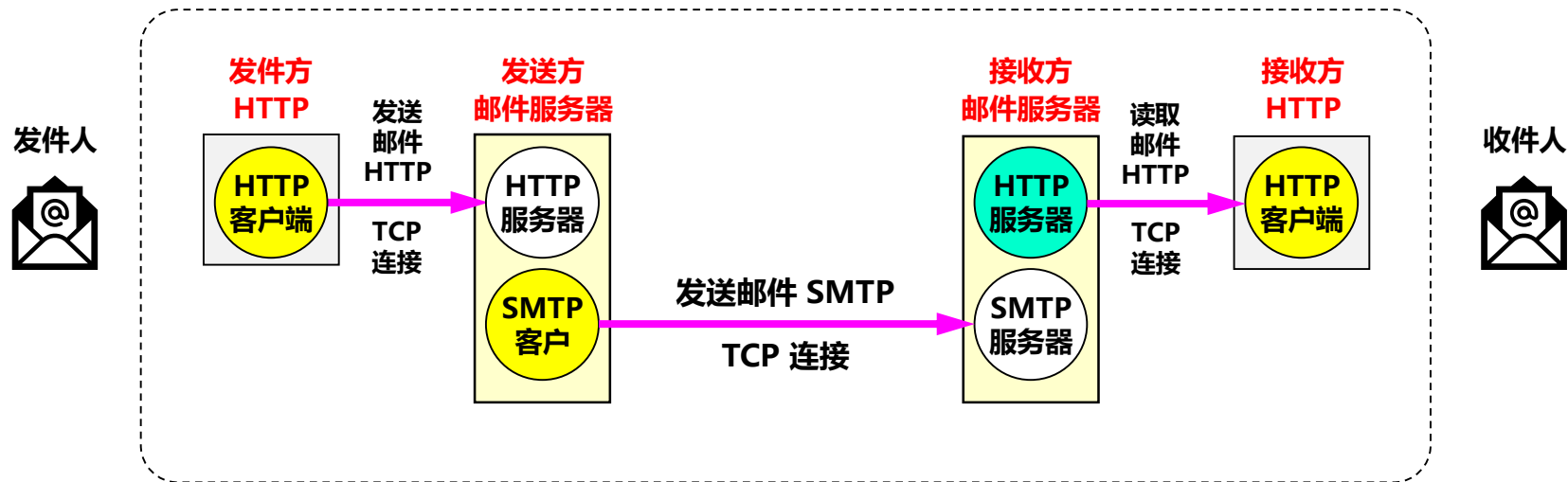
5.5 基于万维网的电子邮件



- 客户端发送、接收电子邮件时使用 SMTP 和 POP3 协议。
- 两个邮件服务器之间传送邮件时使用 SMTP。

5. 电子邮件

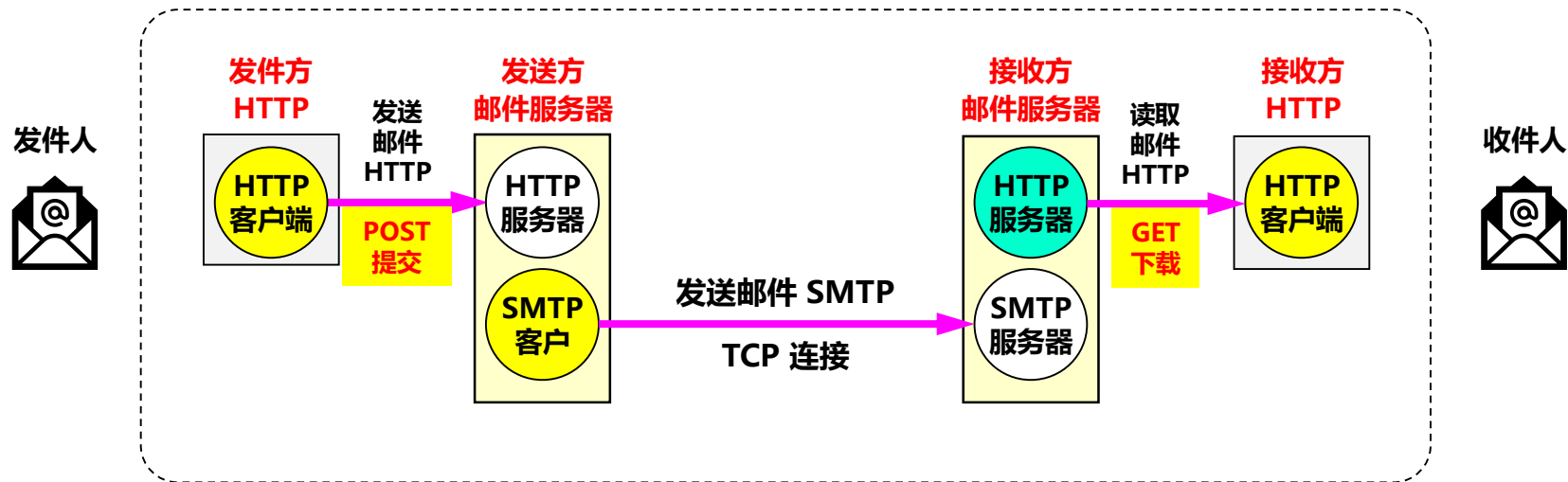
5.5 基于万维网的电子邮件



- 基于 Web 的电子邮件发送、接收电子邮件时使用 HTTP 协议。
- 两个邮件服务器之间传送邮件时使用 SMTP。

5. 电子邮件

5.5 基于万维网的电子邮件



- 使用 HTTP POST 方法提交要发送的邮件。
- 使用 HTTP GET 方法读取邮件。

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

- SMTP 存在的缺点：
 - 不能传送可执行文件或其他的二进制对象。
 - 限于传送 7 位的 ASCII 码，无法传送非 ASCII 编码的信息。
 - 服务器会拒绝超过一定长度的邮件。
 - 某些 SMTP 的实现并没有完全按照 [RFC 821] 的 SMTP 标准。

5. 电子邮件

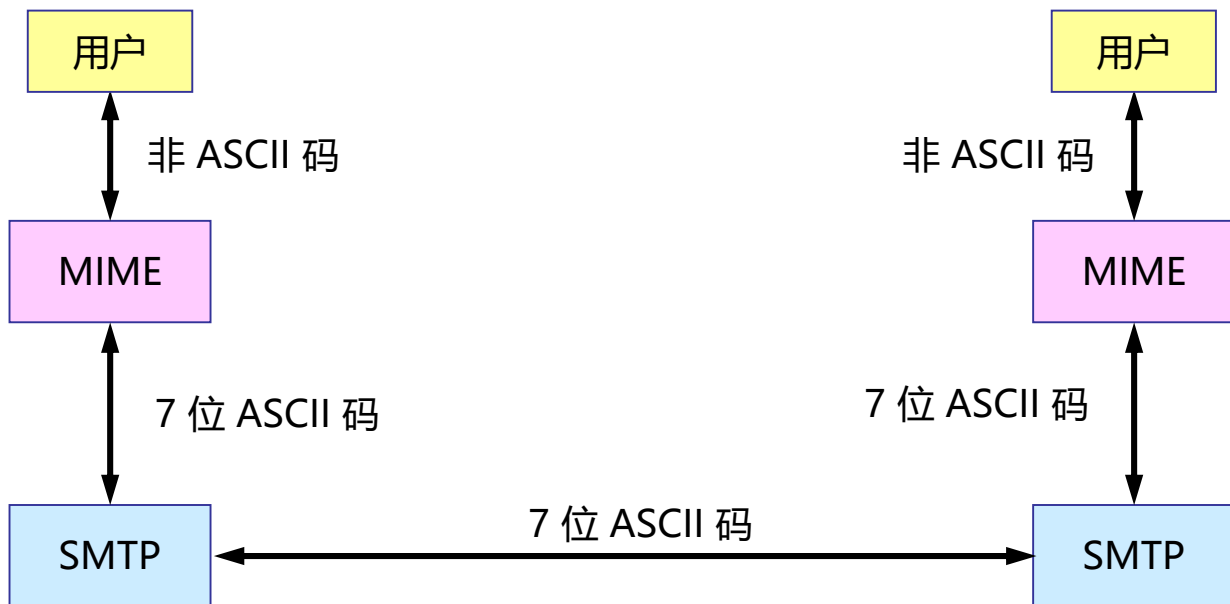
5.6 通用互联网邮件扩充 MIME

- 什么是 MIME:
 - 通用互联网邮件扩充 MIME 并没有改动 SMTP 或取代它。
 - MIME 的意图是继续使用目前的 [RFC 822] 格式, 但增加了邮件主体的结构, 并定义了传送非 ASCII 码的编码规则。

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

- MIME 和 SMTP 的关系:



5. 电子邮件

5.6 通用互联网邮件扩充 MIME

- MIME 主要包括三个部分：
 - 5个新的邮件首部字段，它们可包含在 [RFC 822] 首部中。
 - 这些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，而不会被邮件系统改变。

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

- MIME 增加 5 个新的邮件首部:

MIME 首部

邮件首部

- **MIME-Version:** MIME 的版本。若无此行, 则为英文文本。
- **Content-Description:** 这是可读字符串, 此邮件的说明。
- **Content-Id:** 邮件的唯一标识符。
- **Content-Transfer-Encoding:** 传送时邮件主体使用的编码方法。
- **Content-Type:** 邮件内容类型 / 子类型。

邮件主体

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

- 内容传送编码 (Content-Transfer-Encoding) :
 - 最简单的编码就是 7 位 ASCII 码, 而每行不能超过 1000 个字符。MIME 对这种由 ASCII 码构成的邮件主体不进行任何转换。
 - 另一种编码称为 quoted-printable, 这种编码方法适用于当所传送的数据中只有少量的非 ASCII 码。
 - 对于任意的二进制文件, 可用 base64 编码。

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

内容传送编码 (Content-Transfer-Encoding)

编码方法	说明
7bit	7 位 ASCII 编码，每行不能超过 1000 个字符（包括回车和换行）。缺省编码方法。
8bit	8 位非 ASCII 编码，每行不能超过 1000 个字节（包括回车和换行）。
Binary	8 位非 ASCII 编码，任意长度的字节串。
Base64	将任意长度的字节串转换为用 7 位 ASCII 编码表示的字符串。可用于二进制和非文本数据的编码。
Quoted-printable	将任意长度的字节串转换为 ASCII 编码表示的字符串。可用于二进制和非文本数据的编码。

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

□ 内容类型：

■ MIME 标准规定：

Content-Type 说明必须含有两个标识符：内容类型 (type) 和子类型 (subtype)，中间用 “/” 分开。

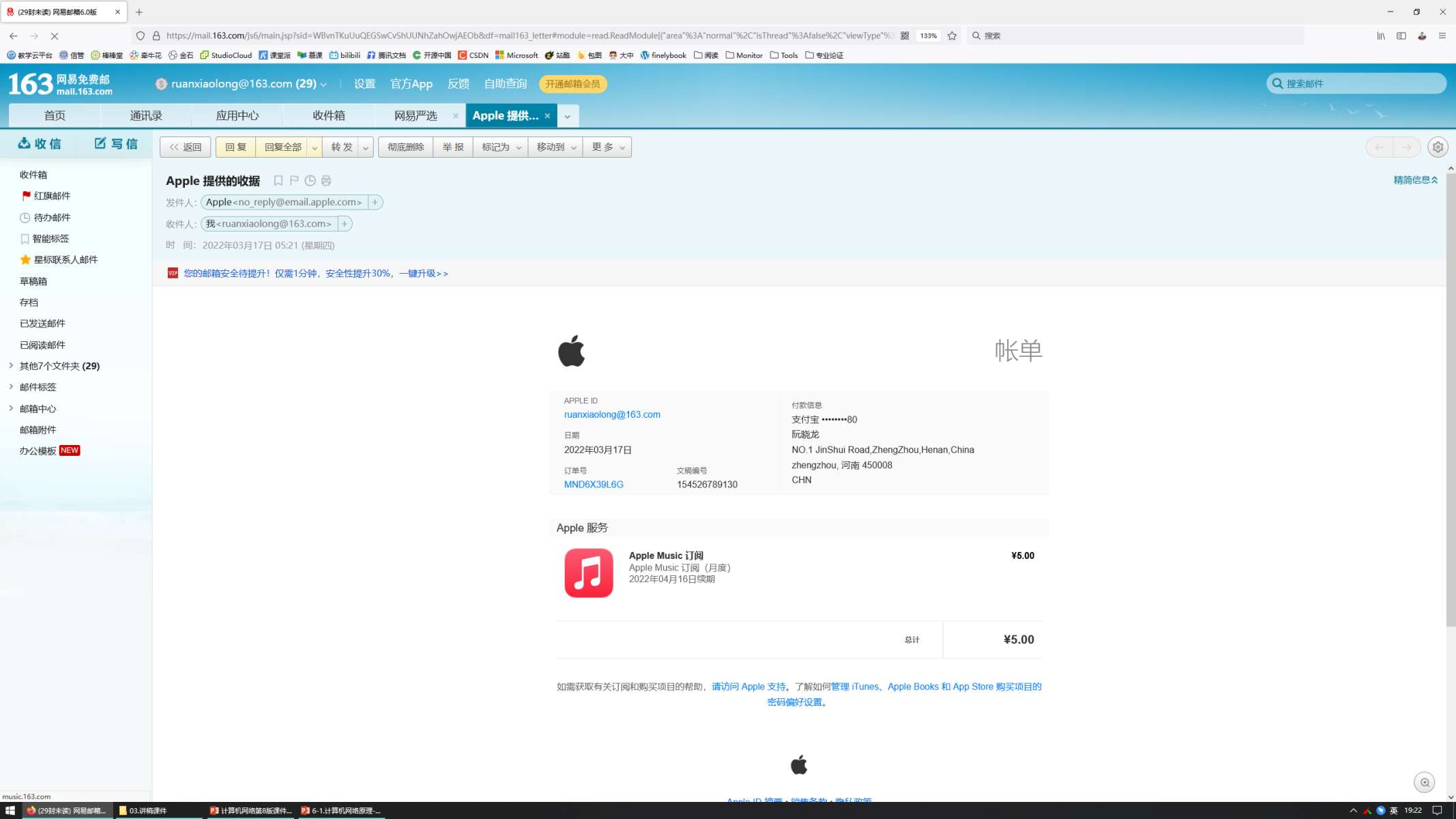
- MIME 标准原先定义了 7 个基本内容类型和 15 种子类型。
- MIME 允许发件人和收件人自己定义专用的内容类型。
 - 但为避免可能出现名字冲突，标准要求为专用内容类型选择的名称要以字符串 X- 开始。

5. 电子邮件

5.6 通用互联网邮件扩充 MIME

MIME Content-Type 说明中的类型及子类型

内容类型	子类型举例	说明
text (文本)	plain, html, xml, css	不同格式的文本
image (图像)	gif, jpeg, tiff	不同格式的静止图像
audio (音频)	basic, mpeg, mp4	可听见的声音
video (视频)	mpeg, mp4, quicktime	不同格式的视频
model (模型)	vrml	3D模型
application (应用)	octet-stream, pdf, javascript, zip	不同应用程序产生的数据
message (报文)	http, rfc822	封装的报文
multipart (多部分)	mixed, alternative, parallel, digest	多种类型的组合



6. 动态主机配置协议 DHCP

- 在协议软件中，给协议参数赋值的动作叫做**协议配置**。
- 一个协议软件在使用之前必须是已正确配置的。
- 具体的配置信息有哪些，取决于协议栈。
- 连接到互联网的计算机的协议软件需要配置的参数包括：
 - IP 地址
 - 子网掩码
 - 默认路由器的 IP 地址
 - 域名服务器的 IP 地址

6. 动态主机配置协议 DHCP

6.1 DHCP

- 动态主机配置协议 DHCP (Dynamic Host Configuration Protocol)
 - 提供了即插即用连网 (plug-and-play networking) 的机制, 允许一台计算机加入网络和获取 IP 地址, 而不用手工配置。
 - DHCP 给运行服务器软件、且位置固定的计算机指派一个永久地址, 给运行客户端软件的计算机分配一个临时地址。

6. 动态主机配置协议 DHCP

6.2 DHCP 的工作方式

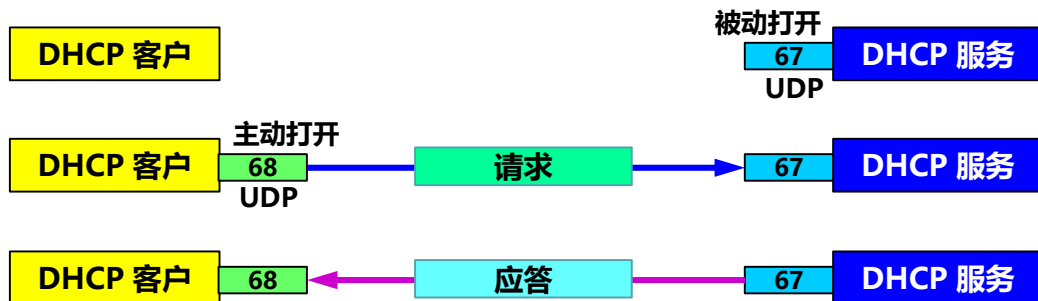
- DHCP 使用客户服务器方式
 - 需要 IP 地址的主机在启动时就向 DHCP 服务器广播发送发现报文 (DHCPDISCOVER) ， 这时该主机就成为 DHCP 客户。
 - 本地网络上所有主机都能收到此广播报文， 但只有 DHCP 服务器才回答此广播报文。
 - DHCP 服务器先在其数据库中查找该计算机的配置信息。
 - 若找到， 则返回找到的信息。
 - 若找不到， 则从服务器的 IP 地址池 (address pool) 中取一个地址分配给该计算机。 DHCP 服务器的回答报文叫做提供报文 (DHCPOFFER) 。

6. 动态主机配置协议 DHCP

6.2 DHCP 的工作方式

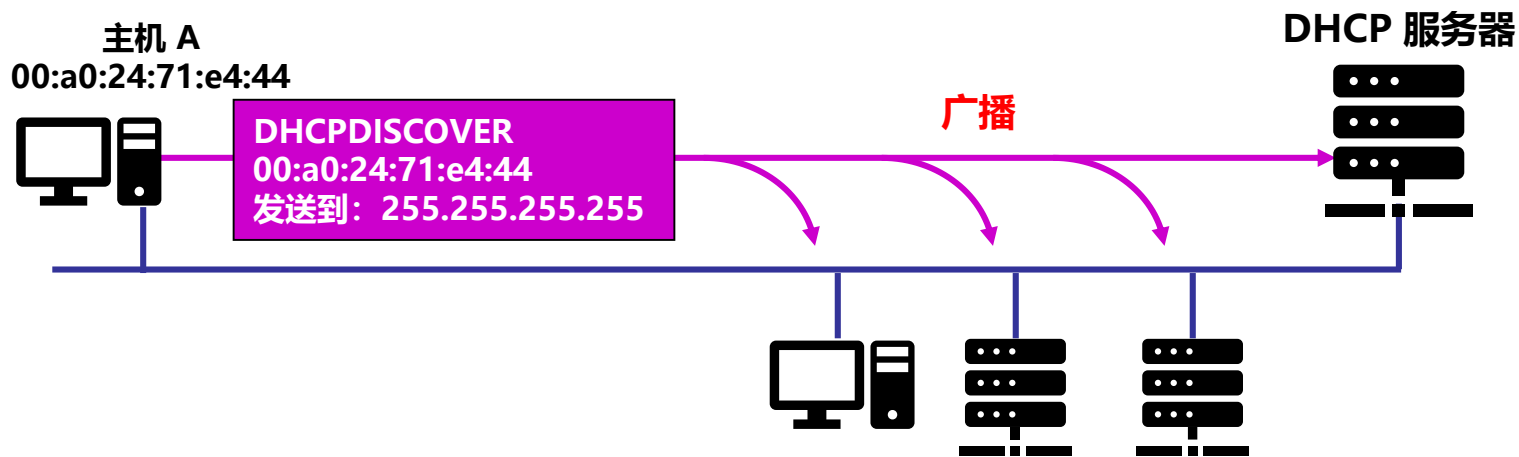
□ DHCP 工作方式

- DHCP 使用客户服务器方式，采用请求/应答方式工作。
- DHCP 基于 UDP 工作，DHCP 服务器运行在 67 号端口，DHCP 客户运行在 68 号端口。



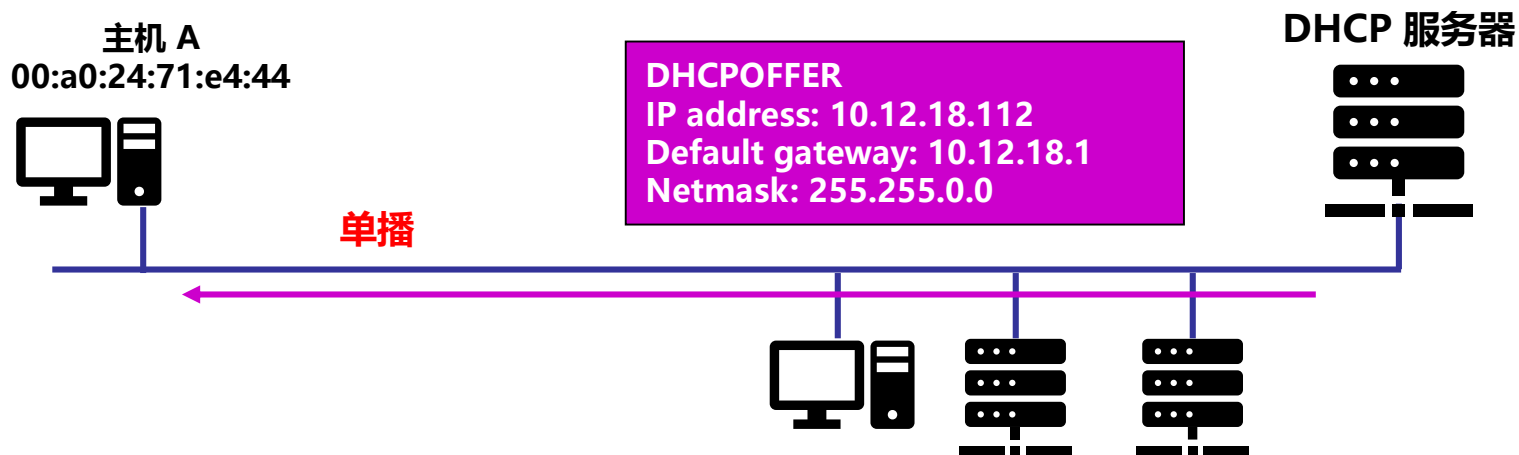
6. 动态主机配置协议 DHCP

6.2 DHCP 的工作方式



6. 动态主机配置协议 DHCP

6.2 DHCP 的工作方式



6. 动态主机配置协议 DHCP

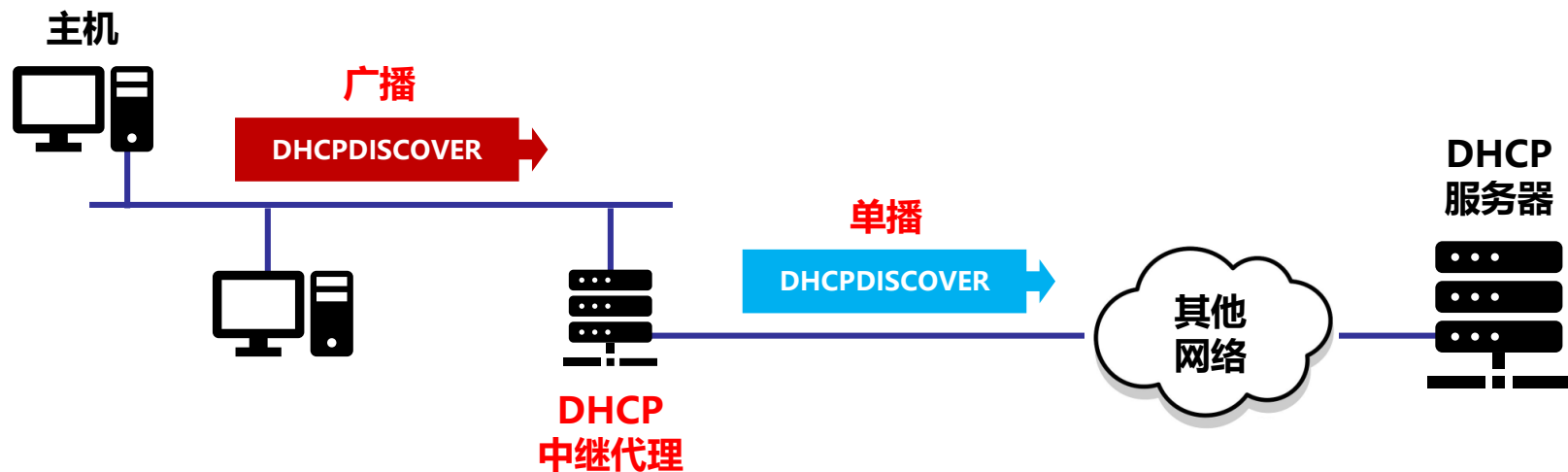
6.3 DHCP 中继代理 (relay agent)

- 问题：每个网络上都需要有 DHCP 服务器吗？
 - 答案：不需要，因为会使 DHCP 服务器的数量太多。

- 问题：若没有 DHCP 服务器，如何自动获得地址？
 - 解决：每一个网络至少有一个 DHCP 中继代理，它配置了 DHCP 服务器的 IP 地址信息。

6. 动态主机配置协议 DHCP

6.3 DHCP 中继代理 (relay agent)



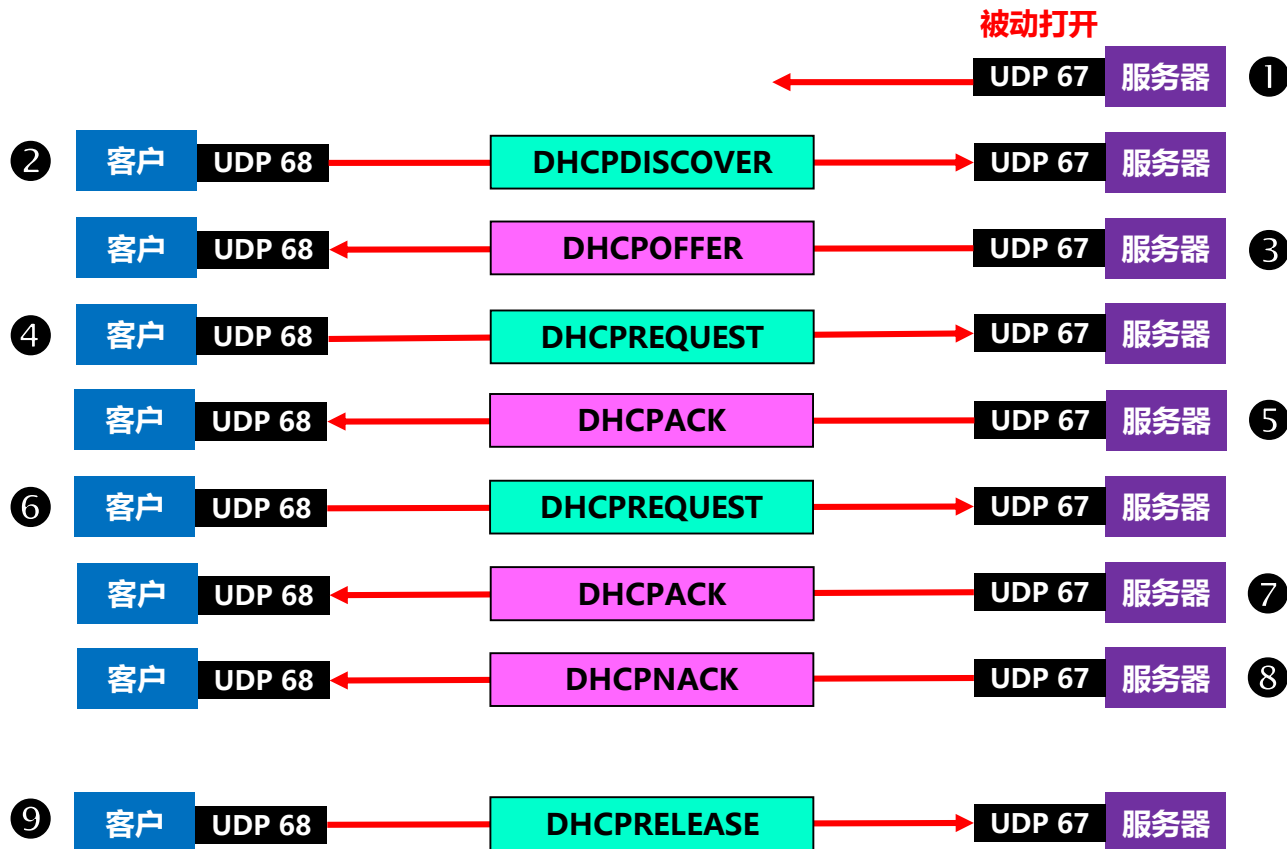
6. 动态主机配置协议 DHCP

6.4 租用期 (Lease Period)

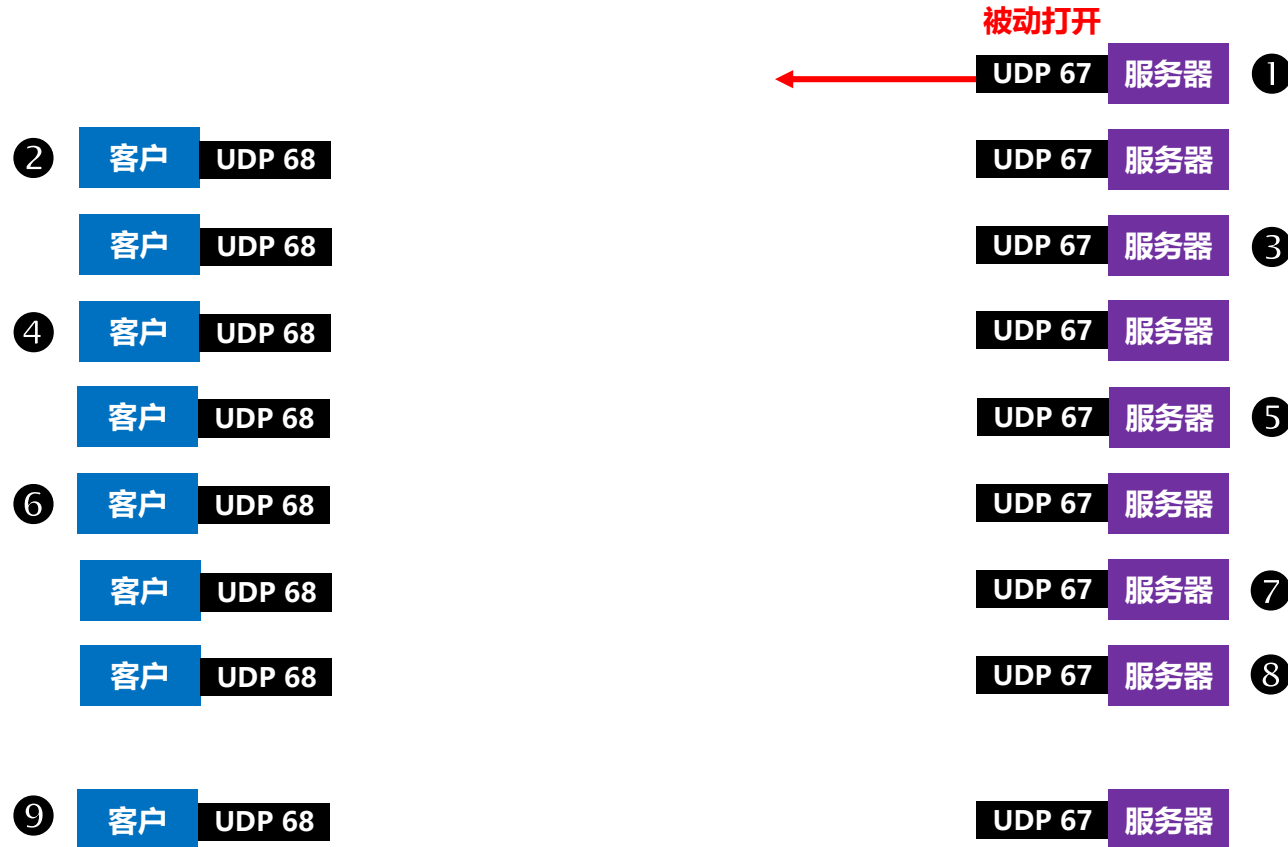
- DHCP 服务器分配给 DHCP 客户的 IP 地址是临时的，因此 DHCP 客户只能在一段有限的时间内使用这个分配到的 IP 地址。
- DHCP 协议称这段时间为租用期。
 - 租用期的数值应由 DHCP 服务器自己决定。
 - DHCP 客户也可在自己发送的报文中（例如，发现报文）提出对租用期的要求。

6. 动态主机配置协议 DHCP

6.5 DHCP 协议的工作过程

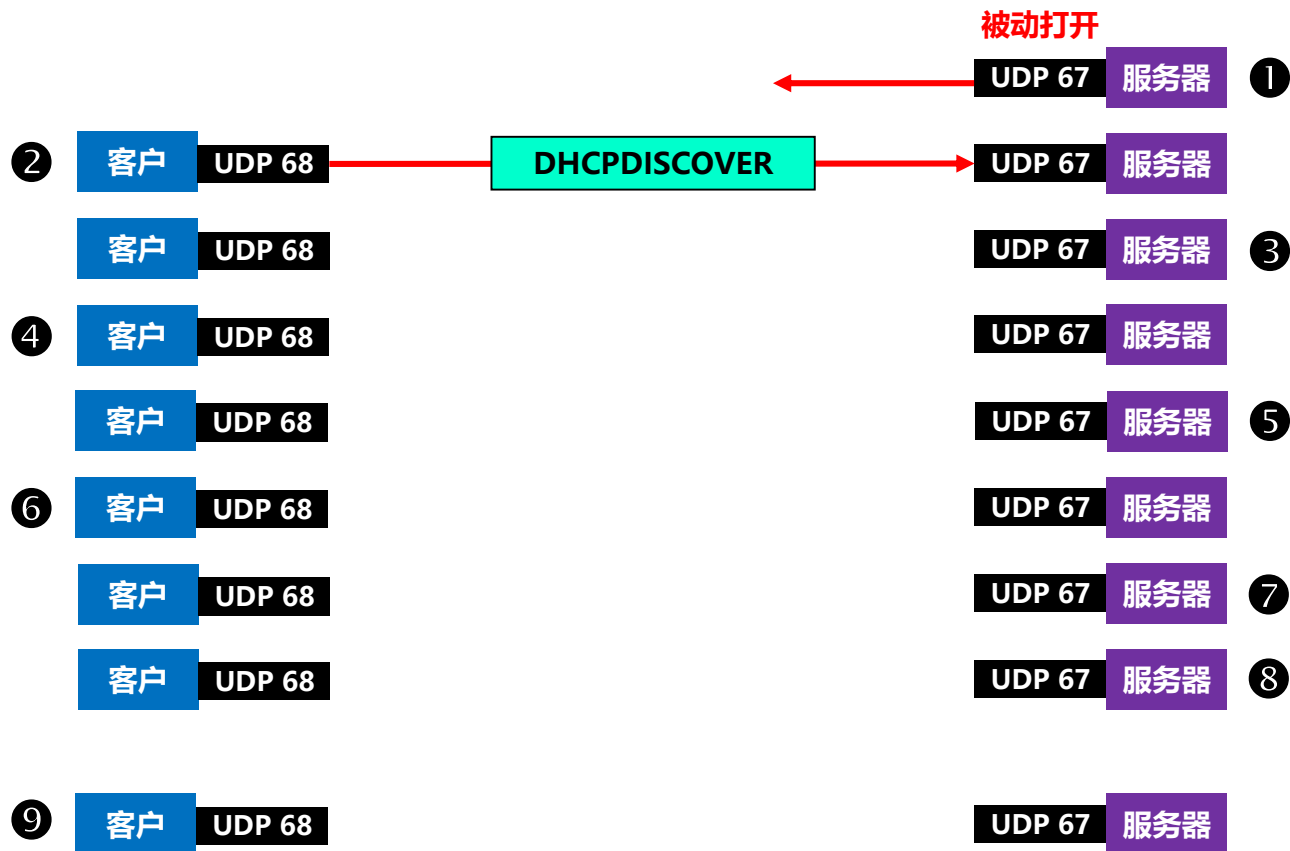


DHCP 协议的工作过程



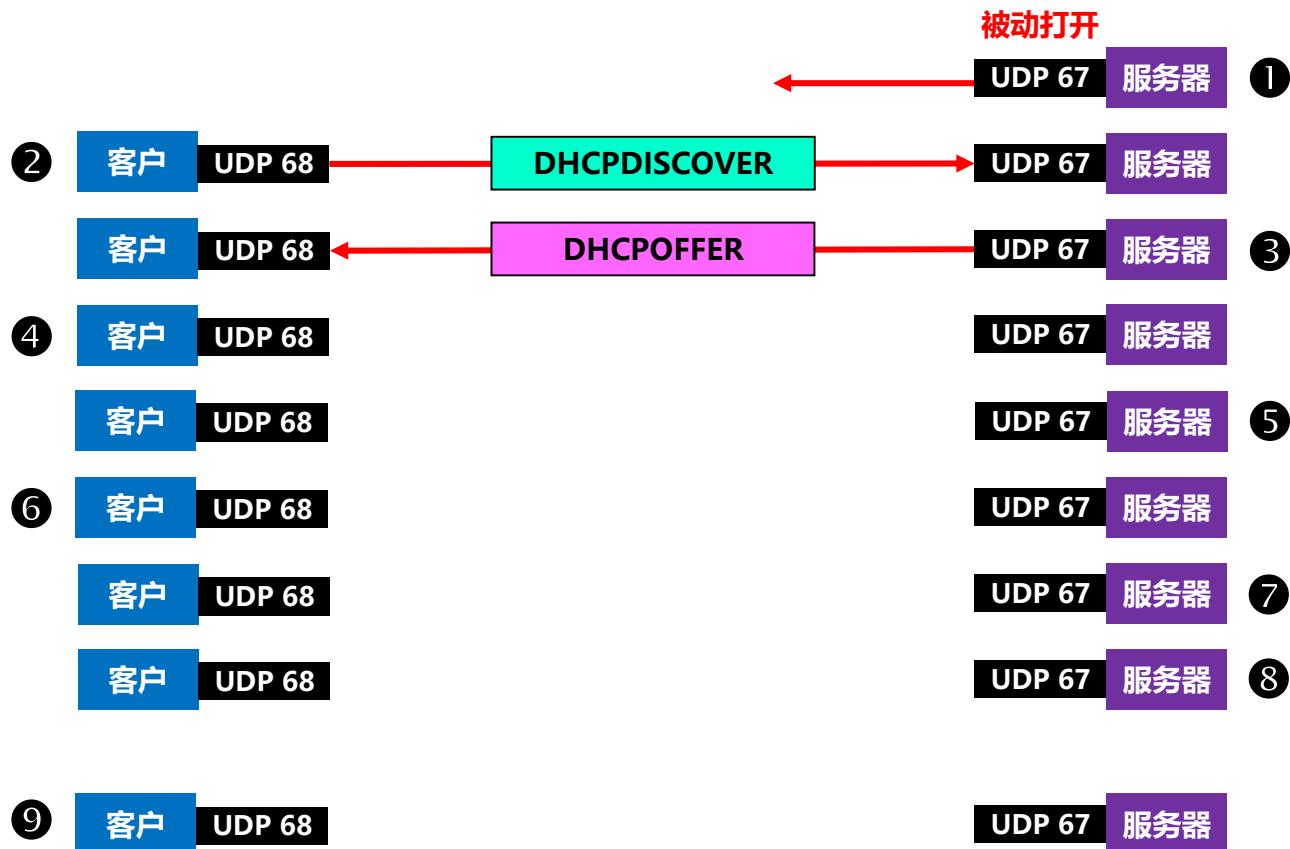
❶: DHCP 服务器被动打开 UDP 端口 67, 等待客户端发来的报文。

DHCP 协议的工作过程



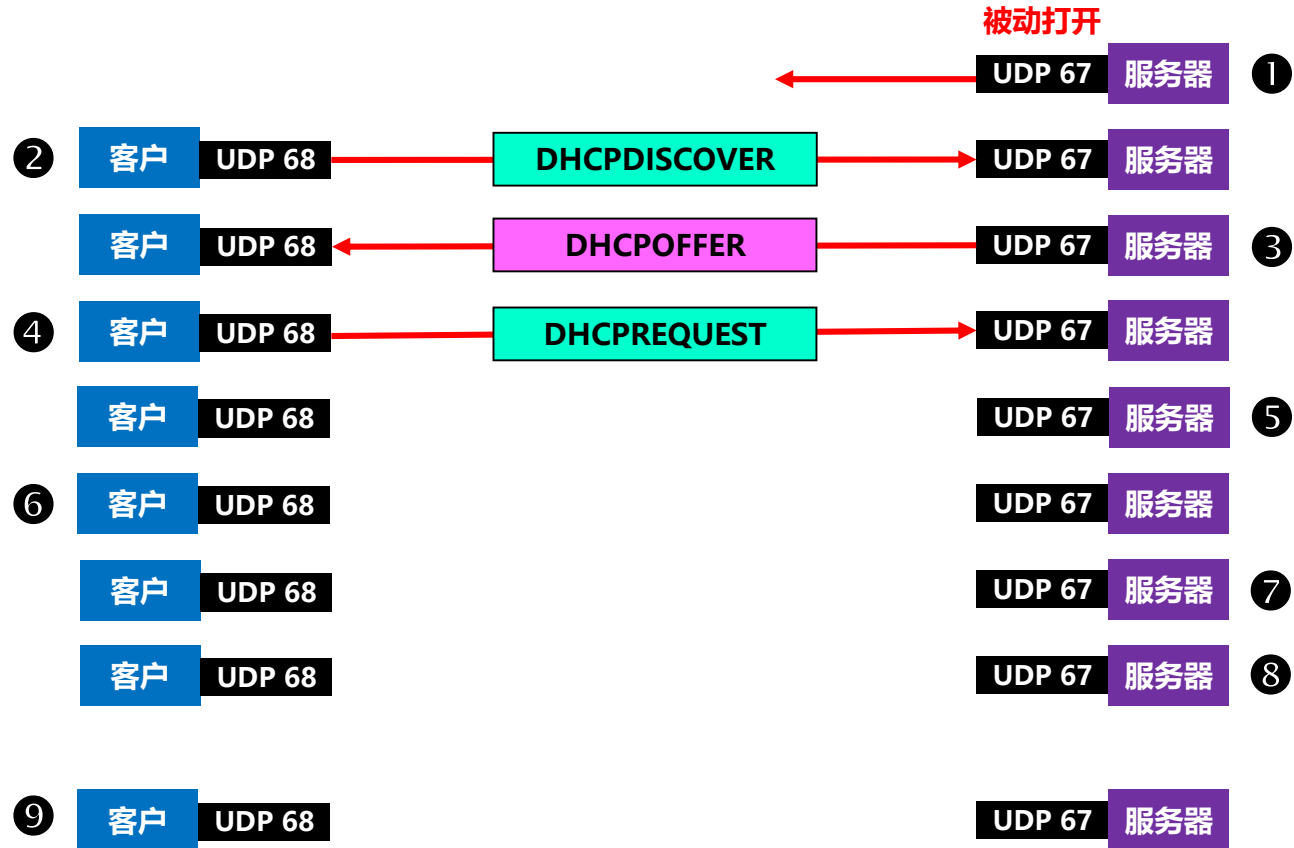
②: DHCP 客户从 UDP 端口 68 发送 DHCP 发现报文 DHCPDISCOVER。

DHCP 协议的工作过程



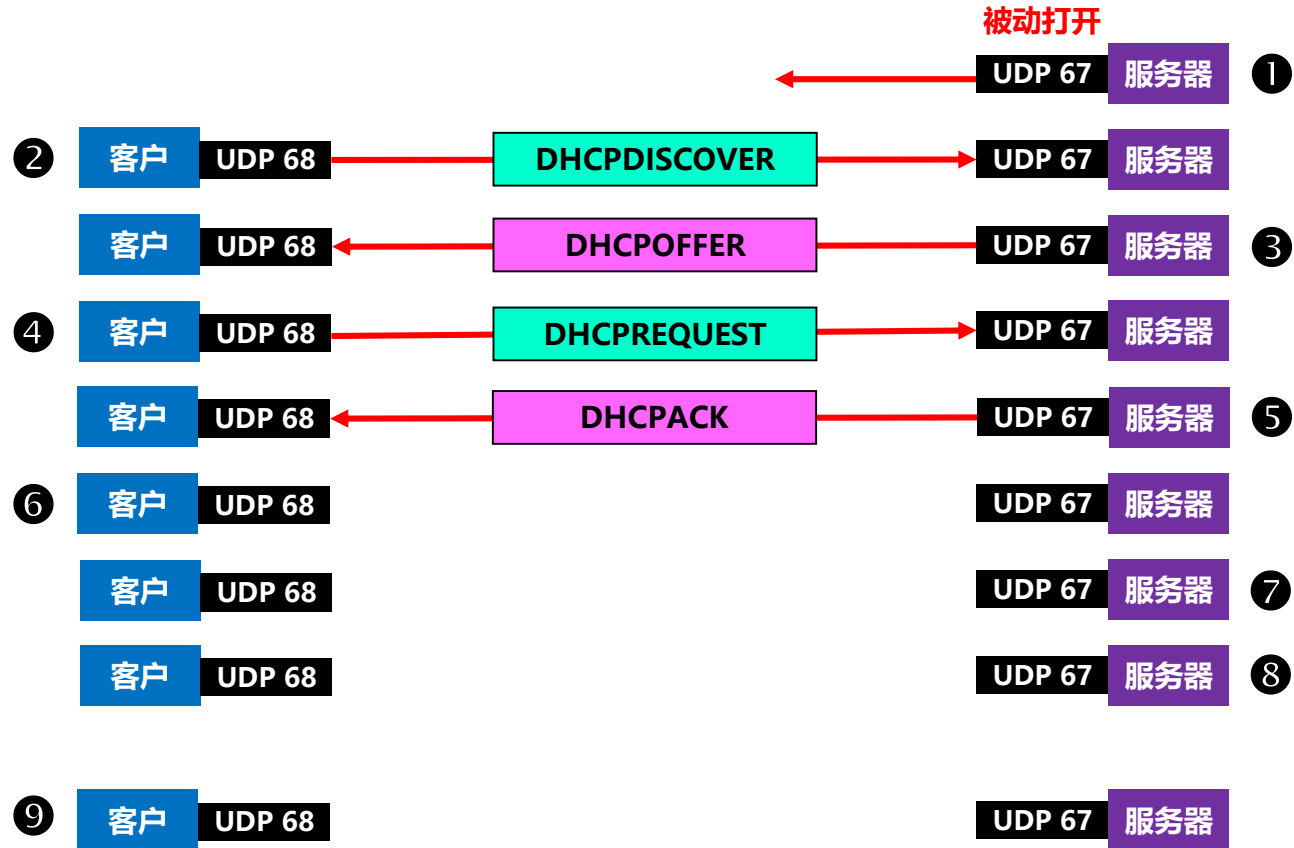
③: 凡收到 DHCP 发现报文的 DHCP 服务器都发出 DHCP 提供报文 DHCPOFFER, 因此 DHCP 客户可能收到多个 DHCP 提供报文。

DHCP 协议的工作过程



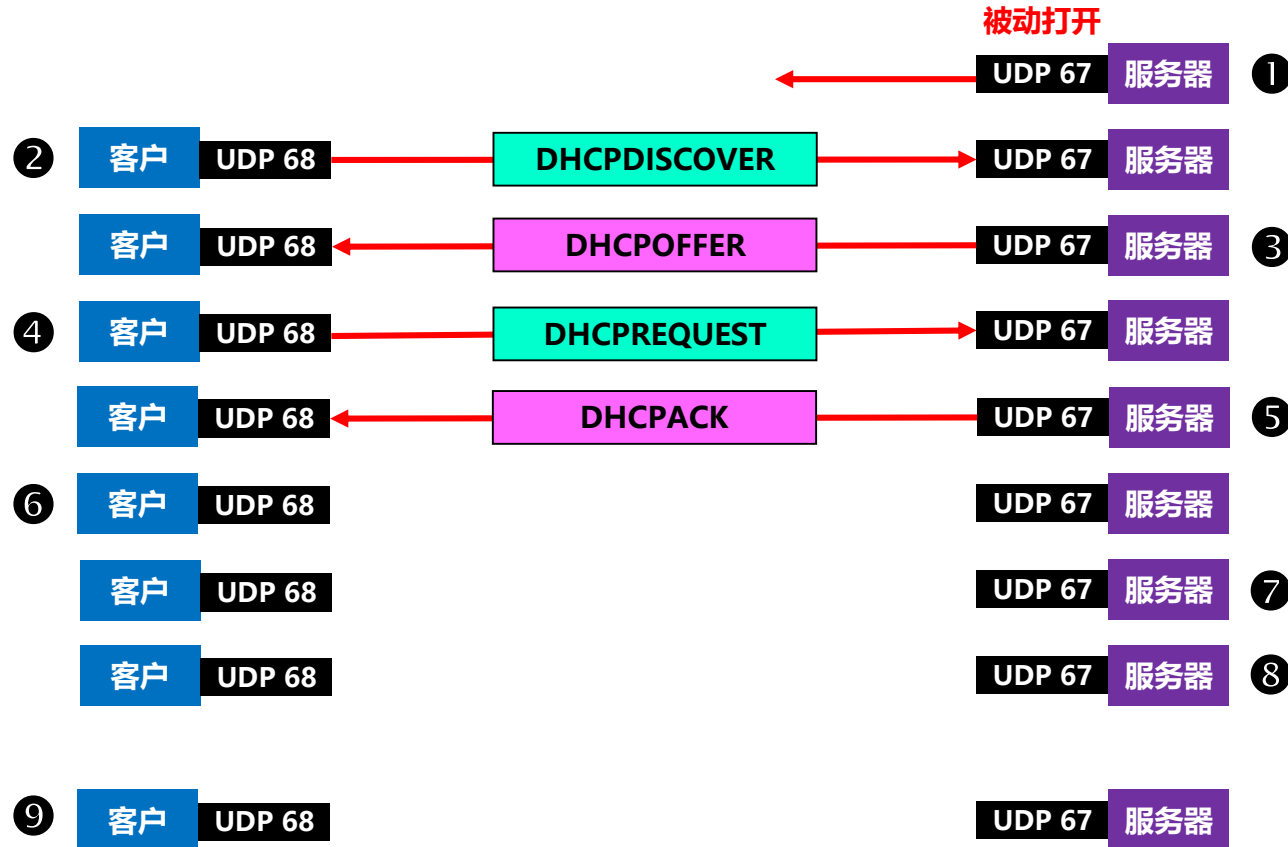
④: DHCP 客户从几个 DHCP 服务器中选择其中的一个, 并向所选择的 DHCP 服务器发送 DHCP 请求报文 DHCPREQUEST。

DHCP 协议的工作过程



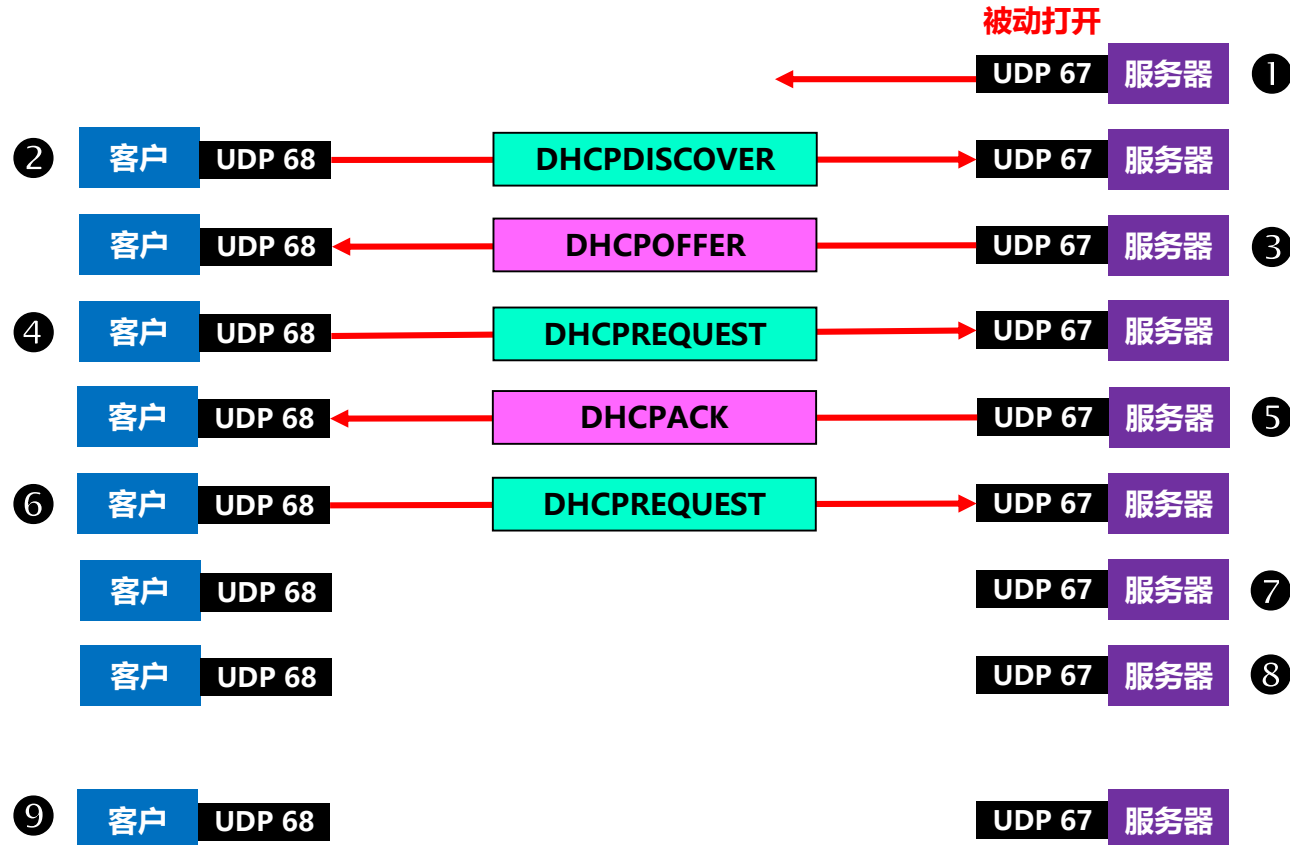
⑤: 被选择的 DHCP 服务器发送确认报文 DHCPACK, DHCP 客户可开始使用得到的临时 IP 地址了, 进入已绑定状态。

DHCP 协议的工作过程



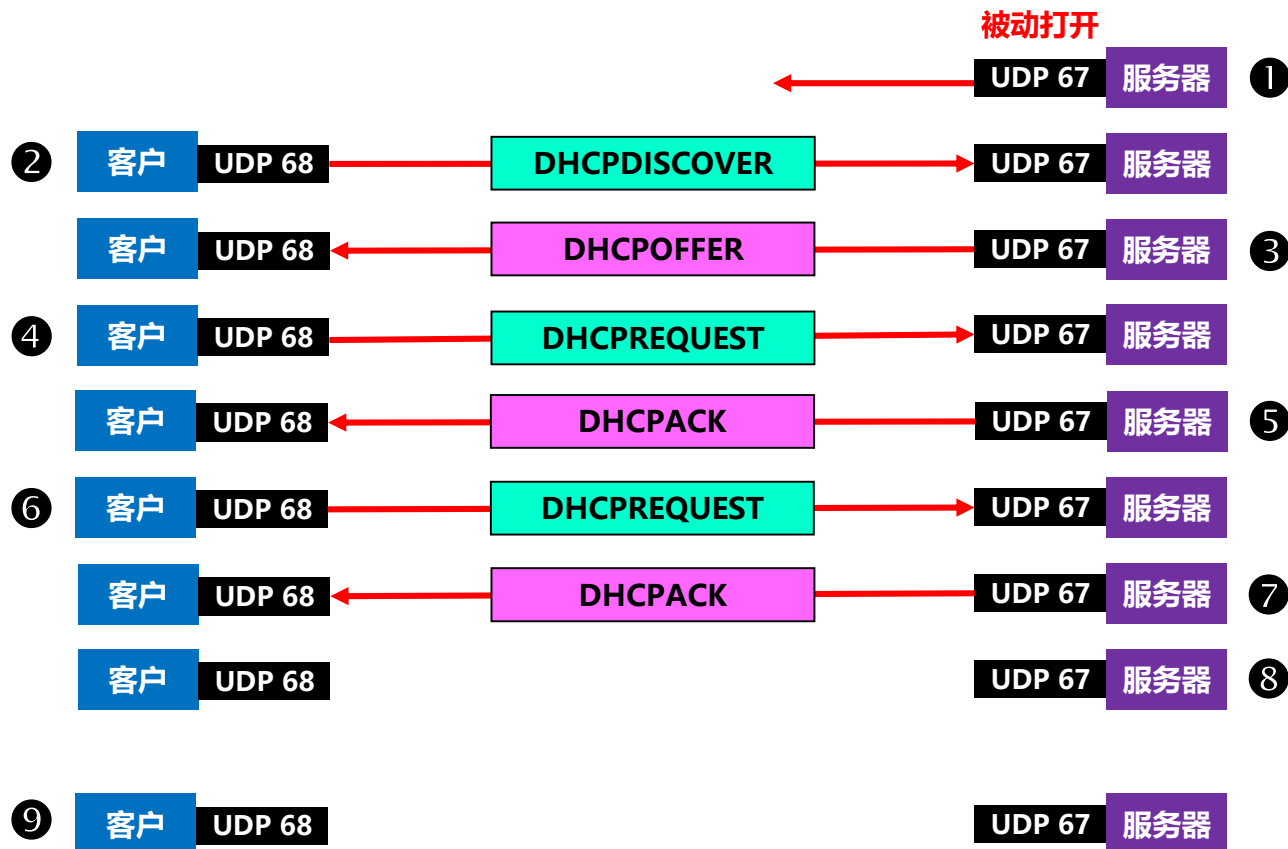
⑤: DHCP 客户现在要根据服务器提供的租用期 T 设置两个计时器 $T1$ 和 $T2$, 它们的超时时间分别是 $0.5T$ 和 $0.875T$ 。当超时时间到时, 就要请求更新租用期。

DHCP 协议的工作过程



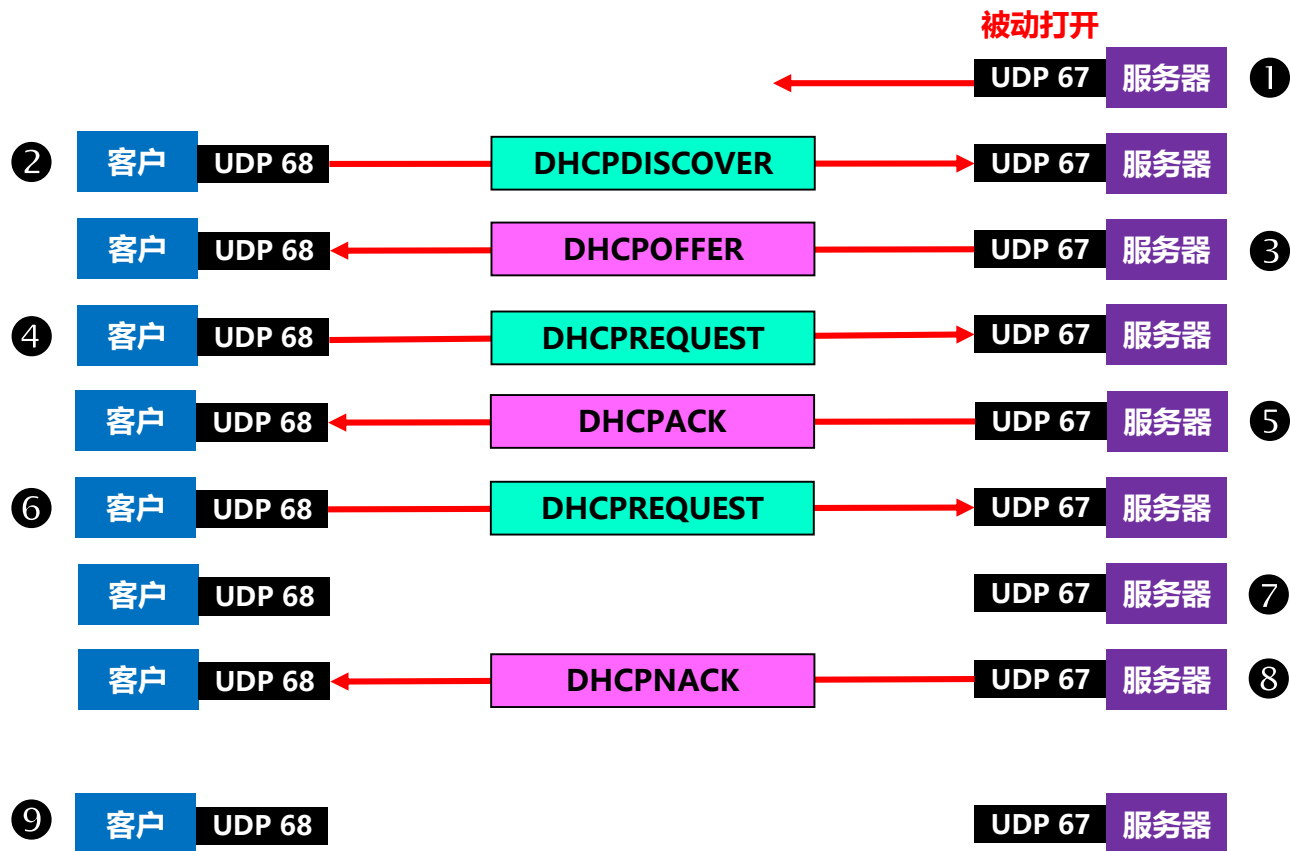
⑥：租用期过了一半 (T1 时间到)，DHCP 发送请求报文 DHCPREQUEST，要求更新租用期。

DHCP 协议的工作过程



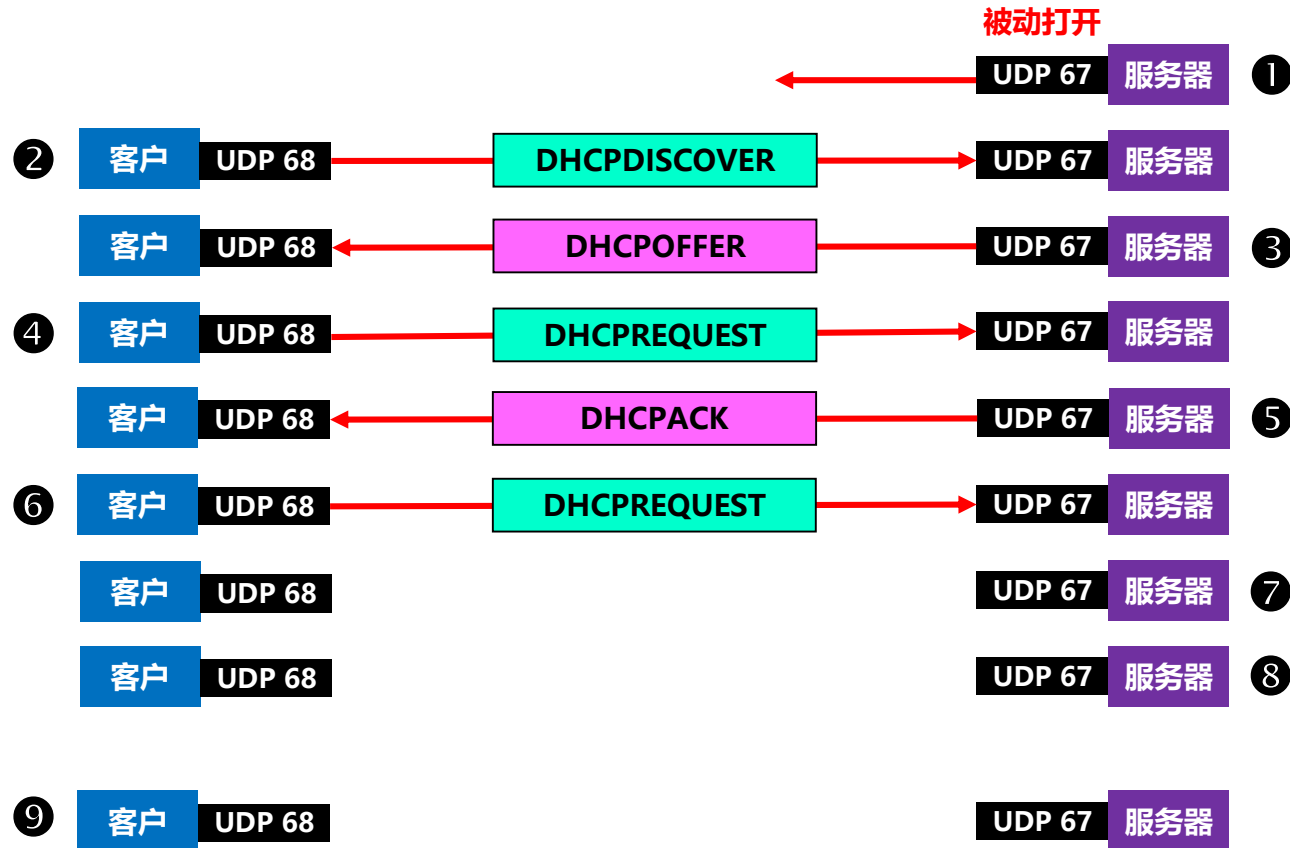
⑦: DHCP 服务器若同意, 则发回确认报文 DHCPACK。DHCP 客户得到了新的租用期, 重新设置计时器。

DHCP 协议的工作过程



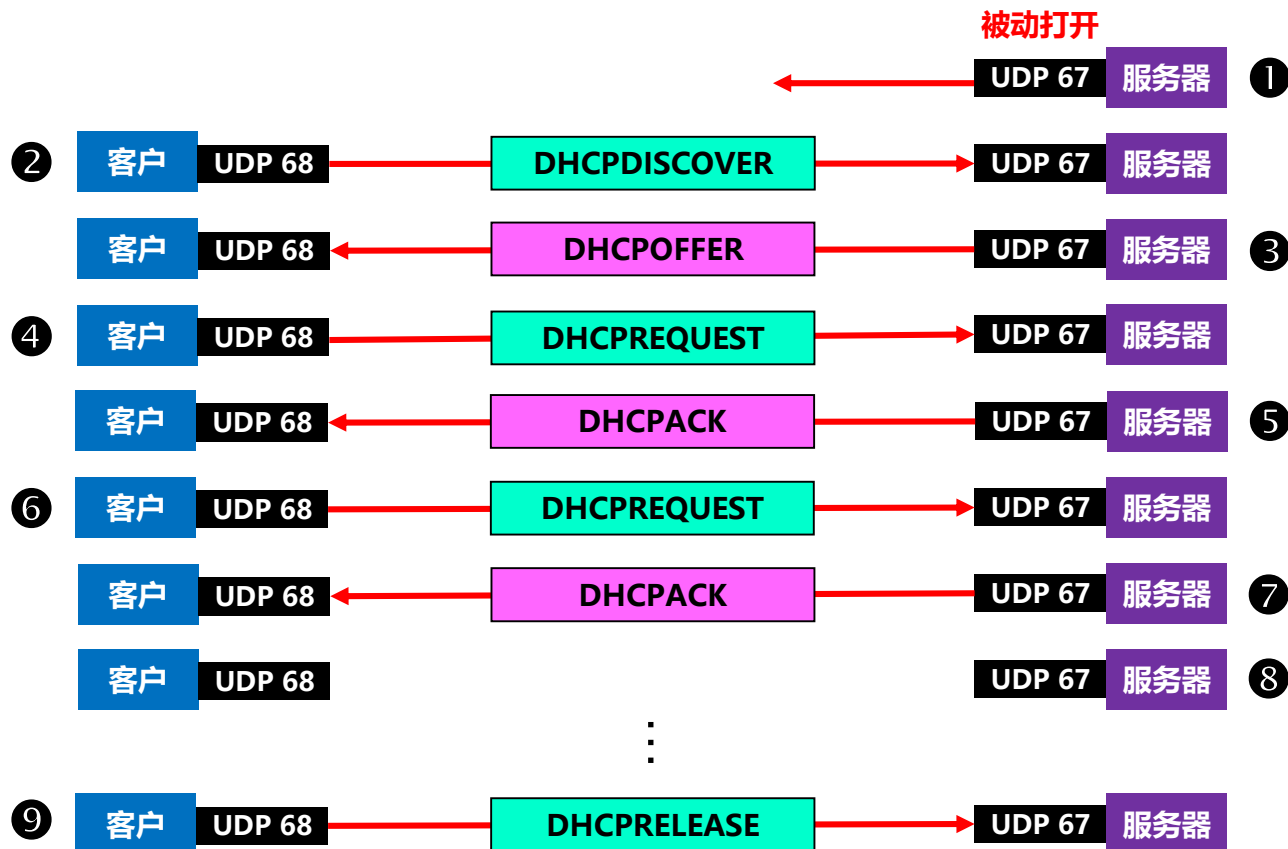
⑧：DHCP 服务器若不同意，则发回否认报 DHCPNACK。这时 DHCP 客户必须立即停止使用原来的 IP 地址，而必须重新申请 IP 地址（回到步骤 ②）。

DHCP 协议的工作过程



若 DHCP 服务器不响应步骤 ⑥ 的请求报文 DHCPREQUEST，则在租用期过了 87.5% 时 (T2 时间到)，DHCP 客户必须重新发送请求报文 DHCPREQUEST (重复步骤 ⑥)，然后又继续后面的步骤。

DHCP 协议的工作过程



⑨：DHCP 客户可随时提前终止服务器所提供的租用期，这时只需向 DHCP 服务器发送释放报文 DHCPRELEASE 即可。

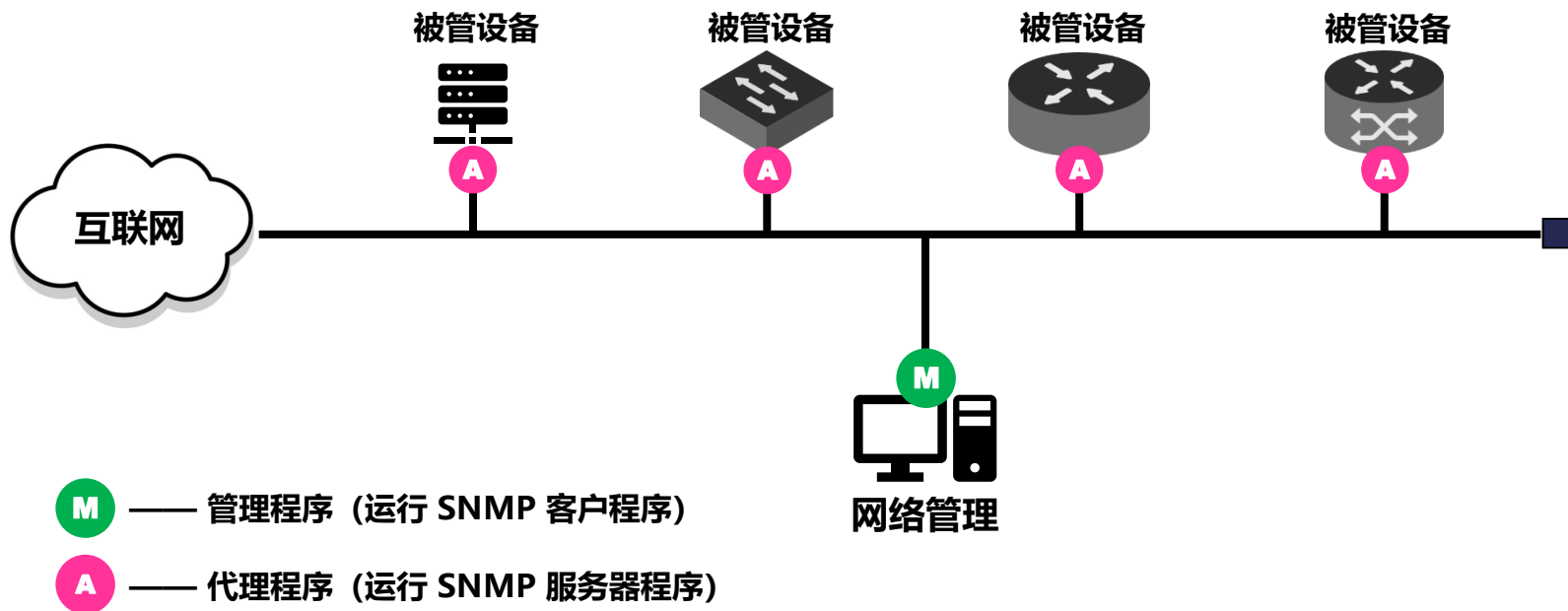
7. 简单网络管理协议 SNMP

- 网络管理简称为网管。
 - 网络管理包括对硬件、软件和人力的使用、综合与协调，以便对网络资源进行监视、测试、配置、分析、评价和控制，这样就能以合理的价格满足网络的一些需求，如实时运行性能，服务质量等。
 - 网络管理并不是指对网络进行行政上的管理，通常分为五大功能。
 - **故障管理**：故障检测、隔离和纠正。
 - **配置管理**：初始化网络、并配置网络。
 - **计费管理**：记录网络资源的使用。
 - **性能管理**：估价系统资源的运行状况及通信效率等。
 - **安全管理**：对授权机制、访问控制、加密和加密关键字的管理。

7. 简单网络管理协议 SNMP

7.1 网络管理

网络管理的一般模型



7. 简单网络管理协议 SNMP

- 网络管理模型中的主要构件：
 - 管理站：
 - 管理站也常称为网络运行中心 NOC (Network Operations Center)，是网络管理系统的核心。
 - 管理程序在运行时就成为管理进程。
 - 管理站（硬件）或管理程序（软件）都可称为管理者 (manager)。Manager 不是指人，而是指机器或软件。
 - 网络管理员 (administrator) 指的是负责网络管理的人员。
 - 大型网络往往实行多级管理，因而有多个管理者，而一个管理者一般只管理本地网络的设备。

7. 简单网络管理协议 SNMP

- 网络管理模型中的主要构件：
 - 被管对象：
 - 网络的每一个被管设备中可能有多个被管对象。
 - 被管设备有时可称为网络元素或网元。
 - 在被管设备中也会有一些不能被管的对象。
 - 代理 (agent)
 - 在每一个被管设备中都要运行一个程序以便和管理站中的管理程序进行通信。这些运行着的程序叫做网络管理代理程序，或简称为代理。
 - 代理程序在管理程序的命令和控制下在被管设备上采取本地的行动。

7. 简单网络管理协议 SNMP

- 网络管理模型中的主要构件：
 - 网络管理协议：
 - 网络管理协议简称为网管协议。
 - 网络管理协议是管理程序和代理程序之间进行通信的规则。
 - 网络管理员利用网络管理协议，通过管理站对网络中的被管设备进行管理。
 - 需要注意的是，网管协议本身不管理网络。

7. 简单网络管理协议 SNMP

- 简单网络管理协议 SNMP (Simple Network Management Protocol)
 - 管理程序和代理程序按客户-服务器方式工作。
 - 管理程序运行 SNMP 客户程序，向某个代理程序发出请求（或命令），代理程序运行 SNMP 服务器程序，返回响应（或执行某个动作）。
 - 在网管系统中，往往是一个（或少数几个）客户程序与很多的服务器程序进行交互。
- 注意：
 - 若要管理某个对象，就必然会给该对象添加一些软件或硬件，但这种“添加”必须对原有对象的影响尽量小些。

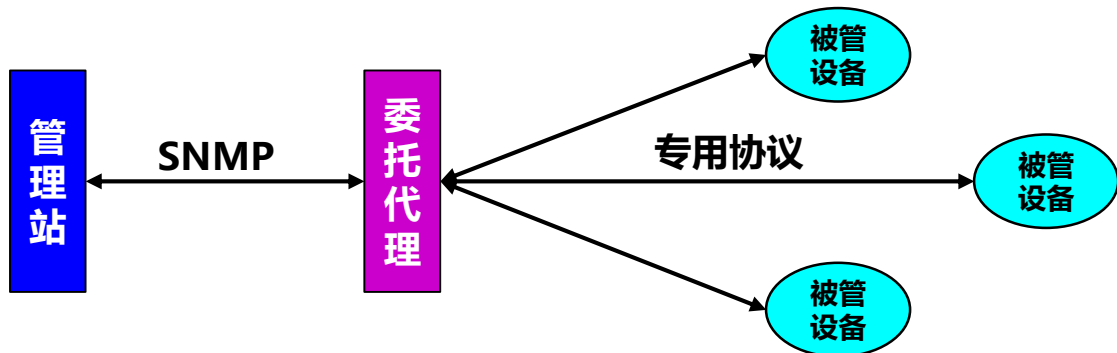
7. 简单网络管理协议 SNMP

- SNMP 最重要的指导思想就是要**尽可能简单**。
- SNMP 的基本功能包括：
 - 监视网络性能
 - 检测分析网络差错
 - 配置网络设备
- 例如：
 - 在网络正常工作时，SNMP 可实现统计、配置、和测试等功能。
 - 当网络出故障时，可实现各种差错检测和恢复功能。
 - 虽然 SNMP 是 TCP/IP 基础上的网络管理协议，但也可扩展到其他类型的网络设备上。

7. 简单网络管理协议 SNMP

□ SNMP 的管理站和委托代理

- 整个系统必须有一个管理站。
- 若网络元素使用的不是 SNMP 而是另一种网络管理协议，SNMP 协议就无法控制该网络元素，此时可使用委托代理 (proxy agent)。
- 委托代理能提供协议转换和过滤操作等功能，对被管对象进行管理。



7. 简单网络管理协议 SNMP

- SNMP 的网络管理由三个部分组成：
 - SNMP 本身
 - 管理信息结构 SMI (Structure of Management Information)
 - 管理信息库 MIB (Management Information Base)。

- 简单网络管理协议 SNMP
 - SNMP 定义了管理站和代理之间所交换的分组格式。
 - 所交换的分组包含各代理中的对象（变量）名及其状态（值）。
 - SNMP 负责读取和改变这些数值。

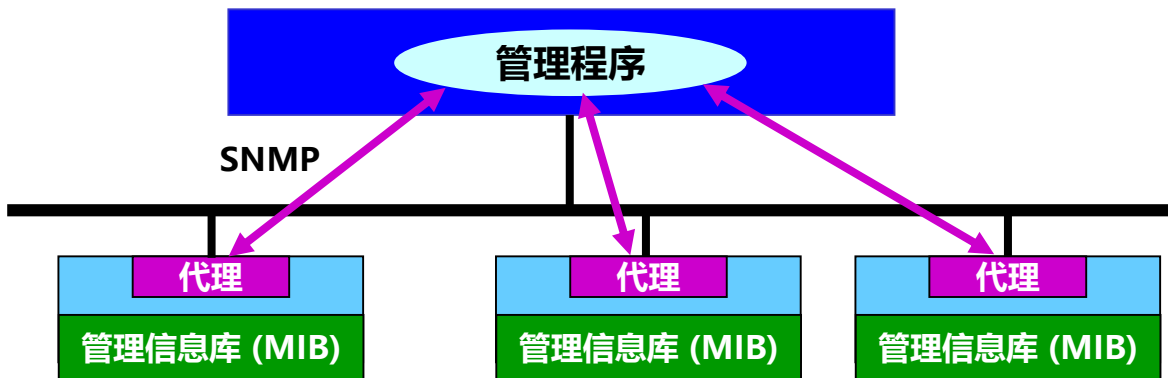
7. 简单网络管理协议 SNMP

□ 管理信息结构 SMI

- SMI 定义了命名对象和定义对象类型（包括范围和长度）的通用规则，以及把对象和对象的值进行编码的规则。
- 这样做是为了确保网络管理数据的语法和语义的无二义性。但从 SMI 的名称并不能看出它的功能。
- SMI 并不定义一个实体应管理的对象数目，也不定义被管对象名以及对象名及其值之间的关联。

7. 简单网络管理协议 SNMP

- 管理信息库 MIB
 - MIB 在被管理的实体中创建了命名对象，并规定了其类型。
 - 管理程序使用 MIB 中的信息，对网络进行管理。



7. 简单网络管理协议 SNMP

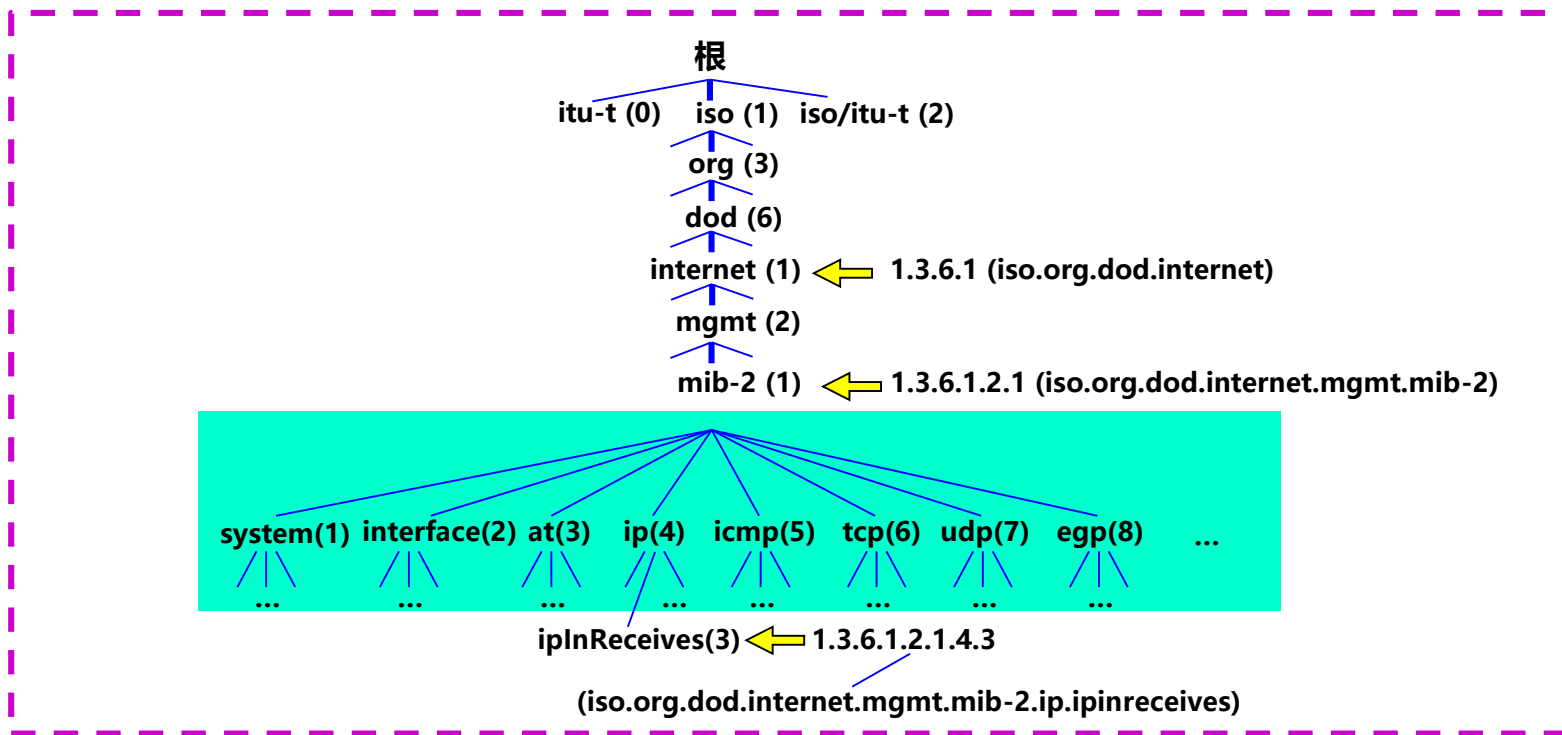
- SMI 的功能：
 - 被管对象应怎样命名
 - 用来存储被管对象的数据类型有哪些种
 - 在网络上传送的管理数据应如何编码

- SMI 标准指明：
 - 所有的 MIB 变量必须使用抽象语法记法 1 (ASN.1) 来定义。
 - SMI 既是 ASN.1 的子集，又是 ASN.1 的超集。
 - ASN.1 记法很严格，它使得数据的含义不存在任何可能的二义性。

7. 简单网络管理协议 SNMP

7.3 SMI

SMI 规定所有被管对象必须在命名树上



7. 简单网络管理协议 SNMP

- SMI 把数据类型分为两大类：
 - 简单类型
 - 结构化类型

数据 { 类型：值集合的名字
值：某个值集合中的一个元素

7. 简单网络管理协议 SNMP

ASN.1 部分数据类型

分类	数据类型名称	含义
简单类型	INTEGER	任意长度整数
	BIT STRING	0 位或多位组成的二进制串
	OCTET STRING	0 位或多位组成的字节串
	NULL	空值
	OBJECT IDENTIFIER	定义的数据类型
结构化类型	SEQUENCE	由多个数据类型按序组成的值
	SEQUENCE OF	由同一数据类型按序组成的值
	CHOICE	可以从多个数据类型中选择一个
	ANY	任何数据类型

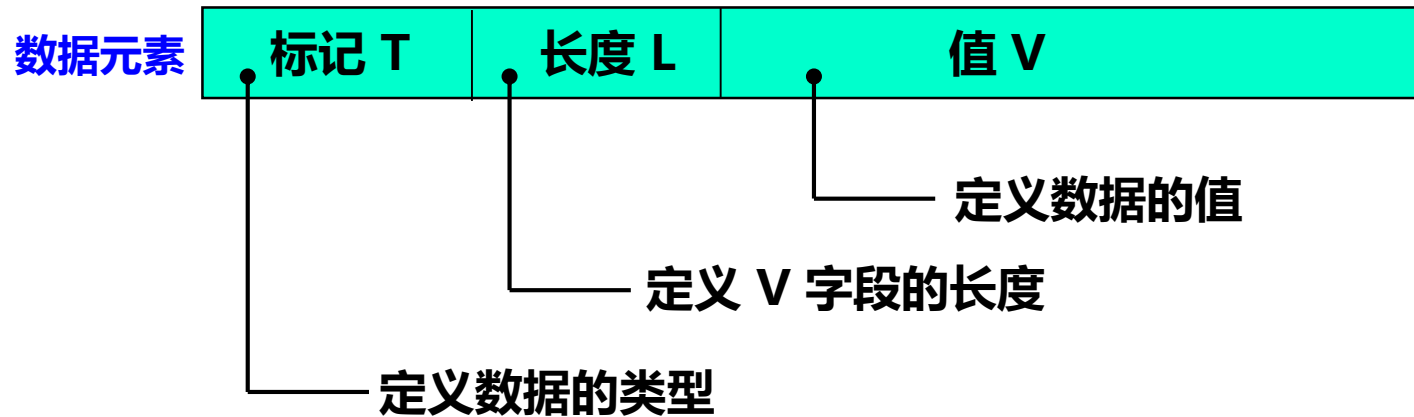
7. 简单网络管理协议 SNMP

□ 基本编码规则 BER

- ISO 在制订 ASN.1 语言的同时也为它定义了一种标准的编码方案，即基本编码规则 BER (Basic Encoding Rule)。
- BER 指明了每种数据类型中每个数据的值的表示。
- 发送端用 BER 编码，可将用 ASN.1 所表述的报文转换成唯一的比特序列。
- 接收端用 BER 进行解码，得到该比特序列所表示的 ASN.1 报文。

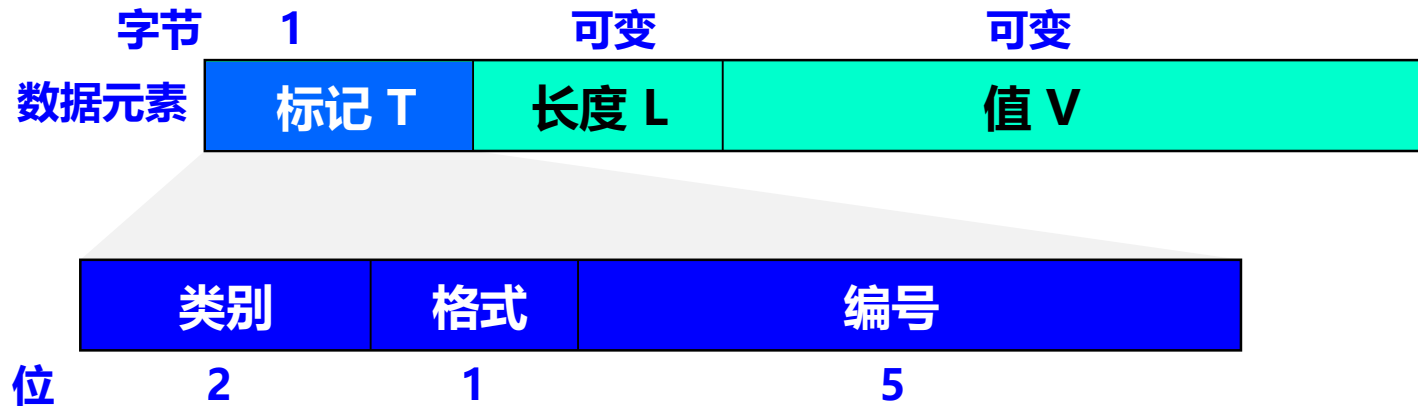
7. 简单网络管理协议 SNMP

- 用 TLV 方法进行编码



7. 简单网络管理协议 SNMP

- TLV 中的 T 字段定义数据的类型



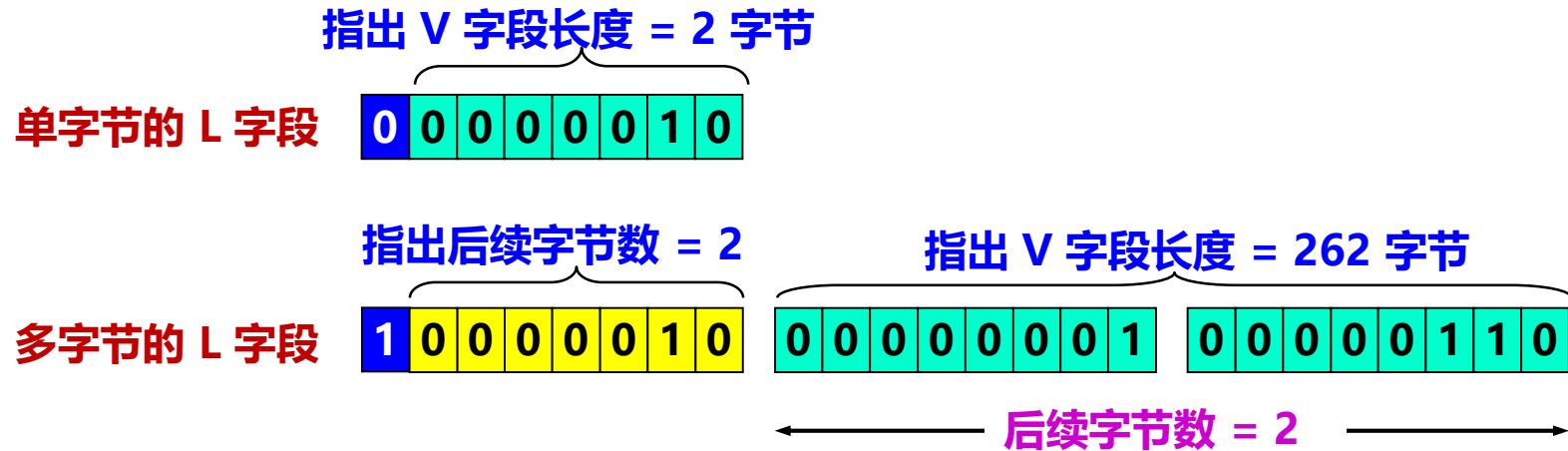
7. 简单网络管理协议 SNMP

□ TLV 中的 T 字段定义数据的类型

数据类型	类别	格式	编号	T字段 (二进制)	T字段 (十六进制)
INTEGER	00	0	00010	00000010	02
OCTET STRING	00	0	00100	00000100	04
OBJECT IDENTIFIER	00	0	00110	00000110	06
NULL	00	0	00101	00000101	05
Sequence, sequence of	00	1	10000	00110000	30
IPAddress	01	0	00000	01000000	40
Counter	01	0	00001	01000001	41
Gauge	01	0	00010	01000010	42
TimeTicks	01	0	00011	01000011	43
Opaque	01	0	00100	01000100	44

7. 简单网络管理协议 SNMP

- TLV 中的 L 字段定义 V 字段的长度



7. 简单网络管理协议 SNMP

□ TLV 中的 V 字段定义数据的值

类 型	大 小	说 明
INTEGER	4 字节	在 -2^{31} 到 $2^{31} - 1$ 之间的整数
Interger32	4 字节	和 INTEGER 相同
Unsigned32	4 字节	在 0 到 $2^{32} - 1$ 之间的无符号数
OCTET STRING	可变	不超过 65535 字节长的字节串
OBJECT IDENTIFIER	可变	对象标识符
IPAddress	4 字节	由 4 个整数组成的 IP 地址
Counter32	4 字节	可从 0 增加到 2^{32} 的整数；当它到达最大值时就返回到 0
TimeTicks	4 字节	记录时间的计数值，以 1/100 秒为单位
BITS	—	比特串
Opaque	可变	不解释的串

7. 简单网络管理协议 SNMP

□ TLV 举例:

- INTEGER 15, 其 T 字段是 02, INTEGER 类型要用 4 字节编码

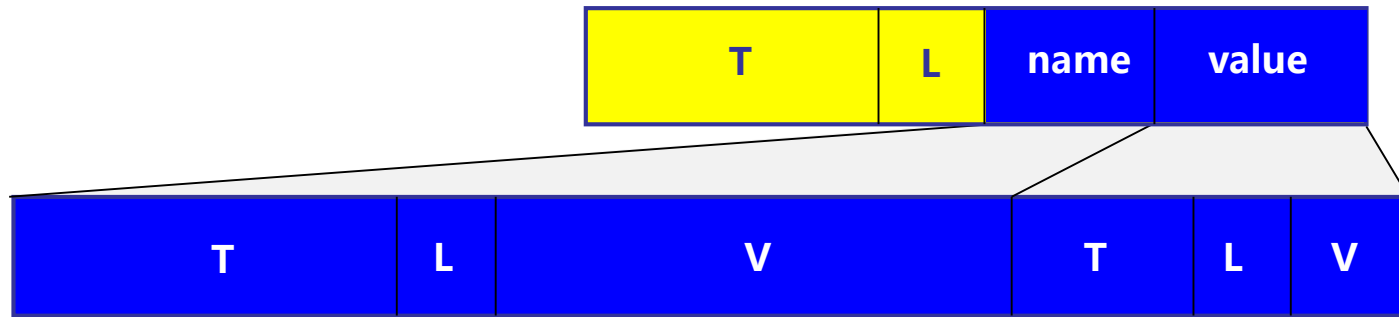
T 02	L 04	V 00 00 00 0F
-----------------------	-----------------------	--------------------------------

- IPAddress 192.1.2.3, 其 T 字段是 40, V 字段需要 4 字节表示

T 40	L 04	V C0 01 02 03
-----------------------	-----------------------	--------------------------------

7. 简单网络管理协议 SNMP

- TLV 中的 V 字段可嵌套其他数据元素的 TLV 字段



7. 简单网络管理协议 SNMP

- 被管对象必须维持可供管理程序读写的若干控制和状态信息。这些信息总称为管理信息库 MIB (Management Information Base) 。
- 管理程序使用 MIB 中这些信息的值对网络进行管理（如读取或重新设置这些值）。
- 只有在 MIB 中的对象才是 SNMP 所能够管理的。

7. 简单网络管理协议 SNMP

7.4 MIB

节点 mib-2 所包含的信息类别举例

类别	标号	所包含的信息
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)	IP 软件
icmp	(5)	ICMP 软件
tcp	(6)	TCP 软件
udp	(7)	UDP 软件
egp	(8)	EGP 软件

7. 简单网络管理协议 SNMP

MIB 变量的例子

MIB 变量	所属类别	意义
sysUpTime	system	距上次重启的时间
ifNumber	interfaces	网络接口数
ifMtu	interfaces	特定接口的最大传送单元 MTU
ipDefaultTTL	ip	IP 在生存时间字段中使用的值
ipInReceives	ip	接收到的数据报数目
ipForwDatagrams	ip	转发的数据报数目
ipOutNoRoutes	ip	路由选择失败的数目
ipReasmOKs	ip	重装的数据报数目
ipFragOKs	ip	分片的数据报数目
ipRoutingTable	ip	IP 路由表
icmpInEchos	icmp	收到的 ICMP 回送请求数目
tcpRtoMin	tcp	TCP 允许的最小重传时间
tcpMaxConn	tcp	允许的最大 TCP 连接数目
tcpInSegs	tcp	已收到的 TCP 报文段数目
udpInDatagrams	udp	已收到的 UDP 数据报数目

7. 简单网络管理协议 SNMP

7.5 SNMP 协议数据单元与报文

- SNMP 的操作只有两种基本的管理功能。
 - “读”操作，用 get 报文来检测各被管对象的状况
 - “写”操作，用 set 报文来改变各被管对象的状况
- SNMP 的这些功能通过**探测**操作来实现。
 - 探测操作：SNMP 管理进程定时向被管理设备周期性地发送探测信息。
 - 探测的好处：
 - 可使系统相对简单。
 - 能限制通过网络所产生的管理信息的通信量。
 - 探测的缺点：
 - 不够灵活，而且所能管理的设备数目不能太多。
 - 开销也较大。

7. 简单网络管理协议 SNMP

7.5 SNMP 协议数据单元与报文

- SNMP 陷阱 (SNMP trap)
 - SNMP 不是完全的探询协议，它允许不经过询问就能发送某些信息。这种信息称为陷阱，表示它能够捕捉“事件”。
 - 当被管对象的代理检测到有事件发生时，就检查其门限值。代理只向管理进程报告达到某些门限值的事件（即过滤）。
 - 过滤的好处是：
 - 仅在严重事件发生时才发送陷阱
 - 陷阱信息很简单且所需字节数很少

7. 简单网络管理协议 SNMP

7.5 SNMP 协议数据单元与报文

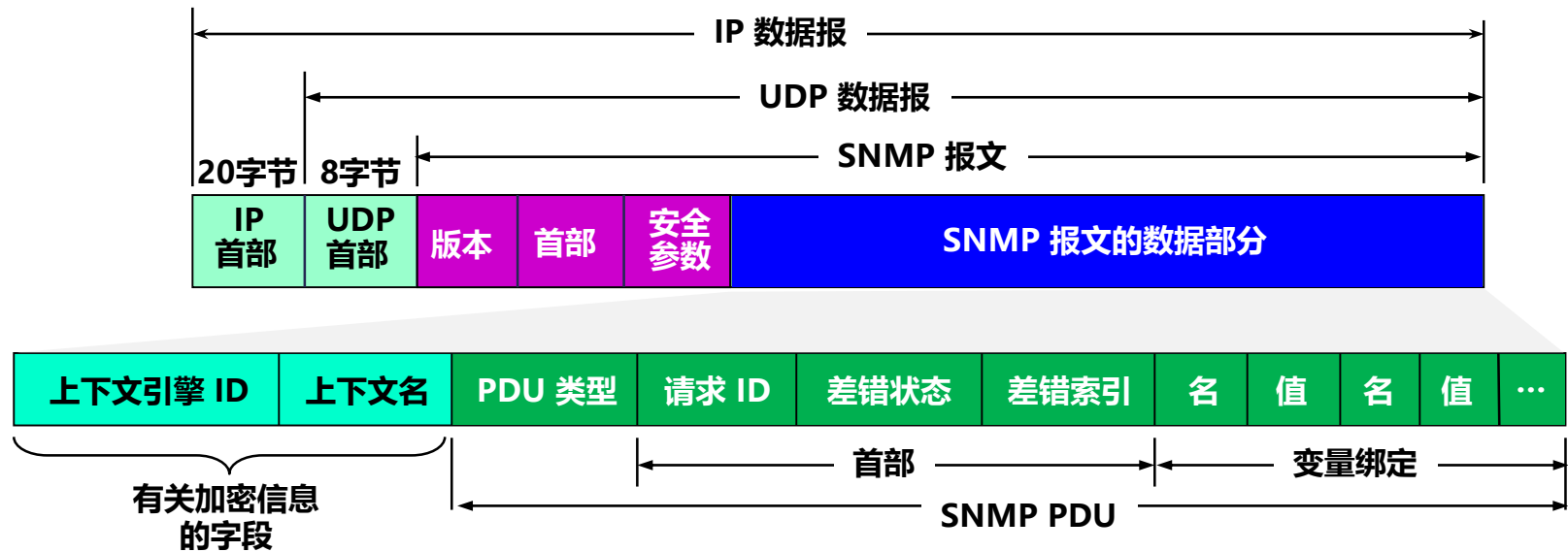
- SNMP 是有效的网络管理协议
 - 使用探询（至少是周期性地）以维持对网络资源的实时监视。
 - 同时也采用陷阱机制报告特殊事件，使得 SNMP 成为一种有效的网络管理协议。

- SNMP 使用无连接的 UDP
 - SNMP 使用无连接的 UDP，不保证可靠交付。
 - 在运行代理程序的服务器端用熟知端口 161 来接收 get 或 set 报文和发送响应报文（与熟知端口通信的客户端使用临时端口）。
 - 运行管理程序的客户端使用熟知端口 162 来接收来自各代理的 trap 报文。

7. 简单网络管理协议 SNMP

7.5 SNMP 协议数据单元与报文

SNMP 的报文格式



7. 简单网络管理协议 SNMP

7.5 SNMP 协议数据单元与报文

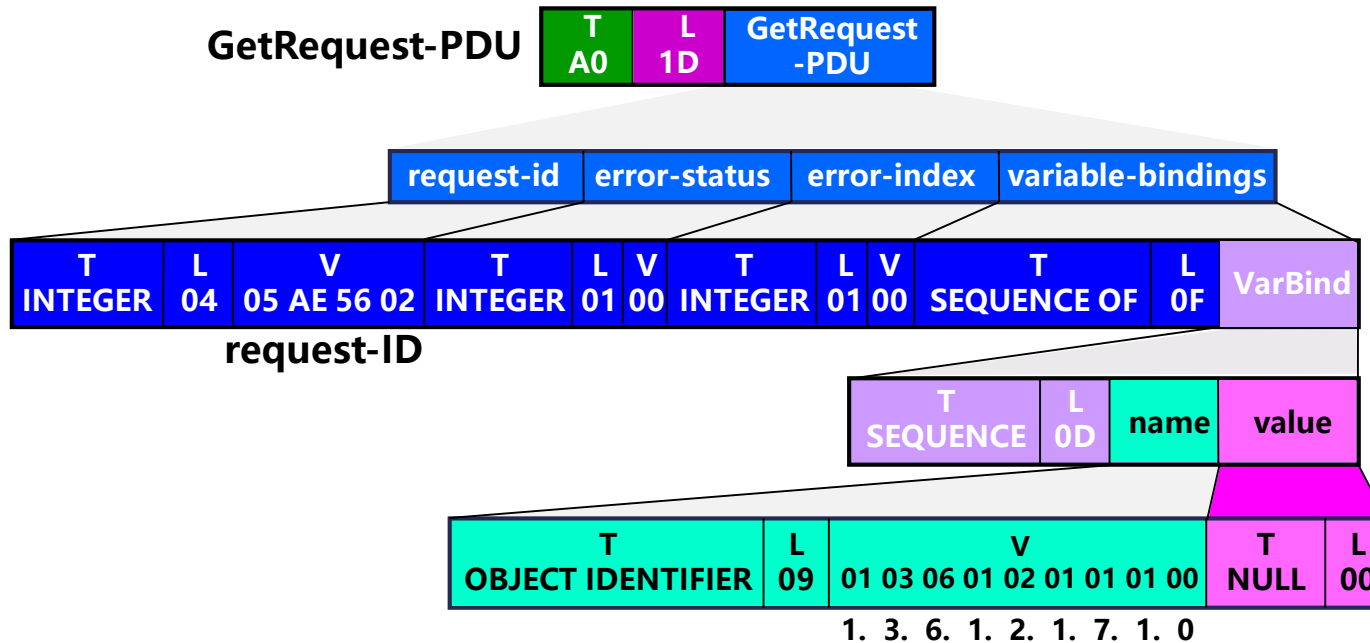
SNMPv1 定义的协议数据单元 (PDU) 类型

PDU 编号 (T字段)	PDU 名称	用途
0 (A0)	GetRequest	用来查询一个或一组变量的值
1 (A1)	GetNextRequest	允许在 MIB 树上读取下一个变量，此操作可反复进行
2 (A2)	Reponse	代理向管理者或管理者向管理者发送响应
3 (A3)	SetRequest	对一个或多个变量值进行设置
5 (A5)	GetBulkRequest	管理者从代理读取大数据块的值
6 (A6)	InformRequest	管理者从另一管理者读取代理的变量
7 (A7)	SNMPv2Trap	代理向管理者报告异常事件
8 (A8)	Report	管理者之间报告某些差错

7. 简单网络管理协议 SNMP

7.5 SNMP 协议数据单元与报文

Get-request 报文 ASN.1 编码



8. 应用进程跨越网络的通信

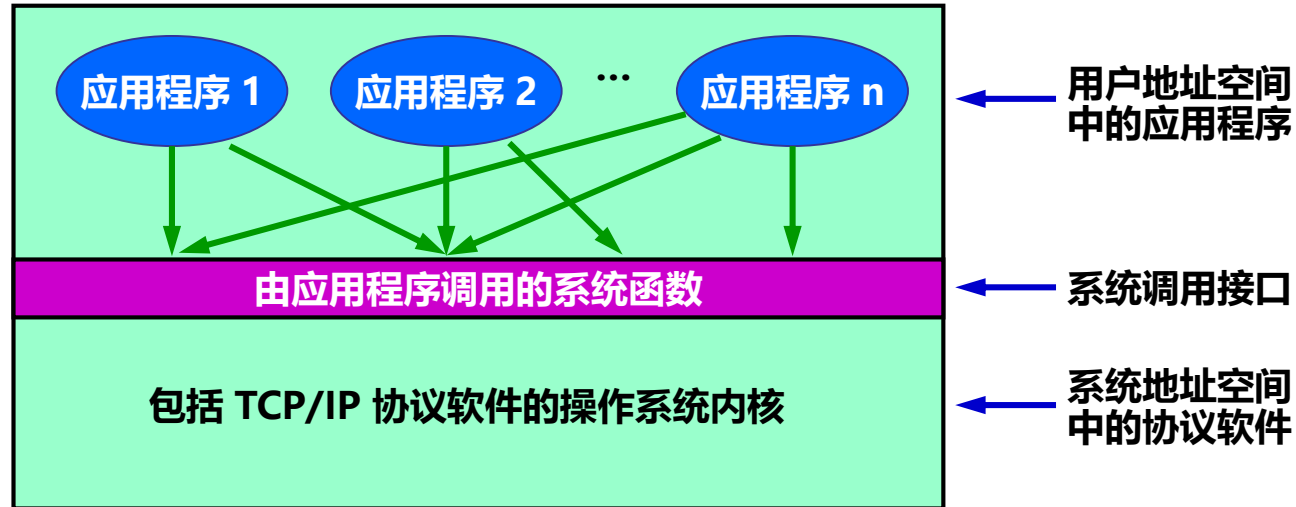
8.1 系统调用和应用编程接口

- 大多数操作系统使用系统调用 (system call) 的机制在应用程序和操作系统之间传递控制权。
- 对程序员来说，每一个系统调用和一般程序设计中的函数调用非常相似，只是系统调用是将控制权传递给了操作系统。

8. 应用进程跨越网络的通信

8.1 系统调用和应用编程接口

多个应用进程使用系统调用的机制



8. 应用进程跨越网络的通信

8.1 系统调用和应用编程接口

□ 应用编程接口 API

- 系统调用接口实际上就是应用进程的控制权和操作系统的控制权进行转换的一个接口。
- 使用系统调用之前要编写一些程序，特别是需要设置系统调用中的许多参数，因此这种系统调用接口又称为应用编程接口 API (Application Programming Interface)。

8. 应用进程跨越网络的通信

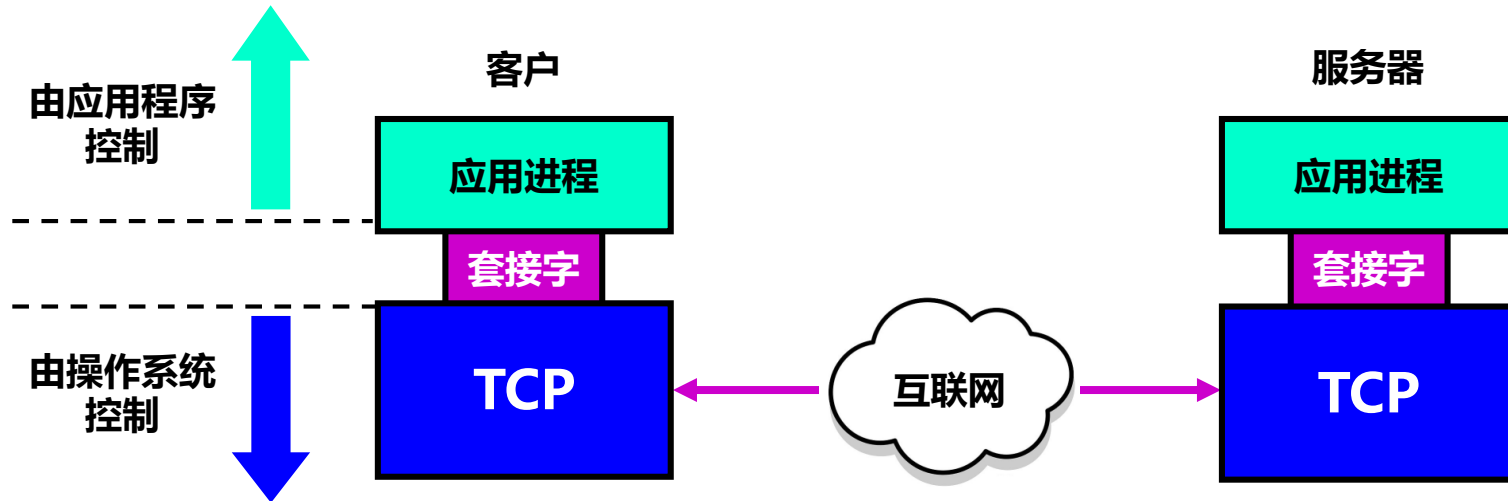
8.1 系统调用和应用编程接口

- 几种应用编程接口 API
 - Berkeley UNIX 操作系统定义了一种 API，它又称为套接字接口 (socket interface)。
 - 微软公司在其操作系统中采用了套接字接口 API，形成了一个稍有不同的 API，并称之为 Windows Socket。
 - AT&T 为其 UNIX 系统 V 定义了一种 API，简称为 TLI (Transport Layer Interface)。

8. 应用进程跨越网络的通信

8.1 系统调用和应用编程接口

应用进程通过套接字接入到网络



8. 应用进程跨越网络的通信

8.1 系统调用和应用编程接口

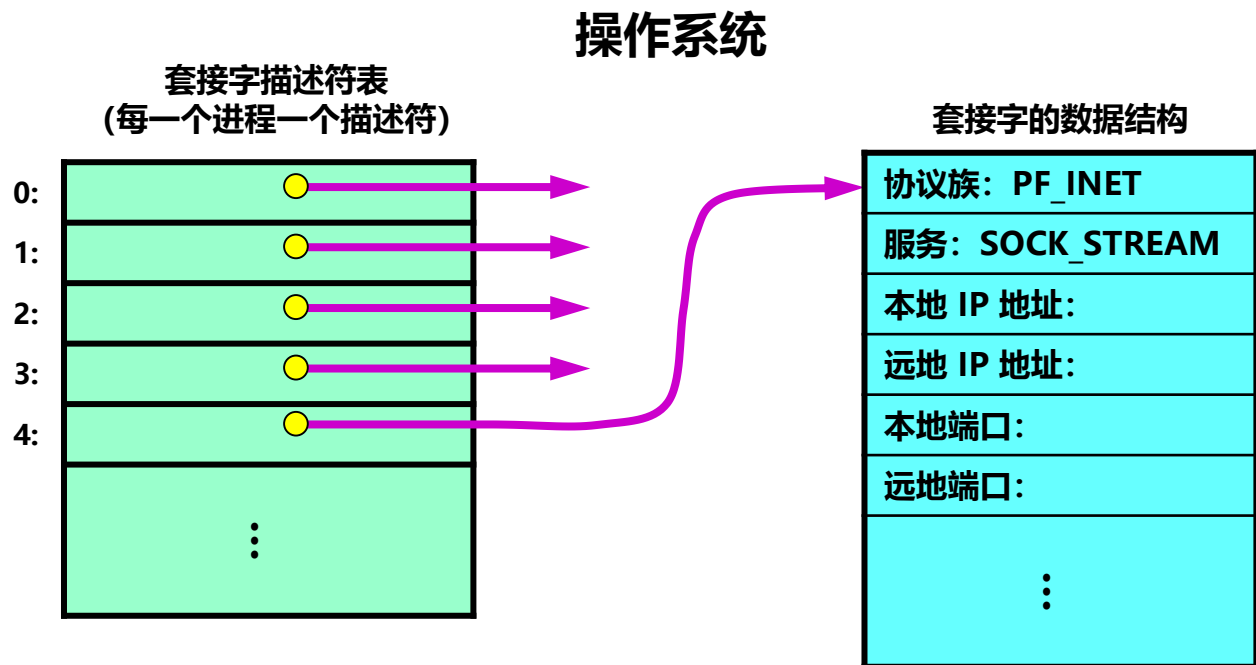
□ 套接字的作用

- 当应用进程需要使用网络进行通信时就发出系统调用，请求操作系统为其创建套接字，以便把网络通信所需要的系统资源分配给该应用进程。
- 操作系统为这些资源的总和用一个套接字描述符的号码来表示。
- 应用进程所进行的网络操作都必须使用这个套接字描述符。
- 通信完毕后，应用进程通过一个关闭套接字的系统调用通知操作系统回收与该套接字描述符相关的所有资源。

8. 应用进程跨越网络的通信

8.1 系统调用和应用编程接口

调用 socket 创建套接字



8. 应用进程跨越网络的通信

8.2 几种常用的系统调用

- 当应用进程需要使用网络进行通信时，就发出系统调用。
 - 使用 TCP/IP 应用编程接口 API，就可编写基于互联网的网络应用程序。
 - 调用 API 时，用户可以使用 TCP 服务，也可以使用 UDP 等其他服务。
- TCP 提供面向连接的服务。
 - 使用 TCP 服务需要经历 3 个阶段：
 - 连接建立阶段
 - 数据传送阶段
 - 连接释放阶段

8. 应用进程跨越网络的通信

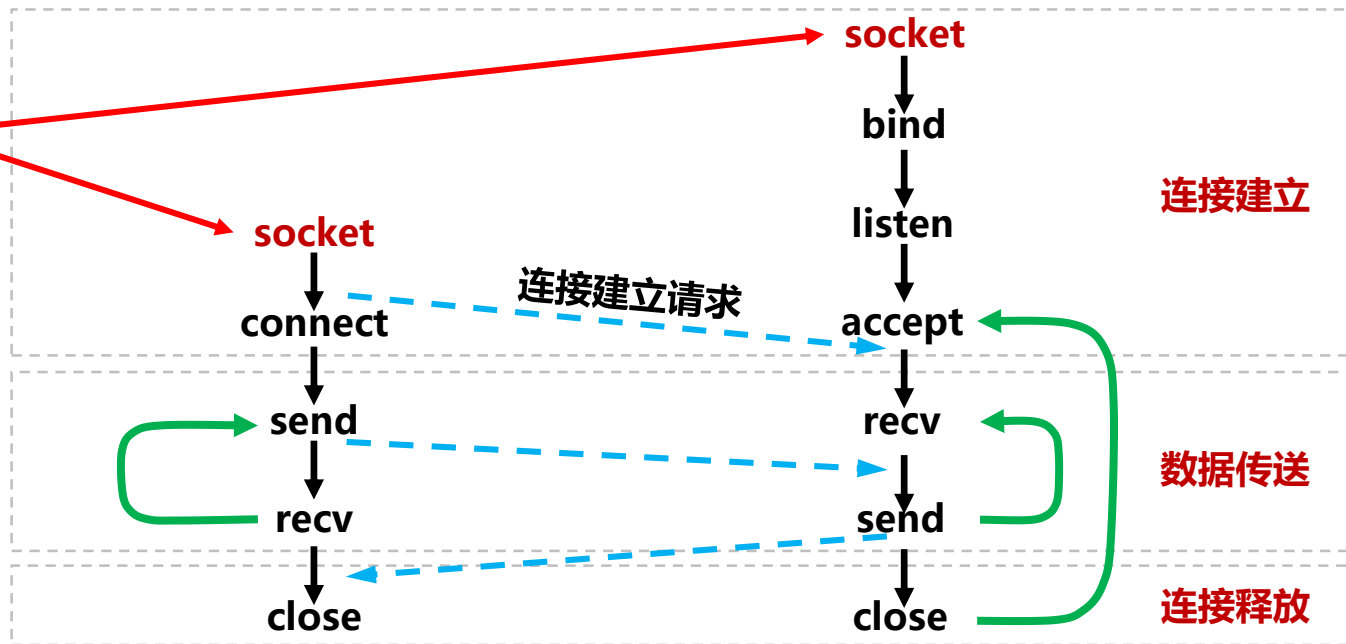
8.2 几种常用的系统调用

1. 连接建立阶段

客户端

服务器端

通信之前，客户和服务先创建套接字



8. 应用进程跨越网络的通信

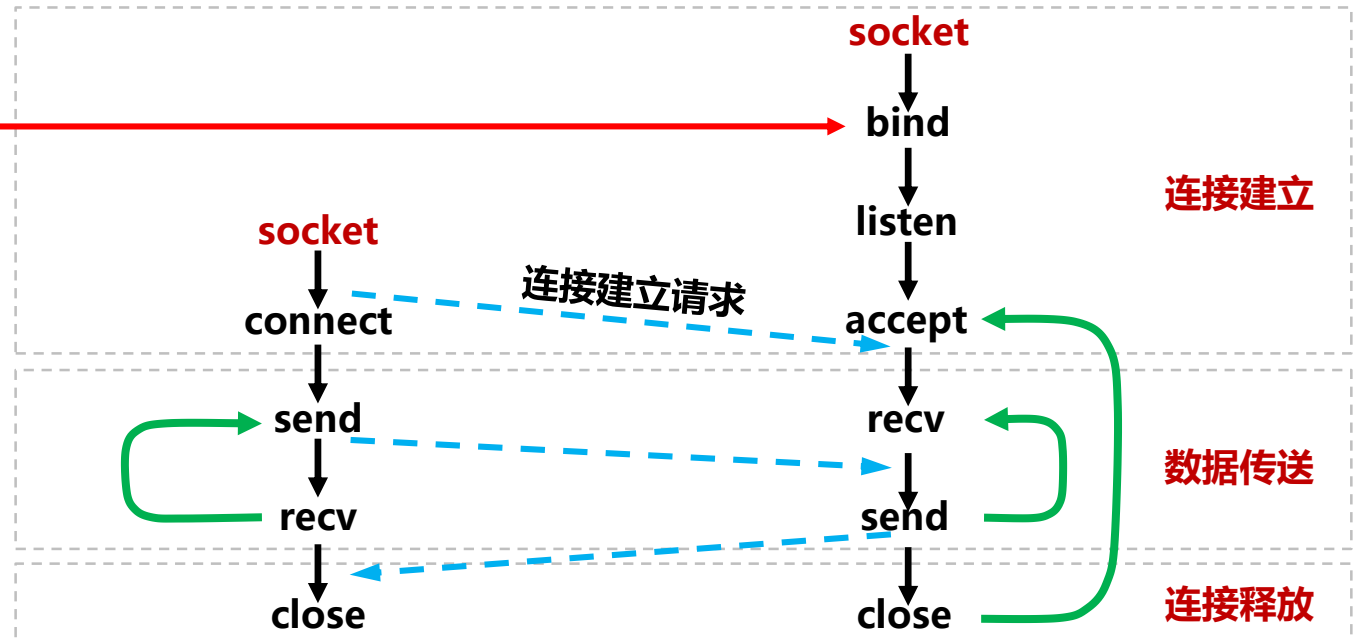
8.2 几种常用的系统调用

1. 连接建立阶段

客户端

服务器端

服务器端调用 bind, 把熟知端口号和本地 IP 地址填写到已创建的套接字中。

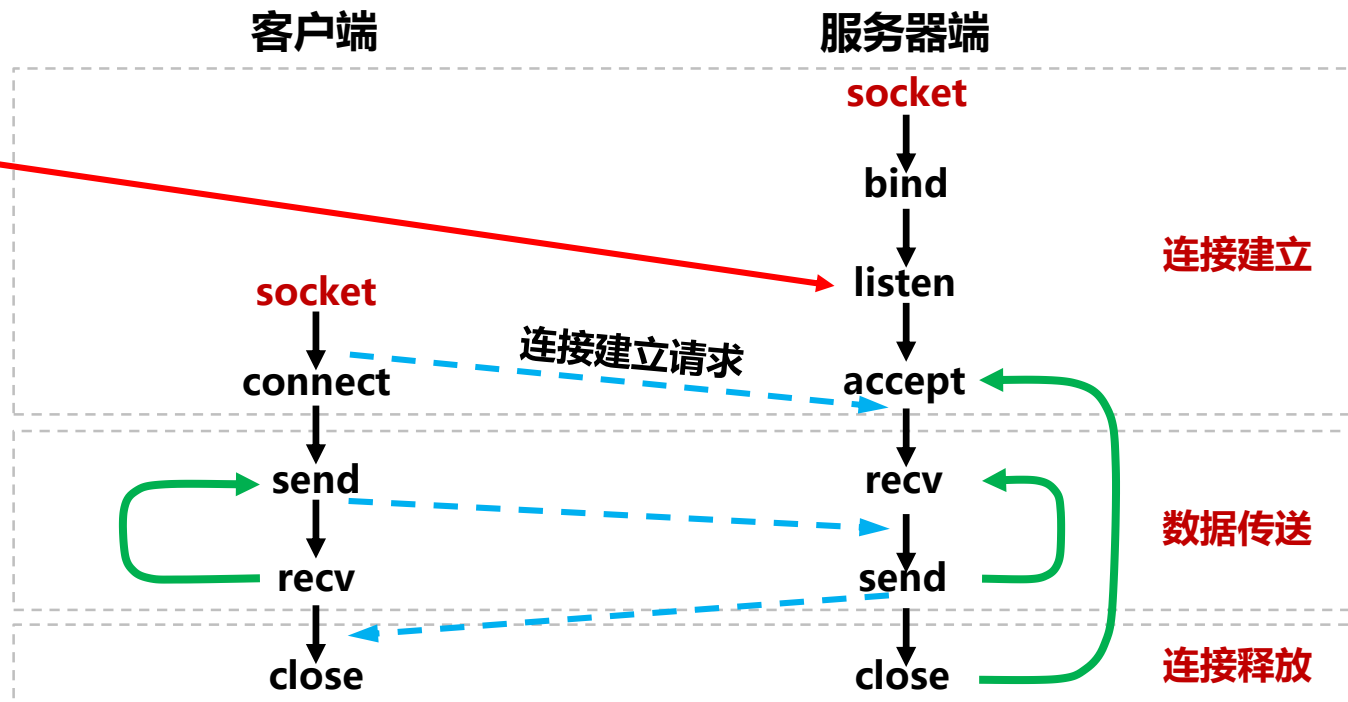


8. 应用进程跨越网络的通信

8.2 几种常用的系统调用

1. 连接建立阶段

服务器端调用 `listen` (收听), 把套接字设置为被动方式, 以便随时接受客户的服务请求。
UDP 服务器不使用 `listen` 系统调用。

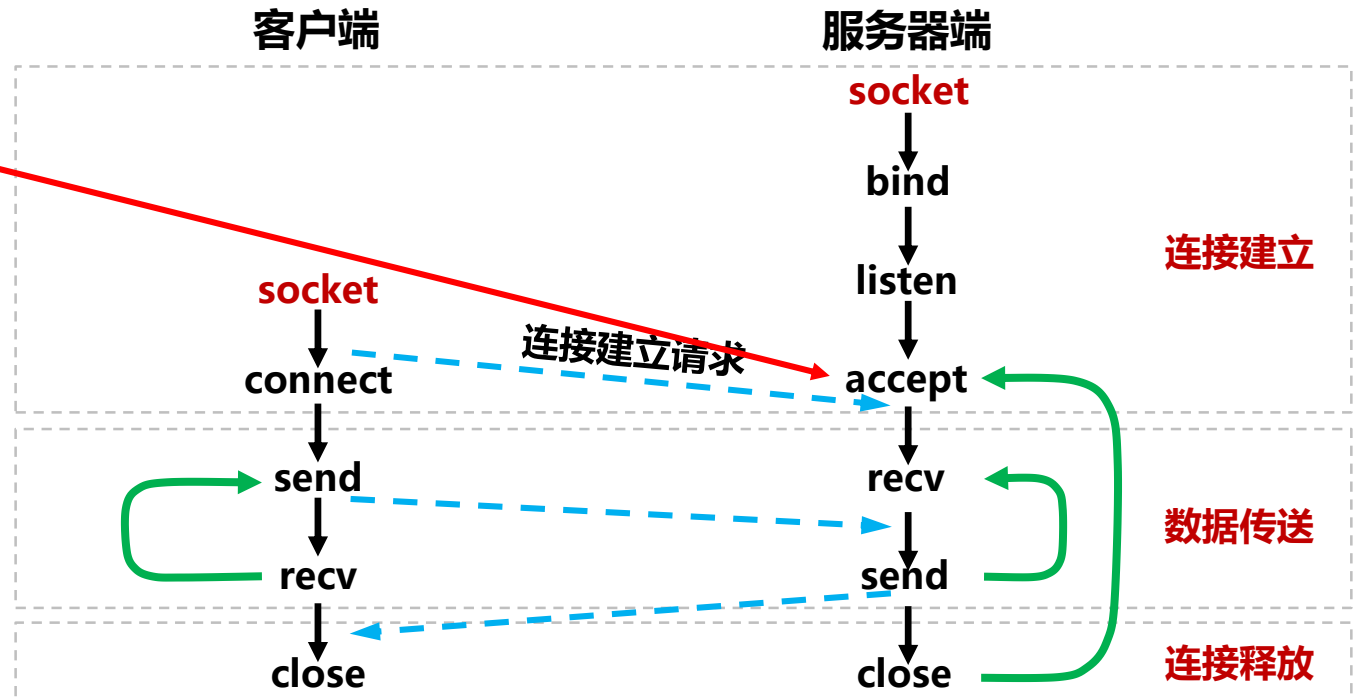


8. 应用进程跨越网络的通信

8.2 几种常用的系统调用

1. 连接建立阶段

服务器端调用 `accept` (接受), 以便把远地客户进程发来的连接请求提取出来。UDP 服务器不使用 `accept` 系统调用



8. 应用进程跨越网络的通信

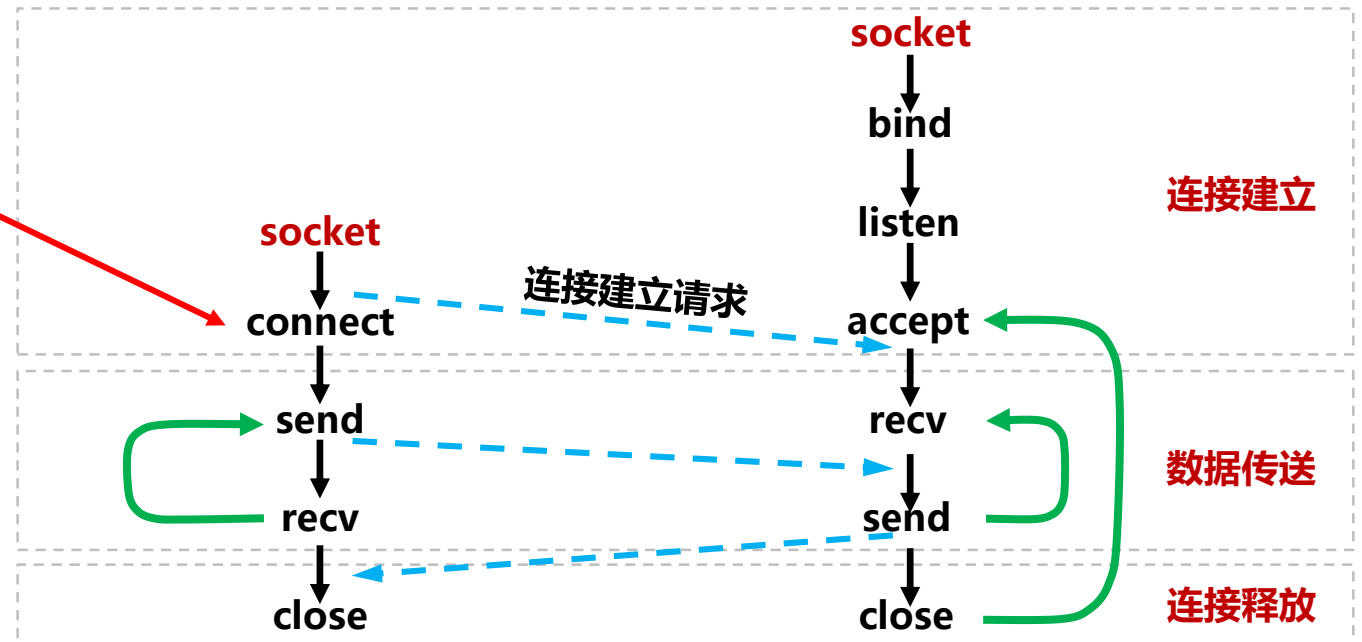
8.2 几种常用的系统调用

1. 连接建立阶段

客户端

服务器端

客户进程调用 `connect`,
以便和远地服务器建立连接
(这就是主动打开)。



8. 应用进程跨越网络的通信

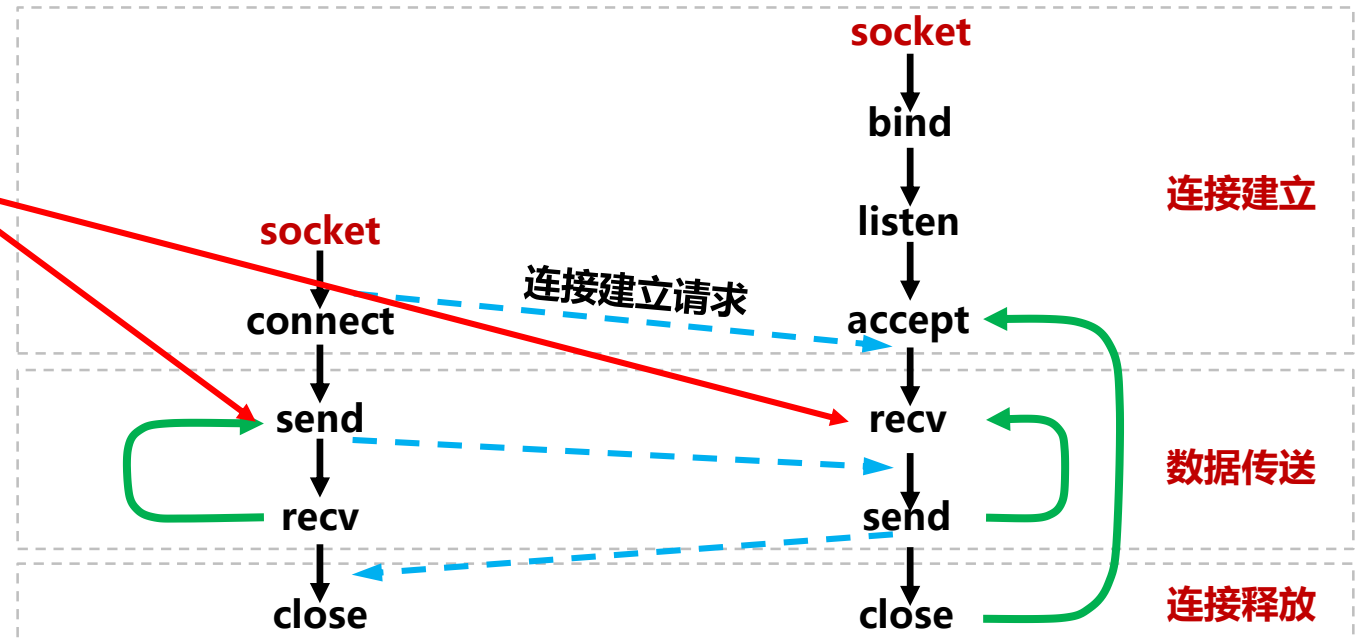
8.2 几种常用的系统调用

2. 传送阶段

客户端

服务器端

客户和服务在 TCP 连接上使用 send 传送数据，使用 recv 接收数据。



8. 应用进程跨越网络的通信

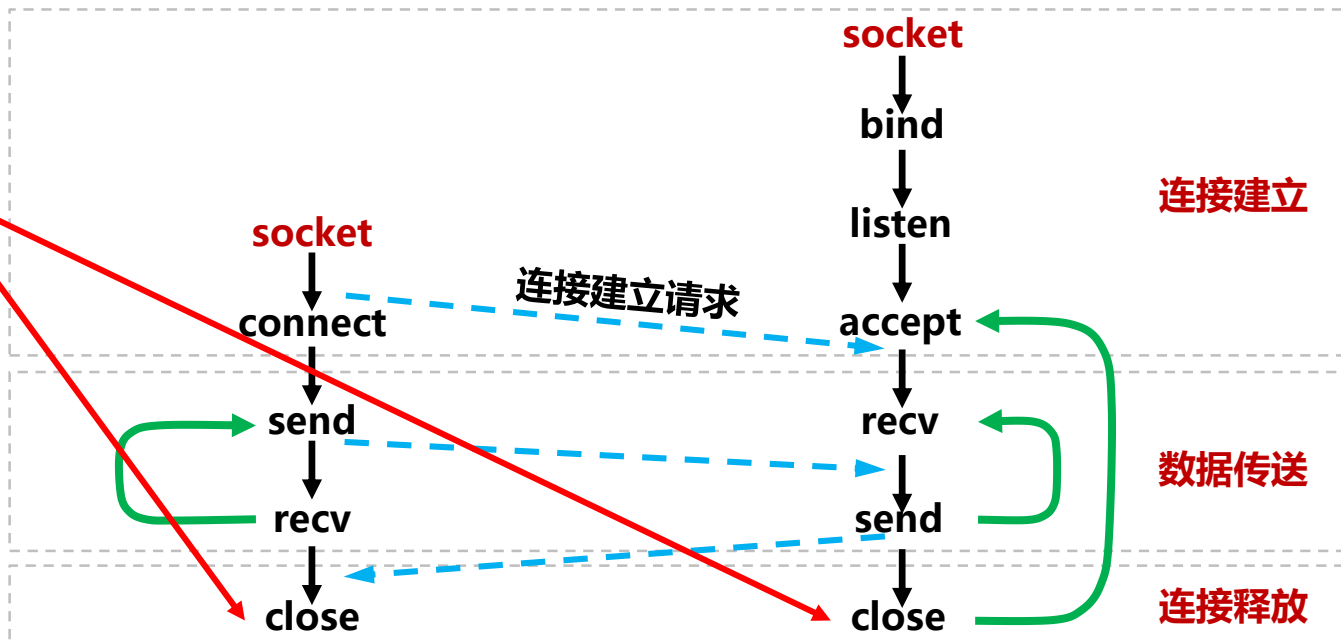
8.2 几种常用的系统调用

2. 传送阶段

客户端

服务器端

客户或服务器通信结束，调用 close 释放连接和撤销套接字。

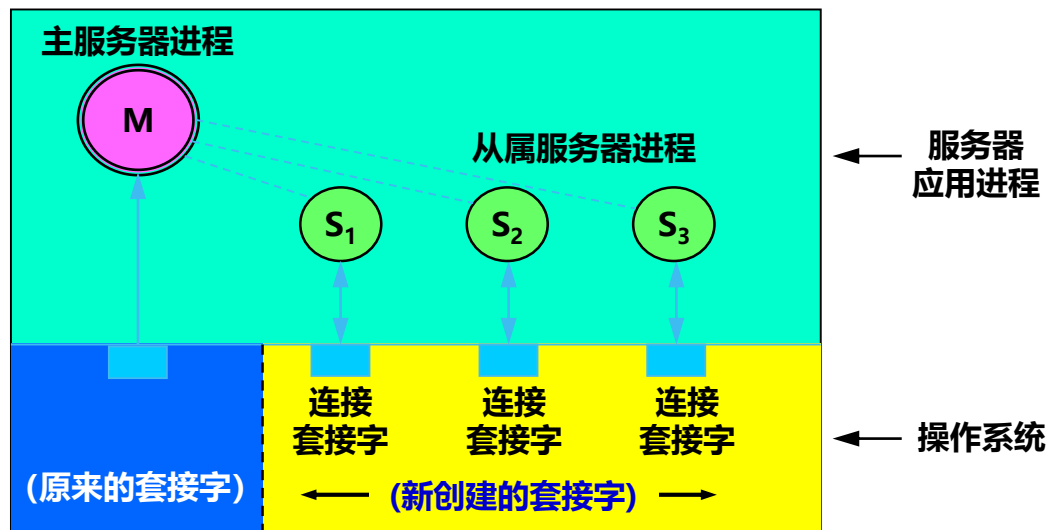


8. 应用进程跨越网络的通信

8.2 几种常用的系统调用

并发方式工作的服务器

- 调用 `accept` 要完成的动作较多。这是因为一个服务器必须能够同时处理多个连接。
- 这样的服务器常称为并发方式 (concurrent) 工作的服务器。





智能运维课程体系

