

6.1网络应用模型

客户/服务器模型

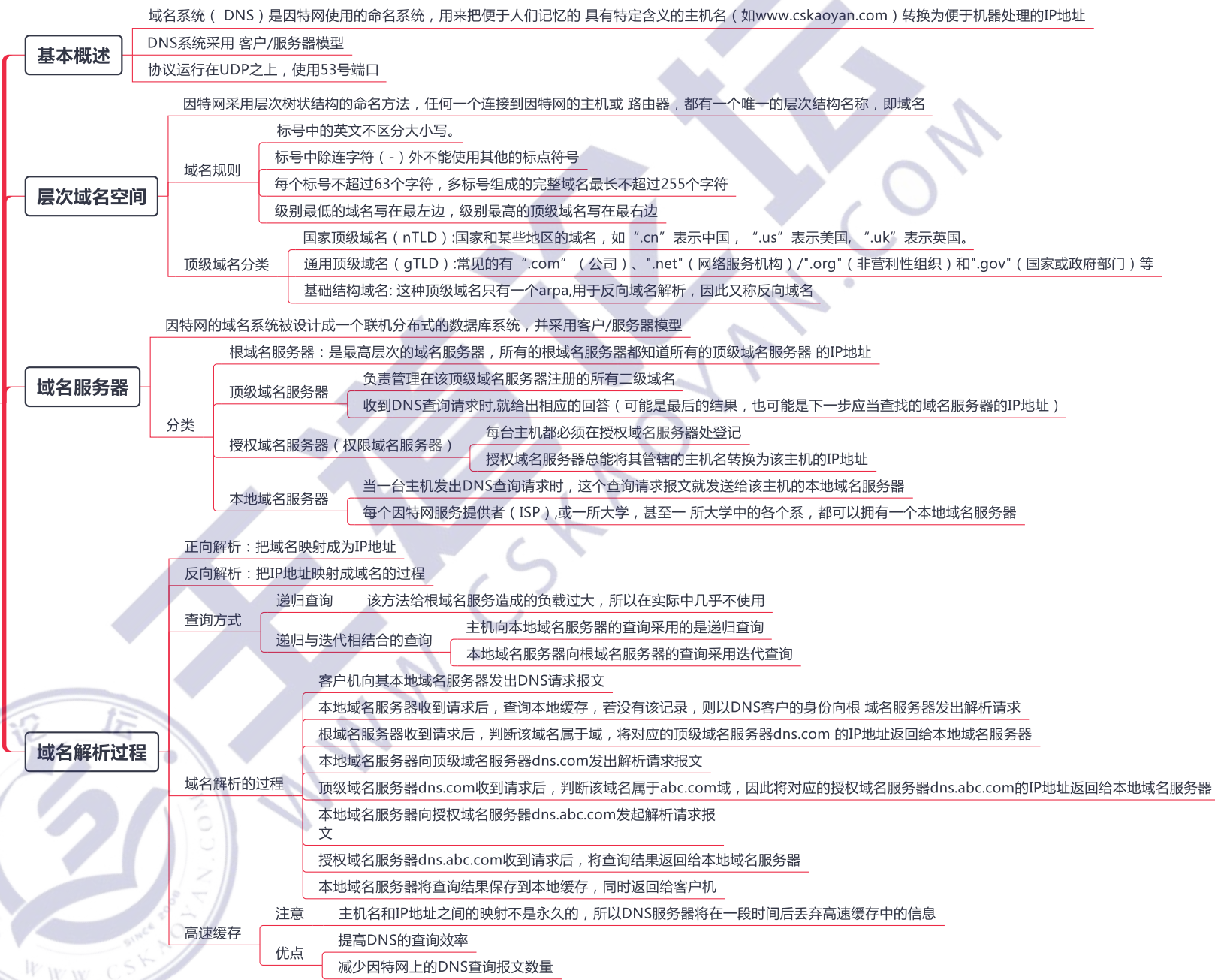
- 概述 在客户/服务器（C/S）模型中，有一个总是打开的主机称为服务器，它服务于许多来自其他称为客户机的主机请求
- 工作流程
 - 服务器处于接收请求的状态
 - 客户机发出服务请求，并等待接收结果
 - 服务器收到请求后，分析请求，进行必要的处理，得到结果并发送给客户机
- 主要特征
 - 客户是服务请求方，服务器是服务提供方
 - 网络中各计算机的地位不平等，服务器可以通过对用户权限的限制来达到管理客户机的目的，使它们不能随意存储/删除数据，或进行其他受限的网络活动
 - 客户机相互之间不直接通信
 - 可扩展性不佳
- 客户/服务器模型的应用
 - Web
 - 文件传输协议（FTP）
 - 远程登录
 - 电子邮件

P2P模型

- 概述
 - 各计算机没有固定的客户和服务划分
 - 任意一对计算机——称为对等方（Peer），直接相互通信
 - 每个结点既作为客户访问其他结点的资源，也作为服务器提供资源给其他结点访问
- P2P 应用
 - PPlive
 - Bittorrent
 - 电驴
- 优点
 - 减轻了服务器的计算压力，消除了对某个服务器的完全依赖，提高了系统效率和资源利用率
 - 多个客户机之间可以直接共享文档
 - 可扩展性好，传统服务器有响应和带宽的限制，只能接受一定数量的请求
 - 网络健壮性强，单个结点的失效不会影响其他部分的结点
- 缺点
 - 占用较多的内存，影响整机速度
 - P2P下载会对硬盘造成较大的损伤
 - 使网络变得非常拥塞

各大isp（互联网服务提供商，如电信、网通等）通常都对p2p应用持反对态度

6.2域名系统（DNS）



6.3文件传输协议（FTP）

FTP的工作原理

- 概述
 - FTP 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的任意计算机之间传送文件
 - FTP采用客户/服务器的工作方式，它使用TCP可靠的传输服务
- FTP功能
 - 提供不同种类主机系统（硬、软件体系等都可以不同）之间的文件传输能力
 - 以用户权限管理的方式提供用户对远程FTP服务器上的文件管理能力
 - 以匿名FTP的方式提供公用文件共享的能力
- FTP进程组成
 - 主进程：负责接收新的请求
 - 若干从属进程：处理单个请求
- 工作步骤
 - 打开熟知端口 21（控制端口），使客户进程能够连接上
 - 等待客户进程发连接请求
 - 启动从属进程来处理客户进程发来的请求。主进程与从属进程并发执行，从属进程对客户进程的请求处理完毕后即终止
 - 回到等待状态，继续接收其他客户进程的请求

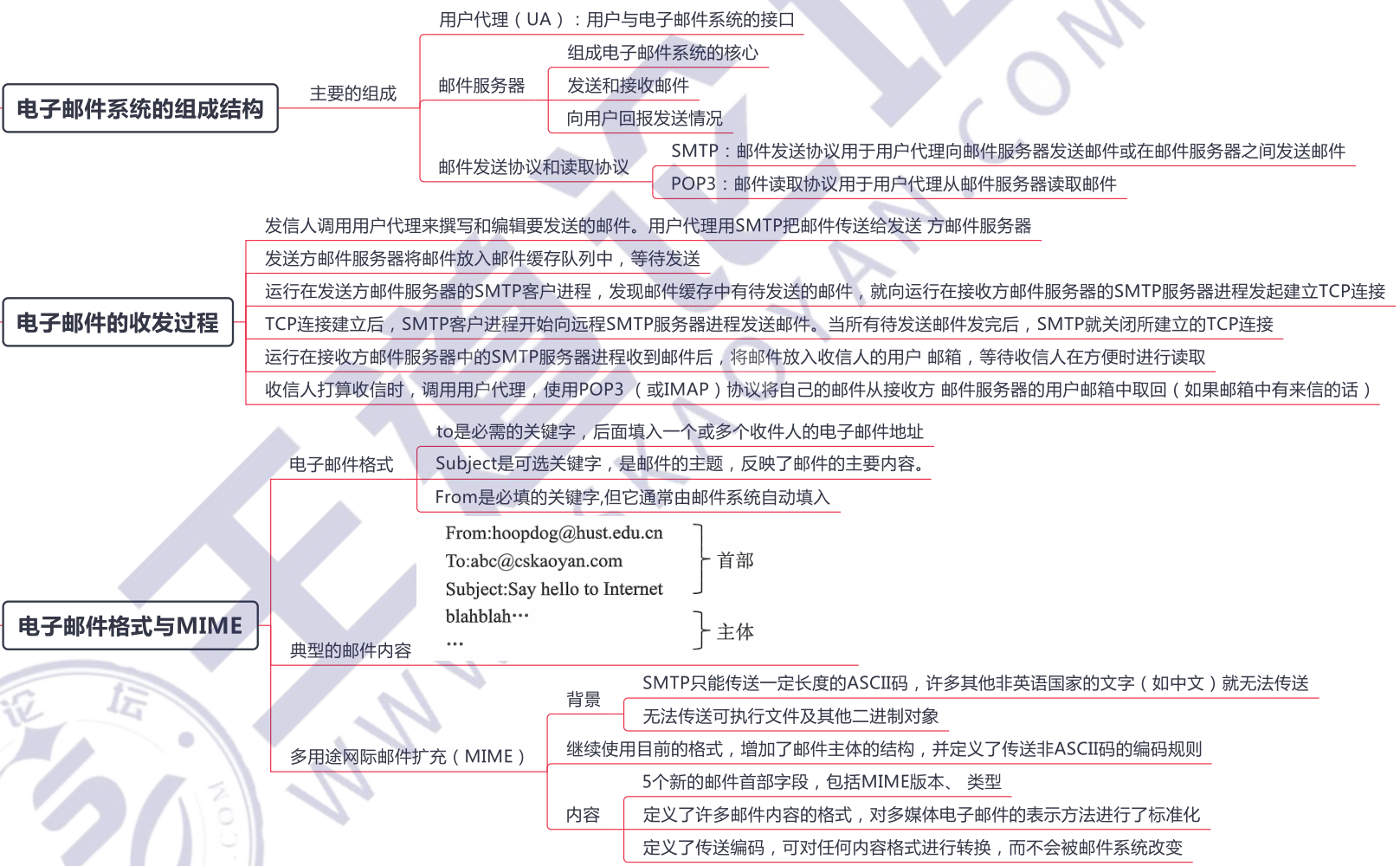
ftp服务器必须在整个会话期间保留用户的状态信息

控制连接与数据连接

- 特点：FTP在工作时使用两个并行的TCP连接
- 控制连接（端口号21）
 - 服务器监听21号端口，等待客户连接，建立在这个端口上的连接称为控制连接
 - 控制连接用来传输控制信息（如连接请求、传送请求等），并且控制信息都以7位ASCII格式传送
 - 控制连接 在整个会话期间一直保持打开状态
- 数据连接（端口号20）
 - 服务器端的控制进程在接收到FTP客户发来的文件传输请求后，就创建数据传送进程和数据连接
 - 传送完毕后关闭“数据传送连接”并结束运行

使用两个不同的端口号可使协议更加简单和更容易实现

6.4电子邮件（上）



6.4电子邮件（下）

SMTP 和 POP3

简单邮件传输协议（SMTP）

概述

一种提供可靠且有效的电子邮件传输的协议，它控制两个相互通信的SMTP进程交换信息
使用客户/服务器方式
发送邮件的SMTP进程是SMTP客户，而负责接收邮件的SMTP进程是SMTP服务器
TCP连接，端口号为25

通信阶段

连接建立

发件人的邮件发送到发送方邮件服务器的邮件缓存中后，SMTP客户就每隔一定时间对邮件缓存扫描一次
发现有邮件，就使用SMTP的熟知端口号（25）与接收方邮件服务器的SMTP服务器建立TCP连接
SMTP不使用中间邮件服务器。TCP连接总是在发送方和接收方这两个邮件服务器之间直接建立

邮件传送

连接建立后，就可开始传送邮件
邮件的传送从MAIL命令开始，MAIL命令后面有发件人的地址
SMTP服务器已准备好接收邮件，则回答250 OK，接着SMTP客户端发送一个或多个RCPT（收件人recipient的缩写）命令
RCPT命令的作用：先弄清接收方系统是否已做好接收邮件的准备，然后才发送邮件，避免浪费通信资源
获得OK的回答后，客户端就使用DATA命令，表示要开始传输邮件的内容

连接释放

邮件发送完毕后，SMTP客户应发送QUIT命令
SMTP服务器返回的信息是221（服务关闭），表示SMTP同意释放TCP连接

缺点

- 1.SMTP不能传送可执行文件或者其他二进制对象。
- 2.SMTP仅限于传送7位ASCII码，不能传送其他非英语国家的文字。
- 3.SMTP服务器会拒绝超过一定长度的邮件。

POP3

概述

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3个版本POP3
当用户读取邮件时，用户代理向邮件服务器发出请求，拉取用户邮箱中的邮件

工作方式

- 下载并保留
- 下载并删除

因特网报文存取协议（IMAP）

邮件接收协议，比POP复杂得多

用户提供了创建文件夹、在不同文件夹之间移动邮件及在远程文件夹中查询邮件的命令，为此IMAP服务器维护了会话用户的状态信息
允许用户代理只获取报文的某些部分

基于万维网的电子邮件

特点

用户浏览器与Hotmail或Gmail的邮件服务器之间的邮件发送或接收使用HTTP
在不同邮件服务器之间传送邮件时使用SMTP

6.5万维网(WWW) (上)

WWW的概念与组成结构

- 概念
 - 万维网 (WWW) 在这个空间中, 有用的事物称为资源, 并由一个全域“统一资源定位符”(URL) 标识。
 - 这些资源通过超文本传输协议 (HTTP) 传送给使用者, 而后者通过单击链接来获取资源
- 内核部分标准构成
 - 统一资源定位符 (URL) : 负责标识万维网上的各种文档, 并使每个文档在整个万维网的 范围内具有唯一的标识符URL
 - 超文本传输协议 (HTTP) : 一个应用层协议, 它使用TCP连接进行可靠的传输, HTTP 是万维网客户程序和服务器程序之间交互所必须严格遵守的协议。
 - 超文本标记语言 (HTML) : 一种文档结构的标记语言, 它使用一些约定的标记对页面上 的各种信息 (包括文字、声音、图像、视频等)、格式进行描述
- 工作流程
 - Web用户使用浏览器 (指定URL) 与Web服务器建立连接, 并发送浏览请求
 - Web服务器把URL转换为文件路径, 并返回信息给Web浏览器
 - 通信完成, 关闭连接

超文本传输协议 (HTTP)

- 概述
 - HTTP定义了浏览器 (万维网客户进程) 怎样向万维网服务器请求万维网文档, 以及服务器怎样把文档传送给浏览器
 - HTTP是面向事务的应用层协议
 - 规定了在浏览器和服务器之间的请求和响应的格式与规则
 - 是万维网上能够可靠地交换文件的重要基础
- HTTP的操作过程
 - 每个万维网站点都有一个服务器进程, 它不断地监听TCP的端口 80 (默认)
 - 当监听到连接请求后便与浏览器建立连接
 - TCP连接建立后, 浏览器就向服务器发送请求获取某个Web页面 的HTTP请求
 - 服务器收到HTTP请求后, 将构建所请求Web页的必需信息, 并通过HTTP 响应返回给浏览器
 - 浏览器再将信息进行解释, 然后将Web页显示给用户
 - TCP连接释放
- HTTP报文
 - 请求报文
 - 响应报文
- 用户单击鼠标后发生的事件按顺序
 - 浏览器分析链接指向页面的 URL。
 - 浏览器向DNS请求解析域名的IP地址。
 - 域名服务器DNS解析出IP地址
 - 浏览器与该服务器建立TCP连接 (默认端口号为80)
 - 浏览器发出 HTTP 请求:GET /chn/index.html
 - 服务器通过HTTP响应把文件index.htm发送给浏览器
 - TCP连接释放
 - 浏览器解释文件index.htm,并将Web页显示给用户

6.5万维网(WWW) (下)

HTTP的特点

- HTTP是无状态的，同一个客户第二次访问同一个服务器上的页面时，服务器的响应与第一次被访问时的相同
- HTTP的无状态特性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
- 使用Cookie加数据库的方式来跟踪用户的活动
- HTTP采用TCP作为运输层协议，保证了数据的可靠传输
- HTTP既可以使用非持久连接，也可以使用持久连接（HTTP/1.1支持）

持久连接与非持久连接

- 非持久连接：每个网页元素对象（如JPEG图形、Flash等）的传输都需要单独建立一个TCP连接
- 持久连接：是指万维网服务器在发送响应后仍然保持这条连接，使同一个客户和服务器可以继续在这条连接上传送后续的HTTP请求与响应报文
- 持久连接
 - 非流水线 客户在收到前一个响应后 才能发出下一个请求
 - 流水线 客户每遇到一个对象引用，就立即发出一个请求

HTTP的报文结构

HTTP是面向文本的

报文类型

- 请求报文：从客户向服务器发送的请求报文。
- 响应报文：从服务器到客户的回答

方法（操作）	意 义
GET	请求读取由 URL 标识的信息
HEAD	请求读取由 URL 标识的信息的首部
POST	给服务器添加信息（如注释）
CONNECT	用于代理服务器

状态码

- 1xx表示通知信息的，如请求收到了或正在处理。
- 2xx表示成功，如接受或知道了。
- 3xx表示重定向，如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