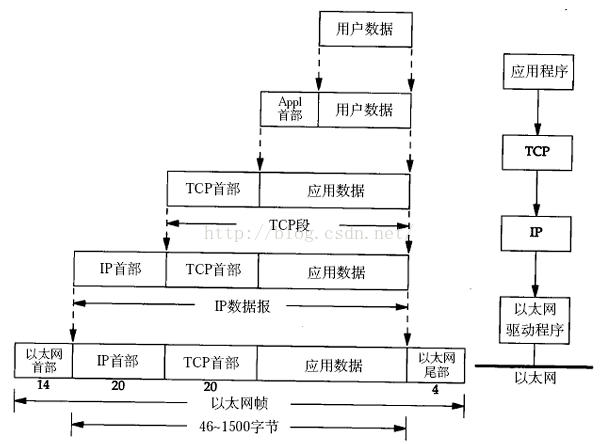
**TCP、UDP数据包大小的限制**

原创 2016年05月29日 10:27:34

* 标签：
* [socket](http://so.csdn.net/so/search/s.do?q=socket&t=blog) /
* [网络通信](http://so.csdn.net/so/search/s.do?q=%E7%BD%91%E7%BB%9C%E9%80%9A%E4%BF%A1&t=blog)
* 19466

**1、概述**

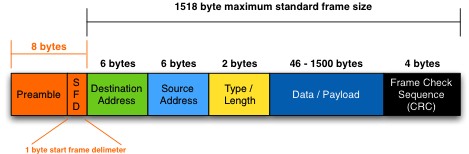
首先要看TCP/IP协议，涉及到四层：链路层，网络层，传输层，应用层。 　　  
其中以太网（Ethernet）的数据帧在链路层 　　  
IP包在网络层 　　  
TCP或UDP包在传输层 　　  
TCP或UDP中的数据（Data)在应用层 　　  
它们的关系是 数据帧｛IP包｛TCP或UDP包｛Data｝｝｝ 　　  
  
    不同的协议层对数据包有不同的称谓，在传输层叫做段(segment)，在网络层叫做数据报(datagram)，在链路层叫做帧(frame)。数据封装成帧后发到传输介质上，到达目的主机后每层协议再剥掉相应的首部，最后将应用层数据交给应用程序处理。

在应用程序中我们用到的Data的长度最大是多少，直接取决于底层的限制。 　　  
我们从下到上分析一下： 　　  
1.在链路层，由以太网的物理特性决定了数据帧的长度为(46＋18)－(1500＋18)，其中的18是数据帧的头和尾，也就是说数据帧的内容最大为1500(不包括帧头和帧尾)，即MTU(Maximum Transmission Unit)为1500； 　  
2.在网络层，因为IP包的首部要占用20字节，所以这的MTU为1500－20＝1480；　  
3.在传输层，对于UDP包的首部要占用8字节，所以这的MTU为1480－8＝1472； 　　  
所以，在应用层，你的Data最大长度为1472。当我们的UDP包中的数据多于MTU(1472)时，发送方的IP层需要分片fragmentation进行传输，而在接收方IP层则需要进行数据报重组，由于UDP是不可靠的传输协议，如果分片丢失导致重组失败，将导致UDP数据包被丢弃。 　　  
从上面的分析来看，在普通的局域网环境下，UDP的数据最大为1472字节最好(避免分片重组)。 　　  
但在网络编程中，Internet中的路由器可能有设置成不同的值(小于默认值)，Internet上的标准MTU值为576，所以Internet的UDP编程时数据长度最好在576－20－8＝548字节以内。  
　

**2、TCP、UDP数据包最大值的确定**

        UDP和TCP协议利用端口号实现多项应用同时发送和接收数据。数据通过源端口发送出去，通过目标端口接收。有的网络应用只能使用预留或注册的静态端口；而另外一些网络应用则可以使用未被注册的动态端口。因为UDP和TCP报头使用两个字节存放端口号，所以端口号的有效范围是从0到65535。动态端口的范围是从1024到65535。

        MTU最大传输单元，这个最大传输单元实际上和链路层协议有着密切的关系，EthernetII帧的结构DMAC+SMAC+Type+Data+CRC由于以太网传输电气方面的限制，每个以太网帧都有最小的大小64Bytes最大不能超过1518Bytes，对于小于或者大于这个限制的以太网帧我们都可以视之为错误的数据帧，一般的以太网转发设备会丢弃这些数据帧。



        由于以太网EthernetII最大的数据帧是1518Bytes这样，刨去以太网帧的帧头（DMAC目的MAC地址48bits=6Bytes+SMAC源MAC地址48bits=6Bytes+Type域2Bytes）14Bytes和帧尾CRC校验部分4Bytes那么剩下承载上层协议的地方也就是Data域最大就只能有1500Bytes这个值我们就把它称之为MTU。

UDP 包的大小就应该是 1500 - IP头(20) - UDP头(8) = 1472(Bytes)  
TCP 包的大小就应该是 1500 - IP头(20) - TCP头(20) = 1460 (Bytes)

注\*PPPoE所谓PPPoE就是在以太网上面跑“PPP”。随着宽带接入（这种宽带接入一般为Cable Modem或者xDSL或者以太网的接入），因为以太网缺乏认证计费机制而传统运营商是通过PPP协议来对拨号等接入服务进行认证计费的，所以引入PPPoE。PPPoE导致MTU变小了以太网的MTU是1500，再减去PPP的包头包尾的开销（8Bytes），就变成1492。不过目前大多数的路由设备的MTU都为1500。

        如果我们定义的TCP和UDP包没有超过范围，那么我们的包在IP层就不用分包了，这样传输过程中就避免了在IP层组包发生的错误；如果超过范围，既IP数据报大于1500字节，发送方IP层就需要将数据包分成若干片，而接收方IP层就需要进行数据报的重组。更严重的是，如果使用UDP协议，当IP层组包发生错误，那么包就会被丢弃。接收方无法重组数据报，将导致丢弃整个IP数据报。UDP不保证可靠传输；但是TCP发生组包错误时，该包会被重传，保证可靠传输。

        UDP数据报的长度是指包括报头和数据部分在内的总字节数，其中报头长度固定，数据部分可变。数据报的最大长度根据操作环境的不同而各异。从理论上说，包含报头在内的数据报的最大长度为65535字节(64K)。

       我们在用Socket编程时，UDP协议要求包小于64K。TCP没有限定，TCP包头中就没有“包长度”字段，而完全依靠IP层去处理分帧。这就是为什么TCP常常被称作一种“流协议”的原因，开发者在使用TCP服务的时候，不必去关心数据包的大小，只需讲SOCKET看作一条数据流的入口，往里面放数据就是了，TCP协议本身会进行拥塞/流量控制。

       不过鉴于Internet(非局域网)上的标准MTU值为576字节，所以建议在进行Internet的UDP编程时，最好将UDP的数据长度控制在548字节 (576-8-20)以内。

**3、TCP、UDP数据包最小值的确定**

在用UDP局域网通信时，经常发生“Hello World”来进行测试，但是“Hello World”并不满足最小有效数据(64-46)的要求，为什么小于18个字节，对方仍然可用收到呢？因为在链路层的MAC子层中会进行数据补齐，不足18个字节的用0补齐。但当服务器在公网，客户端在内网，发生小于18个字节的数据，就会出现接收端收不到数据的情况。

       以太网EthernetII规定，以太网帧数据域部分最小为46字节，也就是以太网帧最小是6＋6＋2＋46＋4＝64。除去4个字节的FCS，因此，抓包时就是60字节。当数据字段的长度小于46字节时，MAC子层就会在数据字段的后面填充以满足数据帧长不小于64字节。由于填充数据是由MAC子层负责，也就是设备驱动程序。不同的抓包程序和设备驱动程序所处的优先层次可能不同，抓包程序的优先级可能比设备驱动程序更高，也就是说，我们的抓包程序可能在设备驱动程序还没有填充不到64字节的帧的时候，抓包程序已经捕获了数据。因此不同的抓包工具抓到的数据帧的大小可能不同。下列是本人分别用wireshark和sniffer抓包的结果，对于TCP 的ACK确认帧的大小一个是54字节，一个是60字节，wireshark抓取时没有填充数据段，sniffer抓取时有填充数据段。

**4、实际应用**

        用UDP协议发送时，用sendto函数最大能发送数据的长度为：65535- IP头(20) - UDP头(8)＝65507字节。用sendto函数发送数据时，如果发送数据长度大于该值，则函数会返回错误。

        用TCP协议发送时，由于TCP是数据流协议，因此不存在包大小的限制（暂不考虑缓冲区的大小），这是指在用send函数时，数据长度参数不受限制。而实际上，所指定的这段数据并不一定会一次性发送出去，如果这段数据比较长，会被分段发送，如果比较短，可能会等待和下一次数据一起发送。

1、数据链路层对数据帧的长度都有一个限制，也就是链路层所能承受的最大数据长度，这个值  
称为最大传输单元，即MTU。以以太网为例，这个值通常是1500字节。  
  
2、对于IP数据包来讲，也有一个长度，在IP包头中，以16位来描述IP包的长度，也就是说，  
一个IP包，最长可能是65535字节。  
  
3、结合以上两个概念，第一个重要的结论就出来了，如果IP包的大小，起过了MTU值，那么就需要  
分片，也就是把一个IP包分为多个，这个概念非常容易理解，一个载重5T的卡车，要拉10T的货，它  
当然就得分几次来拉了。  
  
4、IP分片是很多资料常讲的内容，但是我倒是觉得分不分片其实不重要，重要的是另一个东西。一个  
数据包穿过一个大的网络，它其间会穿过多个网络，每个网络的MTU值是不同的。我们可以设想，如果  
接受/发送端都是以太网，它们的MTU都是1500，我们假设发送的时候，数据包会以1500来封装，然而，  
不幸的是，传输中有一段X.25网，它的MTU是576，这会发生什么呢？我想，这个才是我们所关心的。  
当然，结论是显而易见的，这个数据包会被再次分片，咱开始用火车拉，到了半路，不通火车，只通汽车，  
那一车货会被分为很多车……仅此而已，更重要的是，这种情况下，如果IP包被设置了“不允许分片标志”，那  
会发生些什么呢？对，数据包将被丢弃，然事收到一份ICMP不可达差错，告诉你，需要分片！  
这个网络中最小的MTU值，被称为路径MTU，我们应该有一种有效的手段，来发现这个值，最笨的方法或许是先  
用traceroute查看所有节点，然后一个个ping……  
  
5、到了传输层，也会有一个最大值的限制，当然，对于只管发，其它都不管的UDP来说，不再我们讨论之列。这里  
说的是TCP协议。说到大小，或许会让人想到TCP著名的滑动窗口的窗口大小，它跟收发两端的缓存有关，这里讨论  
的是传输的最大数据包大小，所以，它也不在讨论之列。  
TCP的选项字段中，有一个最大报文段长度（MSS），表示了TCP传往另一端的最大数据的长度，当一个连接立时，连接  
的双方都要通告各自的MSS，也就是说，它说是与TCP的SYN标志在一起的。当然，对于传输来讲，总是希望MSS越大越好，  
现在超载这么严重，谁家不希望多拉点货……但是，MSS总是有个限制的，也就是MTU-IP头长度-TCP头长度，对于以太网来讲  
它通常是1500-20-20=1460，虽然总是希望它能很大(如1460)，但是大多数BSD实现，它都是512的倍数，如1024…  
  
6、回到分片上来，例如，在Win2000下执行如下命令：  
ping 192.168.0.1 -l 1473  
  
按刚才的说法，1473+20(ip头)+8(icmp头)=1501，则好大于1500，它会被分片，但是，我们关心的是：  
  
这个数据包会被怎么样分法？  
可以猜想，第一个包是  
以太头+IP头+ICMP头+1427的数据；  
那第二个分片包呢？  
它可以是：  
以太头+IP头+ICMP头+1个字节的数据  
或者是：  
以太头+IP头+1个字节的数据  
也就是省去ICMP头的封装，当然，IP头是不可以省的，否则怎么传输了……  
  
事实上，TCP/IP协议采用的是后一种封装方式，这样，一次可以节约8个字节的空间。IP包头中，用了三个标志来描述一个分片包：  
1、分片标志：如果一个包被分片了，被置于1，最后一个分片除外；——这样，对于接收端来讲，可以根据这个标志位做为重组的重  
要依据之一；  
2、分片偏移标志：光有一个标志位说明“自己是不是分片包”是不够的，偏移标志位说明了自己这个分片拉于原始数据报的什么位置，  
很明显，这两个标志一结合，就很容易重组分片包了。  
3、不允许分片标志：如果数据包强行设置了这个标志，那么在应该分片的时候，…… err，刚才已经说过了  
  
在以太网中，数据包的大小范围是在64—1518字节之间，如果除去头部开销，则实际的数据大小为46—1500字节之间。 一般情况下，数据包的大小都是在这个范围内，如果数据包小于64字节，称为碎片；而如果大于1518字节，称为特大数据包。这两种类型的数据包都是非正常的以太网数据包，它们将影响网络的正常运行。无论是碎片或特大数据包，都会增加网络的负载，导致网络故障的发生。所以，我们在对网络进行分析的时候，对数据包大小的判断也是不可缺少的一个环节。