

Q：一个基于重传的可靠传输协议通常包含以下要素：差错编码，确认，重传，定 时器，分组序号。请解释为什么需要这些要素

- 差错编码：差错编码提供了一种发现传输错误的机制

- 确认：确认提供了一种反馈机制，使得发送方可以得知接收方的接收情况

- 重传：重传提供了修正错误的方法，即用新的正确的包替换错误的

- 定时器：定时器提供了发现/定义超时的方法，避免死锁无限等待

- 分组序号：分组序号提供了区分不同分组的方式，流水线操作不会引起混乱

小测题目：

1. **面向连接服务的优点缺点？**

优点：(1)可靠传输；(2)有序传输；(3)资源预置(使用)

缺点：需要全局信息

1. **无连接服务的优点与缺点？**

优点：无需知道网络状态(包括网络资源)或只需知道局部网络状态

缺点：具有不确定性(是否有满足服务的网络资源不确定，能否完成服务不确定)

3.**分层网络体系结构的不足：**

上层协议的性能依赖于下层协议

4.**分组交换原理：**

(1)存储转发;(2)动态路由(包括每个分组自带源地址、目的地址，拓扑发现、路由选择);(3)出错交由端系统处理

5**.若一个WWW文档中除有文本外，还有7个图像。试问使用单个http/1.0、4个并行http/1.0与单个http1.1各需要建立几次TCP连接？**

(1)8； (2)3； (3)1

6.假定要传送的报文共有x(单位bit)，从源节点到目的节点共有k跳链路，每条链路的传播时延为d(单位s)，链路带宽为b(单位bit/s)；电路交换(包括连接建立与拆除)使用的控制帧(或信令)长度、在各节点的排队时延忽略不计；分组交换使用的分组头、分组长度分别为h、p(单位bit)，分组在各节点的排队时延q(单位s)。试分析在何种条件下电路交换的总时延要小于分组交换的总时延？

电路交换总时延D(c)：

1. 连接建立时间：kd
2. 连接拆除时间：kd
3. 数据传输时间：x/b
4. 数据传播时间：kd

D(c)=3kd+x/b

-分组交换总时延D(p):

-单个分组传输时间：(p+h)/b

-第1跳传输时间：(x/p).((p+h)/b) (x/p为分组个数)

-传输时间每1跳增加1个分组的传输时间🡪总的传输-时间为x/p\*(p+h)/b+(k-1)\*(p+h)/b

-排队时间：kq

-传播时间：kd

D(p)=x/p\*(p+h)/b+(k-1)\*(p+h)/b+kd+kq

若D(c)<D(p)，则电路交换的总时延要小于分组交换的总时延

1.**回答TCP协议中可能使用的停等、回退N步（Go Back N）、选择重传三个滑动窗口协议有关问题：**

**答：(1)三个协议的发送窗口、接收窗口大小分别是多少？(2)发送窗口最大有效值由什么因素决定？**

答：(1)1、1；N、1：N、M

(2)网络缓冲能力（或RTT\*瓶颈节点带宽，即BDP）

2. **实现TCP连接目标的主要机制。**

答：(1)通过传输层地址(端口号)实现进程间通信

(2)通过确认机制实现可靠传送(3)通过接收方缓存实现按序传送(4)流量控制(5)拥塞控制(6)连接建立与拆除机制

3.在TCP连接中，客户端的初始号215。客户打开连接，只发送一个携带有300字节数据的报文段，然后关闭连接。试问下面从客户端发送的各个报文段的序号分别是多少？

(1)SYN报文段；(2)数据报文段；3)FIN报文段。

答：(1)215；(2)216；(3)516

4.在一条新建的TCP连接上发送一个长度为40KB的文件。发送端每次都发送一个最大长度的段（MSS），MSS的长度为1KB，接收端正确收到一个TCP段后立即给予确认。发送端的初始拥塞窗口门限设为16KB。假设发送端尽可能快地传输数据，即只要发送窗口允许，发送端就发送一个MSS。

(1)已知发生第一次超时后，发送端将拥塞窗口门限调整为4KB。请问发生超时的时候，发送端的拥塞窗口是多大？此时发送端共发送了多少数据？其中有多少数据被成功确认了？

(2)发送端从未被确认的数据开始使用慢启动进行重传。假设此后未再发生超时，当文件全部发送完毕时，发送端的拥塞窗口是多大？

答：(1) 第一次超时发生时，发送端拥塞窗口大小 = 4KB\*2 = 8KB

在新建立的TCP连接上，发送端采用慢启动开始发送，因此当第一次超时发生时，发送端已发送的数据量 = 1KB + 2KB + 4KB + 8KB = 15KB。

此时，除最后一批8个TCP段未获确认外，之前发送的TCP段都被确认，因此成功确认的数据量为7KB。

(2) 此问题是在(1)问题的基础开始的，即已确认过7KB，未确认与未发送还有33KB，**发送端采用慢启动重新开始发送**，在拥塞窗口达到4KB时发送数据量=1KB+2KB+ 4KB = 7KB。然后进入**拥塞避免**阶段：在收到全部4个MSS的确认后，拥塞窗口增至5KB，相应地发送端发送了5KB数据；收到全部5个MSS的确认后，拥塞窗口增至6KB，相应地发送端发送了6KB数据；收到全部6个MSS的确认后，拥塞窗口增至7KB，相应地发送端发送了7KB数据；收到全部7个MSS的确认后，拥塞窗口增至8KB，相应地发送端发送了8KB数据；此时刚好发完。因此，文件发送结束时，发送端的拥塞窗口大小为8KB。

5. **TCP如何发送紧急数据？**

答：(1)紧急标志位U(URG)置1；(2)紧急数据置于TCP段数据(载荷)前部；(3)紧急指针指向紧急数据的最后一个字节。

6. **TCP接收方何种情形需要立即进行确认？**

答：(1)连续两个段按序到达，且前一个未确认；(2)收到失序段(序号比期望的序号大)；(3)收到丢失段；

1. 收到重复段。

7.TCP协议中ACK的作用。

答：(1)建立连接、拆除连接(2)差错控制(或可靠传送)

(3)流量控制 (4)拥塞控制

**Chap3 传输层**

**3,1 概述和运输层服务**

**网络层:** 提供主机之间的逻辑通信；尽力而为的服务，网络层尽最大努力在主机间交付分组，但不提供任何承诺：不保证交付，不保证按序交付，不保证数据完整

**传输层**: 提供**进程**之间的逻辑通信； 依赖并增强网络层服务；

1. 传输协议运行在端系统上：

发送方: 将应用报文封装成报文段，交给网络层发送。

接收方: 从收到的报文段中取出载荷，交给应用层

1. 传输层不能提供的服务: 延迟保证；带宽保证；

传输层可以提供的服务：保证可靠、按序的交付：TCP；

不保证可靠、按序的交付：UDP；

**3.2 多路复用与多路分解（运输层必须提供的服务）**

1. 发送端多路复用：从多个套接字收集数据，交给网络层发送；

2. 接收端多路分解：将接收到的报文段交付到正确的套接字；

3. 如何进行多路复用和多路分解？

为将报文段交付给正确的**套接字**：

主机中每个套接字应分配一个唯一的标识

报文段中有特殊字段指示要交付的套接字

发送方传输层需在报文段中包含目的套接字标识（多路复用）

接收方传输层需将报文段中的目的套接字标识与本地套接字标识进行匹配，将报文段交付到正确的套接字（多路分解）

4.端口号是套接字标识的组成部分，是16比特的数，其中0～1023保留给公共域协议使用

5.UDP套接字标识：(目的IP地址，目的端口号) UDP的源是用来发响应报文的。

TCP 套接字标识（源IP，源端口号，目的IP，目的端口号）

**3.3 UDP（流媒体，DNS，SNMP）**

1. UDP提供的服务：多路复用和多路分解（最基础的传输层服务）；检测报文错误（但不尝试恢复）；

2. UDP不提供的服务：可靠/按序交付；延迟及带宽保证；【若要UDP实现可靠传输，在应用层实现可靠性】

3. UDP 报文结构：用于多路复用/多路分解的字段：源端口号，目的端口号；用于检测报文段错误的字段：

报文段长度，检查和；

4.UDP检验和：所有 16 位比特字相加，若溢出，则相加和+1；然后取反；

检验方式：接收方计算检验和，然后与checksum相加，若所有比特位都是1，正确，否则有错。

**5.优点**：（1）没有建立连接的延迟；（2）协议简单，发送端和接收端不需要保存连接状态；（3）头部开销小 【UDP8B TCP20B】，承载效率高，网络带宽利用率高；（4）没有拥塞控制和流量控制，可以尽可能快的发送报文。

**3.4 可靠数据传输(RDT)原理**

数据通过可靠信道传输，没有损坏和丢失，且按序接收。

**Rdt1.0:** 底层信道完全可靠，只需发送和接收即可。

**Rdt2.0:**下层信道可能产生比特错误。错误检测，接收方反馈ACK/NAK，出错重传。

**Rdt2.1:**ACK/NAK 出错：发送方发现受损就重传，但接收方会出现冗余，发送方给分组添加序号【0，1】，接收方发现序号相同就丢弃。（2.1的ACK不带序号）

**Rdt2.2:** 不用 NAK：接收方收到出错的分组、或不是期待接收的分组，重发对前一个正确接收分组的ACK（带序号）；发送方：若ACK的序号不是所期待的（表明当前分组未被确认），重发当前分组。

**Rdt3.0：**可能丢包：启动定时器，在超时前没收到ACK/ 期待的ACK，重发。

Rdt3.0是停等协议，效率太低，一个传完了才有下一个，需要流水线机制。

**传输时延（发送包用的时间）t=(L/R)\*(N流水线)**

**信道占用时间 RTT+ L/R**

**发送方利用率：U = t /(RTT+L/R)**

**GBN&SR见右边对比图**

**GBN**：发送方只对基序号分组使用定时器，若ACK 不对，不管。超时重传整个发送窗口。

**SR**：自动调整失序，避免不必要的重传。

发送方收到的ACK不是最小未确认分组，则标记。若是，滑动窗口。

接收方一旦收到就ACK，失序就缓存。若收到窗口前冗余分组n，说明发送方超时，发送ACK（n）

接收方窗口大小限制在序号空间的一半以内N<m/2。

**3.5 TCP 协议**

1. 特性：点到点，全双工，面向连接，可靠有序，流水线式发送，流量控制，拥塞控制

2. TCP报文段结构：

源、目的端口号：多路复用/分解

序号：首字节在字节流中的序号，非报文段序号

确认号：希望的下一字节序号，隐含累计确认

首部长度：32bits为单位的首部长度

检验和、紧急数据指针（指向最后一个字节）、数据段

标志位：URG：紧急数据 PSH：立即交给上层 RST：不接受连接 SYN：建立连接 FIN 拆除连接

接收窗口：接收端还可以接受的字节数

重要TCP选项：MSS：TCP 段中可以携带的最大数据字节数；window scale: 窗口比例因子，实际接收窗口大小 = windowsize\*2^windowscale; SACK: 选择确认，在累计确认的基础上允许接收端指出缺失的数据字节

5. 往返时间估计

**估计平均RTT 指数加权移动平均 a = 0.125**

**EstimatedRTT = (1-a)\*EstimatedRTT +a\*SampleRTT**

**安全距离 beta = 0.25 DevRTT =**

**(1- beta)\*DevRTT + beta\*|SampleRTT - EsRTT|**

**超时值：**

**TimeoutInterval = EstimatedRTT + 4\*DevRTT**

1. **TCP 发送方处理的事件**：

- 收到应用数据: (1)创建并发送TCP报文段；(2)若当前没有定时器在运行（没有已发送、未确认的报文段），启动定时器；

- 超时:(1)重传包含最小序号的、未确认的报文段；(2)重启定时器;

- 收到ACK:（1）如果确认序号大于基序号，推进发送窗口（更新基序号，由于接收方采用累计确认机制，故发送方窗口可一次推进多步，避免重发某些丢失了 ACK 的报文段）；（2）如果还有未确认的报文段，启动定时器；（3）**快速重传**：当发送方收到对同一序号的3次重复确认时，立即重发包含该序号的报文段（**why 3次？**Because, 发送方收到冗余 ACK 不仅仅可能是因为报文段丢失，还可能是因为网络中的报文段重新排序，使得到达接收方顺序不同，此时接收方缓存失序报文段，说不定下一个到达报文段就是期望值了。）

event: ACK received, with ACK field value of y

if (y > SendBase) { //收到更新的确认号

SendBase = y；

if (there are currently not-yet-acknowledged segments) start timer }

else { //收到重复的ACK

increment count of dup ACKs received for y

if (count of dup ACKs received for y = 3) {

resend segment with sequence number y }}

**定时器补偿**

- 发送方每重传一个报文段，超时值就**增大一倍**。

- 若连续发生超时事件，**超时值呈指数增长**（至一个规定的上限值）。Karn 算法 & EstimateRTT & 定时器补偿=>timeout；

**推迟发送ACK**：

- 优点：可以减少通信量；

- 缺点：当延迟太大时，会导致不必要的重传

推迟确认造成RTT估计不准确；TCP协议规定：

推迟确认的时间最多为500ms；接收方至少每隔一个报文段使用正常方式进行确认；

1. **TCP 流量控制**：调节发送速度，使得接收缓存不会溢出；

接收缓存中的可用空间= RcvWindow //接收窗口

= RcvBuffer-[LastByteRcvd - LastByteRead]

接收方将RcvWindow放在报文段中，向发送方通告接收缓存的可用空间；

发送方限制未确认的字节数不超过接收窗口的大小，即：LastByteSent-LastByteAcked ≦ RcvWindow

接收窗口为0时，发送方必须停止发送。但是数据不能不传呀，发送方还需要知道窗口大小。此时发送方启动坚持定时器，发送零窗口报文段，从接收方的响应中获知窗口大小。

糊涂窗口综合征：双方处理速度严重失衡：接收方不断发送微小窗口通告，引起发送方不断发送很小的数据分组，导致带宽浪费。

1. **连接管理**：发送方和接收方在交换数据前先握手。

- 两次握手的不可行性：

一个不可靠网络中存在许多干扰连接正常建立的因素：包传输延迟变化很大；存在重传的报文段；存在报文重排序；

- **三次握手建立连接**：

Step1. 客户TCP发送SYN 报文段（SYN=1，ACK=0）；给出客户选择的起始序号；不包含数据；

Step 2: 服务器TCP用SYNACK报文段响应（SYN=ACK=1）; 给出服务器选择的起始序号; 确认客户的起始序号; 不包含数据（服务器端分配缓存和变量）

Step 3: 客户用 ACK报文段响应（SYN=0，ACK=1）; 确认服务器的起始序号; 可能包含数据（客户端分配缓存和变量）

- **四次挥手 closing a connection**：

客户端：向服务器发送 FIN，等待服务器确认；

服务器：向客户端发送 ACK，确认请求；

服务器：向客户端发送 FIN，等待客户端确认；

客户端：向服务器发送 ACK，等待一段时间后结束。

1. **拥塞控制原理：**

流量控制：限制发送速度，使不超过**接收端**的处理能力

拥塞控制**（多个源）**：限制发送速度，使不超过**网络**的处理能力;

- 拥塞造成（可用来判断拥塞现象）：1 丢包（路由器缓存溢出）；2 分组延迟增大（路由器来不及转发，等待转发的分组很多）；

- 拥塞结果：网络负载很重，但网络吞吐量很低。

- TCP 采用从端到端的拥塞控制机制，发送方根据自己感知的网络拥塞程度，限制其发送速度。

**发送方感知到拥塞后，改变其发送速率的算法是什么？**

发送方使用**拥塞窗口CongWin>=（**限制）已发送未确认的数据量，其间接限制了发送速率。**Rate = CongWin/RTT bytes/sec**

TCP拥塞窗口的调节（拥塞控制）： 加性增，乘性减（AIMD）,慢启动。发送方维护变量 Threshold

**超过阈值，慢启动状态转为拥塞控制**

**发生丢包时，Threshold=CongWin/2 转为快速恢复**

- 收到3个冗余ACK：说明网络仍然有一定交付能力

Threshold=CongWin/2

Reno**快速恢复** CongWin = Threshold + 3MSS;

Tahoe **快速恢复** CongWin = 1MSS

- 超时： 网络的交付能力很差

Threshold=CongWin/2

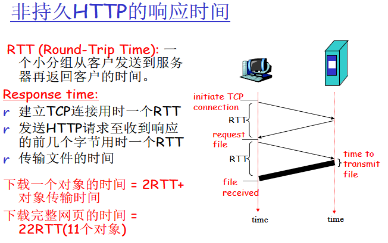
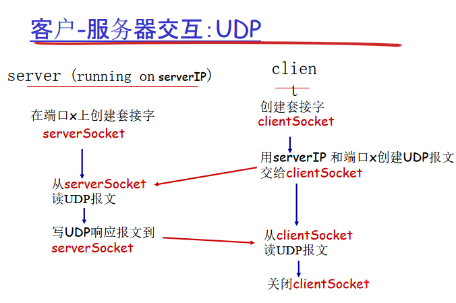
CongWin = 1MSS（发送速率降至最低）

无条件到慢启动，增大CongWin，CongWin < threshold，CongWin 每次加倍; 否则只加 1MSS（拥塞避免：缓慢增大CW）。

1. **TCP 吞吐量**

吞吐量与丢包率L的关系: **[1.22MSS / （RTT 根号 L）]**

TCP的公平性：如果K条TCP连接共享某条带宽为R的瓶颈链路，每条连接具有平均速度R/K。



**套接字编程：**

**服务器使用多个套接字服务客户：**

1 服务器进程在欢迎套接字上等待客户的连接请求；

客户进程需要通信时，创建与服务器欢迎套接字通信的客户套接字：在此过程中，客户TCP向服务器TCP发送连接请求；2 服务器进程创建一个临时套接字（称连接套接字）和一个新的服务器进程，与客户进程通信3 服务器进程回到欢迎套接字上继续等待： 允许服务器同时服务多个客户4 客户服务结束后，服务器销毁进程，关闭连接套接字

**UDP&TCP**

UDP- 报文传输服务- 由于没有建立管道，应用程序发送每个报文必须给出远程进程地址- 服务器使用一个进程和一个套接字为所有客户服务，**一次请求-响应完成一次服务**

TCP- 字节流传输服务- 由于建立了管道，应用程序**只需向套接字中写入字节序列，不需指出远程进程地址**- 服务器为每个客户单独生成一个套接字和一个新进程，允许双方长时间通信；多个传输请求->服务器开多线程建立多个TCP

**Chap1 导言**

**1.1 什么是因特网**

1. 计算机网络向用户提供的重要功能：连通性，共享
2. 因特网的两种描述：
3. 按松散的层次结构组织、并且遵循 TCP/IP 协议的 ISP 集合（2）是为分布式应用提供通信服务的基础设施。
4. 具体构成描述-（1）终端设备：主机或端系统，运行网络应用程序（2）通信链路：光前，铜线，射频等；传输速率常称为带宽。（3）分组交换机：转发分组；路由器和链路层交换机；
5. 因特网提供的通信服务包括：（1）源主机到目的主机的可靠数据交付 (2)尽力而为的数据交付 不可靠
6. 网络协议：定义了通信实体之间交换报文的格式和次序，以及在报文发送、接受或收到其他事件后采取的动作【协议规定了设备之间的通信规则TCP/TP/HTTP/Skype】不同系统的相同层实体之间
7. 路由：分组从发送端到接受中断经过的通信链路以及分组交换机的序列。
8. ISP：由分组交换机和通信链路组成的网络，为终端提供接入因特网的服务。接入 ISP 必须相互连接。

ISP 采用多层结构：局域网连接到区域 ISP，区域ISP间接直连/用IXP相连/连接到更高层的ISP

**1.2 网络边缘**

1. 端系统：与因特网相连的计算机和其他设备

2. 接入网：将终端连接到其边缘路由器的物理链路

物理媒体分类：设备之间通过物理媒体相连，分为引导型（铜线，光纤）/非引导型（无线电）

**1.3 网络核心 （选路，转发）**

Df:由路由器和通信链路组成的网状网络。

任务：将数据从发送终端的边缘路由器，转发到接收终端的边缘路由器。

数据在网络中的传递方式：电路/分组交换

1. **电路交换(固定分配)**：电话网采用电路交换

通信前预留好端-端资源（对比：分组交换不预留资源）

资源独占：保证性能（带宽，延迟）；在通信的静默期，资源被闲置。

多路复用技术实现共享通信链路：频分FDM/时分复用TDM ：TDM 相对FDM的优点：FDM 需要复杂的模拟硬件来将信号转换到合适频带上。

优点：能在请求时间内为端到端保持一个确定量的带宽。

1. **分组交换(按需分配)**：主机将应用报文划分成分组。交换机仅在接受到整个分组后才可以开始转发（存储-转发) [N条链路有N-1台路由器] 端到端传输时延：

P个分组经过N条链路的总耗时：(P+N-1)L/R

分组到达速率大于链路输出速率时，会在缓存中排队，若链路的出书缓存满，分组丢失。

优点：适合突发数据，简单，不需建立电路；

缺点：可能产生严重拥塞：延迟，丢包。需要有保证可靠传输和拥塞控制的协议

1. **分组交换原理**：存储转发，动态路由(包括每个分组自带源地址、目的地址，拓扑发现、路由选择)，出错由端系统处理；

**1.4分组延迟**

R=链路带宽(bps); L=packet length (bits)

a=average packet arrival rate

1. 来源：1节点处理【检查比特错误，确定输出链路】2 排队【在输出缓存等待传输，时间长短取决于链路拥塞程度】3 传输延迟【将分组发送到链路上的时间 L/R】4 传播延迟【数据在物理链路上的时间 d/s】

d=物理链路的长度 s=数据在物理介质上的传播速度

1. **节点延迟：处理+排队+chuanshu +传播**

**d\_nobal=d\_proc+d\_queque+d\_trans+d\_drop**

1. **流量强度：La/R <1 越接近 1 延迟越高**

**4, 吞吐量：**发送端和接收端之间的比特传输速率。

Rs:服务器与路由器之间的链路速率；Rc:路由器与客户

端到端吞吐量: min(Rc,Rs,R/10)

吞吐量与链路速率及链路上的负载有关，限制因素通常是接入网。

**1.5 协议层次和服务模型**

分层：将系统功能组织成一系列水平的层次，每层实现一个功能（服务）每层通过以下方式提供它的服务：

在本层内执行一些动作；依靠下层提供的服务；

**优点：**易于处理复杂系统:显式的层次结构易于确定系统的各个部分及其相互关系；模块化简化了系统的维护和升级；

缺点：一层可能冗余低层功能；

**五层中，应用层在用户态，运输层和网络层在内核态其协议运行在主机的操作系统中；链路层和物理层协议运行在wifi和网卡上。**

OSI 模型：在应用层和传输层间加了表示层和会话层

表示层：使应用层能够解释交换数据的含义，如压缩、解密等。

会话层：提供数据交换的定界和同步功能，包括建立检查点的恢复方案的方法

封装：路由器实现1-3层协议，链路交换机实现1-2

封装形式：分组=首部+有效载荷字段（来自上一层（序号大）的分组）

网络攻击：恶意软件、僵尸网络、自我复制、病毒、蠕虫。拒绝服务攻击（DoS）：弱点攻击、带宽洪泛、连接洪泛。分布式DoS（DDoS）。分组嗅探器。IP哄骗。

协议层次：协议栈：物理层、链路层、网络层、运输层、应用层。

应用层：应用层协议（HTTP、SMTP、FTP、域名系统DNS）报文。

运输层：运输协议（TCP、UDP）。报文段/ 进程-进程。

网络层：IP协议、路由选择协议。数据报/ 主机-主机。

链路层：以太网、WiFi、电缆接入网的DOCSIS协议/ 帧。

物理层：双绞铜线、同轴电缆、光纤的协议。

**电子信箱**：(1)由计算机上的一个存储区域（如磁盘上的一个文件）组成（2）每个信箱均被分配了唯一的电子邮件地址4. **简单邮件传输协议—SMTP**

-邮件服务器之间传输邮件采用客户-服务器模式；

-使用 TCP 作为传输层协议，持久连接，服务器端口25；

-发送服务器和接收服务器之间直接传输邮件-SMTP采用命令/响应交互方式：命令: ASCII文本；

响应: 状态码和短语-报文只能包含7位ASCII码）HTTP无此限制-是个推协议，只能将邮件从用户代理推送到邮件服务器，不能用SMTP从邮件服务器中获取邮件；HTTP是拉协议；SMTP把所有报文对象直接放在一个报文中，HTTP把对象封装到相应报文中；

-SMTP 服务器使用“.”表示报文结束（FTP使用关闭连接表示传输结束，HTTP使用长度域表示报文结束）

-非ASCII内容编解码方式：Base64，quoted-printable

5.**邮件访问协议：允许用户从信箱中提取邮件**

**POP3** 无状态**,IMAP** 有状态，所有邮件保存在服务器上**,HTTP**

**2.5 DNS 主机名-IP地址转换**

1.基本概念：由大量按层次组织的DNS服务器实现的分布式数据库 ；允许主机查询分布式数据库的**应用层协议**

**DNS是实现在应用层的因特网核心功能。**

2. DNS 提供的服务：

-主机名-IP地址转换-主机别名 允许拥有复杂主机名的主机具有一个或多个别名 提供与主机别名对应的规范主机名及IP地址-邮件服务器别名 提供邮件服务器的规范主机名及IP地址 允许用域名作为邮件服务器别名-负载分配 允许一个规范主机名对应一组IP地址 将http请求在一群相同功能的web服务器之间分配

Q: 为什么不使用集中式的DNS？（1）单点失效（2）流量集中：单个DNS服务器需处理全部查询（2）响应时间长：远距离的集中式数据库

（3）需要维护庞大的数据库

**域名：域名的任一后缀也是一个域**

**顶级域：组织域，国家域，反向域(IP映射为名字**

**DNS服务器类型**：

-根服务器13个；知道所有顶级域服务器的IP地址

-顶级域TLD服务器；每个TLD服务器负责一个顶级域知道其所有二级子域的域名服务器地址

-权威DNS服务器：提供机构内服务器（如Web, mail）的主机名-IP地址映射；提供一个主域名服务器、一个或多个辅助域名服务器；可由机构维护，也可ISP维护

3.DNS工作机理：

（1）应用程序（如浏览器）调用一个本地例程（称解析器），主机名作为参数之一传递（2）解析器向网络中的DNS服务器发送DNS查询报文（包含要查询的主机名）（3）解析器收到包含IP地址的DNS响应报文（4）解析器将IP地址返回给调用者（如浏览器）

Q：客户想知道www.amazon.com的IP地址:（1）DNS客户查询根服务器，得到com域的DNS服务器地址（2）DNS客户查询com域的DNS服务器，得到amazon.com域的DNS服务器地址（3）DNS客户查询amazon.com域的DNS服务器，得到 www.amazon.com的IP地址**域名解析的例子：**迭代查询，递归查询；【从请求主机到本地DNS 服务器的查询是递归的，其余查询是迭代的。】

***实际的物理服务器的层次与域名空间的逻辑层次不同***

4.DNS缓存

每当收到一个响应报文，DNS服务器将报文中的映射信息缓存在本地。缓存中的映射在一定时间后被丢弃。

*本地DNS服务器通常会缓存TLD服务器的IP地址，因而很少去访问根服务器*5. DNS资源记录

RR format: (name, type, ttl, value)

Type=A Name：主机名Value：IP地址Type=NS Name：域 (e.g. foo.com)

value：该域的权威DNS服务器的主机名Type=CNAME Name：别名 Value：规范名

Type=MX Name：域(e.g. foo.com) Value：该域的邮件服务器名字6.DNS协议，报文

DNS protocol: 定义了**查询和响应**两种报文，查询和响应使用相同的报文格式**DNS报文的封装**

DNS可以使用UDP，也可以使用TCP，服务器端口53响应报文的长度小于512字节时，使用UDP，超过512字节时，使用TCP。当解析器事先不知道响应报文的长度时，先使用UDP；若响应报文的长度超过512字节，服务器截断这个报文，置DNS报文首部的TC标志为1；解析程序打开TCP连接，并重复这个请求，以便得到完整的响应。

**往DNS中插入资源记录**example: new startup “Network Utopia”向DNS注册机构注册域名“networkuptopia.com”提供权威DNS服务器（主域名服务器，辅助域名服务器）的名字和IP地址对每个权威域名服务器，注册机构往 com TLD 服务器中插入两条资源记录，

建立权威DNS服务器，特别是：建立www.networkuptopia.com的Type A记录 建立networkutopia.com的Type MX记录

4. 报文格式：请求行，首部行，回车（表示结束）

上传方法：post 放在报文体内；get放在URL内

5. cookie：保存状态

存储位置：服务器端：返回ID给客户/客户端:文件中

6. web 缓存器（代理服务器，proxy）

即是客户端又是服务器，保存最近请求过的对象的拷贝；减少客户请求的响应时间，减少机构介入链路上的流量。一般由ISP架设。

**2.3FTP 文件传输协议 TCP 端口21、20**

用户通过FTP用户代理上传和下载远程文件；采用客户-服务器模式

FTP 采用两个并行的TCP 连接传输文件：控制连接（端口21）+数据连接（端口20）是有状态服务。

21 一直保持，20随文件传输结束而关闭

分开控制、数据连接的原因：不会混淆数据与命令/相应，简化协议涉及和实现，在传输文件的过程中可以继续执行其他的操作，便于控制传输过程。

用关闭数据连接的方式结束文件传输：允许动态创建文件

**2.4 电子邮件系统**

1.三部分:用户代理、邮件服务器、简单邮件传输协议

2.用户代理：（1）编辑邮件等e.g., Outlook, elm, Mozilla Thunderbird（2）将要外发的邮件发送到用户的邮件服务器；（3）从用户邮箱中取邮件

3.邮件服务器：（1）用户信箱：存放到来的邮件（2）发送报文队列：存放要发送出去的邮件（3）报文传输代理MTA：运行在服务器后台的系统守护进程，负责在邮件服务器之间传输邮件，及将收到的邮件放入用户信箱。

**Chap2 应用层**

2.1 基本概念

1.网络应用架构：客户-服务器架构；对等架构P2P

服务器：永远在线裕兴服务器程序，具有永久的IP地址；客户机：需要时与服务器通信，动态IP地址，不与其他客户机直接通信。**P2P:** 任意端系统可以直接通信，没有总是运行的服务器，使用动态IP。【优点-高度可伸缩，缺点-难于管理】

2.套接字(门）:进程通过套接字发送和接收报文。是应用层和传输层的接口&应用程序和网络之间的API

3.进程标识包括：IP地址，与该进程关联的端口号端口号的例子:HTTP server: 80；Mail server: 254.因特网提供的传输服务：运输层协议如下

**TCP 面向连接**，发送进程和接收进程的可靠传输，流量控制，拥塞控制；**不提供**：及时性，最低带宽保证

**UDP 无连接**，不可靠传输；**不提供**：连接建立，可靠传输, 流量控制, 拥塞控制, 及时性，最低带宽保证

2.2 WEB HTTP

1.应用层协议定义了：交换的报文类型, request, response；报文语法语义，进程发送/响应规则Web：HTTP（超文本传输协议）；Skype：专用协议；电子邮件：SMTP（简单邮件传输协议）

2. HTTP 协议：采用**客户-服务器**模式，定义浏览器和web服务器之间的通信规则；使用**TCP**作为传输层协议。客户发起到服务器80端口的TCP连接，客户端创建一个套接字；服务器接收TCP连接并创建套接字；浏览器和服务器通过各自的套接字来交换HTTP报文。其中，*服务器不保存有关客户请求的任何信息*。【对象-就是文件，html文件，jpeg图像…】

每个对象通过一个URL访问

3.连接方式：

**非持久HTTP**:HTTP1.0在一个TCP连接上最多发送一个对象，接受完响应报文之后关闭TCP连接

获取每个对象需要2个RTT；每个TCP连接需要消耗操作系统资源；浏览器通常打开多个TCP连接获取引用对象，消耗资源

**持久HTTP:**HTTP1.1一个TCP连接上发送多个对象

**RTT-一个小分组从客户发送到服务器再返回的时间**

服务器在发送响应后保持连接；同一对客户-服务器之间的后续HTTP报文可以在该连接上传输无流水线方式:客户仅当收到前一个响应后再发送新的请求-请求每个对象用时1个RTT--请求一个网页用时12RTT【网页包含文本和10个图像】

流水线方式:HTTP/1.1；客户每解析到一个引用对象就可以发送请求；可在一个RTT时间内请求所有引用对象；请求一个网页用时约3RTT（网页2RTT(建立连接+发送请求）+图片1RTT）