

半夜里，手机突然响起。恍惚中按下接听键，传来的竟然是老板的声音……一番寒暄之后（半夜跑到阳台接电话是挺寒的）：

老板：我司在为 C 电视台实施 **Isilon**，被一个读性能的问题卡了好几天。希望能派一位网络方面的专家，尽快飞到北京，你……

阿满：我们 **team** 有那么多 **CCIE**，能不能派他们去？我老婆正在生病，明天还约好了搬家公司。

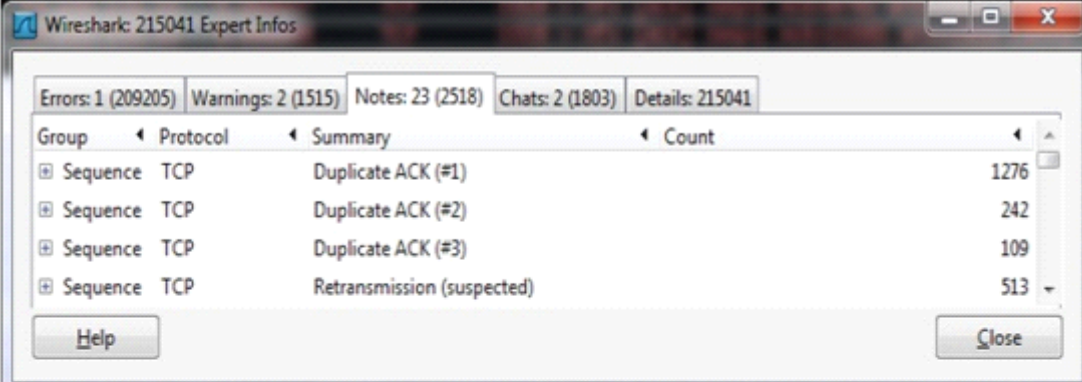
老板： # ¥ % * @ \$ ^ & ……你不用担心，我可以安排几个人帮你搬家。

阿满：（赶在老板安排人帮我照顾老婆之前）好吧，我赶紧准备一下。

回到房间，才想起以前完全没接触过 **Isilon**。**Google** 了一下，才知道是 **EMC** 收购的 **NAS** 产品，性能卓越，怎么现场工程师们会被卡了好几天？看看表，已经夜里 2 点了，还是先睡吧。

5 点起床，司机已经等在楼下了。飞驰到了办公室，现场工程师上传的网络包也已经准备完毕。粗略一看，有很多包发生了重传（**Retransmission**），而且有大量乱序

（**Out-Of-Order**）。我的第一反应就是，乱序导致了重传，从而影响了性能。由于时间有限，我随机挑了几个重传的包检查一下，发现重传的方向都是从 **Isilon** 到 **Windows**，这倒是符合读慢写快的症状。下图是 **Wireshark** 的分析结果。

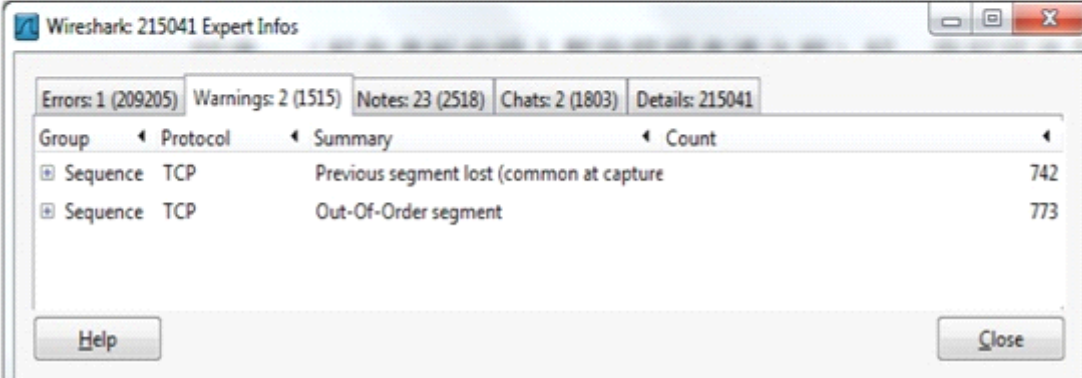


Wireshark: 215041 Expert Infos

Errors: 1 (209205) Warnings: 2 (1515) Notes: 23 (2518) Chats: 2 (1803) Details: 215041

Group	Protocol	Summary	Count
Sequence	TCP	Duplicate ACK (#1)	1276
Sequence	TCP	Duplicate ACK (#2)	242
Sequence	TCP	Duplicate ACK (#3)	109
Sequence	TCP	Retransmission (suspected)	513

Help Close



Wireshark: 215041 Expert Infos

Errors: 1 (209205) Warnings: 2 (1515) Notes: 23 (2518) Chats: 2 (1803) Details: 215041

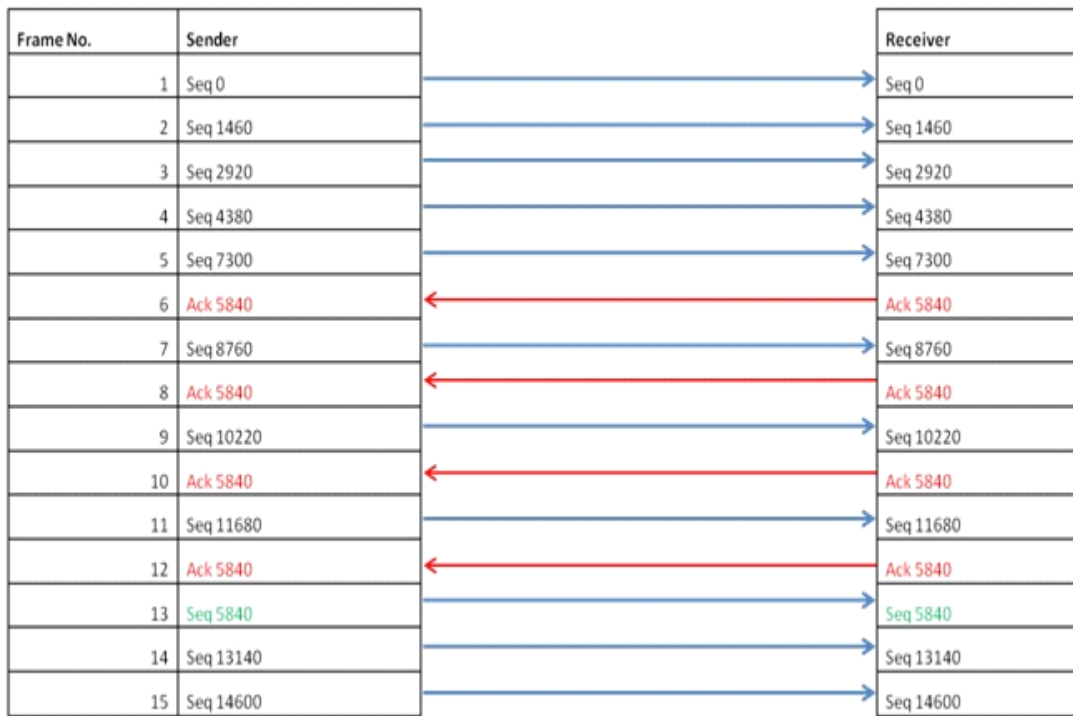
Group	Protocol	Summary	Count
Sequence	TCP	Previous segment lost (common at capture)	742
Sequence	TCP	Out-Of-Order segment	773

Help Close

乱序的网络包为什么会重传呢？下面简单介绍一下：

在正常情况下，接收方收到包的 **Seq** 号总是顺序的，比如在每个包长度为 **1460** 的情况下，**Seq** 号可能是这样的：**0**，**1460**，**2920**，**4380**……（每个相差 **1460**）。接收方知道下一个包的号码应该是什么，比如 **4380** 之后应该是 **4380+1460=5840**。如果收到的不是 **5840**，接

收方就知道包序乱了，它会回复一个包给发送方，说“我要的是5840（即 Ack 5840）”。如果接下来收到的还不是5840，那接收方每收到一个包，就会发一次“我要的是5840”给发送方，直到收到5840为止（如下图所示）。



对于发送方来说，持续收到“我要的是5840”不但意味着5840可能跑到其它包后面了，还可能意味着5840已经丢失。RFC里这样定义：如果发送方收到三次以上的“我要的是X”，即可认为包X丢失，立即启动快速重传。上图演示了这个过程。快速重传不但重新传输了（可能）丢失的包，还会减小TCP发送窗口，对性能的影响仅次于超时重传。分析到这里，我仿佛看到一丝曙光。

乱序一般是由发送方或者网络设备导致的。由于手头只有在Windows客户机抓的包，所以我无法进一步确认。便匆匆赶去机场，飞往北京了。在飞机上拟了一个计划：

1. Isilon和其它服务器一样，估计有类似Large Segment Offload的机制，也许关闭后能减少乱序。
2. 把Isilon和Windows客户端连到同一台空闲的Switch，尽量排除网络设备导致的乱序。
3. 在Isilon上再抓个网络包确认。

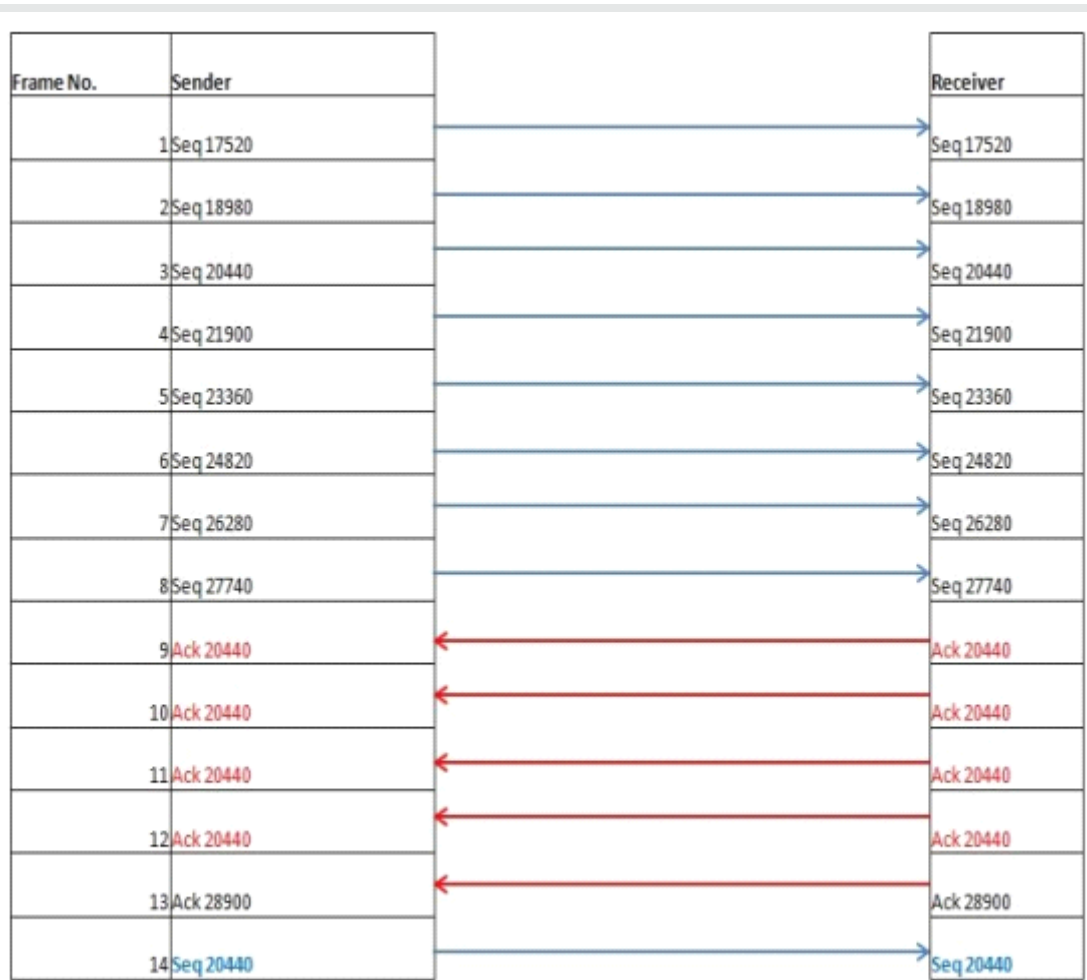
到了北京，已经是晚上。和几位来自香港，美国，北京等地的现场工程师在全聚德边吃边聊。原来他们已经做过很多方面的尝试，包括我计划中的第二点，但客户要求的80MB/s的读性能一直无法达到。客户端也换过几台，结果都差不多。目前看起来网络设备和客户端都没有问题，难道根本原因出在Isilon上了？这和我早上在网络包里看到的现象倒是吻合的。也许明天把Isilon的网络参数调一下，问题就解决了。到这个时候，我还是挺乐观的。

第二天一早，便赶到了C电视台的新大楼，比约定时间还早了2小时。更悲催的是比客户早了3小时……第一次到现场，才体会到现场工程师的苦。搭个测试环境也花了好长时间，等到真正可以测试，已经是下午了。令人兴奋的是，Isilon上果然有Large Segment Offload的设置，我满怀希望的关闭了这个功能。现场工程师立即启动一个新的读测试……

结果令我们大跌眼镜，读性能比之前还差一点，降到了**70MB/s**以下。这究竟是怎么回事？还有哪些设置可能会导致乱序？**LACP**昨天被现场工程师关掉，没什么好查的。我一下子不知道如何是好了。对着等我下一步建议的几位同事，不知道该说些什么。现场工程师习惯性的抓了一个包，叫我过去看。这一看更是一身冷汗，果然没有乱序的包了。我之前的猜测没有错，乱序是由 **Large Segment Offload** 导致的。但为什么消除了乱序之后性能没有得到改善呢？再看重传率，还是一样高。难道我从一开始就走错胡同，重传根本就不是乱序导致的？我不得不一个人坐到角落里，再次打开昨天早上研究过的网络包。一个一个的查看乱序的包，果然看到了一些很有趣的现象。如下图所示，虽然乱序的比例很大，但是每个乱序都是相邻两个包的颠倒，这样接收方永远不会发出三个以上的“我要的是 X”，也不会触发快速重传。



再逐个分析重传的网络包，现象就更加有趣了。如下图所示，接收方明明收到了 **Seq 20440** (**Frame No. 3**)，但它发送了4个“**Ack 20440**”给发送方，触发了发送方的快速重传机制。这说明接收方的 **TCP** 层在收到**20440**之前，已经收到了**21900, 23360, 24820, 26280**。这个结果表明，客户端在把包从网络层交给 **TCP** 层的时候，自己把包打乱了。为什么我们知道客户端只是打乱**20440**，而不是丢弃呢？在发送方重传的 **Seq 20440** (**Frame No. 14**) 到达接收方之前，接收方又发送了“**Ack 28900**”，这表明接收方的 **TCP** 层收到了**28900**之前的所有包，包括**20440**。从以上分析可以得出，重传的根本原因发生在操作系统或者网卡驱动，和 **Switch** 或者 **Isilon** 完全没有关系。



这结果是何等出人意料，以至于我自己都难以接受。不但颠覆了我前一天早上的分析结果，而且无法说服同事和客户：我们已经测试了7台客户端，结果都是一样的，难道7台同时出问题了？这概率低得难以置信。接下来的就是一场辩论，C电视台派出了一位网络专家，企图说服我客户端没有问题。我无法解释为什么会碰到如此低概率的事情，他也无法反驳我在网络包中看到的问题。一直拉锯到夜里12点，客户才答应第二天提供另外7台客户端供我们测试，但是有一个要求，必须给提供他们一个官方的分析报告，证明的确是客户端导致的问题。

和客户吃完晚饭，已经凌晨1点。JW万豪非常贴心，准备好了巧克力，掀好被子，拆好拖鞋。可惜我没有机会享受，等写完报告，已经到了凌晨3点半。没睡多久，morning call就来了……再次感叹现场工程师真辛苦。这天我是真的信心满满的到C电视台的，因为大多数重传的包都被我逐一研究过。现场工程师手脚麻利，很快就测出结果，在7台同时读写的情况下，每台都到100MB/s以上，大大超过了客户的期望值，甚至达到客户机单网卡的理论极限了。在场的现场工程师们异常兴奋，给测试结果拍照，截屏，甚至拍了一段视频。他们被这个项目压抑太久了，需要好好庆祝。

而我也背起笔记本，挥手向这栋造型诡异的建筑，向这个诡异的问题告别。匆匆赶往首都机场，家里还有生病的老婆，断奶的孩子，还要没搬完的家……