

计算机网络作业答案 (2020)

刘威

华中科技大学电子信息与通信学院

Email: liuwei@hust.edu.cn

2020.10

目录

5 第五章 端到端协议	2
5.1 TCP 流量控制	2
5.1.1 补充 1-TCP 首部	2
5.1.2 补充 2-TCP 吞吐量	3
5.1.3 补充 3-TCP 序列号	3
5.1.4 [1] 5-9	3
5.2 TCP 拥塞控制	5
5.2.1 [2] 7-8	5
5.2.2 [2] 7-9	5
5.2.3 [1] 6-16	6

Chapter 5

第五章 端到端协议

5.1 TCP 流量控制

5.1.1 补充 1-TCP 首部

〔题目〕

用 16 进制表示的一个 TCP 报文的头部数据为：

0D28 0015 0000 0006 0000 0000 7002 4000 C029 0000

回答：

- (1) 源端口与目的端口的序号各为多少？
- (2) 发送序列号是多少？
- (3) TCP 连接是由什么应用层协议建立的？
- (4) TCP 连接的状态是什么

〔回答〕

(1) 源端口为 1-2 字节，即 0x0D28，十进制端口号为 3368；目的端口为 3-4 字节，即 0x0015，十进制端口号为 21

(2) 序列号为 5-8 字节，即 0x00000006，seq = 6；确认号为 9-12 字节，即 0x00000000，ack = 0

(3) 目的端口号为 21，上层应用为 FTP，文件传输协议

(4) 标志位所处 14 字节，即 0x02，二进制表达 0000 0010，所以 ACK = 0，SYN = 1，FIN = 0。处于建立 TCP 的客户机与服务器的第一次握手阶段。

5.1.2 补充 2-TCP 吞吐量

〔题目〕

已知通信信道带宽为 1Gbps，端到端的时延为 10ms，TCP 发送窗口为 65535Byte。求该 TCP 连接可能达到的最大的吞吐量，以及信道利用率。

〔回答〕

(1) 往返时延

$$RTT = 2 * 10 * 10^{-3} = 20 * 10^{-3}(\text{s})$$

(2) 最大吞吐量 (Throughput, Th) 的达成条件为，在单个 RTT 内完成发送窗口内的数据传输

$$Th = \frac{win_size}{RTT} = \frac{65535 * 8}{20 * 10^{-3}} = 26.21 * 10^6 \text{ (bps)}$$

(3) 信道利用率 (Link Utilization Ratio, U) 为实际吞吐量与信道带宽的比值

$$U = \frac{Th}{BW} * 100\% = \frac{26.214}{1000} = 2.62\%$$

5.1.3 补充 3-TCP 序列号

〔题目〕

主机 A 和主机 B 建立了 TCP 连接。主机 A 向主机 B 连续发送 3 个 TCP 报文段，长度分别为 100B、200B、300B，第一个报文段的序列号为 201，主机 B 在正确接收 3 个报文段之后，发给主机 A 的确认序列号应该为多少？

〔回答〕

根据序号和长度，可以推知三个报文段的序号分别为：

第一个报文段：201 300；

第二个报文段：301 500；

第三个报文段：501 800；

主机 B 发送的确认序列号是其期待接收的下一个报文段的第一个字节的序号，即 801。

5.1.4 [1] 5-9

〔题目〕

假设你受雇设计一个使用滑动窗口的可靠的字节流协议 (如像 TCP)。这个协议将运行在 100Mbps 的网络上。网络的 RTT 是 100ms，而且数据段的最大生存期是 60 秒。

- (a) 应该在协议首部的 AdvertisedWindow 字段和 SequenceNum 字段中包含多少比特？
 (b) 怎样确定上述数值，哪个值可能不太确定？

〔回答〕

(a) TCP 是一个字节流协议，它的缺省通告窗口 AdvertisedWindow 字段为 16 (bit)，有效的大小为 $2^{16} = 64 * 2^{10}$ (byte) = 64 (KB)，它的序号 SequenceNum 字段为 32 (bit)，是 AdvertisedWindow 大小的两倍。

接收方通告窗口字段应该具有记录发送方的发送窗口中字节数的能力，我们由此可以估测 AdvertisedWindow 字段的长度。发送方的发送窗口的设置需要考虑两点，一是接收方的缓存大小，二是链路的带宽时延积。题目没有提到接收方缓存大小的限制，因此我们假设接收方拥有足够大的缓存作为接收窗口，只考虑保持 100(Mbps)*100(ms) 的链路管道满载的需求来估算发送窗口的最大值。

链路的带宽时延积为 $100(Mbps) * 100(ms) = 10^7$ (bit) = $2^{-3} * 10^7$ (byte)。将 AdvertisedWindow 的字段长度记为 k (bit)，接收窗口可以描述的字节数需要大于链路的带宽时延积的字节数，即：

$$2^k \geq 2^{-3} * 10^7$$

解之， $k \geq \log_2(10^7) - 3 = 20.25$ 。因此，AdvertisedWindow 字段至少需要 $k = 21$ (bit)。

SequenceNum 的设置需要考虑两点，一是必须是 AdvertisedWindow 大小的两倍以保证不出现序号重复；二是支持数据段的最大生存期，即可以描述数据量 $100(Mbps) * 60(s) = 6 * 10^9$ (bit) = $0.75 * 10^9$ (byte)。

将 SequenceNum 的大小记为 s bit，需要满足：

$$(1) s \geq k + 1;$$

$$(2) 2^s \geq 0.75 * 10^9$$

解之， $s \geq 22$ 且 $s \geq 29.48$ 。因此，SequenceNum 字段至少需要 $s = 30$ (bit)。

(b) 上述计算中带宽时延积是对网络传输通道容量的刻画，其中带宽是固定的，但是 RTT 会随着传输路径和传输负载的改变而改变，因此其可能不太确定。另外，最大生存时间 MSL(Maximum Segment Life) 是一个用户自定义的变量，反映了用户对端对端的网络时延、网络复杂性等因素的估计，也是可能不太确定的。

5.2 TCP 拥塞控制

5.2.1 [2] 7-8

〔题目〕

假设 TCP 拥塞控制的 AIMD 算法中，慢开始 SST1 的阈值设置为 8，当拥塞窗口上升到 12 时，发送端检测出超时，TCP 使用慢启动与拥塞避免。试给出第 1 次到第 15 次传输的拥塞窗口分别为多少？

〔回答〕

TCP 的慢启动和拥塞避免的详细讨论，参见吴功宜教材 [2] 的 P310-313 页的案例与讨论。

(1) 慢启动阶段。当 TCP 连接初始化时，将拥塞窗口大小 (CongestionWindow, cwnd) 设置为 1。慢启动的阈值 SST1 (Slow-start Threshold, SST) 设置为 8。在慢启动阶段，当 cwnd 经过 3 个往返传输之后，按照指数算法已经增长到 8 时，进入“拥塞避免”控制阶段。这个过程经过了 3 个往返的过程。

(2) 拥塞避免阶段。进入拥塞避免阶段之后，cwnd 按照线性的方法增长，cwnd 值从 8 上升到 12，经过了 4 个往返过程。

(3) 超时后窗口调整。cwnd=12 时，发送端检测出现超时，认为网络处于拥塞状态，第 8 个往返时拥塞窗口 cwnd 重新回到 1。

(4) 重新进入慢启动与拥塞控制。当出现一次网络拥塞之后的慢开始阈值 SST2 设置是出现超时的 cwnd 最大值的 1/2，即 $SST2=12/2=6$ ，然后重新开始慢启动与拥塞避免的过程。

(5) 慢启动。第 8 个往返的拥塞窗口取 $cwnd=2^0=1$ ；按照指数增长，第 9 个往返的 $cwnd=2^1=2$ ；第 10 个往返 $cwnd=2^2=4$ ；第 11 个往返 cwnd 不能够超过 $SST2=6$ ， $cwnd=6$ 。

(6) 加性增加。第 12 个往返的 cwnd 在 6 的基础上加 1， $cwnd=7$ ；以此类推，第 13、14、15 个往返的 cwnd 分别应该等于 8、9、10。

按照上述分析和计算的 TCP 拥塞控制过程如图 5.1 所示。

第 1 个往返到第 15 个往返的 cwnd 值分别为 2、4、8、9、10、11、12、1、2、4、6、7、8、9、10。

5.2.2 [2] 7-9

〔题目〕

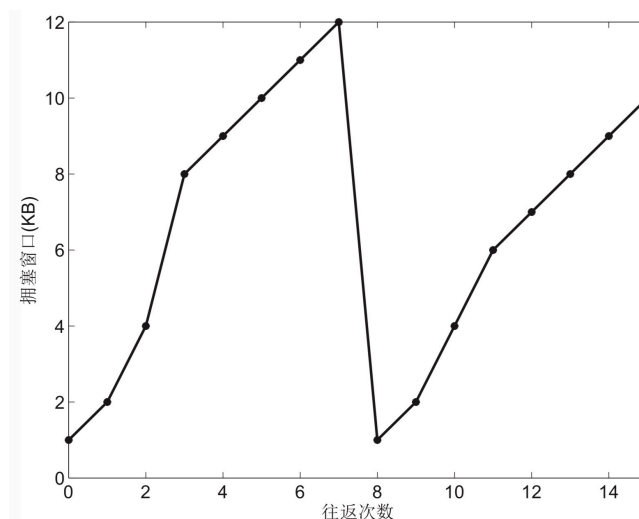


图 5.1: GBN 工作示意图

一个 TCP 连接采用慢开始算法进行拥塞控制。它的最大段长度为 1KB，发送端一直有数据要发送。在拥塞窗口为 16KB 时发生了超时，如果接下来的 4 个 RTT 时间内的 TCP 端传输都成功，那么当第 4 个 RTT 时间内发送的所有 TCP 段都得到肯定的应答。请问：拥塞窗口是多少？

〔回答〕

(1) $MSS=1KB$ ，在拥塞窗口 $cwnd=16KB$ 发生了超时后，拥塞窗口 $cwnd$ 重新回到 1KB，慢开始阈值被设置为 $ssthresh = 16 \div 2 = 8KB$ ，然后重新开始慢启动。

(2) $cwnd < ssthresh$ 时，使用慢开始算法，在第 1 个 RTT 内，TCP 端传输成功，拥塞窗口 $cwnd$ 加倍变为 2KB；在第 2 个 RTT 内，TCP 端传输成功，拥塞窗口 $cwnd$ 加倍变为 4KB；在第 3 个 RTT 内，TCP 端传输成功，拥塞窗口 $cwnd$ 加倍变为 8KB。

(3) 由于 $cwnd = ssthresh = 8KB$ ，所以在第 4 个 RTT 内，TCP 端传输成功后，根据拥塞控制算法，拥塞窗口 $cwnd$ 仅加 1KB 变为 9KB。

故在 4 个 RTT 后拥塞窗口为 9KB。

5.2.3 [1] 6-16

〔题目〕假设 TCP 实现一个扩展，允许窗口大小远大于 64KB。假定你用这一扩展 TCP 在一条延迟为 100 毫秒（往返时延）的 1Gbps 链路上传送一个 10MB 的文件，而且 TCP 接收窗口为 1MB。如果 TCP 发送 1KB 的分组（假设无拥塞，无丢失分组）：

- 当慢启动打开发送窗口达到 1MB 时用了多少 RTT？
- 发送该文件用了多少 RTT？

(c) 如果发送文件的时间由所需的 RTT 的数量与链路延迟的乘积给出, 传输的有效吞吐量是多少? 链路带宽的利用率是多少?

〔回答〕

在 TCP 报文头部的原始设计中, AdvertisedWindow 的长度为 16bit, 即 $2^{16} = 2^6 * 2^{10} = 64 KB$ 。因此, 原始 TCP 的最大有效发送窗口大小为 64KB。题目的第一句话松弛了这一约束。

(a) 这个 TCP 会话通过链路的时延带宽积是 TCP 发送窗口的物理极限值 w_{max} , 即

$$w_{max} = 1(Gbps) * 100(ms) = 10^9 * 100 * 10^{-3} (bit) = 1 * 10^8 (bit) = 11.92 (MB)$$

本小题中, 1MB 的发送窗口和接收窗口并没有超过链路的带宽时延积, 是可以达到的。

TCP 有效发送窗口 w_{send} 大小为流量控制的接收窗口 (w_{ad} , 即协议中的 Advertised-Window) 和拥塞控制的拥塞窗口 (w_{cwnd} , 即 CongestionWindow) 的最小值, 即 $w_{send} = \min(w_{ad}, w_{cwnd})$ 。在不考虑拥塞的情况下, 初始情况下影响 TCP 发送窗口的主要因素是 w_{cwnd} 的增长。

在慢启动阶段, w_{cwnd} 的原始大小是 1 倍的数据段 (segment) 大小, 即 1 KB。成功传输一次 RTT 之后, 窗口变为原来的 2 倍, 成功传输 i 个 RTT 之后, $w_{cwnd} = 1 * 2^i KB$

根据题意, 设 x 个 RTT 后窗口达到 1(MB), 则 $2^x = 1024(KB)$ 。因此 $x = \log_2(1024) = 10$ 。这时, TCP 已经完成了 10 次 RTT 的传输, TCP 发送窗口达到 1MB, 并在即将进行的第 11 次发送生效。

(b) TCP 有效发送窗口 $w_{send} = \min(w_{ad}, w_{cwnd})$ 。当发送窗口超过 1MB 之后, $w_{cwnd} \geq w_{ad}$, 控制 TCP 流速的主要参数为 w_{ad} 。

因此该传输过程可以分为两个阶段: 第一个阶段是拥塞控制的慢启动机制起主要作用, w_{send} 随着 w_{cwnd} 而 2 倍增长; 当 w_{send} 达到 w_{ad} 限制后, 即进入第二阶段, 该阶段是流量控制机制起作用, 此时 w_{send} 保持不变。

第一阶段: 达到 1(MB) 的发送窗口需要 10RTT 时间, 总共传输的数据有 $Y = \sum_{i=1}^{10} (2^{i-1}) * 1(KB) = (1 + 2 + 4 + \dots + 2^9) * 1(KB) = (2^{10} - 1) * 1(KB) = 1MB - 1KB$ 。

第二阶段: 受限接收方的缓存 $w_{ad} = 1 (MB)$, 后续的数据发送过程中, 每个 RTT 只能传输 1(MB)。待发送数据大小为 $10MB - (1MB - 1KB) = 9MB + 1KB$, 则还需要 10 个 RTT 传输剩余的数据。所以总共需要 10+10=20RTT 完成文件的传输。

(c) 在延续 (b) 条件下:

该过程中的吞吐量为 $\frac{10(MB)}{20*100(ms)} = 10 * 2^{20} * 8/2(bps) = 41.94(Mbps)$,

则链路带宽的利用率为 $41.94Mbps/1Gbps = 4.19\%$ 。

另：如果忽略超时，考虑发送窗口主要由带宽时延积给出，取消流量控制，即释放了接收窗口大小为 1MB 的约束。由于 $w_{max} > 10 MB$ ，因此该文件可以在慢启动阶段完成传输。

假设传输文件需要 x 个 RTT，则有

$$(1 + 2 + 4 + \dots + 2^{x-1}) * 1(KB) \geq 10(MB)$$

$$2^x - 1 \geq 10240$$

解之， $x \geq \log_2(10241) = 13.32$ ，因此， $x = 14$ ，即传输这个文件将花费 14 个 RTT。

该过程中的吞吐量为 $\frac{10(MB)}{14*100(ms)} = 10 * 2^{20} * 8/1.4(bps) = 59.92(Mbps)$ ，则链路带宽的利用率为 $59.92MB/1Gbps = 5.99\%$ 。

参考文献

- [1] Larry L. Peterson, Bruce S. Davie [著], 王勇, 张龙飞等 [译] 计算机网络: 系统方法 (第五版). Morgan Kaufmann, 机械工业出版社, 2015.
- [2] 吴功宜. 计算机网络 (第三版). 清华大学出版社, 2011.
- [3] 谢希仁. 计算机网络 (第六版). 电子工业出版社, 2014.