

题目1

1. 什么是HOL阻塞？它出现在输入端口还是输出端口？

- HOL (Head-of-Line) 阻塞是指当一个交换机输入端口队列的队头分组因为其要去的输出端口正忙而被阻塞时，它会阻塞后面所有排在它后面的分组，即使那些分组要去的输出端口是空闲的。这种现象通常发生在**输入端口**。

2. 关于FIFO、优先权、循环 (RR) 和加权公平排队 (WFQ) 分组调度规则。这些排队规则中，哪个规则确保所有分组是以到达的次序离开的？

- **FIFO (先进先出)** 规则确保所有分组以它们到达的次序离开。

3. RR和WFQ分组调度之间的基本差异是什么？存在RR和WFQ将表现得完全相同的场合吗？

- **基本差异：**
 - RR (循环调度) 为每个队列轮流提供服务，每个队列获得相等的时间片，它不考虑分组的大小。
 - WFQ (加权公平排队) 是一种更为智能的调度方式。它会为不同的流量类别分配不同的权重，并根据权重来共享链路带宽，同时它也考虑分组的大小，确保即使在高负载下，每个流也能得到公平的带宽份额。
- **表现相同的场合：**
 - 是的，存在这样的场合。如果所有分组的大小都相同，并且在WFQ中为所有队列分配了完全相同的权重，那么WFQ的行为将和RR非常相似。

题目2

1. 指出TCP慢启动运行时的时间间隔。

- 根据图示，慢启动发生在传输轮回 **1-6** 和 **23-26** 期间，此时拥塞窗口 (cwnd) 呈指数增长。

2. 指出TCP拥塞避免运行时的时间间隔。

- 根据图示，拥塞避免发生在传输轮回 **6-16** 和 **17-22** 期间，此时拥塞窗口呈线性增长。

3. 在第16个传输轮回之后，报文段的丢失是根据3个冗余ACK还是根据超时检测出的？

- 是根据 **3个冗余ACK** 检测出的。因为事件发生后，拥塞窗口减半（从约42降至21），这是TCP Reno在快速恢复中的典型行为。如果是超时，窗口会降为1。

4. 在第22个传输轮回之后，报文段的丢失是根据3个冗余ACK还是根据超时检测出的？

- 是根据 **超时** 检测出的。因为事件发生后，拥塞窗口直接降到了1，这表明TCP进入了慢启动阶段。

5. 在第1个传输轮回里，ssthresh的初始值设置为多少？

- 通常初始 ssthresh 值会设置得非常大（例如65535字节），以确保TCP连接开始时总是处于慢启动阶段。从图中无法直接读出，但它远大于初始窗口大小。

6. 在第18个传输轮回里，ssthresh的值设置为多少？

- 在第16轮发生丢包时，拥塞窗口大小约为42。ssthresh 被设置为当时窗口大小的一半，即 $42 / 2 = 21$ 。所以在第18个传输轮回里，ssthresh 的值是 **21**。

7. 在第24个传输轮回里，ssthresh的值设置为多少？

- 在第22轮发生超时丢包时，拥塞窗口大小约为28。ssthresh 被设置为当时窗口大小的一半，即 $28 / 2 = 14$ 。所以在第24个传输轮回里，ssthresh 的值是 **14**。

8. 在哪个传输轮回内发送第70个报文段？

- 我们来累加每个轮回发送的报文段数：
 - 第1-6轮（慢启动）： $1 + 2 + 4 + 8 + 16 + 21 = 52$
 - 第7轮：发送22个，累计 $52 + 22 = 74$
- 因此，第70个报文段是在**第7个传输轮回**内发送的。
- 9. 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，拥塞窗口的长度和ssthresh的值应当是多少？
 - 在第26轮结束时，拥塞窗口大小约为12。如果此时因3个冗余ACK检测到丢包，ssthresh会设置为当前窗口的一半： $12 / 2 = 6$ 。拥塞窗口也会设置为新的ssthresh值，即 **ssthresh = 6，拥塞窗口 = 6**。
- 10. 假定使用TCP Tahoe（而不是TCP Reno），并假定在第16个传输轮回检测到3个冗余ACK。在第19个传输轮回，ssthresh和拥塞窗口长度是什么？
 - TCP Tahoe会将3个冗余ACK视为与超时一样的丢包事件。
 - 在第16轮丢包时，ssthresh 设置为 $42 / 2 = 21$ 。拥塞窗口立即降为1，并重新开始慢启动。
 - 第17轮：cwnd = 1
 - 第18轮：cwnd = 2
 - 第19轮：cwnd = 4
 - 因此，在第19个传输轮回，**ssthresh = 21，拥塞窗口 = 4**。
- 11. 再次假设使用TCP Tahoe，在第22个传输轮回有一个超时事件。从第17个传输轮回回到第22个传输轮回（包括这两个传输轮回），一共发送了多少分组？
 - 在第16轮丢包后，Tahoe将 ssthresh 设为21，cwnd 设为1。
 - 第17轮：cwnd=1
 - 第18轮：cwnd=2
 - 第19轮：cwnd=4
 - 第20轮：cwnd=8
 - 第21轮：cwnd=16
 - 第22轮：cwnd=21 (达到ssthresh，转为线性增长)
 - 总发送分组数 = $1 + 2 + 4 + 8 + 16 + 21 = 52$ 个。

题目3

1. 这条TCP连接能够取得的最大窗口长度（以报文段计）是多少？
 - 带宽 = 100 Mbps = $100 * 10^6$ bps
 - MSS = 1500 字节 = $1500 * 8$ bits = 12000 bits
 - RTT = 150 ms = 0.15 s
 - 最大窗口长度 (W) = (带宽 * RTT) / MSS
 - $W = (100 * 10^6 * 0.15) / 12000 = 15 * 10^6 / 12000 = 1250$
 - 最大窗口长度是 **1250个报文段**。
2. 这条TCP连接的平均窗口长度（以报文段计）和平均吞吐量（以bps计）是多少？
 - 在稳定的拥塞避免阶段，窗口大小在 W/2 和 W 之间波动。
 - 平均窗口长度 $\approx (3/4) * W = 0.75 * 1250 = 937.5$ 个报文段。

- 平均吞吐量 = (平均窗口长度 * MSS) / RTT
- 平均吞吐量 = $(937.5 * 12000) / 0.15 = 75,000,000 \text{ bps} = \mathbf{75 \text{ Mbps}}$ 。

3. 这条TCP连接在丢失包恢复后，再次到达其最大窗口要经历多长时间？

- 丢包后，`ssthresh` 设为 $W/2 = 1250 / 2 = 625$ ，`cwnd` 也设为625（快速恢复）。
- 连接进入拥塞避免阶段，每个RTT，`cwnd` 增加1。
- 需要增加的窗口大小 = $1250 - 625 = 625$ 个报文段。
- 所需时间 = 需要增加的窗口大小 * RTT = $625 * 0.15 \text{ s} = \mathbf{93.75 \text{ 秒}}$ 。