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；**尽力而为交付服务**（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

2. 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：(1)加密确保应用程序报文不被入侵者读取(2) TCP 保证可靠、按序的交付；UDP 不可靠、无序的交付

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. 多路复用要求：(1)主机中每个套接字应分配一个唯一的标识 (2)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**）

IP 数据报包含源 IP&端口号、目的 IP&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：（伪头部+头部+数据段）

伪头部：源/目的 IP，0，上层协议 ID=17(UDP)，UDP 长度

头部：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(首部加数据)，检查和；

4.UDP 检验和：所有 16 位比特字相加，若溢出，则回卷(相加和+1)；然后取反（所有的 0 转换成 1，1 转换成 0）；

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。(1)错误检测 (2)接收方反馈 ACK/NAK (3)出错重传（**自动重传请求协议 ARQ**）。

Rdt2.1 ACK/NAK 出错：发送方发现受损或 NAK 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，NAK 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,能处理**冗余数据分组**）；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（比特交替协议）：可能丢包：每次发送一个分组前启动**定时器**（发送方无法区分数据分组丢失/ACK 丢失/分组或 ACK 过度延时），在超时前没收到 ACK/ 期待的 ACK，重发。

Rdt2 和 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**（GBN&SR(1)增加序号范围 (2)缓存分组）。

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(信道)利用率：U = t /(RTT+L/R)

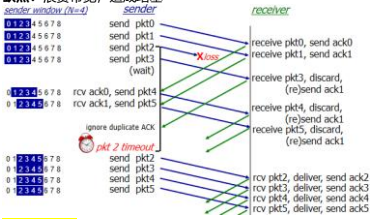
回退 N 步 GBN(滑动窗口协议)：

发送方(1)只对基序号分组使用定时器 (2)超时重传所有已发送但未被确认的分组 (3)仅维护基序号和下一个序号

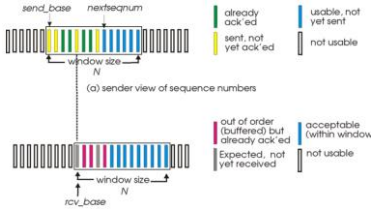
接收方(1)不缓存失序分组 (2)**累计确认**(表明接收方已正确接收到序号 n 及以前的所有分组) (3)对失序分组发送重复 ACK

优点：(1)发送方只需维护一个定时器(2)接收方不需要缓存失序分组(3)易于实现

缺点：浪费带宽，造成堵塞



选择重传 SR：

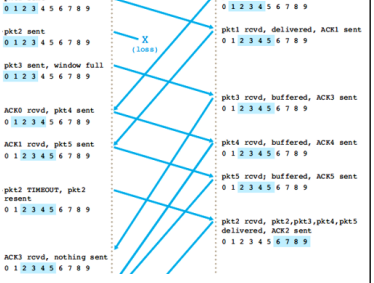


发送方(1)每个分组有自己的定时器 (2)超时仅重传未被确认的分组(3)收到的 ACK 不是最小未确认分组则标记;若是则滑动窗口

接收方(1)缓存失序分组 (2)单独确认每个收到的分组 (3)若收到窗口前冗余分组 n，说明发送方超时，发送 ACK(n)

优点：不浪费带宽

缺点：(1)发送方需为每个未确认分组维护一个定时器 (2)接收方缓存失序分组 (3)复杂



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. TCP 特性：点到点(单个发送方，单个接收方)，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

(以太网/PPP 的**最大传输单元 MTU**=1500bytes，TCP/IP 首部=40bytes，**最大报文段长度 MSS**=1460bytes，TCP 发送大文件时将该文件划分成长度为 MSS 的若干块，最后一块除外)

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，累计确认（第一个丢失字节，即之前的都已确认过，即使 ACK 中途丢失也无需重传）（接收方保留失序字节，并等待缺失字节以填补该间隔）

首部：32bits(首部长度+保留未用+标志位+接受窗口 rwnd)

检验和、紧急数据指针（指向最后一个字节）、**选项、数据段**

（标志位：CWR：明确拥塞通告回显；ECE：拥塞窗口缩减；URG：紧急数据；ACK：确认；PSH：立即交给上层；RST：不接受连接；SYN：建立连接；FIN 拆除连接）

（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）

确认捎带在服务器到客户的数据的报文段中，最后客户向服务器发送的报文段中没有数据还仍有序号

3. 往返时间 RTT 估计

估计平均 RTT 指数加权移动平均 a = 0.125 , b = 0.25

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

DevRTT=(1-b)DevRTT+b|SampleRTT-EstimatedRTT|

超时值：TimeoutInterval = EstimatedRTT + 4*DevRTT

(初始为 1s,超时后加倍,只要收到报文段则用上述公式计算)

4.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）； (2) 如果还有未确认的报文段，启动定时器；

超时间隔加倍：(1)超时事件中重启定时器时，TimeoutInterval 设置为两倍 (2)另两个事件中重启定时器时，由 RTT 公式计算

快速重传：3 个**冗余 ACK** 时（**why 3 次?**

ACK 的产生

接收端事件	接收端动作
收到一个期待的报文段，且之前的报文段均已发送确认	推迟发送ACK. 在500ms时间内若无下一个报文段到来，发送ACK
收到一个期待的报文段，且前一个报文段被推迟确认	立即发送ACK
收到一个失序的报文段（序号大于期待的序号），检测到序号间隙	立即发送重复的ACK
收到部分或全部填充间隙的报文段	若报文段始于间隙的低端，立即发送ACK

GBN-like(1)累积确认 (2)失序报文发送重复 ACK (3)仅维护基序号和下一个序号(区别：缓存失序报文段，超时后仅重传最早未确认的报文段) (4)仅维护一个定时器

SR-like(1)缓存失序报文段 (2)仅重传接收方缺失的数据（区别：在 SACK 选项头给出非连续数据块的上下边界）

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；

5. TCP 流量控制：

速度匹配服务，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:

接收窗口 rwnd= RcvBuffer-[LastByteRcvd - LastByteRead]

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 ≤ rwnd (连接中大小改变)

接收窗口为 0 时，发送方继续发送只有一个字节数据的报文段，从接收方的响应中获知窗口大小。

8. 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**(SYNbit=1，ACKbit=0)；给出客户选择的起始序号 seq=client_isn；不包含数据；

Step 2.服务器 TCP 用 **SYNACK 报文段**响应（SYNbit=1,ACKbit=1)；给出服务器选择的起始序号 seq=server_isn; 确认客户的起始序号 ACKnum=client_isn+1；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（SYNbit=0, ACKbit=1)；确认服务器的起始序号 ACKnum=server_isn+1; seq=client_isn+1；可能包含数据（客户端分配缓存和变量）

- **四次挥手 关闭连接**：

客户端：向服务器发送 FIN，等待服务器确认；

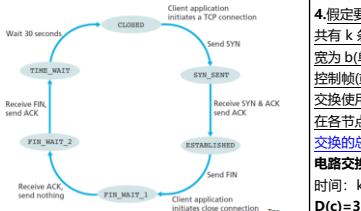
服务器：向客户端发送 ACK，确认请求；

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。

Server TCP FSM

Client TCP FSM



拒绝连接 主机接收到一个 TCP 报文段,其端口号或源 IP 与主机上进行中的套接字都不匹配时，向源发送 **RST=1** 的报文段

nmap：端口扫描工具

6. 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**(因为 IP 层不向端系统提供显式的网络拥塞反馈),发送方根据自己感知的网络拥塞程度,限制发送速度。而不是**网络辅助的拥塞控制**(ABR 中路由器显式通知)。

发送方使用**拥塞窗口 cwnd** 限制，LastByteSent-LastByteAcked≤ min{cwnd,rwnd}。发送速率为**(cwnd/RTT)bytes/sec**

TCP 丢包事件：(1)超时 (2)收到 3 个冗余 ACK

TCP 是**自计时的**：使用确认来触发（或计时）增大 cwnd

TCP 拥塞窗口的调节（拥塞控制）：加性增，乘性减（**AIMD**）。

Tahoe：所有丢包事件都进行相同处理，进入慢启动

Reno：有快速修复

7. TCP 吞吐量

吞吐量与丢包率 L 的关系: **[1.22MSS / (RTT$\sqrt{L}$)]**

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。

田野小测 1

1 一条分组交换链路，带宽 10Mbps，现有 20 位用户先后希望建立并保持通信连接，每位用户产生 1Mbps 的数据流量，20 位用户均建立连接，数据通信产生较多丢包和较大时延；一条线路交换链路，仅有 10 位用户成功建立连接，剩余 10 位用户无法建立连接。

2 哪些层的协议工作在主机上：应用层、传输层、网络层、链路层、物理层；

哪些层的协议工作在主机的操作系统中：传输层、网络层；

哪些层的协议工作在路由器上：网络层、链路层、物理层；

哪些层的协议工作在交换机上：链路层、物理层。

3 以下哪种不是 ISP 的连接方式：双方链接内容提供商网络实现连接。

4 带宽和吞吐率的区别?

带宽：单位时间能够传输的最大数据量

吞吐率：单位时间能够成功实现的最高传输速率

田野小测 2

1 假设你从 USTC 校园网发出 www.baidu.com 的 DNS 询问，在名字解析过程中可能包括了哪些 DNS 服务器：Local DNS server of USTC, root DNS server, ".com" TLD server, Authoritative server of the "baidu.com" domain；一定不包括哪些 DNS 服务器：Local DNS server of Baidu company。

2 进程 A 想与另一个主机上的进程 B 通信时，进程 A 通过进程 B 的所在的主机的 IP 和进程 B 绑定的端口号来识别进程 B。

3 假设 Alice（有个 Web-based e-mail account 比如 qqmail 或 gmail）给 Bob（通过 POP3 访问他的邮件服务器）发送了一个信息，该信息是如何从 Alice 的主机传递到 Bob 的主机的?

Alice 用户代理--(HTTP)->Alice 邮件服务器--(SMTP)->Bob 邮件服务器—(POP3)->Bob 用户代理

田野小测 3

1 TCP 提供了服务：流控制，建立和拆除连接，拥塞控制，丢包修复，按序交付；没有提供服务：延迟保证，带宽保留。

2 主机 B(纯粹接收方)没有数据发给主机 A 时，也可以只发 ACK

3 主机 A 通过 TCP 连接发送了两个报文段来回主机 B，第一个 seqnum=90，第二个 seqnum=110：

(1)第一个报文段含有数据量：20bytes

(2)假设第一个报文段丢失但第二个报文段到达 B，B 给 A 发送的 acknum=90

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：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：提供了发现/定义超时的方法，避免死锁无限等待
- 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

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

2. 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.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

4.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 条链路，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): (1)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+k-1)*(p+h)/b (2)排队时间：kq (3)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小于分组交换的总时延

5.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**）

6. 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

7.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

8.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 cwnd = 4KB*2 = 8KB 在新建立的 TCP 连接上，先慢启动，第一次超时时，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) 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。故文件发送结束时，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 (4)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

田野作业 1

1 服务器包含两条链路的路径，向客户端传输两个数据包，每个长度为 L，第一条和第二条链路的带宽分别为 Rs、Rc，两条链路的传播时延均为 d_prop。

(1)**如果第一条链路时整个路径的瓶颈，即 Rs<Rc。**服务器“背靠背”传输数据，即完成第一个数据包的传输后立刻开始传输第二个数据包。问客户端收到两个数据包的第一个 bit 之间间隔多长时间? 间隔 L/Rs，因为

第一个包的第一个 bit 到达客户端时刻 L/Rs+2d_prop

第二个包的第一个 bit 到达客户端...2L/Rs+2d_prop

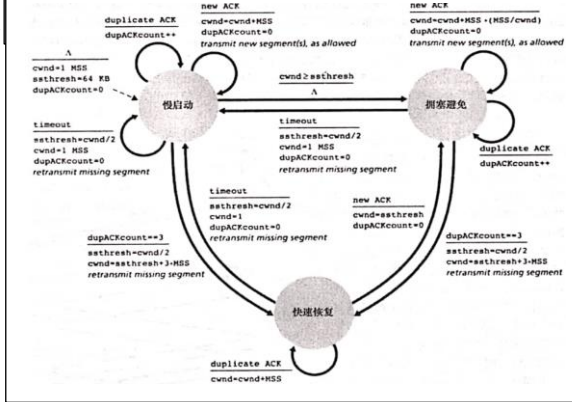
(2)**如果第二条链路是整个路径的瓶颈，即 Rc<Rs。**服务器完成第一个数据包的传输后等待 T，再传第二个数据包，问 T 满足什么条件，第二个数据包在路由器中不会排队等待? 第一个包的最后一个 bit 从路由器发出时刻:

L/Rs+L/Rc+d_prop

第二个包的最后一个 bit 到达路由器的时刻

2L/Rs+d_prop+T

满足后者大于前者，故 T≥L/Rc-L/Rs



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：
   （1）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。
3. 具体构成描述
   （1）**终端设备**：主机=端系统(运行网络应用程序)
   （2）**通信链路**：光纤,铜线,射频,卫星等(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**；
   路由：分组从发送端到接受端经过的通信链路以及分组交换机的序列。
   （4）**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 e.g. WeChat, Skype, 802.11】不同系统的相同层实体之间
   （5）**ISP**（因特网服务提供商）: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（因特网交换点）IXP 相连/连接到更高层的 ISP/PoP/多宿/对等/内容提供商网络
4. 提供服务描述(1)给应用提供服务的基础设施 (2)给 apps 提供编程接口

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(hosts,servers,运行网络应用程序)
2. 接入网：将终端连接到其边缘路由器的物理链路。
(1)家庭接入:数字用户线(DSL,电路交换)(电话,单独线路)和电缆(分组交换)(有线电视,共享媒体),光纤到户(FTTH),卫星
(2)企业接入:局域网(LAN,以太网),Wifi
(3)广域无线接入:3G 和 LTE
物理媒体分类：导引型（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)
(1)通信前预留好端-端资源（对比：分组交换不预留资源）
(2)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：**频分复用 FDM/时分复用 TDM** ；
TDM(时间->帧->时隙):能在请求时间内为端到端保持一个确定量的带宽。
FDM(专用频段)：FDM 需要复杂的模拟硬件来将信号转换到合适频带上。
2. 分组交换(按需分配)：(电缆) 主机将应用报文划分成**分组**。
(1)交换机仅在接受到整个分组后才可以开始转发（**存储转发传输**）[N 条链路有 N-1 台路由器，P 个分组] 端到端传输时延：**(P+N-1)L/R**
(2)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**输出缓存(输出队列)**满，**分组丢失。**
(3)每台路由器有**转发表**，将目的地址映射成为输出链路
优点：适合突发数据，简单有效成本低，不需建立电路；更好的带宽共享
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
3. 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 时延、丢包和吞吐

R=链路带宽(bps); L=packet length (bits)
a=average packet arrival rate
1. 时延类型：1 处理时延【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
2. 节点延迟 nodal=处理 proc+排队 queue+传输 trans+传播 prop
3. 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。瞬时/平均 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(瓶颈链路): **min(Rc,Rs,R/客户服务器对数)**
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；上层协议的性能依赖于下层协议
5 层因特网协议栈：
应用层：HTTP,SMTP,FTP,DNS 报文(消息 message)/应用程序
运输层：TCP、UDP 报文段(分组 segment)/进程-进程
网络层：IP 协议,路由选择协议 数据报(datagram)/主机-主机
链路层:以太网、WiFi、电缆接入网的 DOCSIS 协议/帧(frame)
物理层：双绞铜线、同轴电缆、光纤的协议
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：应用层(7)和传输层(4)间加了表示层(6)和会话层(5)
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：源和目的地是五层，链路交换机是链路层&物理层，路由器是网络层&链路层&物理层。
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）

1.6 网络攻击

(1)恶意软件、僵尸网络、自我复制、病毒、蠕虫。(2)拒绝服务攻击（DoS）：弱点攻击、带宽洪泛、连接洪泛。分布式 DoS（DDoS）。(3)分组嗅探器。(4)IP 哄骗-端点鉴别。

Chap2 应用层

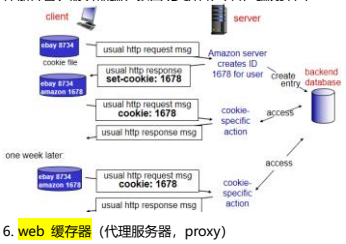
2.1 基本概念

1.网络应用架构：
(1)客户-服务器
服务器：永远在线运行服务器程序；具有永久的 IP 地址；数据中心；
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。
(2)对等架构 P2P: 任意端系统（**对等方**）可以直接通信，没有总是运行的服务器，使用动态 IP。【优点-**自扩展性**(新对等方带来新的服务能力和服务需求)，高度可伸缩，缺点-难于管理，缺乏安全性和可靠性】
每一个主机（对等方）既是服务器又是客户端
2.进程通信
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: 25
4.因特网提供的传输服务 app 对传输服务的需求: data loss, throughput, time sensitive
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,安全性
UDP 无连接,不可靠传输,报文乱序到达；**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,安全性
（**SSL** 在应用层：加密 TCP 连接,数据可靠性,端点认证）

2.2 WEB 和 HTTP

1.应用层协议定义：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 超文本传输协议：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，**无状态协议**(*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（如一个包含 HTML 文本和 5 个 JPEG 图形的 WEB 页面有 6 个对象）

URL:
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 RTT+对象传输时间; 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
（RTT 往返时间-一个小分组从客户发送到服务器再返回确认被收到的时间）
持久 HTTP:HTTP1.1 一个 TCP 连接上发送多个对象
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
流水线方式：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
4. 报文格式：
请求报文：请求行(方法,URL,HTTP 版本)，首部行(Host, Connection, User-agent)，回车（表示结束）
响应报文：状态行(协议版本,状态码,状态信息),首部行,实体体
上传方法：post 放在报文体内；get 放在 URL 内
HTTP1.0: GET,POST,HEAD
HTTP1.1: GET,POST,HEAD,PUT,DELETE
5. cookie：保存状态,无状态 HTTP 上建立用户会话层
存储位置：服务器端：返回 ID 给客户/客户端:文件中



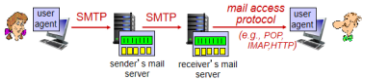
6. web 缓存器（代理服务器，proxy）
即是客户端又是服务器，保存最近请求过的对象的拷贝；减少客户请求的响应时间，减少机构介入链路上的流量。
一般由 ISP 架设。
条件 GET: If-Modified-Since,仅在自指定日期之后该对象被修改过才发送该对象，否则只发送 304

2.3FTP 文件传输协议（有状态协议） 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

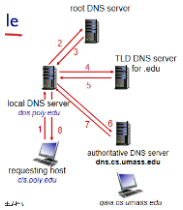
1.三部分:用户代理、邮件服务器、简单邮件传输协议
2.用户代理：（1）编辑邮件等 e.g., Outlook, elm, Mozilla Thunderbird（2）将要外发的邮件发送到用户的邮件服务器；（3）从用户邮箱中取邮件
3.邮件服务器：（1）用户信箱：存放到来的邮件
（2）发送报文队列：存放要发送出去的邮件
（3）报文传输代理 MTA：运行在服务器后台的系统守护进程，负责在邮件服务器之间传输邮件，及将收到的邮件放入用户信箱。
电子信箱(1)由计算机上的一个存储区域(如磁盘上的一个文件)组成(2)每个信箱被分配了唯一的电子邮件地址
4. 简单邮件传输协议—SMTP
-邮件服务器之间传输邮件采用客户-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
SMTP v.s. HTTP:
(1)SMTP 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；
(2)SMTP 要求报文采用 7 比特 ASCII 码格式，HTTP 无限制
(3)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；
(4)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）
5.邮件访问协议：允许用户从信箱中提取邮件



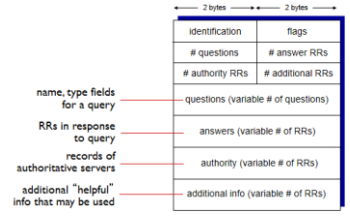
POP3：无状态，特许-事务处理-更新，邮件下载到本地
IMAP：有状态，所有邮件保存在服务器上
HTTP：浏览器到邮件服务器使用 HTTP 而非 SMTP

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（冗余）
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（3）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
-本地 DNS 服务器：不属于服务器层次结构
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）



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 记录和报文
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
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 报文格式如下：



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，超过则使用 TCP。
往 DNS 中插入资源记录 eg: new startup "Network Utopia"
-向注册登记机构注册域名"networkuptopia.com"，提供权威 DNS 服务器（主域名服务器，辅助域名服务器）名字和 IP
对每个权威域名服务器，注册机构往 TLDcom 服务器中插入两条资源记录：
(networkutopia.com, dns1.networkutopia.com,NS)
(dns1.networkutopia.com,212.212.212.1,A)
-在权威 DNS 服务器中输入 www.networkuptopia.com 的 Type A 记录和 mail.networkutopia.com 的 Type MX 记录

2.5 P2P 文件分发 e.g. 文件分发 (BitTorrent,Xunlei), Streaming(PPTV), VoIP(Skype)

u_s:服务器接入链路的上载速率，**u_i:**第 i 对等方接入链路的上载速率，**d_i:**第 i 对等方接入链路的下载速率，**F:**被分发的文件长度，**N:**要获得该文件副本的对等方的数量，**分发时间** 所有对等方得到该文件副本所需要的时间
最小分发时间 D_P2P≥max{F/u_s,F/d_min,NF/(u_s+∑u_i)}
BitTorrent P2P 协议
洪流：参与一个特定文件分发的所有**对等方**的集合
当一个对等方 Alice 加入某洪流时，它向**追踪器**注册自己，并周期性通知追踪器它仍在该洪流中。追踪器随机从参与对等方的集合中选择子集，将其 IP 发给 Alice，Alice 试图与该列表上所有对等放创建并行的 TCP 连接。
请求：**最稀缺**(邻居中副本最少)**优先**
发送：**疏通**（Alice 对每个邻居都持续测量接收到比特的速率，并确定以最高速率流入四个邻居） **一报还一报**

2.6 视频流和内容分发流

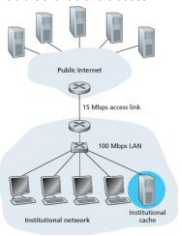
视频编码：spatial (color+repeated num) & temporal (frame differences)
HTTP 流：服务器尽快发送，客户缓存收到的字节至预先设定的门限后，应用程序播放视频
DASH：视频编码为比特率不同的几个版本，对应不同质量，客户根据带宽量动态请求
HTTP 服务器中**告示文件**为每个版本提供 URL&比特率
内容分发网 CDN:管理分布在多个地理位置上的服务器，在他的服务器中存储视频的副本

2.7 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：此时，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
- 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

田野作业 1

2 institutional network user issue 16 requests per second, the average object size is 1.2Mb. 求 minimum cache hit ratio so that the access link not congested?



假设命中率是 x
1.2*16*(1-x)≤15Mbps
x≥0.21875

3 Suppose with your Web browser you click a link to retrieve an HTML Web page that references eight very small objects on the same server. The RTT between Web server and local host is RTT0. Neglecting transmission times, how much time elapses with:
a. Non-persistent HTTP with no parallel TCP connections?
2RTT0+8*2*RTT0
b. Non-persistent HTTP with the browser configured for 5 parallel connections? 2RTT0+2*2*RTT0
c. Persistent HTTP without pipelining? 2RTT0+8RTT0
d. Persistent HTTP with pipelining? 2RTT+RTT

4 假设你创办了公司网址是"www.chuangye.com"。你的公司在地址为 IP1 的主机上运行 web 服务器，在地址为 IP2 和 IP3 的主机上分别运行名为"ns1.chuangye.com"和"ns2.chuangye.com"的权威域名服务器。你的域名注册服务商需要在何种域名服务器注入哪些 RR 信息？"ns1.chuangye.com"和"ns2.chuangye.com"上应该包含什么 RR。写出这些 RR 的三元组。描述第一个访问"www.chuangye.com"的用户在浏览器上打开网站页面的过程。

在 `.com TLD server` 上注入：

```
(chuangye.com, ns1.chuangye.com, NS)
(chuangye.com, ns2.chuangye.com, NS)
(ns1.chuangye.com, 212.212.212.1, A)
(ns2.chuangye.com, 212.212.212.2, A)
```

在 `ns1.chuangye.com` 上应该包含：

```
(www.chuangye.com, 212.212.71.4, A)
```

在 `ns2.chuangye.com` 上应该包含：

```
(www.chuangye.com, 212.212.71.4, A)
```