

班级：2022211301

姓名：卢安来

学号：2022212720

8. In Figure 6-20, suppose a new flow E is added that takes a path from R_1 to R_2 to R_6 . How does the max-min bandwidth allocation change for the five flows?

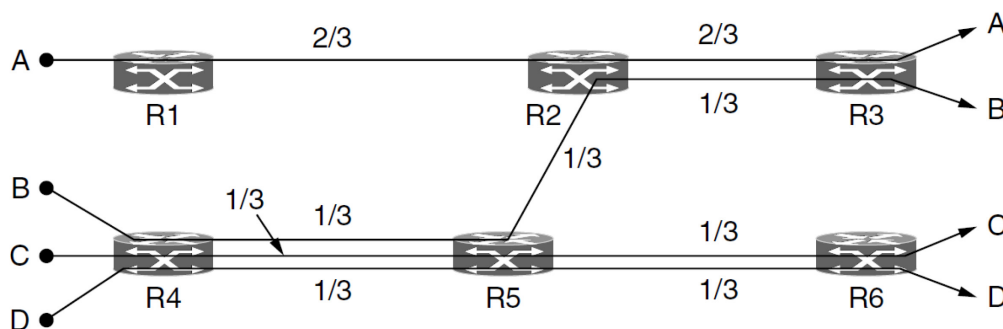


Figure 6-20. Max-min bandwidth allocation for four flows.

解答：

规划可知，流 A 与流 E 将在 $R_1 \rightarrow R_2$ 链路平分带宽，即各占 $\frac{1}{2}$ 。并且，流 A 在 $R_2 \rightarrow R_3$ 链路上占用的带宽将会调整到 $\frac{1}{2}$ ，流 E 将会占用 $\frac{1}{2}$ 的 $R_2 \rightarrow R_6$ 的带宽，流 B, C, D 的带宽分配方案不变。

10. Some other policies for fairness in congestion control are Additive Increase Additive Decrease (AIAD), Multiplicative Increase Additive Decrease (MIAD), and Multiplicative Increase Multiplicative Decrease (MIMD). Discuss these three policies in terms of convergence and stability.

解答：

AIAD 和 MIMD 策略会导致带宽震荡，不会收敛到最优点。特别地，MIAD 策略反而会远离最优点。

AIAD、MIMD 和 MIAD 这些策略都不具有良好的稳定性。在 AIAD 和 MIAD 中，拥塞时的发送速率下降不够快，不能迅速缓解拥

塞。而在 MIMD 中，发送速率的增加不够平稳，容易导致拥塞的发生。

11. Why does UDP exist? Would it not have been enough to just let user processes send raw IP packets?

解答：

UDP 的存在有它的必要性。

必要性之一在于它属于体系结构更高层次——传输层，实现了进程间的通信和复用。UDP 提供的是无连接服务，这与 IP 协议是类似的，但通过 IP 协议只能将数据传送给目的主机，无法确定应交给目的主机上的哪个进程。UDP 增加了端口号，端口号允许网络层将收到的数据包正确地分发到相应的应用程序或进程。同时，一个主机上的多个进程可以同时使用 UDP 进行通信，而不会相互干扰。

必要性之二在于它的简单、高效。UDP 提供了一种无连接的服务，这意味着在数据传输之前，UDP 不需要建立连接。这种设计选择使得 UDP 比其他协议（如 TCP）更加轻量级和高效，尤其适用于那些对实时性要求高、能够容忍一定程度数据丢失的应用场景，如视频会议和在线游戏。

18. What is the total size of the minimum TCP MTU, including TCP and IP overhead but not including data link layer overhead?

解答：

根据 RFC 9293，在默认情况下，TCP MSS 为 536 字节，故考虑 TCP header（最小 20 字节）和 IP header（最小 20 字节）后最小的 MTU 为 576 字节。

19. Datagram fragmentation and reassembly are handled by IP and are invisible to TCP. Does this mean that TCP does not have to worry about data arriving in the wrong order?

解答：

即使 IP 协议处理的数据报的分片和重组，这也并不意味着 TCP 不需要考虑数据的乱序到达，这是两个不相关的问题。

这是因为 IP 只提供尽力而为的服务，并不能保证数据报的顺序到达。在网络中，由于路由的变化，不同的数据报可能会走不同的路径，因此它们可能会以不同的顺序到达目的地。

21. A process on host 1 has been assigned port p , and a process on host 2 has been assigned port q . Is it possible for there to be two or more TCP connections between these two ports at the same time?

解答：

相同主机的两个端口之间不能同时存在多个 TCP 连接。

在 TCP/IP 协议中，一个 TCP 连接是由四个元素唯一确定的：源 IP 地址、源端口、目标 IP 地址、目标端口。因此，题中给定的 (h_1, p, h_2, q) 四元组唯一确定了一个 TCP 连接。

22. In Fig. 6-36 we saw that in addition to the 32-bit acknowledgement field, there is an ACK bit in the fourth word. Does this really add anything? Why or why not?

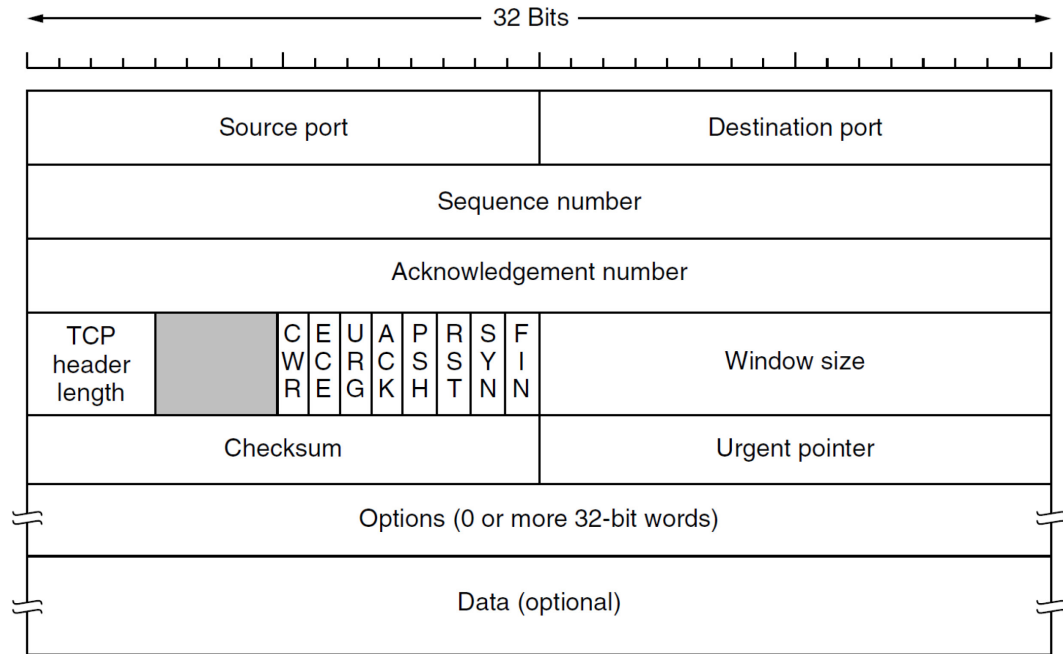


Figure 6-36. The TCP header.

解答：

ACK 标志位用于标识 header 中的确认序号是否有效。若 ACK 标志位被置为 1，则 ACK 字段应该被处理，否则应忽略。

因此，TCP 通信中的 ACK 标志位和 ACK 字段是一起工作的。ACK 标志位了 ACK 字段的有效性。没有 ACK 标志位，接收方就不知道是否应该处理 ACK 字段。

这种模式的作用在 TCP 连接建立的三次握手中表现突出，尤其是在第一次握手和第二次握手时。

25. Consider the effect of using slow start on a line with a 10 ms round-trip time and no congestion. The receive window is 24 KB and the maximum segment size is 2 KB. How long does it take before the first full window can be sent?

解答：

由于慢启动过程中，拥塞窗口初始为 1 个 MSS，每次收到 ACK

后倍增，发送窗口大小为拥塞窗口和接受窗口的最小值，可模拟过程如下。

第一次发送时，拥塞窗口为 2 KB，发送窗口为 $\min\{2 \text{ KB}, 24 \text{ KB}\} = 2 \text{ KB}$ 。收到 ACK 后拥塞窗口倍增为 4 KB。

第二次发送时，拥塞窗口为 4 KB，发送窗口为 $\min\{4 \text{ KB}, 24 \text{ KB}\} = 4 \text{ KB}$ 。收到 ACK 后拥塞窗口倍增为 8 KB。

第三次发送时，拥塞窗口为 8 KB，发送窗口为 $\min\{8 \text{ KB}, 24 \text{ KB}\} = 8 \text{ KB}$ 。收到 ACK 后拥塞窗口倍增为 16 KB。

第四次发送时，拥塞窗口为 16 KB，发送窗口为 $\min\{16 \text{ KB}, 24 \text{ KB}\} = 16 \text{ KB}$ 。收到 ACK 后拥塞窗口倍增为 32 KB。

后续几次发送时，拥塞窗口为 32 KB，发送窗口为 $\min\{32 \text{ KB}, 24 \text{ KB}\} = 24 \text{ KB}$ ，即发送满窗口。由此前共经历四次发送，故所需时间即为 $4 \times \text{RTT} = 40 \text{ ms}$ 。

26. Suppose that the TCP congestion window is set to 18 KB and a timeout occurs. How big will the window be if the next four transmission bursts are all successful? Assume that the maximum segment size is 1 KB.

解答：

发生超时，发送端将拥塞窗口调整为 MSS，设定阈值为原拥塞窗口大小的一半，即 9 KB，后开始慢启动。

发送第一个段时，拥塞窗口大小为 1 KB，传输成功，收到 ACK 后拥塞窗口倍增后未达到阈值，故拥塞窗口大小倍增，调节为 2 KB。

发送第二个段时，拥塞窗口大小为 2 KB，传输成功，收到 ACK 后拥塞窗口倍增后未达到阈值，故拥塞窗口大小倍增，调节为 4 KB。

发送第三个段时，拥塞窗口大小为 4 KB，传输成功，收到 ACK 后拥塞窗口倍增后未达到阈值，故拥塞窗口大小倍增，调节为 8 KB。

发送第四个段时，拥塞窗口大小为 8 KB，传输成功，收到 ACK 后拥塞窗口倍增后超过阈值，故拥塞窗口大小增加一个 MSS，调节

为 9 KB。

因此，四次传输成功后拥塞窗口大小为 9 KB。

27. If the TCP round-trip time, RTT, is currently 30 ms and the following acknowledgements come in after 26 , 32 , and 24 ms , respectively, what is the new RTT estimate using the Jacobson algorithm? Use $\alpha = 0.9$.

解答：

根据 Jacobson algorithm, 有

$$RTT_n = \alpha \cdot RTT_{n-1} + (1 - \alpha) \cdot R_n,$$

代入 $RTT_0 = 30$ ms, $R_1 = 26$ ms, $R_2 = 32$ ms 和 $R_3 = 24$ ms, 得

$$RTT_1 = 0.9 \times 30 \text{ ms} + 0.1 \times 26 \text{ ms} = 29.6 \text{ ms},$$

$$RTT_2 = 0.9 \times 29.6 \text{ ms} + 0.1 \times 32 \text{ ms} = 29.84 \text{ ms},$$

$$RTT_3 = 0.9 \times 29.84 \text{ ms} + 0.1 \times 24 \text{ ms} = 29.256 \text{ ms}.$$

故新的 RTT 计算为 29.256 ms。

28. A TCP machine is sending full windows of 65535 B over a 1 Gbps channel that has a 10 ms one-way delay. What is the maximum throughput achievable? What is the line efficiency?

解答：

由于发送满窗口数据需等确认后才能继续发送，故最大吞吐量

$$b_{\text{throughput}} = \frac{n}{2 \times RTT} = \frac{65535 \times 8 \text{ bit}}{2 \times 10 \text{ ms}} = 26.214 \text{ Mbps}.$$

从而信道利用率

$$\eta = \frac{b_{\text{throughput}}}{b} \times 100\% = 2.6214\%.$$

补充题. 主机甲和乙已建立了 TCP 连接，甲始终以 $MSS = 1\text{ KB}$ 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 10 KB 的确认段。若甲在 t 时刻发生超时拥塞窗口为 8 KB ，则从 t 时刻起，不再发生超时的情况下，请计算经过 10 个 RTT 后，甲的发送窗口是多少？

解答：

甲在 t 时刻超时，阈值设定为此时拥塞窗口的一半，即 4 KB ，拥塞窗口设定为 1 个 MSS ，即 1 KB 。后续过程列举如下。

经过的 RTT 数	拥塞窗口	接受窗口	发送窗口
1	2 KB	10 KB	2 KB
2	4 KB	10 KB	4 KB
3	5 KB	10 KB	5 KB
4	6 KB	10 KB	6 KB
5	7 KB	10 KB	7 KB
6	8 KB	10 KB	8 KB
7	9 KB	10 KB	9 KB
8	10 KB	10 KB	10 KB
9	11 KB	10 KB	10 KB
10	12 KB	10 KB	10 KB

故经过 10 个 RTT 后，甲的发送窗口为 10 KB 。