# 计算机网络阶段测试

老师:	朱敏	日期:	2022.11.9
姓名:		学号:	

- I. 首字母缩略词与描述匹配,每个缩略词仅使用一次: Congestion Control, CNAME, FIN, Flow Control, IEEE, RFC, RST, TCP, TLD, UDP。(10 分, 每题 2 分)
  - 1) 最广泛使用的可靠传输协议。
  - 2) 用于突然终止 TCP 连接的控制标志。
  - 3) 发送方如何避免 TCP 连接中的接收方过载。
- 4) 拥塞控制是一种用来调整传输控制协议(TCP)连接单次发送的分组数量算法。

# **答案:** 1) TCP 2) RST 3) Flow Control

4) Congestion Control

### 解析:

1) TCP: 传输控制协议(Transmission Control Protocol, TCP)是一种面向连接的、**可靠的**传输层协议。

干扰项: UDP, 即用户数据报协议(User Datagram Protocol, UDP), 是一种无连接、**不可靠的**传输层协议。

2) RST,即 TCP 协议字段中的复位位 (Reset)。RST=1时,表明 TCP 连接中出现严重差错(如主机崩溃或其他原因),必须**立即释放连接**,然后再重新建立运输连接。

干扰项: FIN,即 TCP 协议字段中的终止位(Finish)。用来释放一个连接。FIN=1 表明此报文段的发送方的数据已发送完毕,并要求释放传输连接。

3) Flow Control,即流量控制。TCP 提供流量控制服务来消除发送方使接收方缓存区溢出的可能性,因此可以说流量控制是一个速度匹配服务(匹配发送方的发送速率与接收方的读取速率)。

干扰项: Congestion Control,即拥塞控制,指防止过多的数据注入网络,以使网络中的路由器或链路不致过载。

- 二者区别: 拥塞控制是让网络能够承受现有的网络负荷,是一个**全局性**的过程,涉及所有的主机、所有的路由器,以及与降低网络传输性能有关的所有因素。相反,流量控制往往是指**点对点**的通信量的控制,**即接收端控制发送端**,它所要做的是抑制发送端发送数据的速率,以便使接收端来得及接收。
- 4) Congestion Control,即拥塞控制,指防止过多的数据注入网络,以使网络中的路由器或链路不致过载。
- II. 单项选择题。(30 分, 每题 3 分)

(1) 假设 OSI 参考模型的应用层欲发送 400B 的数据(无拆分),除物理层和应用层外,其他各层在封装 PDU 时均引入 20B 的额外开销,则应用层的数据传输效率约为 (A)。

A. 80% B. 83% C. 87% D. 91%

解析: OSI 参考模型共 7 层,除去物理层和应用层,剩五层。它们会向 PDU 引入  $20B \times 5 = 100B$  的额外开销。应用层是最项层,因此其数据传输效率为 400B/500B = 80%,选 A。

(2) 数据链路层采用了后退 N 帧(GBN)协议,发送方已经发送了编号为 0~7 的帧。计时器超时时,若发送方只收到 0~2~3~号帧的确认,则发送方需要重发的帧数是 ( C )。

A. 2 B. 3 C. 4 D. 5

解析:后退 N 帧协议采用累计确认,故收到 3 的确认帧就表示编号为 0、1、2、3 的帧已接收,因此需要重传的帧为编号是 4、5、6、7 的帧,重传帧的数目为 4。

- (3) 主机甲向主机乙发送一个(SYN = 1, seq = 11220) 的 TCP 段,期望与主机乙建立 TCP 连接,若主机乙接受该连接请求,则主机乙向主机甲发送的正确的 TCP 段可能是 (C)。
  - A. (SYN=0, ACK=0, seq=11221, ack=11221)
  - B. (SYN=1, ACK=1, seq=11220, ack=11220)
  - C. (SYN=1, ACK=1, seq=11221, ack=11221)
  - D. (SYN=0, ACK=0, seq=11220, ack=11220)

解析:在确认报文段中,同步位 SYN 和确认位 ACK 都是 1。返回的确认号 seq 是甲发送的初始序号 seq = 11220 加 1,即 ack = 11221;同时乙也要选择并消耗一个初始序号 seq,seq 值由乙的 TCP 进程任意给出,它与确认号、请求报文段的序号没有任何关系。

(4) 一个 TCP 连接总以 1KB 的最大段长发送 TCP 段,发送方有足够多的数据要发送,当拥塞窗口为 16KB 时发生了超时,如果接下来的 4 个 RTT 时间内的 TCP 段的传输都是成功的,那么当第 4 个 RTT 时间内发送的所有 TCP 段都得到肯定应答时,拥塞窗口大小是(C)。

A. 7KB B. 8KB C. 9KB D. 16KB

解析:发生超时后,慢开始门限 ssthresh 变为 16KB/2 = 8KB,拥塞窗口变为 1KB。在发生超时后的  $4 \uparrow RTT$  内,执行慢开始算法,拥塞窗口大小依次为 1KB、 2KB、 4KB、 8KB。

注意到题目描述的是"当第 4 个 RTT 时间内发送的所有 TCP 段都得到肯定应答",而**第 4 个 RTT 发送的 TCP 段得到应答需经过 1 个 RTT**,因此题目实际上询问的是第 5 个 RTT 时的拥塞窗口大小。

由于慢开始门限 ssthresh 为 8KB, 故之后转而执行拥塞避免算法,即拥塞窗口开始"加法增大"。因此第 5 个 RTT 时,拥塞窗口的大小为 9KB。

(5) 主机甲和乙建立了 TCP 连接,甲始终以 MSS=1KB 大小的段发送数据,并一直有数据发送; 乙每收到一个数据段都会发出一个接收窗口为 10KB 的确认段。若甲在 t 时刻发生超时的时候拥塞窗口为 8KB,则从 t 时刻起,不再发生超

时的情况下,经过10个RTT后,甲的发送窗口是(A)。

A. 10KB

B. 12KB

C. 14KB

D. 15KB

解析: 当 t 时刻发生超时时,把 ssthresh 设为 8 的一半,即 4,把拥塞窗口设为 1KB。然后经历 10 个 RTT 后,拥塞窗口的大小依次为 2、4、5、6、7、8、9、10、11、12,然后发送窗口取当时的拥塞窗口和接收窗口的最小值,接收窗口初始为 10KB,所以此时的发送窗口为 10KB,选 A。

实际上该题接收窗口一直为 10KB,可知不管何时,发送窗口一定小于等于 10KB,选项中只有 A 选项满足条件,可直接得出选 A。

## **III. 判断正误 (T or F)。**(10 分, 每题 2 分)

1) 假设主机 A 希望通过 TCP 向主机 B 发送数据,而主机 B 希望通过 TCP 向主机 A 发送数据,则需要两个独立的 TCP 连接(每个方向各一个)。( $\mathbf{F}$ )

解析:仅需建立一个TCP连接。

2) 在 TCP 中, 发送方窗口的大小可以与接收方窗口的大小相同。(T)

解析:发送窗口上限值 = min{接收窗口大小,拥塞窗口大小},因此发送方窗口的大小可以与接收方窗口的大小相同。

- 3) 字节流、全双工、可靠、支持广播都是 TCP 服务的特点。(F) 解析: TCP 并不支持广播。
- 4) A 和 B 建立了 TCP 连接, 当 A 收到确认号为 100 的确认报文段时,表示 末字节序号为 99 的报文段已收到。( )

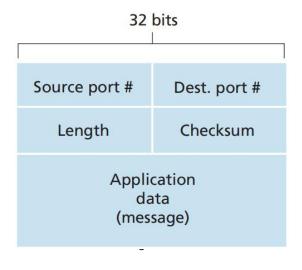
解析: TCP 的确认号是指明接收方下一次希望收到的报文段的数据部分第一个字节的编号,可以看出,前一个已收到的报文段的最后一个字节的编号为 99, 所以此题是正确的。

# IV. 综合题。(36 分)

1. UDP 报头的信息(十六进制)如表 1 所示。UDP 数据报的格式如图 2 所示。请回答以下问题: (8 分)

表 1: UDP 报头的信息(十六进制)

Number	1	2	3	4	5	6	7	8
Data	F7	21	00	45	00	2C	E8	27



#### 图 2: UDP 数据报的格式

- 1) 源端口、目的端口分别是什么? (4分)
- 2) 数据报总长度、数据部分长度分别是什么?(4分)

## 答案:

- 1)源端口: F7 21,转换成十进制数为 63265;目的端口: 00 45,转换成十进制数为 69。
  - 2) 数据报总长度: 44B; 数据部分长度: 36B。

#### 解析:

- 1)第1、2个字节为源端口,即 F721,转换成十进制数为 63265。第3、4个字节为目的端口,即0045,转换成十进制数为69。
- 2) 第 5、6 个字节为 UDP 长度(包含首部和数据部分)即 00 2C, 转换成十进制数为 44。数据报总长度为 44B,减去 UDP 固定首部长度(8B)即为数据部分长度,即 44-8 = 36B。
- 2. 主机 A 基于 TCP 向主机 B 连续发送 3 个 TCP 报文段。第 1 个报文段的序号为 90, 第 2 个报文段的序号为 120, 第 3 个报文段的序号为 150。(8 分)
  - 1) 第1、2个报文段中有多少数据? (4分)
- 2) 假设第 2 个报文段丢失而其他两个报文段到达主机 B, 在主机 B 发往主机 A 的确认报文中,确认号应是多少? (4分)

#### 筌室.

- 1) 第1个报文段: 30B; 第2个报文段: 30B。
- 2) 确认号: 120。

## 解析:

- 1) TCP 传送的数据流中的每个字节都有一个编号,而 TCP 报文段的序号为 其数据部分第一个字节的编号。因此第1个报文中的数据有120-90 = 30B,第2 个报文中的数据有150-120 = 30B。
- 2) **TCP 使用累计确认**,因此当第 2 个报文段丢失后,第 3 个报文段就成了失序报文,B 期望收到的下一个报文段是第 2 个报文段,所以确认号为 120。
- 3. 假设 TCP Reno 是一个经历如图 3 所示行为的协议,回答下列问题。(20分)

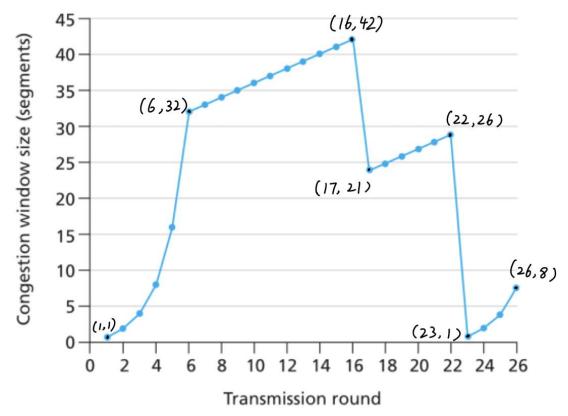


图 3: TCP 窗口大小与时间的关系

- 1) 在第 1 个传输轮回里, ssthresh 的初始值设置为多少? (2 分)
- 2) 在第 18 个传输轮回里, ssthresh 的值设置为多少? (2 分)
- 3) 在第24个传输轮回里, ssthresh 的值设置为多少? (2分)
- 4) 在哪个传输轮回内发送第70个报文段?(2分)
- 5) 假定在第 26 个传输轮回后,通过收到 3 个冗余 ACK 检测出有分组丢失, 拥塞窗口长度和 ssthresh 的值应当是多少? (4 分)
- 6) 假定使用 TCP Tahoe (而不是 TCP Reno),并假定在第 16 个传输轮回收到 3 个冗余 ACK。在第 19 个传输轮回, ssthresh 和拥塞窗口长度是什么?如果在第 22 个传输轮回有一个超时事件,从第 17 个传输轮回到第 22 个传输轮回,一共发送了多少分组?(8 分)

## 答案:

- 1) 32.
- 2) 21.
- 3) 13.
- 4) 第7个传输轮回。
- 5) 拥塞窗口长度为 4(或拥塞窗口长度为 7), ssthresh=4。
- 6) 拥塞窗口长度为 4, ssthresh=21; 共发送 52 个分组。

#### 解析:

1) ssthresh 最初是 32。因为在此窗口大小下,慢启动算法停止并开始执行拥塞避免算法。

- 2) 当检测到分组丢失时,将 ssthresh 设置为拥塞窗口值的一半。当在传输回 合 16 中检测丢失时,拥塞窗口大小为 42。因此,在第 18 个传输回合中, ssthresh 为 21。
- 3) 当检测到分组丢失时,将 ssthresh 设置为拥塞窗口值的一半。在传输回合 22 中检测到丢失时,拥塞窗口大小为 26。因此,在第 24 传输回合中, ssthresh 为 13。
- 4) 在第一轮发送中,发送分组 1; 在第二轮发送中,发送分组 2~3; 在第三轮发送中,发送分组 4~7; 在第四轮发送中,发送分组 8~15; 在第五轮发送中,发送分组 16~31; 在第六轮发送中,发送分组 32~63。分组 64~96 在第七轮发送中发送。因此,分组 70 在第七个传输轮回中被发送。
- 5) 当发生丢失时, ssthresh 将被设置为拥塞窗口 8 的一半,并且拥塞窗口将被设置为新的 ssthresh 或设置为新的 ssthresh+3(详见计算机网络[第 7 版][谢希仁]P235)。因此, ssthresh 和拥塞窗口分别为 4 和 4(或 7)。
- 6)使用 TCP Tahoe 意味着发生拥塞时将始终采用拥塞避免算法,即 ssthresh 减半,并将拥塞窗口重置为 1。在第 16 个传输轮回检测到拥塞, ssthresh 设置为 21。第 17 个传输轮回时,拥塞窗口为 1。此后执行慢开始算法,拥塞窗口依次变为 2、4。因此,在第 19 个传输轮回 ssthresh 为 21,拥塞窗口大小为 4。

第 17 轮回, 1 个分组; 第 18 轮回, 2 个分组; 第 19 轮回, 4 个分组; 第 20 轮回, 8 个分组; 第 21 轮回, 16 个分组; 第 22 轮回,继续执行慢开始算法拥塞窗口大小将超过 ssthresh(32>21),故**此时停止慢开始算法而改用拥塞避免算法**,传送 21 个分组。因此,从第 17 个传输轮回到第 22 个传输轮回,一共发送了 52 个分组。