Документация к лабораторной работе №1.

Сбор статистики по сетевому трафику.

Степанова Екатерина

stepkate125@gmail.com

Цель задачи: создать набор программного обеспечения, который мог бы собирать и отображать статистику по трафику на заданном сетевом интерфейсе.

Требования:

1. ПО должно работать на ПК под управлением Debian GNU\Linux (версии 10 и новее)
2. Для реализации использовать язык программирования C.
3. Сборка должна осуществляться GNU Toolchain
4. Дистрибуция должна осуществляться при помощи deb-пакета
5. Сбор статистики должен вестись только по входящим UDP пакетам
6. Должна быть реализована возможность указывать конкретные параметры учитываемых в статистике пакетов:
    1. IP-адрес источника
    2. IP-адрес назначения
    3. Порт источника
    4. Порт назначения
7. В статистике должно присутствовать количество принятых пакетов и суммарное количество байт в этих пакетах

ПО организовать в виде двух отдельных утилит: первая читает данные с сетевого интерфейса и собирает статистику по пакетам, вторая при запуске получает собранную статистику у первой утилиты и выводит её на экран.

Утилита для сбора статистики. Нужно реализовать 2 варианта данной утилиты:

1. Два потока (pthread): первый читает пакеты при помощи Raw Socket (OSI L2) с интерфейса, проверяет параметры пакета и для подходящих по заданным параметрам, передаёт статистику во второй поток. Второй суммирует статистику и отдаёт её по запросу извне.

2. Два потока (pthread): первый читает пакеты при помощи Raw Socket (OSI L2) с интерфейса, проверяет параметры пакета и для подходящих по заданным параметрам, суммирует статистику. Второй отдаёт её по запросу извне.

Самостоятельно провести профилирование обоих вариантов, оценить какой вариант эффективнее, с каким вариантом можно обеспечить большую пропускную способность.

В обоих вариантах: передача данных между потоками осуществляется любым способом, на усмотрение разработчика.

Утилита для вывода статистики на экран: запрашивает статистику у первой утилиты через ubus или через POSIX Message Queues. Рекомендация: попробовать реализовать оба варианта.

Программное обеспечение должно сопровождаться документацией, содержащей следующие разделы:

1. Описание – общая информация, что и как делает ПО
2. Сборка – инструкции по сборке ПО из исходников: что установить в систему, какой командой запустить сборку, что должно получиться в итоге
3. Запуск – как запустить ПО, как подать трафик на интерфейс, чтобы убедиться в корректности работы, что пользователь программ должен увидеть на экране
4. Результаты профилирования двух реализованных вариантов утилиты для сбора статистики
5. Авторство и лицензия – указать имя и электронную почту автора, указать лицензию

Результат работы: архив git репозитория, содержащего исходники ПО и сопроводительную документацию.

Выполнение:

1) Описание

ПО состоит из двух утилит. Первая собирает UDP пакеты, а вторая выводит их кол-во и их общее кол-во байт на экран. Было реализовано два варианта перовой утилиты.

Первая утилита (собирает статистику по каждому пакету, суммирует ее и передает во второй поток):

В начале функции *main* глобальным переменным *iface* и *parm* присваиваются значения сетевого интерфейса и параметрам, по котором будет собираться статистика, соответственно. Далее идет инициализация *thread* потоков и их запуск. Они работают параллельно.

Первый поток открывает *raw* сокет, происходит привязывания этого сокета к определенному интерфейсу и далее в бесконечной цикле сокет начинает принимать пакеты через *recvfrom*. Когда он получает пакет, сокет проверяет является ли он UDP пакетом, если да, то дальше идет проверка по параметрам. Это осуществимо благодаря структурам: *iphdr*, где мы можем посмотреть ip адрес источника и назначения, и udphdr, который предоставляем нам доступ к портам источника и назначения, а также дает информацию о количестве байт в каждом пакете.

Статистика собирается с помощью двух глобальных счетчиков: *packets* и *packets_len.* Если какой-то заданный параметр совпадает с характеристиками пакета, то счетчики считывают информацию и отправляют ее во второй поток. Запись статистики защищено средством синхронизации *mutex*.

Второй поток в начале открывает очереди *mq_server* и *mq_client* для передачи статистики второй утилите. Далее в бесконечном цикле отправляет полученную статистику в очередь.

Чтобы завершить первую утилиту, нужно нажать CTRL + C.

Первая утилита (собирает статистику по каждому пакету и передает во второй поток, который потом суммирует эту статистику):

В общем и целом работает все также, единственное отличие - это то, что сбор статистики происходит не с помощью двух глобальных счетчиков, а с помощью кастомного вектора *vec*, который содержит в себе кол-во байт каждого полученного пакета.

Вторая утилита:

Утилита состоит из всего одной функции *main.* В самом начале он отрывает очередь *mq_server*, в бесконечном цикле считывает сообщения из Message Queue от первой утилиты и выводит их на экран.

Чтобы завершить вторую утилиту, нужно нажать CTRL + C.

2) Сборка

Для сборки только утилит:

```
$ make
```

Чтобы собрать ПО, в систему нужно установить:

```
$ sudo apt-get update
```

```
$ sudo apt-get install dpkg-dev devscripts wget
```

Для запуска сборки можно воспользоваться командой:

```
$ dpkg-buildpackage -uc -us
```

Либо

```
$ make deb
```

В результате у нас получился deb-пакет, и чтобы его установить, можно воспользоваться следующей командой:

```
$ sudo dpkg -i <package_name>.deb
```

3) Запуск

Сначала запускаем вторую утилиту:

```
$ ./util2
```

Первую утилиту(двух вариантов) запускаем от root, также чтобы ее запустить, нужно передать определенные параметры: <сетевой интерфейс> <IP-адрес источника> <IP-адрес назначения> <Порт источника> <Порт назначения>

Это может выглядеть так:

```
$ ./util1_1 eth0 192.168.1.69 192.168.1.100 23 10
```

Если какие-то параметры не учитываются, то они обозначаются нулями, например учитывать ip адрес источника не надо:

```
$ ./util1_1 eth0 0 192.168.1.100 23 10
```

А если вообще никакие параметры не должны учитываться в статистике, то запуск утилиты будет выглядеть так:

```
$ ./util1_1 eth0 0 0 0 0
```

Запуск второй утилиты:

```
mordekay@mordekay:~/test$ ./util2
Total packets: 1         Total bytes: 224
^C
```

Запуск первой утилиты:

```
root@mordekay:/home/mordekay/test# ./util1_1 enp0s3 0 0 0 0
Reciveing packets...
^C
```

4) Результаты профилирования двух реализованных вариантов утилиты для сбора статистики

Вариант первой утилиты, которая просто собирает статистику и отправляет ее во второй поток на суммирование является более эффективным, чем тот вариант, где первый поток суммирует статистику. Именно с этим вариантом можно обеспечить большую пропускную способность, так как в этом случае статистика собирается атомарно, а в другом случае два отдельных действия.

5) Авторство и лицензия

Степанова Екатерина

stepkate125@gmail.com