# Pktgene接口说明文档

*jzheng 14125224@bjtu.edu.cn*

## pktgene介绍

pktgene是一款工作在Linux kernel模式下的发包工具[[1]](#footnote-1)，源自Linux pktgen模块。仅支持以太网帧格式，支持必要的快速校验和。

原始的pktgen在linux源码树的位置为: *net/core/pktgen.c*，采用模块来编译。通常的发行版中该模块位于*/lib/modules/`uname -r`/kernel/net/core/pktgen.ko*，且pktgen采用proc文件系统给出了用户空间的配置接口。其proc文件目录位于*/proc/net/pktgen*下，至此，用户可以通过读写下面的文件来控制发包行为。

原始的pktgen中也只支持ethernet帧格式，且格式固定，只支持ip/udp格式上层协议，这也是pktgen的在功能测试中的一个限制因素。

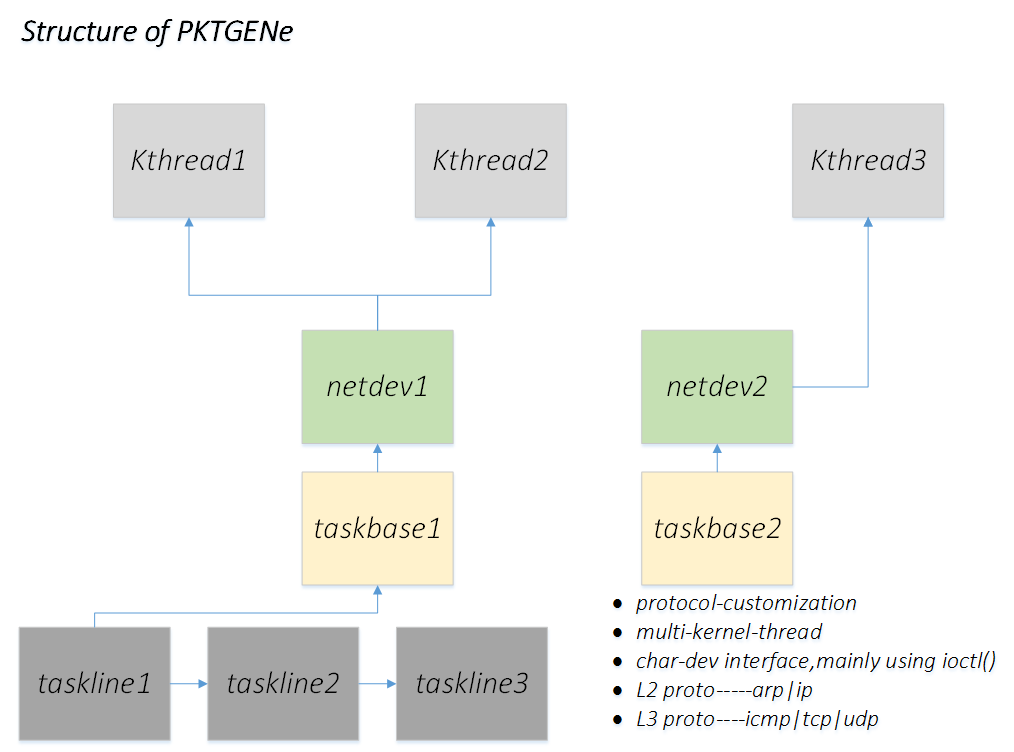
Pktgen只校验IP头部分，不校验其以外部分，且每当更新完一个字段的时候，整个ip头部参与校验计算。udp校验和始终未0，即不参与校验，这是udp协议允许的。所以原始的pktgen的校验部分也是功能有限。

针对pktgen对多协议支持的不足，pktgene扩展了协议使之支持常见的ip,arp二层协议和icmp/tcp/udp三层协议，以及快速校验功能。

## pktgene功能实现

Pktgene在结构上和pktgen一致，都采用多个kernel线程来填充帧并发送帧，每个线程通过绑定到到每个cpu上，利用cpu的亲和性，减少上下文的切换开销。且每个线程可以绑定一个网络接口，一个接口的发包任务可以指派给多个线程。报文传输通过获取内核中注册的网络设备中包传输函数的调用完成。

帧的内容修改作为pktgene的核心部分，其修改作为通过任务数据结构来描述的。一个完整的任务有一个任务基础和若干任务线组成。任务基础指定了帧最基础的内容，之后的任务线描述了随着发帧数的推移，在某一帧上对基础内容所做出的修改。其具体结构如下图所示：



图中展示了各个数据结构之间的从属关系，这与用户接口是一致的。用户负责生成和指定各个结构之间的从属关系。

任务线作为帧修改最基本的单元，描述了在发第几个帧的时候需要以何种方式修改的内容。其结构体的定义如下

|  |
| --- |
| struct pktgene\_task\_line{/\*we using little endian here\*/  int frame\_start\_no;  int frame\_end\_no;/\*-1 indicate never end\*/  int data\_type;/\*1,2,4 supported\*/  int frame\_offset;  int data\_lseek;  int data\_acton;  /\*int data\_length;\*/  union {  uint32\_t qword\_val;  uint16\_t word\_val;  uint8\_t byte\_val;  unsigned char ustuff[4];  };  struct list\_head lh\_task\_line\_node;  }; |

其中知识了该任务线的有效范围[frame\_start\_no, frame\_end\_no]，以及修改帧偏移多少字节的内容，其中偏移以帧的开始计数。之所以选择1.2.4作为基本的数长度类型，因为基本的数据长度中很少有超过4个字节的PDU字段。

但是上述任务中并不要求按照字来对齐，为了进一步计算校验和，其偏移必须想对于帧的起始地址按字对齐，所以在实际填充包的时候，要进一步拆分任务成更多的对其的子任务，其拆分过程如下:



对于一个字节的数据类型来说，若其地址按字对齐了，则与紧接着的后一字节组成一个字进行更新。否则与前一个字节组成字进行内容的更新。

对于字和双字类型来，按照其首字节的对齐情况决定是否需要扩展，并且进一步拆分成字类型存储。更新完数据后，进一步重新计算可能的校验和字段。Pktgene中仅支持ip首部校验和计算，tcp和udp的校验和计算。

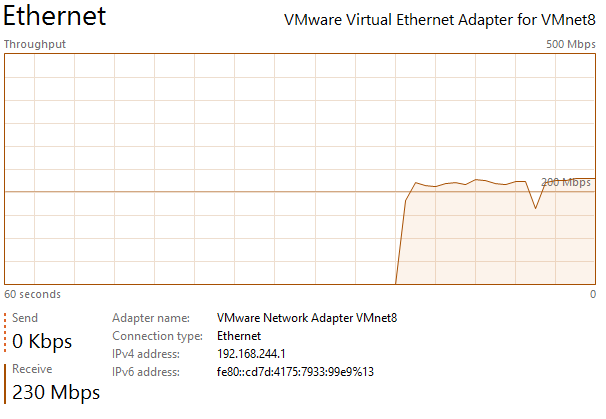
快速计算校验和不必重新以根据所有的的数据内容来计算，而是根据现有的校验和以及更改数据的内容和更新后的数据的内容进行其差值的计算。按照常规的校验和计算方法，先在校验和的位置0，然后按字并且以wrap around的方式计算所有的区域后，最后16位校验和取反填充在校验和字段。所以在快速计算校验和的过程中，认为将要计算的位置通过加上原始值的反码并得到新的校验值，然后加上新的值，从而得到更新后的最终校验值。

在pktgene引出了一个字符设备作为配置接口，利用该接口的ioctl方法可以完成所有的配置，详见工程中的sample目录。

## 功能测试

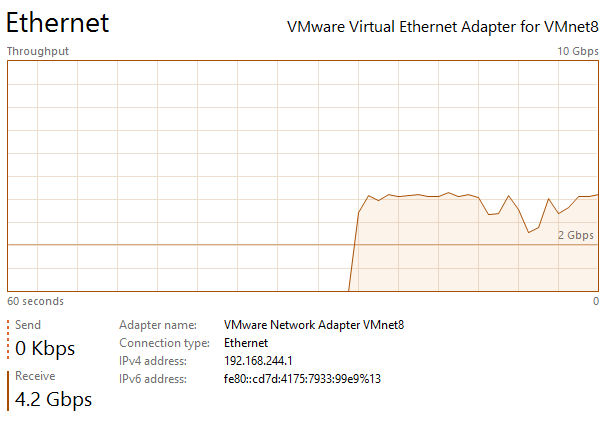
所有测试均使用一个线程，在虚拟机下测试，虚拟机网卡采用nat模式。

小包(60字节)吞吐率测试：



其峰值吞吐率为3.8 Mpps。

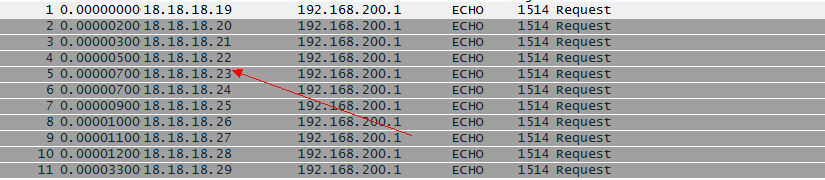
大包(1514字节)性能测试：



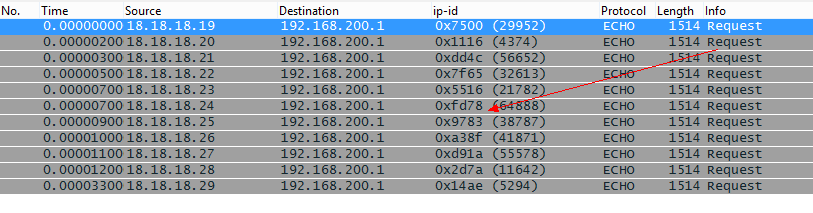
其吞吐率大约为:2.8 Mpps。

任务测

在任务中主要有两部分，在帧的偏移为18字节处，对于后续两个字节采取随机变化，即指定IP首部的标识符；且在26字节偏移处，在0x12121212的基础上每帧加一。也就是指定IP首部的源地址变化序列为递增。



这是源地址的变化序列，递增。



这是IP首部的标识符，随机。

1. http://www.linuxfoundation.org/collaborate/workgroups/networking/pktgen [↑](#footnote-ref-1)