第十九章 TCP的交互数据流

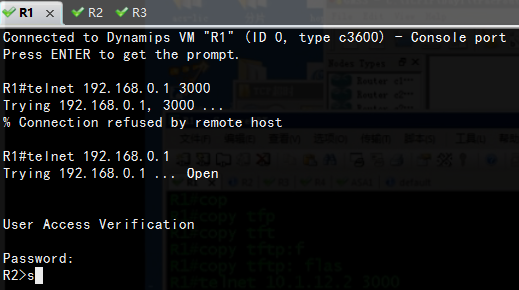
19.1 交互式输入

通常每一个交互按键都会产生一个数据分组

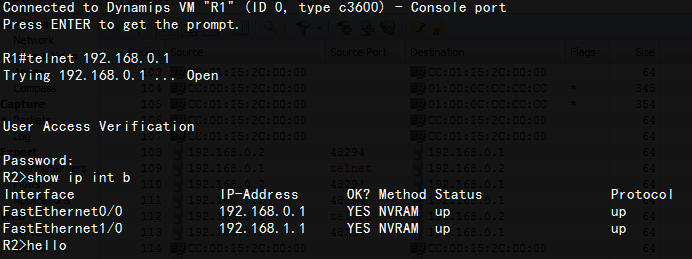
这样就会产生4个报文段：

1. 来自客户的交互按键；
2. 来自服务器的按键确认；
3. 来自服务器的按键回显；
4. 来自客户的按键回显确认

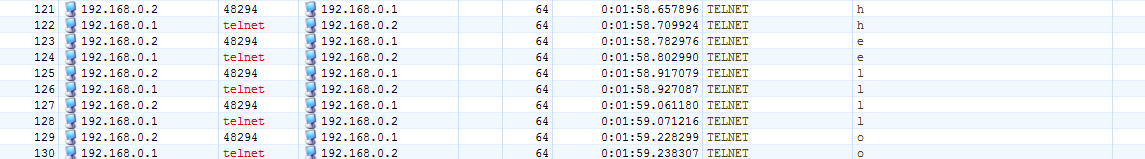
举例子：在secureCRT中的R1路由器telnet连接R2后，输入s为什么会在R1命令行显示s呢?



在secureCRT应用层的输入会交给TCP->IP->telnet client(R1)->网络->telnetserver(R2)->网络回显->telnet client(R1)->IP->TCP->看到数据并打印到屏幕上。



在secure中输入hello，抓包如下图：



交换式输入(抓包)



一种可能处理远程交互按键回显的方法

19.2 经受时延的确认

通常TCP在接收到数据时，并不立即发送ACK；相反，他推迟发送，以便将ACK与需要沿该方向发送的数据一起发送(有时称这种现象为数据捎带ACK，也就是19.1回显方法中将数据字节的确认和数据字节的回显并成一个包发送，如果按键连续发送数据字节，则回显数据字节的确认也可以和数据字节并成一个包发送)。绝大多数采用的时延为200ms，也就是说，TCP将以最大200ms的时延等待是否有数据一起发送。

19.3 Nagle算法

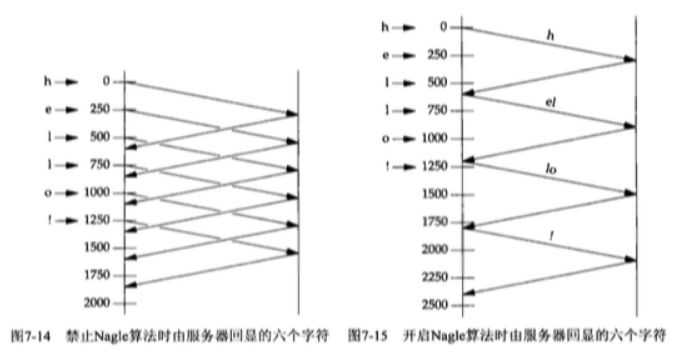
该算法要求一个TCP连接上最多只能有一个未被确认的未完成的小分组，在该分组的确认到达之前不能发送其他的小分组。相反，TCP收集这些少量的分组，并在确认到来时以一个分组的方式发出去。该算法的优越之处在于它是自适应的：确认到达得越快，数据也就发送得越快。而在希望减少微小分组数目的低速广域网上，则会发送更少的分组。

注意到从左到右待发送数据的长度是不同的，分别为：1、1、2、1、2、2、3、1和3个字节。这是因为客户只有收到前一个数据的确认后才发送已经收集的数据。通过使用Nagle算法，为发送16个字节的数据客户只需要使用9个报文段，而不是16个。

19.4 关闭Nagle算法

有时我们也需要关闭Nagle算法。一个典型的例子是X窗口系统服务器：小消息(鼠标移动)必须无延时地发送，以便为进行某种操作的交互用户提供实时的反馈。

这里将举另外一个更容易说明的例子，在一个交互注册过程中键入终端的一个特殊功能键。这个功能键通常可以产生多个字符序列，经常从ASCII码的转义(escape)字符开始。如果TCP每次得到一个字符，它很可能会发送序列号中的第一个字符(ASCII码的ESC)，然后缓存其他字符并等待对该字符的确认



19.5 窗口大小通告

报文段5通告的窗口大小为4095个字节，这意味着在TCP的缓冲区中仍然有一个字节等待应用程序读取。同样，来自客户的下一个报文段声明其窗口大小为4094个字节，这说明仍有两个字节等待读取。

服务器通常通告窗口大小为8192个字节，这是因为服务器在读取并回显接收到的数据之前，其TCP没有数据发送。当服务器已经读取了来自客户的输入后，来自服务器的数据将被发送。

然而，在ACK到来时，客户的TCP总是有数据需要发送。这是因为它在等待ACK的过程中缓存接收到的字符。当客户TCP发送缓存的数据时，客户没有机会读取来自服务器的数据，因此，客户通告的窗口大小总小于4096。

