

# 西北工业大学



## 计算机网络作业集

### Computer Networking

软件学院 钱锋 编

2024 年 4 月 13 日

## 前言

这本作业集的结构是按照教材 [1] 的章节结构来划分的, 请老师在批改的时候通过目录定位到对应的章节作业. 此外, 由于作业设计的习题只是教材习题中的一部分, 因此题号与教材上略有差异——本作业集的习题按照其在教材上的顺序, 从 1 开始重新编号.

钱锋

西北工业大学 软件学院

2024 年 3 月



## 目录

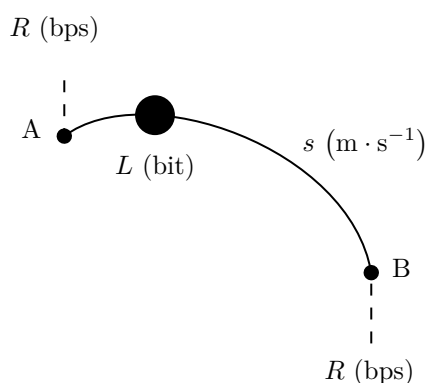
前言	i	第二章 应用层	7
		2.1 习题	7
第一章 计算机网络和因特网	1		
1.1 习题	1	参考文献	11



## 第1章 计算机网络和因特网

### 1.1 习题

**习题 1.1.1.** 考虑两台主机 A 和 B 有一条速率为  $R$  (bps) 的链路相连. 假定这两台主机相隔  $m$  (m), 沿该链路的传播速率为  $s$  (m/s). 主机 A 向主机 B 发送长度为  $L$  (bit) 的分组.



- a) 用  $m$  和  $s$  来表示传播时延  $d_{\text{prop}}$ .

**解.** 根据传播时延的定义可知, 传播时延是节点间距离与传播速度的比值, 即  $d_{\text{prop}} = m/s$ .

- b) 用  $L$  和  $R$  来表示该分组的传输时间  $t_{\text{trans}}$ .

**解.** 传输时延是指数据从节点被推入通信链路中所花的时间, 它是数据长度与链路传输速率的比值, 因此  $d_{\text{trans}} = L/R$ .

- c) 忽略处理和排队时延, 得出端到端时延的表达式.

**解.** 在该问题中, 端到端时延由传输时延和传播时延两部分组成, 而这两个部分在前两问中都已经计算过了, 因此

$$d_{\text{end-end}} = d_{\text{prop}} + d_{\text{trans}} = \frac{m}{s} + \frac{L}{R}.$$

- d) 假定主机 A 在时刻  $t = 0$  开始传输该分组, 在时刻  $t = t_{\text{trans}}$ , 该分组的最后一个比特在什么地方?

**解.** 该时刻已经经历了一个传输时延, 即第一个分组已经完全传输到链路当中了. 该分组的最后一个比特在链路中已经传播了  $d_{\text{trans}} - d_{\text{trans}} = 0$  的时间. 在这段时间内其走过的距离为

$$l = s \times 0 = 0,$$

即此时该分组的最后一个比特位于链路的 A 端点处.

e) 假定  $d_{\text{prop}}$  大于  $d_{\text{trans}}$ . 在时刻  $t = d_{\text{trans}}$ , 该分组的第一个比特在何处?

**解.** 根据已知条件, 该时刻该分组的第一个比特依然在链路中传播, 由于第一个比特是直接进入链路开始传播的, 因此它已经传播了  $d_{\text{trans}}$  个单位时间, 于是它所处的位置是  $l = sd_{\text{trans}} = sL/R$ .

f) 假定  $d_{\text{prop}}$  小于  $d_{\text{trans}}$ . 在时刻  $t = d_{\text{trans}}$ , 该分组的第一个比特在何处?

**解.** 根据已知条件, 该时刻第一个比特已经到达 B 主机了.

**习题 1.1.2.** 考虑从主机 A 向主机 B 通过分组交换网发送语音 (VoIP). 主机 A 将模拟语音转换为传输中的 64kbps 数字比特流, 然后主机 A 将这些比特分为 56 字节的分组. A 和 B 之间有一条链路: 它的传输速率是 2Mbps, 传播时延是 10ms. 一旦 A 收集了一个分组, 就将它向主机 B 发送. 一旦主机 B 接收到一个完整的分组, 它将该分组的比特转换成模拟信号. 从比特产生 (从位于主机 A 的初始模拟信号起) 的时刻起, 到该比特被解码 (在主机 B 上作为模拟信号的一部分), 花了多少时间?

**解.** 我们先统一单位, 统一用比特作为单位来进行计算. 于是, 主机 A 发送的每个分组的大小为  $56 \times 8 = 448\text{b}$ . 这就是说, 主机 A 只需要  $448/64000$  秒就能产生一个完整的分组, 这就是传输过程中的处理时延. 这就是说,

$$d_{\text{proc}} = \frac{56\text{B}}{64\text{kbps}} = \frac{56 \times 8\text{b}}{64 \times 10^3\text{b} \cdot \text{s}^{-1}} = 7 \times 10^{-4}\text{s} = 0.7\text{ms}.$$

接下来考虑传输时延, 事实上, 根据传输时延的定义,

$$d_{\text{trans}} = \frac{56\text{B}}{2\text{Mbps}} = \frac{56 \times 8\text{b}}{2 \times 10^6\text{b} \cdot \text{s}^{-1}} = 2.24 \times 10^{-4}\text{s} = 0.224\text{ms}.$$

于是最终的端到端时延为

$$d_{\text{end-end}} = d_{\text{proc}} + d_{\text{trans}} + d_{\text{prop}} = 0.7\text{ms} + 0.224\text{ms} + 10\text{ms} = 10.924\text{ms}.$$

**习题 1.1.3.** 假设用户共享一条 3Mbps 的链路. 又设每个用户传输时要求 150kbps, 但是每个用户仅有 10% 的时间传输.

a) 当使用电路交换时, 能够支持多少用户?

**解.** 使用电路交换时, 每个用户都会占用一定的带宽, 因此我们可以直接用链路带宽和用户需求做商, 则

$$\text{userNum} = \frac{3\text{Mbps}}{150\text{kbps}} = \frac{3 \times 10^6\text{bps}}{150 \times 10^3\text{bps}} = 20,$$

因此在使用电路交换时, 能够支持最多 20 名用户.

b) 当使用分组交换时, 用户正在传输的概率为?

**解.** 由于每个用户只有 10% 的时间传输, 所以其正在传输的概率为 0.1.

c) 假设使用分组交换, 且有 120 名用户. 在任何给定时刻, 实际有  $n$  个用户正在传输的概率是?

**解.** 显然, 我们可以将这个问题视为连续进行 120 次随机事件发生概率为 0.1 的 Bernoulli 试验, 正在传输数据的用户数量  $X \sim B(0.1, 120)$ . 那么

$$P(X = n) = C_{120}^n \times 0.1^n \times 0.9^{120-n}.$$



其中  $C_m^k = \binom{m}{k}$  为组合数, 即有限集合  $A$  ( $A$  中含有  $m$  个元素, 记作  $|A| = m$ ) 的满足  $|S| = k$  的子集  $S \subset A$  的数量.

**习题 1.1.4.** 在一个电路交换网络中, 每台端系统与一台交换机直接相连, 当两台端系统要通信时, 该网络在两台端系统之间建立一条专用的端到端连接. 用户在忙时以 1000kbps 的速率产生数据, 但忙时仅以  $p = 0.1$  的概率产生数据, 假定电路交换链路的带宽为 1Gbps.

a) 在采用电路交换技术时, 能被支持的最大用户数量  $N$  是多少?

**解.** 可以用链路带宽与用户需求带宽做商直接得到能够支持的最大用户数量

$$N = \frac{1\text{Gbps}}{100\text{kbps}} = \frac{1 \times 10^9 \text{bps}}{100 \times 10^3 \text{bps}} = 10^4.$$

b) 在使用分组交换时, 假设有  $M$  个用户, 多于  $N$  个用户发送数据的概率为?

**解.** 设正在发生数据的用户数量为  $X$ , 显然  $X \sim B(0.1, M)$ . 同时, 由于两个用户发送数据与否是独立的, 因此随机事件  $\{X = N + 1\}, \{X = N + 2\}, \dots, \{X = M\}$  是随机事件  $\{X > N\}$  的一个划分, 由全概率公式得

$$P(X > N) = \sum_{k=N+1}^M P(X = k) = \sum_{k=N+1}^M C_M^k \times 0.1^k \times 0.9^{M-k}.$$

**习题 1.1.5.** 令  $a$  表示在一条链路上分组的到达率, 其单位为 pkt/s,  $\mu$  为一条链路上分组的传输率, 基于  $d = d_{\text{queue}} + d_{\text{trans}}$ , 用  $a$  和  $\mu$  表示  $d$ .

**解.** 要用  $a$  和  $\mu$  表示  $d$ , 只需用  $a$  和  $\mu$  表示  $d_{\text{queue}}$  和  $d_{\text{trans}}$ . 我们只考虑 1 个分组, 那么

$$d_{\text{queue}} = \frac{1}{a}, \quad d_{\text{trans}} = \frac{1}{\mu}.$$

于是总时延

$$d = \frac{1}{a} + \frac{1}{\mu}.$$

Little 公式是排队论中的一个著名公式, 设  $N$  是传输队列中待传输分组的平均数量,  $a$  为分组到达率,  $d$  为一个分组经历的平均总时延 (即排队时延加传输时延). Little 公式表明,

$$N = ad.$$

其实呢, Little 公式的意思就是说, 每分钟有  $a$  个人来排队, 处理一个人平均需要  $d$  分钟, 那么这个队伍的长度就应该是  $ad$ .

**习题 1.1.6.** 考虑一台路由器缓存前面的一条出链路. 假定某队列平均包含 10 个分组, 平均分组排队时延是 10ms, 链路传输速率为 100pkt/s. 不考虑丢包情况, 使用 Little 公式计算平均分组到达率.

**解.** 由已知条件,  $N = 10$ ,  $d_{\text{queue}} = 10\text{ms}$ ,  $d_{\text{trans}} = \frac{1}{100}\text{s}$ , 那么

$$N = ad = a(d_{\text{queue}} + d_{\text{trans}}),$$

代入数值,

$$\begin{aligned} 10\text{pkt} &= a\text{pkt/s} \left( 10\text{ms} + \frac{1}{100}\text{s} \right) = a\text{pkt/s} \left( \frac{1}{100}\text{s} + \frac{1}{100}\text{s} \right) \\ &= a \times 0.02\text{s}, \end{aligned}$$

解得平均分组到达率  $a = 10/0.02 = 500\text{pkt/s}$ .

**习题 1.1.7.** 假设你希望从波士顿向洛杉矶紧急传送  $40 \times 10^{12}\text{B}$  的数据, 你有一条  $1\text{Gbps}$  的专用链路可用于传输数据. 你是愿意通过这条链路传输数据, 还是愿意使用 FedEx 夜间快递来交付?

**解.** FedEx 夜间快递是指 FedEx 提供的在晚间送达包裹的服务, 这种服务通常适用于那些需要在次日早晨或者特定时间内送达的包裹<sup>1</sup>. 我们不妨假设 FedEx 可以在 72 小时内将包裹送达洛杉矶. 那么使用专用链路的传输时间

$$\begin{aligned} t &= \frac{40 \times 10^{12}\text{B}}{1\text{Gbps}} = \frac{8 \times 40 \times 10^{12}\text{b}}{10^9\text{bps}} \\ &= 320,000\text{s} = \frac{320,000\text{s}}{3600\text{s/h}} \approx 89\text{h} > 72\text{h}. \end{aligned}$$

于是我选择 FedEx 夜间快递.

**习题 1.1.8.** 在包括因特网的现代分组交换网络中, 源端系统将长应用层报文分段为较小的分组并向网络发送, 接收方将这些分组重新装配为原始报文, 这个过程称为报文分段. 考虑一个长度为  $8 \times 10^6\text{bit}$  的报文, 经过两个分组交换机到达目的端系统, 每段链路的传输速率是  $2\text{Mbps}$ . 忽略传播、排队和处理时延.

- a) 考虑从源端系统到目的端系统发送该报文且没有报文分段, 从源端系统到第一台分组交换机需要多长时间? (每台分组交换机均适用存储转发分组交换) 从源端系统移动该报文到目的端系统需要多长时间?

**解.** 没有报文分段时, 从源端系统到第一台分组交换机需要的时间为

$$t_1 = \frac{8 \times 10^6\text{bit}}{2\text{Mbps}} = 4\text{s},$$

该文件最终到达目的端系统的时间为

$$t = 3t_1 = 12\text{s}.$$

- b) 假设报文被分段为 800 个分组, 每个分组长度为  $10,000\text{bit}$ . 从源端系统移动第一个分组到第一个分组交换机需要多长时间? 从第一台交换机发送第一个分组到第二台交换机, 从源端系统发送第二个分组到第一台交换机各需要多长时间? 什么时候第二个分组能被第一台交换机全部收到?

**解.** 有报文分段时, 从源端系统移动第一个分组到第一台分组交换机需要的时间为

$$t'_1 = \frac{10,000\text{bit}}{2\text{Mbps}} = 5\text{ms}$$

由于每个分组的传输速率相同, 且不考虑出了传输时延以外的各种时延, 因此从第一台交换机到第二个交换机需要  $t'_1$  时间, 第二个分组源端系统到第一台交换机需要  $t'_1$  时间. 于是在整个过程经历  $2t'_1$ , 即  $10\text{ms}$  后, 第二个分组被第一台交换机全部收到.

- c) 使用报文分段时, 该文件从源端系统到目的端系统需要多长时间?

<sup>1</sup>这段材料由 ChatGPT 生成, ChatGPT 可能会生成错误的信息, 请谨慎参考.

**解.** 每个分组需要  $3t_1'$  的时间才能完全送到目的端系统, 但是由于在整个过程中三条链路都在不停地传输数据, 因此事实上传输 800 个分组只需要  $(800 + 2)t_1$  的时间, 这是因为我们需要  $2t_1$  的时间才能让三条链路全部工作起来. 所以他需要的时间为

$$t' = (800 + 2) \times 5\text{ms} = 4010\text{ms} < 4\text{s},$$

发现使用报文分段时, 传输速率能够大幅减小时延.

d) 除了减小时延以外, 使用报文分段还有什么原因?

**解.** 除了减小时延以外, 使用报文分段还能够提高链路的使用率. 事实上, 在不使用报文分段的时候, 传输的第二阶段 (即从交换机 1 到交换机 2) 是源端系统到交换机 1 的链路和交换机 2 到目的端系统的链路是空闲的. 在传输的第一阶段和第三阶段同理, 而当网络中接入的用户数量增加, 链路的利用率将会变得至关重要, 因此采用报文分段的方式.

e) 讨论报文分段的缺点.

- 由于报文分段把一段报文分解为若干分组, 因此在拆分报文和重组报文的过程中会产生一定的时间开销;
- 由于分组当中包含一些头部信息, 所以报文分段会产生额外的头部信息, 增加了传输数据量;
- 分组在遇到丢包的时候需要重传 (虽然重传一个分组总比重传一整个文件更容易, 但是若干个分组中有一个分组丢包的概率是要远远大于 1 整个文件丢包的概率的);
- 报文分段可能会降低实时性.



## 第2章 应用层

### 2.1 习题

**习题 2.1.1.** 考虑当浏览器发送一个 HTTP GET 报文时, 通过 Wireshark 俘获到下列 ASCII 字符串 (即这是一个 HTTP GET 报文的实际内容)<sup>1</sup>. 字符 `<cr><lf>` 是回车和换行符 (即下面文本中的字符串 `<cr>` 表示了单个回车符, 该回车符包含在 HTTP 首部中的相应位置).

```
1 GET /cs453/index.html HTTP/1.1<cr><lf>
2 Host: gaia.cs.umass.edu<cr><lf>
3 User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.2) Gecko
  /20040804 Netscape/7.2 (ax) <cr><lf>
4 Accept: ext/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;
  q=0.8,image/png,*/*;q=0.5<cr><lf>
5 Accept-Language: en-us,en;q=0.5<cr><lf>
6 Accept-Encoding: zip,deflate<cr><lf>
7 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>
8 Keep-Alive: 300<cr><lf>
9 Connection: keep-alive<cr><lf>
10 <cr><lf>
```

回答下列问题, 指出你在下面 HTTP GET 报文中找到答案的地方.

- a. 由浏览器请求的文档的 URL 是什么?

HTTP 请求报文由**请求行** (request line)、**首部行** (header line) 和**实体体** (entity body) 构成. 请求行的第二个部分即为将要请求的文档的 URL, 不难看出, 浏览器将要请求的是 `/cs453/index.html`.

- b. 该浏览器运行的是 HTTP 的何种版本?

在请求行的最后一个部分可以看到运行的是 HTTP 1.1.

- c. 该浏览器请求的是一条非持续连接还是一条持续连接?

根据 `Connection: keep-alive<cr><lf>` 首部行容易得到, 它请求的是一个持续连接.

- d. 该浏览器所运行的主机的 IP 地址是什么?

HTTP 请求报文一般不包含主机的 IP 地址.

- e. 发起该报文的浏览器的类型是什么? 在一个 HTTP 请求报文中, 为什么需要浏览器类型?

---

<sup>1</sup> 以下报文通过 OCR 识别得到, 在答案不涉及的部分可能会有细微的出入, 敬请见谅. 同时, 可读性起见, 我在排版的时候对每一个 `<cr><lf>` 都执行了换行操作 (当然在真实的 ASCII 报文中这是不可能的).

根据 `User-Agent: Mozilla/5.0` 首部行容易得到, 这是一个 Mozilla 的浏览器, 很可能是 Firefox. 在 HTTP 请求报文中包含浏览器类型的原因是, 服务器可以根据具体的浏览器类型来提供更具针对性的服务.

**习题 2.1.2.** 下面文本中显示的是来自服务器的回答, 以相应上述问题中的 HTTP GET 报文.

```

1 | HTTP/1.1 200 OK<cr><lf>
2 | Date: Tue, 07 Mar 2008 12:39:45GMT<cr><lf>
3 | Server: Apache/2.0.52 (Fedora)<cr><lf>
4 | Last-Modified: Sat, 10 Dec 2005 18:27:46 GMT<cr><lf>
5 | ETag: "526c3-f22-a88a4c80" <cr><lf>
6 | Accept-Ranges: bytes<cr><lf>
7 | Content-Length: 3874<cr><lf>
8 | Keep-Alive: timeout=max=100<cr><lf>
9 | Connection: Keep-Alive<cr><lf>
10 | Content-Type: text/html; charset=ISO-8859-1<cr><lf>
11 | <cr><lf>
12 | <!doctype html public "-//w3c//dtd html 4.0 transitional//en" ><lf><html><lf><
    head><lf><meta http-equiv= "Content-Type" content= "text/html; charset=iso
    -8859-1" ><lf><meta name= "GENERATOR" content= "M0zilla/4.79 [en] (Windows
    NT 5.0; U) Netscape]" ><lf> <title>CMPSCI 453 / 591 / NTU-ST550ASpring 2005
    homepage</titie><lf></head><lf><much more document text following here (not
    shown)>

```

a. 服务器能否成功地找到那个文档? 该文档提供回答是什么时间?

HTTP 响应报文由**状态行**(status line)、首部行和实体体组成. 在状态行的 `200 OK` 中我们能够看出, 服务器找到了所请求的文档, 这个文档将被包含在返回的响应报文中. 从响应报文的 `Date:` 首部行中我们能得知该文档提供回答的时间是 2008 年 3 月 7 日 GMT 12:39:45.

b. 该文档最后修改是什么时间?

从 `Last-Modified:` 首部行中得知该文档上一次修改是在 2005 年的 9 月 10 日, GMT 18:27:46.

c. 文档中被返回的字节有多少?

由 `Content-Length:` 首部行得知文档的字节数为 3874.

d. 文档被返回的前 5 个字节是什么? 该服务器同意一条持续连接吗?

被返回的前五个字节是 `<! do`, 从 `Connection:` 首部行得知服务器同意一条持续连接.

**习题 2.1.3.** 考虑向  $N$  个对等方分发  $F = 20\text{Gb}$  的一个文件. 该服务器具有  $u_s = 30\text{Mbps}$  的上载速率, 每个对等方具有  $d_i = 2\text{Mbps}$  的下载速率和上载速率  $u$ . 对于  $N = 10$ 、100 和 1000 并且  $u = 300\text{kbps}$ 、700kbps 和 2Mbps, 为  $N$  和  $u$  的每种组合绘制出确定最小分发时间的图表. 需要分别针对客户—服务器分发和 P2P 分发两种情况制作.

由于

$$D_{\text{cs}} = \max \left\{ \frac{NF}{u_s}, \frac{F}{d_{\min}} \right\}, \quad D_{\text{P2P}} = \max \left\{ \frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + Nu} \right\},$$

方便起见, 在二者的表达式中, 我们只选择最能体现二者特征的一项, 即体现 C-S 模式分发时间随着用户数量线性增长的  $NF/u_s$  一项和能体现 P2P 文件分发自拓展性的  $NF/(u_s + Nu)$  一项进行数值计算. 事实上, 可以证明总是存在这样的正整数  $N$  使得  $NF/u_s$  称为其中的最大项, 而根据多项式的极限法则, 我们知道  $NF/(u_s + Nu) \rightarrow F/u_s (N \rightarrow \infty)$ .

同时为了防止数值过大, 我们按小时计算文件分发时间. 为此, 我们首先把所有的文件大小、传输速率置为原来的  $1/1000$ , 然后把计算得到的时间再除以  $3.6$ , 这是因为  $1s = 1/3600h$ . 我们首先在 Python 中定义计算分发文件时间的函数:

```
1 def client_server_delivery(file_length,num_of_peer,server_upload,peer_download):
2     return (num_of_peer * file_length / server_upload) / 3.6
3
4 def p2p_delivery(file_length, num_of_peer, server_upload, peer_upload):
5     return ((file_length*num_of_peer)/(server_upload+peer_upload*num_of_peer))
        /3.6
```

记得, 我们把所有的速率全部变成原来的  $1/1000$  了, 而最后的  $3.6$  是为了将单位千秒转化为小时. 接下来, 代入数值进行计算并调用 `matplotlib` 进行绘图操作, 得到结果:

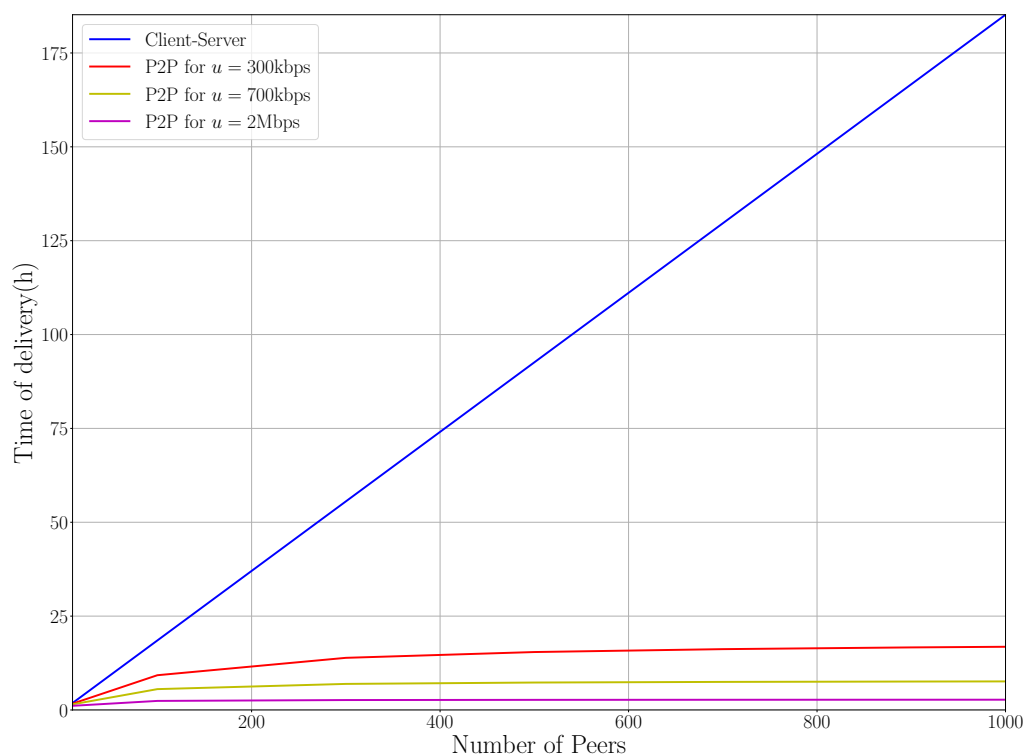


图 2.1: 文件分发时间估计





## 参考文献

- [1] [美] James F.Kurose, [美] Keith W. Ross. 计算机网络: 自顶向下方法 (Computer Networking: A Top Down Approach, 8e)[M]. 陈鸣译. 北京: 机械工业出版社, 2022.
- [2] 王道论坛组. 2024 年计算机网络考研复习指导 [M]. 北京: 电子工业出版社, 2022.

I ♥ NPU

公诚勇毅 永矢毋忘  
中华灿烂 工大无疆

本文档由**钱锋**编写, 钱锋保留一切权利.

文档中出现的部分素材来源于网络, 笔者承诺这些素材仅供学习交流之用, 它们的原作者保留一切权利.

2023 年 西北工业大学 中国西安