



发布新内容
原力计划



Beta



Beta



举报



今岁成蹊

关注

12



73



1



专栏目录

【计算机网络 (谢希仁) 习题题解】第5章 运输层

(2)——连续ARQ协议；TCP报文段

原创

今岁成蹊 于 2021-05-29 08:51:31 发布

版权

阅读量8.9k 收藏 73 点赞数 12

分类专栏: 计算机网络 文章标签: 网络 计算机网络 arq



计算机网络 专栏收录该内容

87 订阅 14 篇文章

订阅专栏

传输控制协议 TCP

TCP 最主要的特点:

- TCP 是**面向连接的运输层协议**。应用程序在使用 TCP 协议之前,必须先建立 TCP 连接。在传送数据完毕后,必须释放已经建立的 TCP 连接。
- 每一条 TCP 连接只能有两个**端点** (endpoint), 每一条 TCP 连接只能是**点对点的** (一对一)。
- TCP 提供**可靠交付**的服务。通过 TCP 连接传送的数据,无差错、不丢失、不重复,并且按序到达。
- TCP 提供**全双工通信**。TCP 允许通信双方的应用进程在任何时候都能发送数据。TCP 连接的两端都设有发送缓存和接收缓存。
- **面向字节流**。TCP 中的流 (stream) 指的是**流入到进程或从进程流出的字节序列**。
“面向字节流”的含义是:虽然应用程序和 TCP 的交互是一次一个数据块 (大小不等),但 TCP 把应用程序交下来的数据仅仅看成是一连串的**无结构的字节流**。

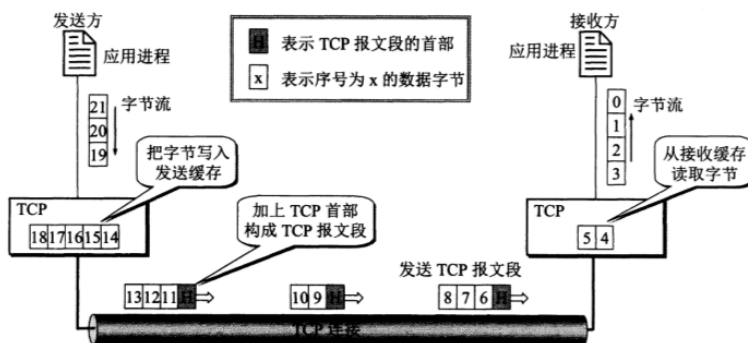


图 5-8 TCP 面向字节流的概念 https://blog.csdn.net/HPP_CSDN

TCP 的连接

TCP 把**连接**作为**最基本的抽象**。

每一条 TCP 连接有两个**端点**。TCP 连接的端点叫做**套接字** (socket) 或**插口**。
端口号**拼接到** (concatenated with) IP 地址即构成了套接字 [RFC 793]。因此,套接字的表示方法是在点分十进制的 IP 地址后面写上端口号,中间用冒号或逗号隔开。

套接字 socket = (IP 地址: 端口号)

每一条 TCP 连接唯一地被通信两端的两个端点 (即两个套接字) 所确定。即:

TCP 连接 ::= {socket₁, socket₂} = {(IP₁: port₁), (IP₂: port₂)}

TCP 连接就是由协议软件所提供的一种抽象。**TCP 连接的端点是个很抽象的套接字**。
同一个 IP 地址可以有多个不同的 TCP 连接,同一个端口号也可以出现在多个不同的



今岁成蹊

关注

12



73



1



专栏目录



Beta



Beta



举报

随着互联网的不断发展以及网络技术的进步，同一个名词 socket 可表示多种不同的意思。例如：

- 允许应用程序访问连网协议的**应用编程接口** API，即运输层和应用层之间的一种接口，称为 socket API，并简称为 socket。
- 在 socket API 中使用的一个**函数名**也叫做 socket。
- 调用 socket 函数的端点称为 socket，如“创建一个数据报 socket”。
- 调用 socket 函数时，其返回值称为 socket 描述符，可简称为 socket。
- 在操作系统内核中连网协议的 Berkeley 实现，称为 socket 实现。

上面的这些 socket 的意思都和 RFC 793 定义的 socket 不同。

可靠传输的工作原理

理想的传输条件有两个特点：

- 传输信道不产生差错。
- 不管发送方以多快的速度发送数据，接收方总是来得及处理收到的数据。

实际的网络都不具备以上两个理想条件。但可以使用一些可靠传输协议，当出现差错时让发送方重传出现差错的数据，同时在接收方来不及处理收到的数据时，及时告诉发送方适当降低发送数据的速度。

停止等待协议

“停止等待”是每发送完一个分组就停止发送，等待对方的确认。在收到确认后再发送下一个分组。

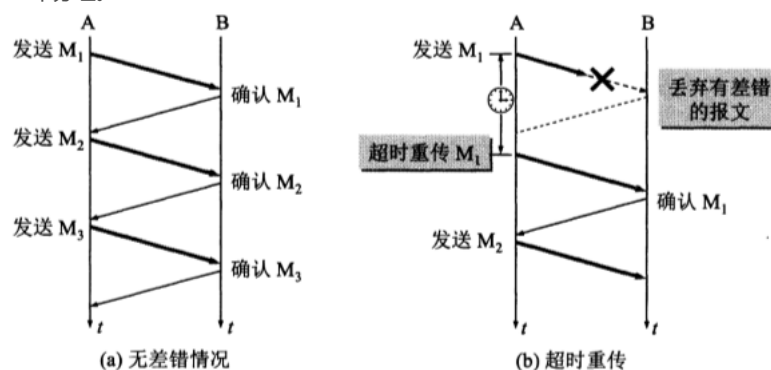


图 5-9 停止等待协议

B 接收 M₁ 时检错出了差错，就丢弃 M₁，其他什么也不做。

可靠传输协议是这样设计的：A 只要超过了一段时间仍然没有收到确认，就认为刚才发送的分组丢失了，因而重传前面发送过的分组。这叫做**超时重传**。

要实现超时重传，就要在每发送完一个分组时设置一个**超时计时器**。如果在超时计时器到期之前收到了对方的确认，就撤销已设置的超时计时器。

注意三点：

- A 在发送完一个分组后，**必须暂时保留已发送的分组的副本**。只有在收到相应的确认后才能清除暂时保留的分组副本。
- 分组和确认分组都必须进行**编号**。这样才能明确是哪一个发送出去的分组收到了确认。
- 超时计时器设置的重传时间**应当比数据在分组传输的平均往返时间更长一些**。在运输层重传时间的准确设定是非常复杂的，这是因为已发送出的分组到底会**经过哪些网络**，以及这些网络将会**产生多大的时延**（取决于网络当时的拥塞情况），这



些都是不确定因素。

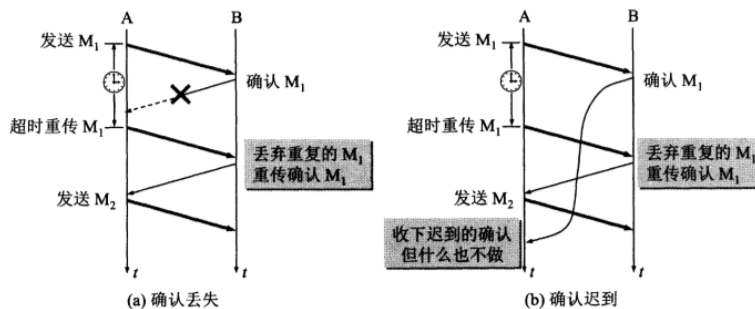


图 5-10 确认丢失和确认迟到

使用上述的确认和重传机制，就可以在不可靠的传输网络上实现可靠的通信。

像上述的这种可靠传输协议常称为**自动重传请求** ARQ (Automatic Repeat ReQuest)。意思是重传的请求是自动进行的。接收方不需要请求发送方重传某个出错的分组。

停止等待协议的优点是简单，缺点是信道利用率太低。

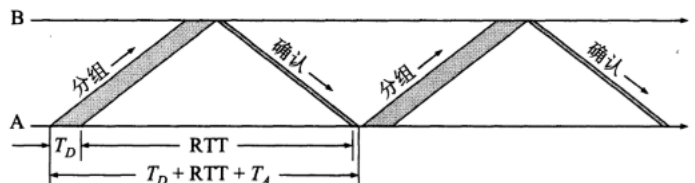


图 5-11 停止等待协议的信道利用率太低

仅仅是在时间 T_D 内才用来传送有用的数据 (包括分组的首部)，因此信道的利用率 U 为：

$$U = \frac{T_D}{T_D + RTT + T_A}$$

为了提高传输效率，发送方可以不使用低效率的停止等待协议，而采用**流水线传输**。流水线传输就是发送方可连续发送多个分组，不必每发完一个分组就停顿下来等待对方的确认。

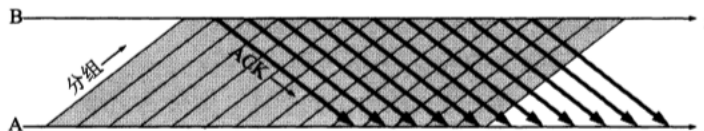


图 5-12 流水线传输可提高信道利用率

当使用流水线传输时，使用**连续 ARQ 协议**和**滑动窗口协议**。

连续 ARQ 协议



图 5-13 连续 ARQ 协议的工作原理

连续 ARQ 协议规定，发送方每收到一个确认，就把发送窗口向前滑动一个分组的位置。



今岁蹊

关注

12

73

1

专栏目录

接收方一般采用**累计确认**的方式。接收方不必对收到的分组逐个发送确认，而是在收到几个分组后，**对按序到达的最后一个分组发送确认**，这就表示：到这个分组为止的所有分组都已正确收到了。

累计确认的优点是：容易实现，即使确认丢失也不必重传。缺点是不能向发送方反映出接收方已经正确收到的所有分组的信息。

例如，如果发送方发送了前 5 个分组，中间的第 3 个分组丢失了。这时接收方只能对前两个分组发出确认。发送方无法知道后面三个分组的下落，只好把后面的三个分组都再重传一次。这叫做 Go-back-N (回退 N)，表示需要再退回来重传已发送过的 N 个分组。

TCP 报文段的首部格式

TCP 虽然是面向字节流的，但 TCP 传送的数据单元却是报文段。

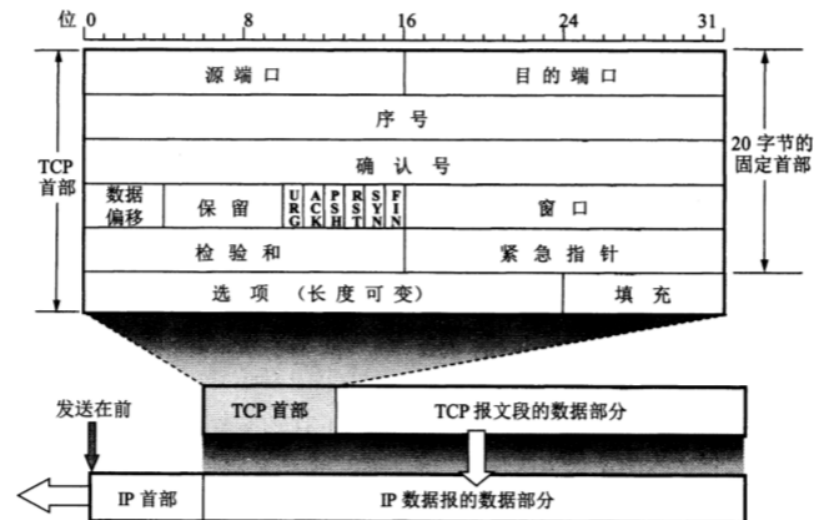


图 5-14 TCP 报文段的首部格式

- 源端口、目的端口：分别写入源端口号和目的端口号。TCP 的分用功能是通过端口实现的。
- 序号：范围是 $[0, 2^{32} - 1]$ ，使用 $\text{mod } 2^{32}$ 运算。在一个 TCP 连接中传送的字节流中的**每一个字节都按顺序编号**。整个要传送的字节流的起始序号必须在连接建立时设置。
首部中的序号字段值则指的是**本报文段**所发送的数据的第一个字节的序号。例，一报文段的序号字段值是 301，携带的数据共有 100 字节。这表明：本报文段的数据的第一个字节的序号是 301，下一个报文段 (若有的话) 的序号字段值应为 401。
这个字段的名称也叫做**报文段序号**。
- 确认号：**期望收到对方下一个报文段的第一个数据字节的序号**。
若确认号 = N，则表明：到序号 N-1 为止的所有数据都已正确收到。
- 数据偏移：指出 TCP 报文段的数据起始处距离 TCP 报文段的起始处有多远。实际上是指出 TCP 报文段的首部长度。“数据偏移”的单位是 32 位字 (即 4 字节)。
- 保留：保留为今后使用，目前应置为 0。
- 窗口：窗口值是 $[0, 2^{16} - 1]$ 之间的整数。窗口指的是发送本报文段的一方的**接收窗口** (而不是自己的发送窗口)。
窗口字段明确指出了现在允许对方发送的数据量。窗口值经常在动态变化着。之所以要有这个限制，是因为接收方的数据缓存空间是有限的。
- 检验和：检验和字段检验的范围包括首部和数据这两部分。在计算检验和时，要在 TCP 报文段的前面加上 12 字节的伪首部。
伪首部的格式与图 5-5 中 UDP 用户数据报的伪首部一样。把伪首部第 4 个字段中的 17 改为 6 (TCP 的协议号)，把第 5 字段中的 UDP 长度改为 TCP 长度。



- 选项：最长可达 40 字节。
TCP 最初只规定了一种选项，即**最大报文段长度** MSS (Maximum Segment Size) [RFC 879]。MSS 是**每一个** TCP 报文段中的**数据字段的最大长度**。在连接建立的过程中，双方都把自己能够支持的 MSS 写入这一字段，以后就按照这个数值传送数据，两个传送方向可以有不同的 MSS 值。MSS 的默认值是 536 字节长。
随着互联网的发展，又陆续增加了几个选项。如**窗口扩大**选项、**时间戳**选项等 [RFC 7323]。以后又增加了有关**选择确认** (SACK) 选项。
窗口扩大选项：为了扩大窗口。窗口扩大选项占 3 字节，其中有一个字节表示**移位值** S。新的窗口值等于 TCP 首部中的窗口位数从 16 增大到 (16 + S)。移位值允许使用的最大值是 14。窗口扩大选项可以在双方初始建立 TCP 连接时进行协商。如果连接的某一端实现了窗口扩大，当它不再需要扩大其窗口时，可发送 S = 0 的选项，使窗口大小回到 16。
时间戳选项：占 10 字节，其中最主要的字段是**时间戳值**字段 (4 字节) 和**时间戳回送回答**字段 (4 字节)。时间戳选项有两个功能。①用来计算往返时间 RTT。发送方在发送报文段时把当前时钟的时间值放入时间戳字段，接收方在确认该报文段时把时间戳字段值复制到时间戳回送回答字段。发送方在收到确认报文后，可准确计算出 RTT。②用于处理 TCP 序号超过 2^{32} 的情况，这又称为**防止序号绕回** (Protect Against Wrapped Sequence numbers)。TCP 报文段每增加 2^{32} 个序号就会重复使用原来用过的序号。当使用高速网络时，在一次 TCP 连接的数据传送中序号很可能会被重复使用。为了使接收方能够把新的报文段和迟到很久的报文段区分开，可以在报文段中加上这种时间戳。

有 6 个**控制位**，用来说明本报文段的性质：

- 紧急 URG (URGeNT)：当 URG = 1 时，表明紧急指针字段有效。它告诉系统此报文段中有紧急数据，应尽快传送 (相当于高优先级的数据)，而不要按原来的排队顺序来传送。
当 URG 置 1 时，发送应用进程就告诉发送方的 TCP 有紧急数据要传送。于是发送方 TCP 就把紧急数据插入到本报文段数据的最前面，而在紧急数据后面的数据仍是普通数据。这时要与首部中**紧急指针**字段配合使用。
- 确认 ACK (ACKnowledgement)：仅当 ACK = 1 时确认号字段才有效。当 ACK = 0 时，确认号无效。TCP 规定，在连接建立后所有传送的报文段都必须把 ACK 置 1。
- 推送 PSH (PuSH)：当两个应用进程进行交互式的通信时，有时在一端的应用进程希望在键入一个命令后立即就能够收到对方的响应。在这种情况下，TCP 就可以使用推送 (push) 操作。发送方 TCP 把 PSH 置 1，并立即创建一个报文段发送出去。接收方 TCP 收到 PSH = 1 的报文段，就尽快地 (即“推送”向前) 交付接收应用进程，而不再等到整个缓存都填满了后再向上交付。
- 复位 RST (ReSeT)：当 RST = 1 时，表明 TCP 连接中出现严重差错 (如由于主机崩溃或其他原因)，必须释放连接，然后再重新建立运输连接。RST 置 1 还用来拒绝一个非法的报文段或拒绝打开一个连接。RST 也可称为重建位或重置位。
- 同步 SYN (SYNchronization)：在连接建立时用来同步序号。当 SYN = 1 而 ACK = 0 时，表明这是一个连接请求报文段。对方若同意建立连接，则应在响应的报文段中使 SYN = 1 和 ACK = 1。因此，SYN 置为 1 表示这是一个连接请求或连接接受报文。
- 终止 FIN (FINis)：用来释放一个连接。当 FIN = 1 时，表明此报文段的发送方的数据已发送完毕，并要求释放运输连接。

5-16 在停止等待协议中如果不使用编号是否可行？为什么？

解：不行。这样才能明确是哪一个发送出去的分组收到了确认，而哪一个分组还没有收到确认。

5-17 在停止等待协议中，如果收到重复的报文段时不予理睬 (即悄悄地丢弃它而其他什么也没做) 是否可行？试举出具体的例子说明理由。



Beta



Beta



举报



今岁成蹊

关注

12



73



1

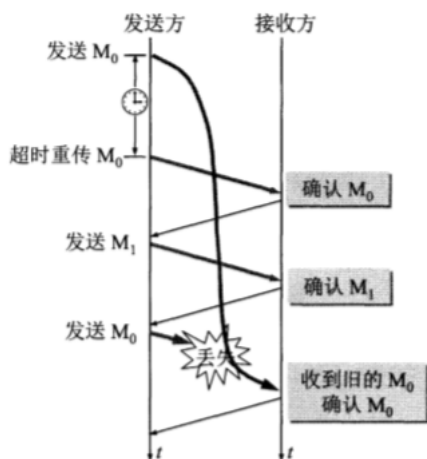


专栏目录

②向发送方发送确认。因为发送方之所以重传报文段，就表示发送方没有收到相应的确认。

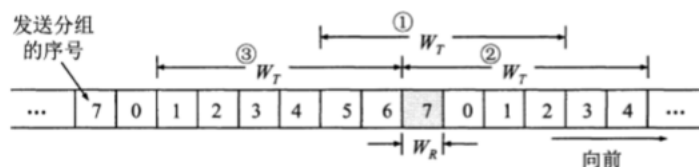
5-18 假定在运输层使用停止等待协议。发送方在发送报文段 M_0 后在设定的时间内未收到确认，于是重传 M_0 ，但 M_0 又迟迟不能到达接收方。不久，发送方收到了迟到的对 M_0 的确认，于是发送下一个报文段 M_1 ，不久就收到了对 M_1 的确认。接着发送方发送新的报文段 M_0 ，但这个新的 M_0 在传送过程中丢失了。正巧，一开始就滞留在网络中的 M_0 现在到达接收方。接收方无法分辨 M_0 是旧的。于是收下 M_0 ，并发送确认。显然，接收方后来收到的 M_0 是重复的，协议失败了。

试画出类似于图5-9所示的双方交换报文段的过程。



旧的 M_0 被当成是新的 M_0 ! CSDN

5-19 试证明：当用 n 比特进行分组的编号时，若接收窗口等于 1 (即只能按序接收分组)，则仅在发送窗口不超过 $2^n - 1$ 时，连续 ARQ 协议才能正确运行。窗口单位是分组。



设发送窗口记为 W_T ，接收窗口记为 W_R 。假定用 3 比特进行编号。设接收窗口正好在 7 号分组处 (有阴影的分组)。发送窗口 W_T 的位置不可能比②更靠前，也不可能比③更靠后，也可能不是这种极端位置，如①。

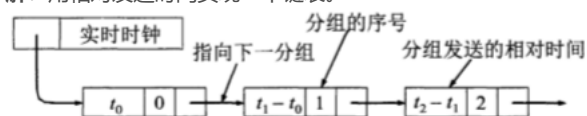
对于①和②的情况，在 W_T 的范围内无重复序号，即 $W_T \leq 2^n$ 。

对于③的情况，在 $W_T + W_R$ 的范围内无重复序号，即 $W_T + W_R \leq 2^n$ 。

现在 $W_R = 1$ ，故发送窗口的最大值 $W_T \leq 2^n - 1$ 。

5-20 在连续 ARQ 协议中，若发送窗口等于 7，则发送端在开始时可连续发送 7 个分组。因此，在每一分组发出后，都要置一个超时计时器。现在计算机里只有一个硬时钟。设这 7 个分组发出的时间分别为 $t_0, t_1 \dots t_6$ ，且 t_{out} 都一样大。试问如何实现这 7 个超时计时器 (这叫软时钟法)？

解：用相对发送时间实现一个链表。



5-21 假定使用连续 ARQ 协议中，发送窗口大小是 3，而序号范围是 $[0, 15]$ ，而传输



今岁成蹊

关注

12



73



1



专栏目录



Beta



Beta



- (1) 在发送方的发送窗口中可能出现的序号组合有哪些？
(2) 接收方已经发送出去的、但在网络中 (即还未到达发送方) 的确认分组可能有哪些？
说明这些确认分组是用来确认哪些序号的分组。

解：(1) 序号到 4 为止的分组都已收到。若这些确认都已到达发送方，则发送窗口的范围是 [5, 7]。假定所有的确认都丢失了，发送方没有收到这些确认。这时，发送窗口应为 [2, 4]。因此，发送窗口可以是 [2, 4], [3, 5], [4, 6], [5, 7] 中的任何一个。
(2) 接收方期望收到序号 5 的分组，说明序号为 2, 3, 4 的分组都已收到，并且发送了确认。对序号为 1 的分组的确认肯定被发送方收到了，否则发送方不可能发送 4 号分组。可见，对序号为 2, 3, 4 的分组的确认有可能仍滞留在网络中。这些确认是用来确认序号为 2, 3, 4 的分组。

5-22 主机 A 向主机 B 发送一个很长的文件，其长度为 L 字节。假定 TCP 使用的 MSS 为 1460 字节。

- (1) 在 TCP 的序号不重复使用的条件下，L 的最大值是多少？
(2) 假定使用上面计算出文件长度，而运输层、网络层和数据链路层所使用的首部开销共 66 字节，链路的数据率为 10Mbit/s，试求这个文件所需的最短发送时间。

解：(1) 序号字段占 32 位，范围为 $[0, 2^{32} - 1]$ ，在一个 TCP 连接中传送的字节流中的每一个字节按顺序编号，故 L 的最大值为 2^{32} ，即长度为 4 GB (4294967296 字节)。

(2) 这个文件至少需要分为 $N = L / \text{MSS} = 2941759$ 个报文段发送。

所以首部开销共有 $L_2 = 66N = 194156094$ 字节。

所以传送文件的总共发送的字节数为 $L_3 = L + L_2 = 4489123390$ 字节。

发送 L_3 字节所需时间为 $L_3 / \text{数据率} = 3591.3$ 秒，即 59.85 分，约 1 小时。

5-23 主机 A 向主机 B 连续发送了两个 TCP 报文段，其序号分别为 70 和 100。试问：

- (1) 第一个报文段携带了多少个字节的数据？
(2) 主机 B 收到第一个报文段后发回的确认中的确认号应当是多少？
(3) 如果 B 收到第二个报文段后发回的确认中的确认号是 180，试问 A 发送的第二个报文段中的数据有多少字节？
(4) 如果 A 发送的第一个报文段丢失了，但第二个报文段到达了 B。B 在第二个报文段到达后向 A 发送确认。试问这个确认号应为多少？

解：(1) 第一个报文段的数据序号是 70 到 99，共 30 字节的数据。

(2) 确认号应为 100。

(3) 第二个报文段的数据序号是 100 到 179，共 80 字节的数据。

(4) 确认号应为 70。

5-25 为什么在 TCP 首部中要把 TCP 的端口号放入最开始的 4 个字节？

解：在 ICMP 的差错报文中要包含 IP 首部后面的 8 个字节的内容，而这里面有 TCP 首部中的源端口和目的端口。当 TCP 收到 ICMP 差错报文时需要用这两个端口来确定是哪条连接出了差错。

5-26 为什么在 TCP 首部中有一个首部长度字段，而 UDP 的首部中就没有这个这个字段？

解：TCP 首部除固定长度部分外，还有选项，因此 TCP 首部长度是可变的。UDP 首部长度是固定的。

5-27 一个 TCP 报文段的数据部分最多为多少个字节？为什么？如果用户要传送的数据的字节长度超过 TCP 报文段中的序号字段可能编出的最大序号，问还能否用 TCP 来传送？

解：65495 字节。此数据部分加上 TCP 首部的 20 字节，再加上 IP 首部的 20 字节，



Beta



Beta



举报



今岁成蹊

关注

12



73



1



专栏目录

数据的字节长度超过 TCP 报文段中的序号字段可能编出的最大序号，还能用 TCP 来传送。因为序号可以循环使用。如果在一次 TCP 连接的数据传送中序号被重复使用，可以在报文段中加上时间戳选项，使接收方能够把新的报文段和迟到很久的报文段区分开。

5-28 主机 A 向主机 B 发送 TCP 报文段，首部中的源端口是 m 而目的端口是 n。当 B 向 A 发送回信时，其 TCP 报文段的首部中源端口和目的端口分别是什么？

解：分别是 n 和 m。

5-48 网络允许的最大报文段长度为 128 字节，序号用 8 位表示，报文段在网络中的寿命为 30 秒。求发送报文段的一方所能达到的最高数据率。

解：具有相同编号的报文段不应该同时在网络中传输，必须保证当序列号循环重复使用的时候，具有相同序列号的报文段已经从网络中消失。

序号用 8 位表示，序号范围是 $[0, 2^8 - 1]$ ，共 256 个序号。

报文段的寿命为 30 s，那么在 30 s 的时间内发送方发送的报文段的数目不能多于 255 个。

网络允许的最大报文段长度为 128 B，一共发送的比特数为 $255 \times 128 \times 8 \text{ bit} = 261120 \text{ bit}$ 。

每一条 TCP 连接所能达到的最高数据传输速率为 $261120 \text{ bit} / 30 \text{ s} = 8.704 \text{ kbit/s}$ 。

5-65 假定主机 A 向 B 发送一个 TCP 报文段。在这个报文段中，序号是 50，而数据一共有 6 字节长。试问，在这个报文段中的确认字段是否应当写入 56？

解：在这个报文段中的确认字段应当写入的是 A 期望下次收到 B 发送的数据中的第一个字节的编号，而这个数值是 A 已经收到的数据的最后一个字节的编号加 1。

然而这些在题目中并未给出。因此，现在无法知道这个报文段中的确认字段应当写入什么数值。

5-66 主机 A 通过 TCP 连接向 B 发送一个很长的文件，因此这需要分成很多个报文段来发送。假定某一个 TCP 报文段的序号是 x，那么下一个报文段的序号是否就是 x + 1 呢？

解：首部中的序号字段值指的是本报文段所发送的数据的第一个字节的序号。

仅在此报文段中仅有 1 个字节的数据使，下一个报文段的序号才是 x + 1。

5-67 TCP 的吞吐量应当是每秒发送的数据字节数，还是每秒发送的首部和数据之和的字节数？吞吐量应当是每秒发送的字节数，还是每秒发送的比特数？

解：TCP 的吞吐量本来并没有标准的定义，TCP 的吞吐量可定义为每秒发送的数据字节数。

计算机内部的数据传送是以每秒多少字节作为单位的，而在通信线路上的数据率则常用每秒多少比特作为单位。

5-70 假定用 TCP 协议在 40 Gbit/s 的线路上传送数据。

(1) 如果 TCP 充分利用了线路的带宽，那么需要多长的时间 TCP 会发生序号绕回？

(2) 假定现在 TCP 的首部中采用了时间戳选项。时间戳占用了 4 字节，共 32 位。每隔一定的时间 (这段时间叫做一个嘀嗒) 时间戳的数值加 1。假定设计的时间戳是每隔 859 微秒，时间戳的数值加 1。试问要经过多长时间才发生时间戳数值的绕回。

解：(1) TCP 报文段中序号字段占 32 位，共 $2^{32} = 4294967296$ 个序号。

表示可以传输 4294967296 字节，即 34359738368 bit。

所求时间为 $34359738368 \text{ bit} / (40 \text{ Gbit/s}) = 859 \text{ ms}$ 。



Beta



Beta



今岁成蹊

关注

12



73



1



专栏目录

5-71 在 5.5 节中指出：例如，若用 2.5 Gbit/s 速率发送报文段，则不到 14 秒钟序号就会重复。请计算验证这句话。

解：TCP 首部中的序号字段占 32 位，共 $2^{32} = 4294967296$ 个序号，可为 2^{32} 个字节编号而不会重复。

不会重复发送的时间是 $2^{32} \times 8\text{bit} / (2.5\text{Gbit/s}) = 13.74\text{s} < 14\text{s}$ 。


【计算机网络 (谢希仁) 习题题解】目录

文章知识点与官方知识档案匹配，可进一步学习相关知识

网络技能树 首页 概览 41817 人正在系统学习中

(谢希仁)计算机网络第五版习题答案 11-12
(谢希仁)计算机网络第五版习题答案(谢希仁)计(谢希仁)计算机网络第五版习题答案算机网络第五...

计算机网络谢希仁课后答案 06-25
计算机网络课后答案。是谢希仁第五版的课后答案。

1 条评论  盼盼编程 **热评** 学习佳作，顺手点赞与关注 写评论

计算机网络spoc作业4·主机在配置IP地址时,其正确的子网掩码和... 11-23
第六周题目1:假设主机A向主机B发送5个连续的报文段,主机B对每个报文段进行确认,其中第二个...

计算机网络作业5 12-19
1、假设主机A向主机B发送5个连续的报文段,主机B对每个报文段进行确认,其中第二个报文段丢...

TCP序号绕回 qq_46146530的博客 221
tcp绕回的基本概念

计算机网络教程第5版-第5章运输层 (传输层) ZMiao的博客 1956
解答：从通信和信息处理的角度看，运输层向它上面的应用层提供端到端通信服务，它属于面向...

计算机网络第八版——第一章课后题答案(超详细)_计算机网络第八版答... 12-21
发送x比特的报文所需的时间是报文长度除以数据率b。因此发送时延是x/b。总的传播时延...

【计算机网络(微课版)】第3章 数据链路层 课后习题及答案_计算机网络微课... 12-23
设发送窗口发送的是0,1,2,3,4,5,接收窗口接收的是6,7,0,则不能确定0是新帧还是重传的 3.14一...

计算机网络第8版第四章网络层习题答案.pdf 11-11
计算机网络第8版第四章网络层习题答案

计算机网络大题 Echo的博客 3万+
如图所示网络。A在t=0时刻开始向C发送一个2Mbits的文件；B在t=0.1+e秒（e为无限趋近于0的...

【计算机网络】习题(三)——数据链路层_数据链路层习题 12-23
6.数据链路层采用了后退N帧的(GBN)协议,如果发送窗口的大小是32,那么至少需要()位的序列号...

【网络】必备知识点 ly甲烷的博客 750
应用层工作在用户态，传输层及以下工作在内核态 应用层、传输层、网络层、数据链路层、物理...

网络-TCP协议详解自学笔记（例题、代码、实战） 关注网络安全、云原生安全 4104
简介 TCP协议又叫传输控制协议(Transport Control Protocol)，是面向连接的，可靠的字节流服...

计算机网络考试复习的错题整理 kingJamesbond的博客 692
计算机网络面临的安全性威胁主要有四种：截获、中断、篡改以及伪造（标红色为考察过） 防火...

【计算机网络 (谢希仁) 习题题解】目录 热门推荐 HPP_CSDN的博客 3万+
计算机网络 第7版 (谢希仁) 习题题解目录

《计算机网络》谢希仁版--第五章运输层.pdf 12-24
《计算机网络》谢希仁版--第五章思维导图（适合期末复习或系统理解知识点

计算机网络期末复习电子版资料（谢希仁第8版） 04-16
csdn上有文章，可以直接看，需要下载打印的可以用这个电子版。



今岁成蹊

关注

12



73



1



专栏目录



Beta



Beta



举报

计算机网络第七版(谢希仁)第五章——传输层课后习题答案... LSYZWF的博客 5052
文章目录5-01解答5-02解答5-03解答5-04解答5-05解答5-06解答5-07解答5-08解答5-09解答5-10...

计算机网络课程总复习例题讲解 qq_64053451的博客 217
解：（1）长度为100字节的应用层数据交给传输层传送，需加上20字节的TCP首部。经过0.642...

TCP协议-TCP连接管理 yunfan 5321
一、TCP概述 TCP协议是 TCP/IP 协议族中一个非常重要的协议。它是一种面向连接、提供可靠...

计算机网络MOOC 作业5 junruit的博客 1万+
1.假设主机A向主机B发送5个连续的报文段，主机B对每个报文段进行确认，其中第二个报文段...

计算机网络第五章课后答案第七版（谢希仁著） Sunflower的博客 2万+
计算机网络第五章课后答案 简介：平时老师布置的作业自己做的答案，仅是做参考，希望有帮...

计算机网络谢希仁第五章 最新发布 11-25
计算机网络谢希仁第五章主要讲述了运输层的相关内容，包括传输控制协议TCP和用户数据报协...

“相关推荐”对你有帮助么？

非常没帮助 没帮助 一般 有帮助 非常有帮助

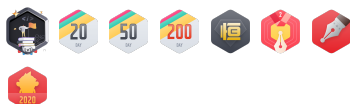
关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息
北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范
版权与免责声明 版权申诉 出版物许可证 营业执照 ©1999-2023北京创新乐知网络技术有限公司



今岁成蹊
码龄7年 暂无认证

102 4万+ 200万+ 19万+ 等级
原创 周排名 总排名 访问

1994 301 283 50 1430
积分 粉丝 获赞 评论 收藏



私信

关注



搜博文文章



热门文章

【计算机网络 (谢希仁) 习题题解】目录 37920
【计算机网络 (谢希仁) 习题题解】第4章 网
络层 (2)——划分子网; CIDR 13658
【计算机网络 (谢希仁) 习题题解】第2章 物
理层 13399
【计算机网络 (谢希仁) 习题题解】第5章 运
输层 (1)——UDP 12022



今岁成蹊

关注

12 73 1

专栏目录



Beta



Beta



举报

分类专栏

	计算机网络	14篇
	C#学习笔记	5篇
	C++ 学习笔记	30篇
	Windows via C/C++	4篇
	Python学习笔记	19篇
	通信	5篇

最新评论

- 【计算机网络 (谢希仁) 习题题解】第5章 ...
A_xxxxxx: 我觉得博主这个答案是正确的，除了建立连接那次消耗掉了一个序号，7 ...
- 【计算机网络 (谢希仁) 习题题解】第4章 ...
今岁成蹊: 是的，已修改
- 【计算机网络 (谢希仁) 习题题解】第4章 ...
嗯嗯.....: 4-55 第二题是不是应该160 AND 1 92才可以啊
- 【计算机网络 (谢希仁) 习题题解】第5章 ...
m0_74021670: 我理解是A->B的ack=202，因为A希望B发送的序号是202，但是B一 ...
- 【python 让繁琐工作自动化】第16章 处...
今岁成蹊: 代码的第23的作用其实就是将 response.text 中的内容赋值给 weatherDa ...

您愿意向朋友推荐“博客详情页”吗？



强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

- 【C# 7.0 in a Nutshell】第4章 C#的高级特性——委托
- 【C# 7.0 in a Nutshell】第3章 在C#中创建类型——类
- 【C# 7.0 in a Nutshell】第2章 C#语言基础——数组

2022年 2篇	2021年 34篇
2020年 51篇	2019年 15篇

目录

- 传输控制协议 TCP
- TCP 的连接
- 可靠传输的工作原理
- 停止等待协议
- 连续 ARQ 协议





举报