Генератор пакетов, встроенный в ядро Linux

При тестировании сетей и отдельных устройств зачастую возникает потребность в генерации набора пакетов с заданными характеристиками. Существует достаточно много программ для генерации пакетов, но большинство таких программ работает на уровне пользовательских приложений, что может существенно ограничивать создаваемый генератором поток трафика. Современные версии ядра Linux (начиная с версии 2.4) включают генератор пакетов, позволяющий создавать и передавать в сеть Ethernet пакеты без использования дополнительных программ. Все операции выполняются на уровне ядра, что позволяет генерировать поток пакетов практически с полной скоростью сетевой среды.

Для использования генератора пакетов потребуется ядро со включенной при компиляции опцией **Packet Generator** (**NET_PKTGEN**). Если функции генератора пакетов реализованы в виде модуля (**NET_PKTGEN=m**) для работы с генератором нужно загрузить модуль с помощью команды

```
modprobe pktgen
```

После загрузки модуля (или сразу после загрузки ОС, если генератор встроен в ядро) в дереве **/proc** появится каталог **/proc/net/pktgen**, содержащий файлы **pg** с номерами от 0 до n-1 (n - число ядер) и n файлов **pb_busy** с идентичными номерами. Каждая пара файлов соответствует одному процессу генерации пакетов (будем называть такие процессы “виртуальными генераторами”). Процессы могут активизироваться параллельно, что позволяет создать множество управляемых потоков трафика. Файлы **pg*** содержат сведения о номере версии генератора, параметрах генерации пакетов и результатах предыдущего цикла , как показано ниже

```
pktgen version 1.33
Params: count 100000  pkt_size: 60  frags: 0  ipg: 0  clone_skb: 0 odev ""
     dst_min:   dst_max:   src_min:   src_max:
     src_mac: 00:00:00:00:00:00  dst_mac: 00:00:00:00:00:00
     udp_src_min: 9  udp_src_max: 9  udp_dst_min: 9  udp_dst_max: 9
     src_mac_count: 0  dst_mac_count: 0
     Flags:
Current:
     pkts-sofar: 0  errors: 0
     started: 0ms  stopped: 0ms  now: 1105536041221ms  idle: 0ns
     seq_num: 0  cur_dst_mac_offset: 0  cur_src_mac_offset: 0
     cur_saddr: 0x0  cur_daddr: 0x0  cur_udp_dst: 0  cur_udp_src: 0
Result: Idle
```

Отметим, что в приведенном примере не содержится результатов и указаны принятые по умолчанию значения, поскольку содержимое файла выведено до использования виртуального генератора.

Файлы **pb_busy** содержат просто флаги использования для каждого из виртуальных генераторов 0 – 7. Значение 1 в файле говорит о том, что соответствующий генератор в данный момент используется. Наличие файлов с флагами позволяет создавать shell-сценарии для длительной генерации пакетов с изменяющимися параметрами.

Встроенный в ядро или загружаемый как модуль код генератора пакетов по сути является интерпретатором простого языка сценариев, позволяющего задать параметры генерации пакетов независимо для каждого процесса. Язык описания параметров генерации достаточно прост и включает незначительное число команд, описанных ниже.

Параметр	*Описание*
clone_skb <целое число>	Определяет число буферов **skb** (“копия” пакета, создаваемая в ядре), создаваемых для генерации пакетов. При нулевом значении параметра (используется по умолчанию) буферы выделяются для каждого генерируемого пакета, что позволяет использовать в пакетах временные метки. Значение, отличное от нуля задает передачу соответствующего числа пакетов с использованием одного буфера **skb**.
count <целое число>	Этот необязательный параметр задает количество передаваемых пакетов. Нулевое значение счетчика обеспечивает генерацию неограниченного числа пакетов (процесс генерации продолжается до тех пор, пока не будет получена команда **stop**). По умолчанию программа генерирует 100000 пакетов.
dst X.X.X.X	Этот параметр является обязательным и задает IP-адрес получателя пакетов.
dst_min X.X.X.X	Задает нижнюю границу диапазона адресов получателей. По умолчанию используется значение параметра **dst**.
dst_max X.X.X.X	Задает верхнюю границу диапазона адресов получателей.
dstmac <MAC-адрес>	Устанавливает MAC-адрес получателя. По умолчанию используется значение 00:00:00:00:00:00. При использовании генератора в коммутируемой сети следует явно задавать MAC-адрес получателя или указывать в этом поле широковещательный адрес, поскольку в противном случае пакеты не уйдут дальше коммутатора. Если же адресат пакетов находится в другой локальной сети (за маршрутизатором), в этом поле следует указывать MAC-адрес интерфейса маршрутизатора, через который будут доставляться пакеты.
dst_mac_count <целое число>	Задает количество MAC-адресов получателей в используемом при генерации пакетов диапазоне с начальным значением **dstmac**.

Параметр	*Описание*
flag [имя]	Задает флаг управления генерацией. Возможны значения флагов: **IPSRC_RND** - использовать случайный адрес IP для отправителя; **IPDST_RND** - использовать случайный адрес IP для получателя; **UDPSRC_RND** - использовать случайный номер порта UDP для отправителя; **UDPDST_RND** - использовать случайный номер порта UDP для получателя; **MACSRC_RND** - использовать случайный MAC-адрес для отправителя; **MACDST_RND** - использовать случайный MAC-адрес для получателя. Адреса выбираются и номера портов из диапазона, заданного минимальным и максимальным значением. При установке соответствующего флага выбор значений осуществляется случайным способом, при отсутствии флага значения перебираются по порядку.
ipg <целое число>	Задает интервал между пакетами в наносекундах. При нулевом значении параметра (используется по умолчанию) интервал между последовательными пакетами определяется загрузкой процессора и его тактовой частотой. Максимальное значение параметра ipg может составлять 4294967295 (2^32), что обеспечивает генерацию пакетов с интервалом 4.295 сек. При попытке указать значение за пределами этого диапазона будут считаны первые 10 знаков числа (слева направо) и произведено преобразование к типу unsigned long (32 бита) с возможной потерей старших битов.
odev <имя устройства>	Задает имя интерфейса, используемого для генерации пакетов. Параметр является обязательным даже при наличии в системе единственного интерфейса.
pkt_size <целое число>	Задает размер генерируемых кадров Ethernet в байтах без учета поля FCS. По умолчанию используется значение pkt_size=ETH_ZLEN=60.
src_min X.X.X.X	Задает IP-адрес отправителя или нижнюю границу диапазона адресов.
src_max X.X.X.X	Задает верхнюю границу диапазона IP-адресов отправителей.
srcmac <MAC-адрес>	Устанавливает MAC-адрес отправителя. По умолчанию устанавливается MAC-адрес интерфейса, используемого для передачи пакетов.
src_mac_count <целое число>	Задает количество MAC-адресов отправителя в используемом диапазоне с начальным значением **srcmac**.
stop	Останавливает генерацию пакетов. Этот параметр передается без кавычек.
udp_src_min <целое число>	Задает минимальный номер порта UDP для отправителя. По умолчанию используется порт 9 (discard).
udp_src_max <целое число>	Задает максимальный номер порта UDP для отправителя. По умолчанию используется порт 9 (discard).
udp_dst_min <целое число>	Задает минимальный номер порта UDP для получателя. По умолчанию используется порт 9 (discard).
udp_dst_max <целое число>	Задает максимальный номер порта UDP для получателя. По умолчанию используется порт 9 (discard).

Для генерации пакетов нужно создать shell-сценарий, который содержит приведенные ниже строки:

```
#! /bin/sh
# загрузка модуля генерации пакетов
modprobe pktgen

# укажите "виртуальный генератор" pgN (N = 0 .. 7) в переменной PGDEV
PGDEV=/proc/net/pktgen/pg7       # будем использовать виртуальный генератор 7

function pgset() {
    local result
    echo $1 > $PGDEV
    result=`cat $PGDEV | fgrep "Result: OK:"`
    if [ "$result" = "" ]; then
         cat $PGDEV | fgrep Result:
    fi
}

function pg() {
    echo inject > $PGDEV
    cat $PGDEV
}

#поместите ниже команды управления генерацией
```

и команды установки параметров генерации, каждая из которых имеет вид:

```
pgset "параметр значение"
```

Пары параметр-значение следует заключать в кавычки. После команд установки параметров следует включить команду pg, инициирующую процесс генерации. В качестве примера ниже приведен набор команд для генерации пакетов в адрес хоста 172.16.33.2, который находится в другом сегменте сети и подключен через коммутатор сетевого уровня (L3).

```
pgset "odev eth0"                 # будем передавать пакеты в сеть через интерфейс eth0
pgset "count 1000000"             # задаем число генерируемых пакетов
pgset "dst 172.16.33.2"           # указываем IP-адрес получателя пакетов
pgset "dstmac 00:05:5d:00:33:44" # указываем MAC-адрес интерфейса маршрутизатора
pg                                # начать генерацию пакетов
```

После создания файла сценария сохраните его и установите для него возможность записи (сделать это можно, например, с помощью команды **chmod 755 <имя файла>**). Для простоты будем называть такие файлы по именам виртуальных генераторов (например, **pg7.sh**). Далее в командной строке набираем

```
./pg7.sh
```

и наблюдаем за процессом. На хосте, которому направляется поток пакетов также разумно включить ту или иную программу сбора пакетов (например, **tcpdump**).

После генерации пакетов на экран будет выведена статистика работы генератора. Эти же параметры и результаты будут сохранены в файле **/proc/net/pktgen/pg0** и могут использоваться в качестве принятых по умолчанию параметров при следующем запуске генератора для того же **pg**. Пример вывода для рассмотренного сценария генерации приведен ниже

```
pktgen version 1.33
Params: count 1000000  pkt_size: 60  frags: 0  ipg: 0  clone_skb: 0 odev "eth0"
     dst_min: 172.16.33.2  dst_max:   src_min: 192.168.77.33  src_max:
     src_mac: 00:A0:CC:60:6F:B7  dst_mac: 00:05:5D:00:33:44
     udp_src_min: 9  udp_src_max: 9  udp_dst_min: 9  udp_dst_max: 9
     src_mac_count: 0  dst_mac_count: 0
     Flags:
Current:
     pkts-sofar: 1000000  errors: 0
     started: 1105547763535ms  stopped: 1105547777329ms  now: 1105547777329ms  idle:
1952323633ns
     seq_num: 1030000  cur_dst_mac_offset: 0  cur_src_mac_offset: 0
     cur_saddr: 0x21c830d4  cur_daddr: 0x22110ac  cur_udp_dst: 9  cur_udp_src: 9
Result: OK: 13790262(c10040403+d3749859) usec, 1000000 (64byte,0frags)
  72514pps 37Mb/sec (37127168bps) errors: 0
```

На хосте 172.16.33.2 для мониторинга была использована команда

```
tcpdump -n -vvv host 172.16.33.2
```

фрагмент вывода которой приведен ниже

```
19:36:00.609255 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609280 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609304 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609327 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609349 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609372 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609394 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
19:36:00.609418 192.168.77.33.9 > 172.16.33.2.9:  [no cksum] udp 18 (ttl 2, id 23130, len 46)
```

Мы не задавали интервал между пакетами и используемое по умолчанию значение 0 обеспечило их генерацию с интервалом около 25 мксек, как можно видеть из приведенных результатов. Размер пакетов также не был указан, поэтому генерировались пакеты минимального размера 60 байт. При генерации в заголовках IP устанавливается значение TTL=3, что полностью соответствует результатам мониторинга. В принятых пакетах TTL=2, поскольку между генератором и получателем находится коммутатор сетевого уровня (маршрутизатор). В качестве номера порта использовались принятые по умолчанию значения 9 (порт службы discard).

Усложним сценарий генерации пакетов, добавив диапазоны значений для адреса отправителя и номеров портов отправителя и получателя, а также зададим случайный характер выбора значений. Укажем также размер кадров (1400 байт) и интервал между генерируемыми пакетами (50 нсек). Фрагмент сценария с установкой параметров пакетов показан ниже

```
pgset "odev eth0"
pgset "pkt_size 1400" # размер кадра
pgset "count 1000000"
pgset "ipg 50" # интервал между пакетами
pgset "dst 172.16.33.2"
pgset "dstmac 00:05:5d:00:33:44"
pgset "src_min 192.168.77.1"
pgset "src_max 192.168.77.254"
pgset "udp_src_min 1"
pgset "udp_src_max 1024"
pgset "udp_dst_min 1"
pgset "udp_dst_max 1024"
pgset "flag IPSRC_RND"
pgset "flag UDPSRC_RND"
pgset "flag UDPDST_RND"
pg
```

Фрагмент вывода tcpdump на хосте 172.16.33.2 показывает

```
20:06:57.617840 192.168.77.225.743 > 172.16.33.2.945:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.617889 192.168.77.191.662 > 172.16.33.2.720:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618033 192.168.77.103.726 > 172.16.33.2.378:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618097 192.168.77.140.79 > 172.16.33.2.948:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618145 192.168.77.2.828 > 172.16.33.2.239:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618197 192.168.77.79.1003 > 172.16.33.2.195:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618240 192.168.77.49.27 > 172.16.33.2.837:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618576 192.168.77.85.969 > 172.16.33.2.940:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618619 192.168.77.169.430 > 172.16.33.2.865:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618662 192.168.77.79.97 > 172.16.33.2.130:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618705 192.168.77.66.181 > 172.16.33.2.355:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618930 192.168.77.45.710 > 172.16.33.2.214:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.618976 192.168.77.213.594 > 172.16.33.2.155:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.619305 192.168.77.250.692 > 172.16.33.2.665:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.619355 192.168.77.93.853 > 172.16.33.2.99:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.619398 192.168.77.242.609 > 172.16.33.2.520:  [no cksum] RIPv155-#190 1358 (ttl 2, id 23130, len
1386)
20:06:57.619448 192.168.77.8.14 > 172.16.33.2.577:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.619490 192.168.77.144.306 > 172.16.33.2.335:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.619667 192.168.77.175.141 > 172.16.33.2.30:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.620138 192.168.77.47.558 > 172.16.33.2.87:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.620183 192.168.77.143.293 > 172.16.33.2.236:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
20:06:57.620225 192.168.77.72.670 > 172.16.33.2.813:  [no cksum] udp 1358 (ttl 2, id 23130, len 1386)
```

Из приведенных значений видно, что действительно генерировались кадры размером 1400 байт, содержащие пакеты IP и поле FCS общим размером 1386 байт (14 байт занимает заголовок канального уровня), в которых содержались дейтаграммы UDP размером 1358 байт (24 байта занимает заголовок IP и 4 байта поле FCS канального уровня). Адреса отправителя, а также номера портов отправителя и получателя случайным образом изменялись в заданном диапазоне.

Результаты работы генератора показаны ниже

```
pktgen version 1.33
Params: count 1000000  pkt_size: 1400  frags: 0  ipg: 50  clone_skb: 0 odev "eth0"
     dst_min: 172.16.33.2  dst_max:   src_min: 192.168.77.1  src_max: 192.168.77.254
     src_mac: 00:A0:CC:60:6F:B7  dst_mac: 00:05:5D:00:33:44
     udp_src_min: 1  udp_src_max: 1024  udp_dst_min: 1  udp_dst_max: 1024
     src_mac_count: 0  dst_mac_count: 0
     Flags: IPSRC_RND  UDPSRC_RND  UDPDST_RND
Current:
     pkts-sofar: 999998  errors: 0
     started: 1105549619362ms  stopped: 1105549762716ms  now: 1105549762716ms  idle:
118927868357ns
     seq_num: 2469997  cur_dst_mac_offset: 0  cur_src_mac_offset: 0
     cur_saddr: 0xefc830d4  cur_daddr: 0x22110ac  cur_udp_dst: 241  cur_udp_src: 706
Result: OK: 143321561(c139571702+d3749859) usec, 999998 (1404byte,0frags)
  6977pps 78Mb/sec (78365664bps) errors: 0
```

Как вы можете видеть, в данном случае поток трафика составил 78 Мбит/с, что для устройства 100Base-TX достаточно хороший результат. В зависимости от настройки параметров мне удавалось получать значения до 90 Мбит/с для одного процесса генерации пакетов.

Как уже было сказано выше, вы можете создать множество сценариев генерации для одного или нескольких “виртуальных генераторов” pg и запускать их последовательно или одновременно (до 8) вручную или с помощью дополнительных shell-сценариев, выбирающих скрипты запуска генерации. Таким образом можно протестировать различные конфигурации хостов, маршрутизаторов, коммутаторов для проверки производительности их работы. Генератор пакетов достаточно эффективно можно использовать и при тестировании систем обеспечения безопасности (пакетные фильтры, IDS, средства обнаружения сканеров, системы мониторинга трафика и т. п.). Ваши возможности тестирования систем канального и сетевого уровня, а также работающих по протоколу UDP сетевых приложений реально могут быть ограничены только вашей фантазией.

Опишу здесь один интересный и не совсем очевидный вариант использования генератора для прецизионного измерения вариаций задержки в сети. При установке параметра **clone_skb=0** для каждого генерируемого пакета создается новый буфер skb и в поле данных пакета UDP включается временная метка. Метки содержат время в формате timeval, принятом в Linux. Первая пара байтов метки содержит время генерации буфера в секундах, вторая – дробную часть момента генерации в микросекундах. Найти эту метку можно по смещению - +8 от начала поля данных UDP, +16 от начала заголовка UDP, +36 от начала заголовка IP или +50 от начала кадра Ethernet. Как уве было сказано, временная метка содержит 8 байтов, из которых 4 байта с меньшим смещением содержат число секунд, а 4 байта с большим смещением – число микросекунд в момент генерации буфера skb. Эти сведения в комбинации с временными параметрами, полученными при сборе пакетов позволяют весьма точно измерить вариации задержки при доставке пакетов через сеть. Если же часы в исходной и конечной точках синхронизированы между собой, вы можете с высокой точностью измерять не только вариации задержки, но и ее абсолютные значения.

Указывается в пакетах и порядковый номер. Нумерация ведется независимо для каждого pg, причем при активизации нового теста (без перезагрузки модуля pktgen) нумерация продолжается с того значения, которое было достигнуто в предыдущем сеансе для этого pg. Наличие порядковых номеров позволяет использовать пакеты для тестирования систем со множеством путей доставки, проверяя сохранность порядка доставки при использовании разных путей. Порядковый номер (4 байта) расположен перед временными метками (+4 от начала поля данных UDP). Первые 4 байта поля данных занимает сигнатура генератора 0xbe9be955, которую также можно использовать в программах анализа для идентификации полей порядкового номера и временных меток.

Если же установить для параметра clone_skb отличное от нуля значение, на основе каждого буфера skb будет создаваться заданное параметром количество идентичных пакетов UDP, в которых значения номера и временных меток будут совпадать. При создании нового буфера значение номера будет увеличиваться в таких случаях не на 1, а на количество копий буфера, используемых для генерации пакетов (т. е., счетчик пакетов будет продолжать работу, увеличив лишь размер шага)

При работе с генератором следует учитывать некоторые специфические для реализации аспекты, перечисленные ниже.

1. Генерируемые пакеты передаются ядром непосредственно в устройство (точнее буфер skb), поэтому большинство средств сетевого мониторинга на локальном компьютере просто не сможет видеть этих пакетов.
2. Пакеты генерируются с малым временем жизни (TTL = 3), что ограничивает дальность их распространения. Это сделано осознанно в целях ограничения области распространения генерируемых пакетов. Вы можете изменить это значение, внеся изменения в исходный код модуля и собрав заново модуль. Можно поменять значение TTL в пакетах и другими способами, но следует быть осторожным, поскольку при неаккуратном обращении выпущенные за пределы вашей сети пакеты могут доставить кому-либо серьезные неприятности.
3. По умолчанию кадры Ethernet, содержащие созданные генератором пакеты, имеют в поле MAC-адреса получателя значение 00:00:00:00:00:00, поэтому в коммутируемой сети эти пакеты просто не попадут к получателю. Для задания MAC-адреса получателя служат параметры **dstmac**, **dst_mac_count** и **flag**, описанные выше. Напомним еще раз, что при наличии между генератором и адресатом маршрутизаторов и иных устройств, обрабатывающих пакеты на сетевом уровне, в качестве MAC-адреса получателя следует указывать значение адреса интерфейса такого устройства.
4. При указании диапазона номеров порта для отправителя или получателя следует различать два случая:
   - если значение **udp_*_min** >= **udp_*_max**, пакеты генерируются с номерами портов от **udp_*_min** до **udp_*_max**, включая граничные точки [**udp_*_min**, **udp_*_max**];
   - если **udp_*_min** < **udp_*_max**, используются номера портов из диапазонов [0, **udp_*_max**) и (**udp_*_min**, 65535].
5. Значения некоторых параметров, устанавливаемых при помощи команды **pgset** в целях упрощения не проверяются на этапе установки. Для обеспечения корректной работы генератора параметры проверяются в процессе генерации пакетов с приведением значений к заданному типу в соответствии с правилами преобразования языка C. Поэтому при вводе слишком больших значений для некоторых параметров (в частности, ipg) вы просто будете терять старшие двоичные разряды заданного значения. Например, задав для ipg значение 4294967296 (2^32), вы получите на самом деле **ipg=0**.
6. Если при повторном запуске генератора с тем же номером **pg** тот или иной параметр не был установлен с помощью команды **pgset**, при генерации пакетов будет использовано предыдущее значение данного параметра, которое сохранено в файле **/proc/net/pktgen/pg***.

В заключение следует отметить, что входящий в состав ядра исходный код генератора (**net/core/pktgen.c**) и файл документации (**Documentation/networking/pktgen.txt**) содержит целый ряд ошибок. Обнаруженные ошибки исправлены и корректный исходный код генератора пакетов будет включен в новые версии ядра. При желании вы сможете загрузить исправленный файл, воспользовавшись приведенной ссылкой http://bugme.osdl.org/show_bug.cgi?id=4037. Исправленный вариант обеспечивает корректную установку и обработку значений всех параметров в соответствии с приведенной в данной статье и документации модуля информацией. Отметим также, что включенный в официальное ядро исходный код генератора можно использовать, но в некоторых ситуациях могут возникать проблемы с установкой желаемых параметров, что затруднит проведение тестов с помощью этого генератора.

Николай Малых

nmalykh@protocols.ru