计算机网络总结笔记(上)

这篇日志将按照名教材《计算机网络——自顶向下方法和 Internet 特色》第四版总结。因为原书是英文版，这个总结可能写的很慢。还是先开预留位。同时我会试着用尽量简单调理的少量语言总结完。最后将给出总结中的中英文对照表。

因为内容难以预料到地多，本文将分成两部分，并用额外日志增加题解和部分实验数据。

总结笔记（上）将包括概述，应用层和运输层内容。

A.概述

1.1 概述

首先我们把一组独立互联的计算机称作计算机网络。而 Internet 与之不同。下面通过两种描述手段阐述 Internet 的概念。

从具体构成的角度来看，Internet 互联了众多称作**主机**或者**终端系统**的设备，他们被认为位于网络边缘，主机又可分为客户机和服务器。这些设备通过通信链路（也就是物理线缆）连接，链路中间还有分组交换机。主机之间交换数据时将数据**分组**发送，并通过分组交换机分配包数据。常见的分组交换机有路由器和链路层交换机。主机通过 ISP（Internet 服务提供商）接入 Internet 。所有 Internet 通讯必须遵循一定的协议。其规范称作互联网标准。

服务角度来说，Internet 是提供这么一些服务的计算机网络：允许离散的应用在终端上交换数据，比如下载软件和浏览器等等。他还为这些应用提供两种服务：面向连接的服务以及无连接的无依赖性服务。简单地说，面向连接的服务有握手程序，还必须确认包是否正确发送和接受，但是无连接服务不需要担心另一方的问题，只要发或者接就行了。

1.2 网络核心

我们知道通过链路和交换机可以让主机之间交换信息。其两种基本方法是电路交换和分组交换。前者的应用主要是电话系统。下面简单分析两种交换信息的方法：

电路交换：每个主机都直接与一个交换机直接相连，各个交换机之间有物理线缆，如果两台主机要传送信息，其对应的交换机之间必须有一条预留电路。假定每个交换机都有 n 条电路，那么连接期间该连接获得链路带宽的 1/n。（想象一下电话系统就明白了）

分组交换：各种应用在完成任务时要交换报文，，报文包含协议要求的内容。主机会把较大的报文分组并发送到分组交换机。交换机使用存储转发传输机制，简单地说就是接受一个报文的全部分组后才输出，这样就会产生存储转发时延，同时，对于每个输出链路，分组交换机还为之生成一个输出缓存或输出队列，因为同一时刻只能向一条链路输出一组信息，其他信息只能在队列中等待，这样会产生排队时延。如果队列已满，新到达的报文分组无法入队，就会产生丢包。此外还有节点处理时延，传播时延等，总称节点总时延。

1.3 协议层次：网络中引入协议栈概念，自顶向下分别是：应用、运输、网络、链路、物理层。对应的主要协议是 HTTP&SMTP, UDP&TCP, IP, PPP, ...等。

----------B.应用层----------

首先我们要从一个 web 应用开始研究网络协议。一个进程进行通讯时，必须将报文发送到套接字（socket），这一过程发生在应用层中，套接字把报文送到操作系统控制的运输层，并通过

Internet 传送到接收方的运输层，通过对方套接字到达目的进程。所以可以把套接字当作一个应用和网络之间的 API。运输层协议的服务应该包括可靠数据传输，吞吐量，定时和安全性。不同的应用对于这些服务的要求不同，可以选择不同的运输层协议。常见的运输层协议就是 TCP 和 UDP 协议。

TCP 是面向连接的可靠数据传输服务，还有拥塞控制机制。简单地说，当双方之间网络阻塞时，TCP 可以抑制发送进程。这是一个有利有弊的技术，之后还会讨论。

UDP 不面向连接，也没有拥塞控制，所以速度比 TCP 快，当然可靠性无法保证。

在选择了协议以后，应用还需要考虑目的主机的地址（用 IP 地址说明）和主机上的什么进程，后一个问题是用端口解决的。即不同类型的 web 程序，其端口不同。

----------HTTP----------

应用层协议规定应用程序之间如何相互传递报文。最著名的就是 HTTP 协议（超文本传输协议）

HTTP 用 TCP 作为其支撑运输层协议，由于其服务器并不存储客户端信息，HTTP 也是一个无状态协议。此外 HTTP 默认使用持久连接。我们知道如果一个请求或相应建立在 TCP 上，就有这样的问题：是每个请求都用一个新的 TCP 还是用一个 TCP 管理所有请求，这两种手段分别称作非持久连接和持久连接。因为非持久连接对于服务器端来说压力巨大，一般默认为持久连接，当一个链接在一段时间内没有被使用，服务器就会终止这个 TCP 连接。

⊞⊟代码

----------cookies----------

用来帮助服务器识别用户。cookies 技术包括四部分：HTTP 的请求报文和响应报文中有一个 cookies 行，在客户端系统中保存有 cookies 文件，在 web 站点后台有数据库。其工作原理是，用户第一次访问某服务器时，其响应报文中包含一个 Set-cookie 字段，该字段的值往往是服务器设定的唯一标识码，浏览器将该字段和服务器主机名等信息保存在本地，以后用户每次访问时，浏览器都会从 cookies 中找到该站点的标识码放在请求报文中发送，这样服务器就能通过这个唯一标识辨认这个用户，通过用户在网站上的浏览记录采取行为

----------web 缓存服务器---------

也叫做代理服务器，当浏览器申请一个文件时，先建立一个到 web 缓存的 TCP 连接，如果缓存服务器上有这个文件就返回，否则，他就建立一个到原始服务器的 TCP 获得该文件并存储，之后将该文件发回客户端。web 缓存可以大幅减少对客户端的响应时间，减少机构内部网络和 internet 之间的链路通信量并节约带宽。这样会带来的新的问题是，web 缓存上的文件可能是陈旧的。为此 HTTP 引入了条件 get，即 web 缓存在保存一个文件时会记录其最后修改时间，客户端每次请求文件时，web 缓存会向原始服务器发送一个带有 if-modified-since 的报文，如果该文件没有改动就直接把缓存中的文件返回给用户，否则重新下载该文件。

----------FTP----------

用两个并行 TCP 连接，一个控制连接，一个是数据连接。

----------邮件系统----------

邮件系统由三个部分组成：用户代理，邮件服务器和邮件传输协议。SMTP 是邮件的主要应用层协议。用户 A 的用户代理发送邮件给 A 的邮箱所在的邮件服务器，A 的服务器通过 SMTP 协议

和 B 的服务器建立 TCP 连接传输邮件，B 的服务器再把邮件发往 B 的用户代理。如果 B 的服务器没有开机，邮件将一直在 A 的服务器的报文发送队列中等待。下面比较 SMTP 和 HTTP 协议：

首先，HTTP 是一个拉协议，也就是说，HTTP 主要是获取信息，而 SMTP 是一个推协议，主要用于发送信息。

其次，SMTP 要求每个报文包括主体在内都使用 7 位 ASCII 码，HTTP 没有这个限制。

另外，在对于多媒体的处理上，HTTP 把每个对象封装在自己的响应报文中，而 SMTP 则把所有对象放在一个报文中。

因为 SMTP 要求所有对象用 ASCII 编码，对于图片、视频或者非 ASCII 字符，就要使用多用途 Internet 邮件扩展（MIME）对其添加支持。需要在报文的首部中添加 Content-Type 和 Content-Transfer-Encoding。

SMTP 的接收方在受到报文后，会在该报文中添加一个 Received 行表示接受信息。

下面考虑邮件如何从接收方的服务器到达接收方主机上的用户代理，也就是邮件访问协议，流行的有 POP3 和 IMAP 协议以及 HTTP。当用户代理打开一个到邮件服务器的 110 端口的 TCP 后，POP3 就开始工作了。第一阶段：特许阶段，用户代理以明文发送用户名和密码，第二阶段：事务处理，这个阶段中用户代理将取出服务器上的报文，还可以将一些报文标记为删除。第三阶段：更新，服务器将删除那些标记了删除的报文。

----------DNS----------

简单地说，DNS 用于将用户提供的主机名（域名）解析为 IP。当主机需要解析一个 URL 时，发出一个 DNS 请求，主机上的 DNS 接收到后发送给 DNS 服务器，服务器返回解析 IP。DNS 提供的其他服务包括主机别名，邮件服务器别名和负载分配。DNS 基于 UDP 协议。DNS 服务器也有分级设定，并通过递归和迭代查询获取 IP。DNS 缓存技术则是，当 DNS 服务器得到一个 DNS 回答时将在一段时间内保存该解析至缓存，这样可以有效减少递归查询的数量。

----------P2P 略----------

----------C.运输层----------

运输层的工作就是把应用层需要发送的信息交给网络层发送。是一个过渡性的协议层。开发者在做 web 应用的时候，必须指定运输层协议是 TCP 还是 UDP。换个角度来看，运输层协议把主机之间的交互扩展到了主机上进程之间的交互（因为他们负责把进程的信息交给主机），这叫做运输层的多路复用和多路分解。接着 B 部分的第一句话讨论就是，将报文从运输层送到网络层，称作多路复用，把报文从运输段交付到正确的套接字称作多路分解。这也是运输层的两个工作。而为了达到这个目标，我们需要唯一标识套接字。实际上，每个报文中都有特殊字段指明了报文所要交付的套接字，也就是源端口和目标端口。端口是 0～65535 之间的数字，其中 0～1023 称作周知端口号，也就是被占用的。应用可以使用其后的端口号。UDP 的套接字就是发出目的 IP 及端口号。TCP 套接字因为需要确认信息，套接字中额外有源端口和源 IP。

----------UDP----------

UDP 的优点包括：应用层可以更好控制要发送的数据和发送时间，无需建立连接，没有连接状态，分组首部开销小。

UDP 的报文段的首部有 4 个 16bit 字段，分别是源端口，目的端口，长度和校验值，之后的内容则是被应用层数据占用的数据段。所谓的校验值是 UDP 的一种验错机制，将他前面的三个

16bit 字相加取反就是校验值，这个工作是发送方 UDP 进行的，接收方将四个字段都相加应该得到 1111111111111111，否则说明出错。

----------可靠数据协议----------

首先还是要讨论所谓可靠数据传输协议，这一部分挺麻烦。我们只讨论单向数据传输的情形，那么一般情况就是，发送方通过该协议把数据交给更底层（比如运输层交给网络层），底层负责传输，接收方再通过该协议把数据取出。我们把这个协议称作 rdt(reliable data transfer)，从简入繁讨论下这个问题：

1.假如该协议的下层底层是完全可靠的

那么 rdt 的发送方只从高层接收数据，产生一个包含该数据的分组，将数据发入信道，接收方只从下层接受一个分组，从分组中获取数据并传给高层。

2.假如下层信道有比特差错

如果是两个人类传递消息，听话的人在听的过程中，听到正确消息会说 ok，听到模糊的消息会要求对方重说一遍。这之中包含了三个要素：

差错检测：首先必须有一种机制让接收方判断出消息出现了比特差错。*接收方反馈*：协议还必须允许接收方反馈消息，反馈包括肯定确认 ACK 和否定确认 NCK。*重传*：收到否定确认的话发送方将重新发送。

基于重传机制的协议称作自动重传请求（ARQ）协议。如果发送方每发完一块数据就等待接收方反馈，否则不发送下一块数据，这种协议称作**停等协议**。

该协议存在的问题是，如果 ACK 和 NCK 出错，比如接收方发送了 ACK，但是发送到发送方时被解读成 NCK，发送方重发分组，却被接收方认为是新分组，这样就会导致一个分组被接受两次。为了解决这个问题，现在通用手段是使用分组序号。这样接收方通过比较序号就能知道这是新包还是重发包了。因为发送方每发一个包就停等，序号只需要 01 来区分新旧包就够了。

3.下层信道具有比特差错并丢包

这是最普遍的情形，不但是分组可以丢，ACK 和 NCK 也可以丢。对于发送方来说，不管是什么情况只要重发就行了。这时引入倒计数定时器，如果定时器出发，就说明包没有按时反馈，发送方就会重发。讨论到这里我们基本得到了一个可靠的协议，也被称作比特交替协议（因为序号只有 01 两个）。该协议的问题是停等的效率问题。利用停等协议发送方的吞吐率只有 0.027%，如果能不停等而是让发送方不断发送数据，基于这种思想的协议称作流水线协议。要想实现流水线协议需要：必须增加序号范围，双方必须加大缓存。常见的方法有退回 N 步(GBN)和选择重传(SR)。

GBN 中，如果将最早的未确认分组序号定义为基序号，将下一个将发送的分组序号定义为下一个序号，整个发送组可以分为四个部分，就是已经发送并确认的，发送但未确认的，未发送的和不可用的。之所以将一部分作为不可用是为了实现对发送方的流量控制。发送未确认的和未发送但可用的这两部分的数量称窗口长度 N，GBN 也被称作滑动窗口协议。其发送方必须相应三种事件：上层调用（发送方首先检查窗口是否满，之后发送一个分组并更新），接受 ACK（如果接受某分组的 ACK，必须保证该分组之前的所有分组都正确收到 ACK，也就是说只处理未确认分组最底端的那一个分组，之后将他移出队列），超时事件（也就是没有收到 ACK，将未确认分组中所有分组重发）。

SR 可以避免 GBN 中重发产生的浪费。需要做的改动就是需要为每一个分组都设定一个定时器了。对于发送方和接收方来说，基序号一旦成功发送或者接受，窗口都会相应移动。

----------TCP----------

1. TCP 是面向连接的，提供全双工服务，也就是说 TCP 连接是双向的，同时也是点对点的。建立 TCP 需要三次握手，发送方通过套接字向 TCP 的发送缓存中传输数据，当数据达到最大报文段长（MSS）时 TCP 就将缓存加上一个 TCP 首部形成报文段发送给接收方的 TCP 接收缓存。

2. TCP 首部一般是 20 字节，包括源端口(16bit)，目的端口(16)，序号(32)，确认号(32),首段长(4),标志字段(6,ACK,URG,PSH,RST,SYN,FIN)，接收窗口(16)，校验和(16)，紧急数据指针(16)以及长度可变的选项。

其中，确认号是接收方希望从发送方收到的下一报文的序号，它代表接收方收到了确认号之前的所有分组，因为 TCP 是双向的，如果 1－100 和 150－200 已经收到，接收方就可以在确认号中写 101 了。ACK 是接受确认标志，URG 是紧急信息标志，对应紧急数据指针，PSH 希望接收方收到后立即向上层反馈，这些根本没有实用价值。剩下三个标志字段用于三次握手。接收窗口表示接收方愿意接受的字节数量，用于对发送方进行流量控制，首段长是因为有可变长的选项字段，需要声明首段长。选项段一般是空的。

3. TCP 还有很多细节问题，包括超时判断和可靠传输实现等等。

TCP 使用在可靠传输一节中得到的倒数计时器来判断一个包是否成功发送。但是超时设定为多少是需要解决的，起码也要大于往返时延 RTT，因此 TCP 有估计往返时延技术：在任一时刻对一个分组取样得到 sampleRTT（也就是从把分组交给底层到收到 ACK 的时间），并用得到的 sampleRTT 维护一个均值 EstimatedRTT,公式是 ERTT=0.875*ERTT+0.125*SRTT.又把 devRTT 称作一个 SRTT 偏离均值的程度，计算公式是 DevRTT=(1-b)*DevRTT+b*|SRTT-ERTT|,b 的推荐值是 0.25。偏离值跟随每次得到的取样 RTT 更新，显示目前 RTT 的变化程度。

超时时间＝ERTT+4*DevRTT

可靠传输也有些细节问题，比如为每个分组都设定定时器开销很大，解决方法就是给当前的基序号分配定时器，基序号发送成功后将定时器重设给新的基序号。此外大部分 TCP 实现采用了加倍超时，即一个报文在超时后，之后每次重发该报文，超时时长都增加一倍。同时，TCP 还有一种快速重传冗余 ACK 机制。对于接收方来说，如果收到的分组是按顺序的 1&2，那么他的下一个期望序号就是 3，但是如果这时候收到了 5，就能意识到数据流产生了间隔（没有 3 和 4），那么他就会重复发送冗余 ACK，表明期望序号 3 。对于接收方来说，如果收到 3 个冗余 ACK，就快速重传。

4. TCP 为应用提供了流量控制服务，因为 TCP 的一个问题在于，发送方不必等待每个分组的 ACK 信息就可以发送新分组，那么接收方就可能接到过多分组而导致缓存溢出，为了解决这个问题，在前面的分析中可以看到，规定一个窗口长度 N，一旦有一个分组没有被确认，就可以有效卡死发送方可以发送的分组数量。此外，发送方可能被 IP 网络拥塞遏制，这种控制称作拥塞控制。

5. TCP 连接的建立：客户机端 TCP 发送一个特殊报文段，不包含应用层数据，但首部的 SYN 置 1，并选择一个起始序号 client_isn 放置到该报文序号字段中。服务器接收后为该 TCP 建立分配 TCP 缓存和变量，并发送允许报文段，SYN 置 1，确认号为 client_isn+1，序号段中则是自己的初始序号 server_isn，客户端收到这个报文后也会分配缓存和变量。并发送第三个报

文，SYN 置 0，确认字段为 server_isn+1，目的是确认信息。这个过程被称作三次握手。连接终止时客户端发出 FIN 置 1 的报文，服务器也发挥 FIN 置 1 报文，客户端确认后双方释放资源。

6.拥塞控制

如果没有拥塞控制，我们可以看到的连接中的问题包括，当分组传输数率接近链路容量时，分组将会经历巨大的时延，因为大量分组涌入路由，但是能从路由出去的分组却被链路容量限制了。而队列缓存溢出时路由将丢包，因此发送方只能重发分组造成恶性循环，或者因为排队时间太长发送方重发了包，但是路由中该包并没有被抛弃，这样接收方会收到两个一样的包，重复收包也增加了网络压力，最后，如果路径上有不止一个路由器，那么一个包被丢弃时，将会造成所有分发他的路由的容量浪费。

拥塞控制可以分为端到端的和网络辅助两种，对应网络层是否为运输层拥塞机制提供显式帮助。TCP 是端到端的拥塞机制，因为其网络下层 IP 协议并不提供拥塞机制。按照前面说的，首先 TCP 可以通过控制拥塞窗口的值控制发送方流量，通过超时和连收 3 个冗余 ACK 判断网络拥塞，如果出现就缩小窗口，否则就趁机放大窗口利用空闲带宽。

具体的窗口调节算法称作 TCP 拥塞控制算法。

总体上来说，如果没有拥塞，TCP 每收到一个正常的确认报文，就将窗口扩大一个 MSS（最大报文段），也就是多发一个报文，如果出现拥塞就将窗口大小缩小一半。该算法称作加性增，乘性减。

首先，因为在开始阶段窗口初值一般仅为一个 MSS，这样在很长一段时间内速率会很低，而可用带宽可能很大，为了尽快探明最大可用带宽，启动阶段每经过一个 RTT，窗口值就增大一倍，直到出现一个报文丢失，这个特殊过程才结束。该过程称作慢启动。其次，遇到一个超时事件的时候，TCP 将窗口值设成 1 个 MSS，并慢启动到丢失报文前窗口值的一半。但是收到 3 个冗余 ACK 的时候并不慢启动而是直接减半，这种行为叫快速恢复。

----------中英文对照表----------

主机 host 端系统 end system 通信链路 communication link 分组交换机 packet switch 路由 router 链路层交换机 link-layer switch

互联网服务商 Internet Service Provider,ISP 互联网标准 Internet Standard 局域网 intranet 协议 Protocol 电路交换 circuit switching

报文 message 分组 packet 缓存 buffer 排队时延 queue delay 丢包 packet lost 报文段 segment