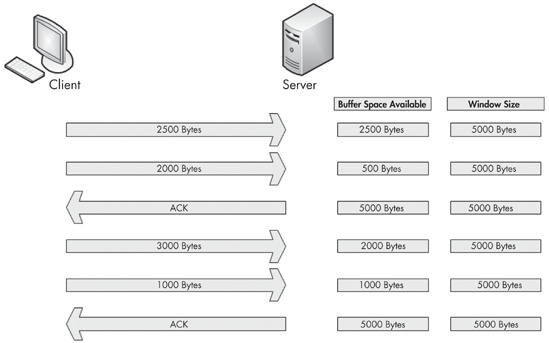
**一站式学习Wireshark（五）：TCP窗口与拥塞处理**

**转载请在文首保留原文出处：EMC中文支持论坛**<https://community.emc.com/go/chinese> [image001.gif](http://service.weibo.com/share/share.php?title=%23ECN%e4%b8%ad%e6%96%87%e6%94%af%e6%8c%81%e8%ae%ba%e5%9d%9b%23%20%e4%b8%80%e7%ab%99%e5%bc%8f%e5%ad%a6%e4%b9%a0Wireshark%ef%bc%88%e4%ba%94%ef%bc%89%ef%bc%9aTCP%e7%aa%97%e5%8f%a3%e4%b8%8e%e6%8b%a5%e5%a1%9e%e5%a4%84%e7%90%86@EMC%e6%98%93%e5%ae%89%e4%bf%a1%e4%b8%ad%e5%9b%bd%e6%8a%80%e6%9c%af%e7%a4%be%e5%8c%ba&url=https://community.emc.com/message/821593#821593)

**介绍**

TCP通过滑动窗口机制检测丢包，并在丢包发生时调整数据传输速率。滑动窗口机制利用数据接收端的接收窗口来控制数据流。

接收窗口值由数据接收端指定，以字节数形式存储于TCP报文头，并告知传输设备有多少数据将会存储在TCP缓冲区。缓冲区就是数据暂时放置的地方，直至传递至应用层协议等待处理。因此，发送端每次只能发送Window Size字段指定的数据量。为了使发送端继续传送数据，接收端必须发送确认信息：之前的数据接收到了。同时必须对占用缓冲区的数据进行处理以释放缓存空间。下图显示了接收窗口是如何工作的：

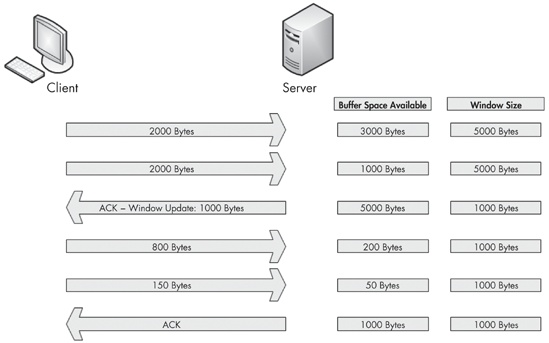
[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87023/image002.jpg)

上图中，客户端向服务器发送数据，服务器接收窗口是5000字节。客户端发送了2500字节，服务器缓冲区还剩2500字节，之后又发送了2000字节，从而缓冲区只剩500字节。服务器发送确认信息。对缓存中数据进行处理并清空缓存。此过程重复进行，客户端又发送3000字节和1000字节，服务器缓存减少至1000字节，客户端再次确认数据并处理缓存中内容。

**更多信息**

**调整窗口大小：**

当TCP堆栈接收到数据的时候，生成一个确认信息并以回复的方式发送，但是放置在接收端缓存中的数据并不总是立即被处理。当服务器忙于处理从多个客户端接收的报文，服务器很有可能因为清理缓存而变得缓慢，无法腾出空间接收新的数据，如果没有流控，则可能会造成丢包和数据损坏。好在，接收窗口所设定的速率无法使服务器正常处理数据时，能够调整接收窗口大小。通过减小返回给发送端的ACK报文的TCP头窗口大小值来实现。如下图所示：

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87027/image003.jpg)

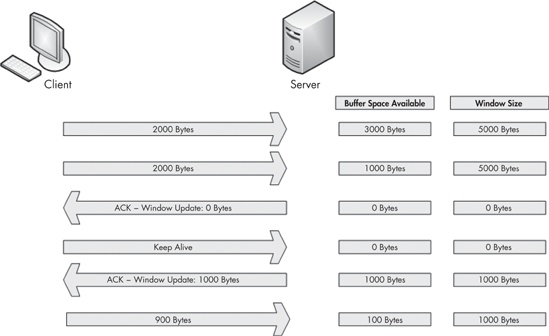
上图中，服务器初始窗口大小为5000字节。客户端发送2000字节，之后又发送了2000字节，缓冲区中只有1000字节可用。服务器意识到缓冲区正在快速填满，它知道如果数据继续以此速率传输，很快会有报文丢失。为了防止报文丢失，服务器发送确认信息给客户端，更新窗口大小为1000字节。结果，客户端减少数据发送，服务器以可以接受的速率处理缓存内容，即保持数据流以稳定的速率传输。

调整窗口大小在两个方向都是可行的。当服务器能够更加快速的处理报文时，它会发送一个较大窗口的ACK报文。

**零窗口暂停数据流：**

某些情况下，服务器无法再处理从客户端发送的数据。可能是由于内存不足，处理能力不够，或其他原因。这可能会造成数据被丢弃以及传输暂停，但接收窗口能够帮助减小负面影响。

当上述情况发生时，服务器会发送窗口为0的报文。当客户端接收到此报文时，它会暂停所有数据传输，但会保持与服务器的连接以传输探测（keep-alive）报文。探测报文在客户端以稳定间隙发送，以查看服务器接收窗口状态。一旦服务器能够再次处理数据，将会返回非零值窗口大小，传输会恢复。下图示例了零窗口通知过程。

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87028/image004.jpg)

服务器初始接收数据窗口为5000字节大小。从客户端接收4000字节数据之后，服务器负载变得非常繁重，无法继续处理客户端任何数据。服务器于是发送窗口大小值为0的报文。客户端暂停数据传输并发送一个探测报文。探测报文之后，服务器回复以告知客户端现在可以接收数据的报文，以及窗口大小为1000字节。客户端恢复传送数据。

**TCP滑动窗口实战：**

本例中，开始从192.168.0.20发送至192.168.0.30。我们关心的是窗口大小字段，可以从Packet List面板的Info栏以及Packet Details的TCP报文头看到。前三个报文后，可看到该值立刻减小，如下图所示：

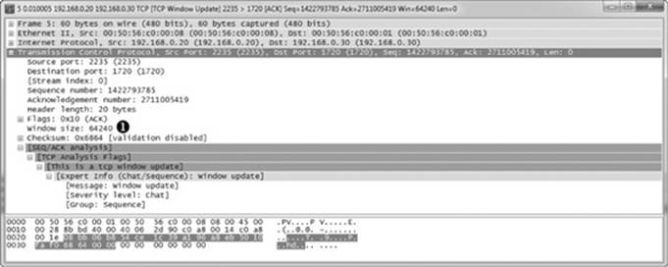
[image005.jpg](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87029/image005.jpg)

窗口大小值从第一个报文的8760字节变成第二个报文的5840字节到第三个报文的2920字节①。窗口大小值的减小是主机延时的典型标志。在时间栏注意到这一过程发生的非常迅速②。当窗口大小迅速减小的时候，通常就有可能下降为零。这就是第四个报文所发生的，如下图所示：

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87030/image006.jpg)

第四个报文从192.168.0.20发送至192.168.0.30，目的是告诉192.168.0.30它不再接收任何数据。0值见于TCP报文头①，Wireshark的Packet List面板Info栏，以及TCP报文头的SEQ/ACK Analysis字段②也告诉我们这是一个0窗口报文。

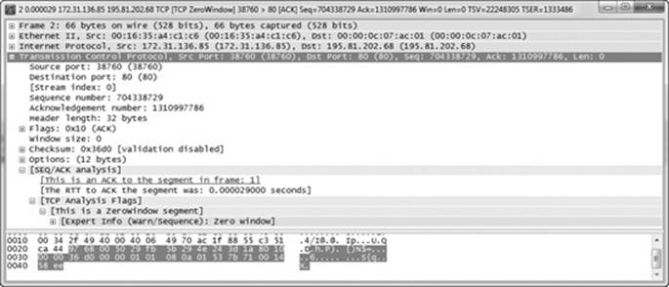
一旦发送了零窗口报文，192.168.0.30的设备不会再发送任何数据，直到收到从192.168.0.20的窗口更新，告知窗口大小已经增加了。本例中导致零窗口的问题是暂时的，所以在下一个报文中发送了窗口更新信息，如下图所示。

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87031/image007.jpg)

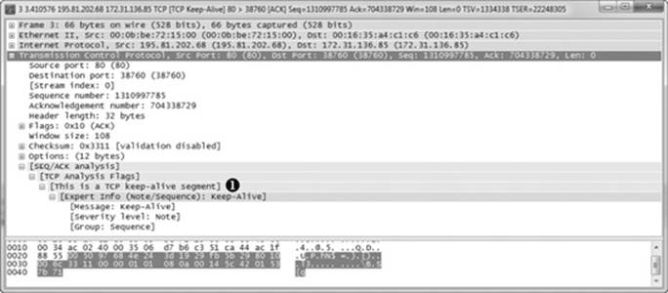
本例中，窗口大小增加到一个非常健康的数值64，240字节①。Wireshark再次在SEQ/ACK Analysis告诉我们这是一个窗口更新。

一旦收到更新报文，192.168.0.30的主机就再次开始发送数据，在报文6和报文7中。这一过程发生很快。如果它持续时间再长一点，就可能会导致网络的潜在中断，引起数据传输减慢或失败。

下一个关于滑动窗口的例子，第一个报文是正常HTTP，从195.81.202.68至172.31.136.85。此报文之后立刻跟随一个从172.31.136.85发送的零窗口报文，如下图所示：

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87032/image008.jpg)

这与上一个例子中的零窗口报文十分类似，但结果显著不同，172.31.136.85主机不是发送一个窗口更新并回复通讯，而是一个探测报文，如下图所示：

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87033/image009.jpg)

此报文被Wireshark标注为探测报文①。时间栏告诉我们这一报文发生于最后一个接收到的报文3.4秒之后。这一过程持续若干次，一端发送零窗口报文另一端发送探测报文，如下图所示：

[](https://community.emc.com/servlet/JiveServlet/showImage/2-821593-87034/image010.jpg)

探测报文发送间隙为3.4，6.8，13.5秒。这一过程可能会持续相当长一段时间，取决于通讯设备的操作系统。该情况下，把时间栏的值加起来，通讯暂停了25秒。

**TCP差错控制和流控排查总结：**

**重传报文**

重传的发生是由于客户端检测到服务器没有接收到它所发送的数据。因此，取决于你所分析的是通讯的哪一端，有可能是看不见重传的。如果从服务器端抓取数据，并且它确实没有接收到客户端所发送的和重传报文，可能会一无所获因为无法看见重传报文。如果怀疑并不是服务器端导致的报文丢失，可以考虑在客户端尝试抓取报文，以查看实际是否有重传发生。

**重复ACK**

可以将重复ACK看作重传的“所谓相反面”，因为它是在服务器检测到客户端发送报文丢失的时候产生的。大多数情况下，在通讯两端抓取流量时都可以看到重复ACK。需记住当接收报文乱序时会触发重复ACK。例如，如果服务器之接收到发送的第一个和第三个报文，就会导致发送重复ACK引起客户端对第二个报文的快速重传，因为你已经收到了第一个和第三个报文，因此不管导致第二个报文丢弃的原因是什么，都很有可能是暂时的，因此大多数情况下重复ACK都会成功发送和接收。当然，这种情形并不一定永远会发生，因此当你怀疑在服务器端丢失报文而又看不到任何重复ACK，考虑从通讯的客户端抓取报文。

**零窗口和探测报文**

滑动窗口直接与服务器无法接收和处理报文有关，任何窗口大小的缩小以及零值都是服务器问题的直接结果。所以如果你在哪里看到这两者之一发生，就应该在那里深入研究。通常应当在网络通讯两端一直主机窗口更新报文。