[TCP层的分段和IP层的分片之间的关系 & MTU和MSS之间的关系](http://blog.csdn.net/yusiguyuan/article/details/22782943)

标签： [网络编程](http://www.csdn.net/tag/%e7%bd%91%e7%bb%9c%e7%bc%96%e7%a8%8b)[c](http://www.csdn.net/tag/c)[以太网](http://www.csdn.net/tag/%e4%bb%a5%e5%a4%aa%e7%bd%91)

2014-04-02 10:16 5312人阅读 [评论](http://blog.csdn.net/yusiguyuan/article/details/22782943#comments)(6) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/yusiguyuan/article/details/22782943#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

TCP/IP详解（56） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

**首先说明：数据报的分段和分片确实发生，分段发生在传输层，分片发生在网络层。但是对于分段来说，这是经常发生在UDP传输层协议上的情况，对于传输层使用TCP协议的通道来说，这种事情很少发生。**

**1，MTU（Maximum Transmission Unit，MTU），最大传输单元**

（1）以太网和802.3对数据帧的长度都有一个限制，其最大 值分别是1500和1492个字节。链路层的这个特性称作MTU。不同类型的网络大多数都有一个上限。如果IP层有一个数据要传，且数据的长度比链路层的 MTU还大，那么IP层就要进行分片（fragmentation），把数据报分成若干片，这样每一个分片都小于MTU。

（2）把一份IP数据报进行分片以后，由到达目的端的**IP层**来进行重新组装，其目的是使分片和重新组装过程对运输层（TCP/UDP）是透明的。由于每一分片都是一个独立的包，当这些数据报的片到达目的端时有可能会失序，但是在IP首部中有足够的信息让接收端能正确组装这些数据报片。

（3）**尽管IP分片过程看起来透明的，但有一点让人不想使用它：即使只丢失一片数据也要重新传整个数据报。**why？因为IP层本身没有超时重传机制------由更高层（比如TCP）来负责超时和重传。当来自TCP报文段的某一片丢失后，TCP在超时后会重发整个TCP报文段，该报文段对应于一份IP数据报（而不是一个分片），没有办法只重传数据报中的一个数据分片。

（4）**使用UDP很容易导致IP分片，TCP试图避免IP分片**。 那么TCP是如何试图避免IP分片的呢？其实说白了，采用TCP协议进行数据传输是不会造成IP分片的，因为一旦TCP数据过大，超过了MSS，则在传输 层会对TCP包进行分段（如何分，见下文！），自然到了IP层的数据报肯定不会超过MTU，当然也就不用分片了。而对于UDP数据报，如果UDP组成的 IP数据报长度超过了1500，那么IP数据报显然就要进行分片，因为UDP不能像TCP一样自己进行分段。**总结：UDP不会分段，就由我IP来分。TCP会分段，当然也就不用我IP来分了！**

**2，MSS（Maxitum Segment Size）最大分段大小的缩写，是TCP协议里面的一个概念**

（1）MSS就是TCP数据包每次能够传输的最大数据分段。为了达到最佳的传输效能TCP协议在**建立连接的时候通常要协商双方的MSS**值，这个值TCP协议在实现的时候往往用MTU值代替（需要减去IP数据包包头的大小20Bytes和TCP数据段的包头20Bytes）所以往往**MSS为1460**。通讯双方会根据双方提供的MSS值得最小值确定为这次连接的最大MSS值。

        （2）相信看到这里，还有**最后一个问题：TCP是如何实现分段的呢**？其实TCP无所谓分段，因为每个TCP数据报在组成前其大小就已经被MSS限制了，所以TCP数据报的长度是不可能大于MSS的，当然由它形成的IP包的长度也就不会大于MTU，自然也就不用IP分片了。

简而言之：

        1.IP分片产生的原因是网络层的MTU；TCP分段产生原因是MSS.

        2.IP分片由网络层完成，也在网络层进行重组；TCP分段是在传输层完成，并在传输层进行重组.   //透明性

        3.对于以太网，MSS为1460字节，而MUT往往会大于MSS.

        故采用TCP协议进行数据传输，是不会造成IP分片的。若数据过大，只会在传输层进行数据分段，到了IP层就不用分片。

        所以可以看成是这种情况：传输层协议想发送一个超过了MTI的数据报，这个时候网络层就需要对其进行分片，一般UDP和ICMP会出现分片情况，但是TCP不会出现这种情况！因为TCP使用了MSS来避免分片！

        IP分片只有第一个带有传输层或ICMP首部，其余的分片只有IP头。至于怎么重组就是到对端以后IP层的事情了。

若TCP报文非常长那么在IP层传输时就有可能要分解成多个短数据报片。（计算机网络谢希仁）

TCP分段每个都有完整首部。

PS:所以我觉得是这样的，TCP的分段是针对应用层的数据来说的，比如使用TCP发送70KB的数据，这个时候就需要将70KB分成若干个MSS，到了网络层就不需要分片了。MSS的存在就避免了网络层分片的发生，

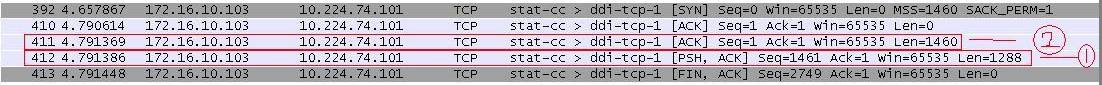
        IP层的分片是针对传输层中使用UDP协议来说的，如果使用UDP发送数据，UDP并不知道如何分段，那么到了IP层就需要进行分片，分片的原则根据MTU，那么分UDP最大的数据负载就是1500-8=1492

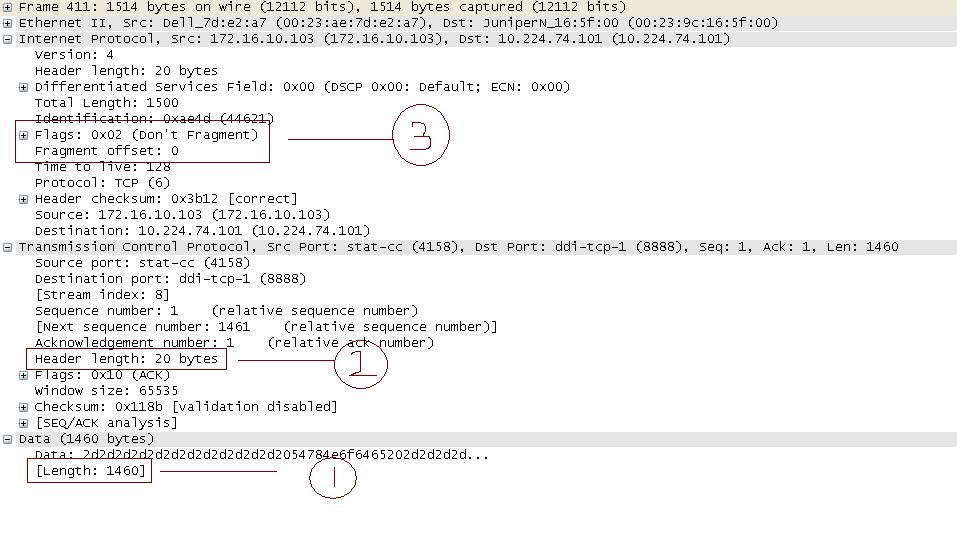
TCP/UDP实验

看完了理论，让我们实践一把，看是否与以上的理论相符。

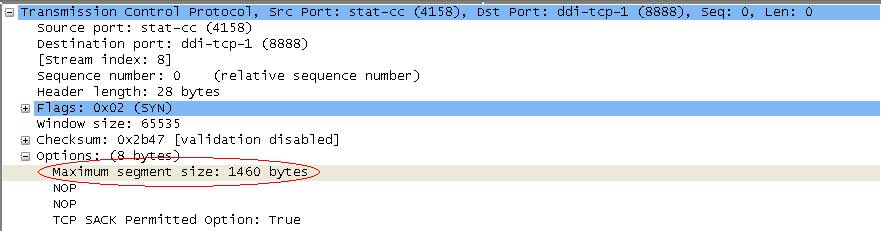
对于TCP来说，它是尽量避免分片的。假设我们这里要发送给TCP层的数据大小为2748个字节，这个大小是明显大于链路层的发送数据的大小的，在这个情况

下我们来看，对于来自TCP层的数据，IP会不会进行分片。

  
   
从第一张图看来，应用层的2748个字节在TCP层就进行了分段，分层了两个TCP段，一个1460字节，一个1288字节。那么到IP层的时候，自然就不会在进行分片了。

  
从第二张图片看出，在这两个TCP分段中，在序号3处，IP的头部字段*（Don ' t Fragment）* 被设置了，用于告诉IP层不要对该数据进行分片。

而对于MSS大小的协商，我们可以从下面这张图片看到，下面的图片是TCP CLIENT发出的第一个SYN TCP分段：



对于UDP来说，假设我们要发送的一个UDP数据包大小为1600个字节，那么在实际上通过UDP/IP分发出去的时候，会不会进行分片呢? 看如下的图片：  
  
   
 从上面的图片可以看出，我们发送的数据包的大小为1600字节（序号1处），在UDP层，长度为1608字节（序号2处），这里的8个字节是UDP的头部字段的长度， 到了IP层（序号3处），我们可以清楚的看到IP对UDP数据包进行了分片，一个大小为1480字节，一个为128字节。