**计算机网络复习**

**第一章：概述章**

没啥说的，当故事看看吧，弄懂啥是分组交换，电路交换即可。

**1、电路交换**： 在通信进行过程中，网络为数据传输在传输路径上预留资源，这些资源只能被这次通信双方所使用；

**2、分组交换**：数据被分成一个一个的分组，每个分组均携带目的地址，网络并不为packet传输在沿途packet switches上预留资源，packet switches为每个packet独立确定转发方向.

**与电路交换不同**，链路、交换机/路由器等资源被多个用户所共享，交换机在转发一个分组时的速度为其输出链路的**full速度**。

注：分组交换一般采用**存储转发**技术，分组在分组交换机中会经历一个**排队(queuing)延迟**。**排队延迟**与交换机的忙闲有关，大小可变。 如果分组到达时缓存已满，则交换机会丢掉一个分组。**分组交换网络有两大类**1、Datagram（数据报)网络2、Virtual Circuit虚电路网络

**第二章：应用层**

重点：HTTP SMTP DNS

http 超文本传输协议

熟悉http报文

1）B/S的通讯过程、无状态 2）流水线协议和非流水线协议

3）持续和非持续方式 4）代理服务器、cookie

1) Browser首先建立与Server的TCP连接

2) 连接建立起来后，browser和server就向/从Socket发送/接收HTTP的消息。借助TCP的reliable data transfer，HTTP知道消息肯定会到达对方，这就是协议分层的好处。

HTTP是一种**stateless(**无状态)**协议**，**server不保存任何client的任何状态信息**。**如果server在很短的时间内从browser接收到对某个object的两次请求**，server就会发送两次**response**。

2）非流水线方式：客户在收到前一个响应后才能发出下一个请求。这比非持续连接的两倍 RTT 的开销节省了建立 TCP 连接所需的一个 RTT 时间。但服务器在发送完一个对象后，其 TCP 连接就处于空闲状态，浪费了服务器资源。

流水线方式：客户在收到 HTTP 的响应报文之前就能够接着发送新的请求报文。一个接一个的请求报文到达服务器后，服务器就可连续发回响应报文。使用流水线方式时，客户访问所有的对象只需花费一个 RTT时间，使 TCP 连接中的空闲时间减少，提高了下载文档效率。

3）1、非持续连接：建立一次TCP连接，browser和server通过此连接只传输一个request消息和一个respond消息

2、持续连接：建立一次TCP连接，browser和server通过此连接可以传输多个request消息和多个respond消息

4）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。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 的地址再去因特网访问该资源。

Cookie定义如下：Cookie是Web服务器保存在用户硬盘上的一段文本，Cookie允许一个Web站点在用户的电脑上保存信息并且随后再取回它。信息的片断以‘名/值’对(name-value pairs)的形式储存。

**注：Web Cache比Server更靠近Client，即使只从延迟上将也会减小服务响应时间；**

利用Cache可以减小响应延迟，但Web Cache引入了一个新问题:即Web Cache中保存的对象可能与原始服务器中保存的对象不同。

SMTP 25号端口 依靠TCP提供可靠数据传输

与HTTP相比，HTTP是一个“拉协议”，SMTP是一个“推协议”，**具体看看书和那个实验吧。**

DNS 主机名到IP地址转换的目录服务

重要的是两种查询方式：递归查询，迭代查询。

心理要清楚整个DNS的查询过程！

两种查询方式：

1、主机向本地域名服务器的查询一般都是采用**递归查询**。如果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。

2、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。

**以及啥上根DNS服务器，啥是顶级DNS服务器，啥是权威DNS服务器等。以及实验中出现的问题，例如NSLOOKUP的使用。**

**第三章 运输层**

**这章就有点儿意思了，重点：TCP TCP TCP！**

**可靠数据传输：**

可靠性传输原理是由rdt1.0 rdt2.0 rdt2.1 rdt2.2 rdt3.0一步步累加而来的。

rdt1.0：接收方无返回确认信息 rdt2.0：接收方进行检错，并发送ACK 或NAK反馈给发送方

rdt2.1：加入序列号0和1 rdt2.2：接收方不再发NAK而将ACK中加入序列号

rdt3.0：发送方引入定时器

以上都是停等式（stop-and-wait）协议**为了解决stop-and-wait协议低效问题的方法非常简单，就是允许发送方可以在等待Receiver的ACK之前连续发送多个分组。这种技术叫做流水线。**

流水线技术对可靠数据传输协议的影响：

1、更大的序列号范围。连续发送的并且是还没有得到ACK的多个分组必须要有唯一的序列号，否则引起混乱。

2、Sender和Receiver方需要存储空间来缓存分组。对于Sender来说，需要缓存已经发送出去但还没有得到ACK的分组；为了实现按序递交，接收方一般也需要存储空间。

序列号的范围和Buffer的大小取决于传输层协议如何相应分组丢失、差错以及过度延迟分组的方式。

**解决流水线的差错恢复有两种基本方法：回退N步(Go-Back-N)和选择性重传(Selective Repeat)**

GBN（Go-Back-N）允许发送方发送N个分组而无需确认，流水线中最多有N个等待确认消息的分组，允许使用的序列号范围可以看作是长度为N的一个窗口。随着协议的运行，这个窗口在序列号空间内向前滑动，因此这种协议也叫滑动窗口协议（sliding-window protocol) 在此系统中，一个分组或其ACK的丢失可能造成GBN重传太多的分组。当信道差错率逐渐变大，信道会被不必要的重传分组所塞满。

SR（Selective Repeat）选择性重传就是Sender只重传那些出现错误的分组，而不是窗口中的所有分组。

**以下这些都看看吧，心理有个印象**

**TCP的流量控制原理**

**TCP的服务特点、流的概念**

**TCP的拥塞控制原理**

**TCP连接建立和拆除的过程**

**UDP的服务特点**

**UDP是一种无连接的、轻量级传输层协议，提供了最最健的服务模型。没有连接，直观上就应该比TCP更高效。**

1、***不可靠的数据传输：***发送端将数据Push入UDP Socket后，UDP并不保证数据最终会到达接收端，即使到达也不保证是按序到达；

2、**没有congestion control机制**：发送方可以以任意的速率向网络中发送数据，不管网络的拥塞状况。但发送的数据可能最终到达不了接收方，产生丢包。

**第四章 网络层**

**为什么这么大字？因为这章太重要了!!!**

**重点：虚电路和数据报网络（p207-p211）、子网划分(p220-242)、路由器选择算法（LS算法和DV算法p243-p257）、RIP、OSPF、BGP(AS的划分和协议特点p257-p268)**

**由于这章重点太多，书上讲的更为详细和清楚，建议大家仔细阅读书！手敲实在太累了……还有仔细看看老师的PPT，PPT都发班群里了！**

**所有重点必须理解！而且要记住！记住！记住！！！！！**

**重点复习这章！！！**

**重点复习这章！！！**

**重点复习这章！！！**

**第五章 链路层**

**这章一样很重要！！！！**

**重点：差错检测（奇偶校验、检验和、循环x余p292-p296）、信道划分协议（不知道考不考，有时间的看一下，没时间就放弃吧）、ARP**

**这章也很重要！仔细看书和幻灯片！！！重点复习这章！！！**

**最后：压题**

**根据老师自己说的三种题型**

**简答题5到6道：**

1，HTTP报文简述  
2，描述一个DNS查询的过程

3，为什么实际带宽会小于链路带宽也就是拥塞的解释  
4，描述TCP连接创建和断开

5，数据报网络和虚电路网络的不同  
6，数据报网络中发送方和接受方各自的特点

7，AS的划分和各自的协议特点（08简单第三题）

9，ARP原理

10.节点间的通信过程（08分析第二题）

**计算题2到3道：**

1，ls选路算法（课本，ppt）  
2，ip网段的划分（课本，ppt）  
3，循环x余检验和（08计算第一题）

**综合题1道：**

**1 WEB页面的请求过程（书p329，幻灯片也有）**

其它的大家自己补充吧。  
还有老师书上留过的题，还有幻灯片，还有资源共享协会给的08年考题。

以上为我通过心理学，社会工程学押的..押对了不图啥，押不对别骂娘=。=

**祝大家取得优异成绩！祝大家取得优异成绩！祝大家取得优异成绩！**