

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний педагогічний університет
імені Володимира Гнатюка

Кафедра інформатики та
методики її викладання

ІНДЗ

Управління маршрутизацією ОС Linux засобами iproute2

Виконав:

Студент групи І - 24

Мандебура Андрій

Науковий керівник:

Олексюк В. П.

Тернопіль – 2013

ЗМІСТ

Оглавление

| | |
|---|----|
| ЗМІСТ..... | 1 |
| ВСТУП..... | 3 |
| РОЗДІЛ 1 | 4 |
| 1.1. Налаштування пакету iproute2..... | 4 |
| 1.2. Основні команди утиліти iproute2..... | 5 |
| 1.3. Загальні приклади використання команд | 8 |
| 1.4. Використання маршрутизації на основі правил..... | 10 |
| ВИСНОВКИ..... | 11 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 12 |
| ДОДАТКИ..... | 13 |

ВСТУП

Напевно кожен з нас хоча б віддалено знає, що таке маршрутизація. Отже маршрутизація – як би це банально не звучало, є вибором маршруту. У нашому випадку це процес вибору маршруту прямування інформації між мережами.

У цій роботі я описую управління маршрутизацією у ОС Linux за допомогою утиліти `iproute2`.

Більшість дистрибутивів Linux, як і більшість ОС UNIX, в даний час використовують досить древні утиліти `arp`, `ifconfig` та `route`. Поки що ці інструменти працюють досить адекватно, але іноді, на ядрах Linux версії 2.2 і вище, вони можуть вести себе досить несподівано.

Мережева підсистема, в ядрах 2.2 і вище, була повністю переписана. Новий мережевий код дав збільшення продуктивності і більш високі експлуатаційні характеристики, що робить Linux ще більш привабливим на ринку операційних систем.

Фактично, реалізація мережевої підсистеми в Linux, що виконує класифікацію, маршрутизацію і фільтрацію, виявилася навіть більш повною, ніж у спеціалізованих маршрутизаторах, між мережевих екранах і інших пристроях управління трафіком.

У міру появи нових розробок, вони "нашаровувалися" поверх існуючих реалізацій в існуючих операційних системах. Це постійне нашаровування призвело до того, що код, який вирішував завдання управління мережевим трафіком, часом проявляв досить дивну поведінку.

Ця, заново переписана, реалізація мережевої підсистеми дозволила досягти таких характеристик, які раніше були просто недоступні.

Linux має досить складну систему управління пропускну здатністю, названою TrafficControl (Керування Трафіком). Вона підтримує різні методи класифікації, поділу за пріоритетами, спільного використання, і обмеження як вхідного, так і вихідного трафіку.

РОЗДІЛ 1

Маршрутизація буває статичною та динамічною . Відмінність в тому , що при статичній маршрутизації всі правила передачі пакетів прописуються статично і можуть бути змінені тільки вручну , динамічна маршрутизація застосовується , коли в мережі існує кілька маршрутизаторів , і знаходження шляху до віддаленого хоста стає нетривіальним завданням . Динамічна маршрутизація більше підходить для часто мінливих мереж зі складною структурою. Хоча GNU / Linux підтримує обидва типи маршрутизації , ми розглядатимемо статичну маршрутизацію за допомогою пакету `iproute2`

1.1. Налаштування пакету `iproute2`

Ми розпочнемо розгляд нашої теми з короткого огляду `iproute2` і її можливостей.

Перед початком роботи нам потрібно відповідним чином налаштувати ядро.

У ядрі необхідно включити ряд опцій маршрутизації. На сторінці `NetworkingOptions` необхідно включити наступні елементи:

IP: `advancedrouter` - включення розширених можливостей маршрутизації;

IP: `policyrouting` - маршрутизація по деяких внутрішніх полях пакетів (зазвичай застосовується спільно з брандмауером), а також для розширених можливостей маршрутизації, наприклад, маршрутизація згідно адресі-джерелу пакету (`source-based routing`);

IP: `usenetfilter MARK valueasroutingkey` - включення можливості маршрутизації згідно з маркуванням пакету брандмауером;

IP: `use TOS valueasroutingkey` - маршрутизація пакетів на основі заголовка тип сервісу (TOS), допомагає збільшити пропускну спроможність мережі за наявності декількох шляхів проходження пакетів;

IP: `largeroutingtables` - включення великих (>64 правих) таблиць маршрутизації ядра.

Можна також включити підтримку тунелів. Після налаштування ядра необхідно встановити `iproute2`. У більшості дистрибутивів GNU/Linux ця програма входить в дистрибутив, наприклад, для Debian GNU/Linux команда виглядатиме так:

```
# apt-getinstall iproute
```

Пакет `iproute` складається фактично з двох утиліт управління трафіком:

`ip` - управління власне маршрутизацією;

`tc` - управління чергами маршрутизації.

1.2. Основні команди утиліти `iproute2`

Для початку розглянемо основні принципи команди `ip`:

Синтаксис команди такий:

```
ip [опції] { об'єкт маршрутизації } { команда або HELP }
```

Одними з найважливіших є команди вибору сімейства IP:

-4 - IPv4;

-6 - IPv6.

Об'єкти маршрутизації представлені наступним списком:

- link - мережевий пристрій (реальне фізичне або віртуальне, наприклад, `vlan` або тунель);
- address - ip-адреса пристрою;
- neighbour - кеш ARP;
- route - таблиці маршрутизації;
- rule - правила маршрутизації;
- maddress - ширококомовна адреса;
- mroute - ширококомовні таблиці маршрутизації;

- tunnel - IP-тунель.

Команди для різних об'єктів різні, але для всіх об'єктів існує стандартний набір команд: add (додати), delete (видалити) і show (показати; можна також list або ls). Розглянемо деякі з цих команд:

iplink - конфігурація мережевого пристрою. (show або list показують доступні мережеві пристрої)

iplinkset - зміна параметрів мережевого пристрою.

iplinkshow (list, ls, sh, lst, l) - показує інформацію про мережевий інтерфейс. (рис.1)

Та аргументи команд:

dev– [Ім'я] інтерфейсу з яким проводитимемо якісь маніпуляції.

up (включити) або down (вимкнути) - включити або вимкнути мережевий інтерфейс.

arp on або arpo ff - змінює значення прапора NOARP на пристрої.

multicast on або multicast off - змінює прапор MULTICAST на пристрої.

dynamic on або dynamic off - змінює прапор DYNAMIC на пристрої.

name - Змінює ім'я пристрою

txqueuelen[Число] або txqlen[Число] - змінює довжину передаваної черги.

mtu[Число] - змінює значення MTU на пристрої.

address[Адреса] - змінює адресу на пристрої.

broadcast[Адреса] або brd[Адреса] - змінює широкомовну адресу на пристрої.

Наведу декілька прикладів використання цих команд:

Також досить важлива для нашого завдання команда address (addr, a). Команда ipaddress показує адреси, їх властивості, а так само додає нові або видаляє старі.

ipaddressadd - додає нову адресу.

ipaddressdelete - видаляє адреси.

ipaddressshow - виводить інформацію про адресу.

Аргументи команди:

Dev[Ім'я] - ім'я пристрою.

local[Адреса] - адреса інтерфейсу.

peer[Адреса] - адреса видаленої машини при використанні протоколу PPP.

broadcast[Адреса] - ширококомвна адреса на інтерфейсі.

Також важлива команда:

iproute (route, ro, r) – керування таблицею маршрутизації.

iprouteadd - додати новий маршрут

iproutechange - змінити маршрут

iproutereplace - замінити маршрут

iproutedelete - видалити маршрут.

Аргументи:

to - призначення маршруту.

metric [Число] - завдання метрики маршруту.

table [Ідентифікатор таблиці] - таблиця пов'язана з маршрутом. Ідентифікатором таблиці може бути номер або рядок з файлу /etc/iproute2/rt_tables.

dev[Ім'я] - ім'я пристрою.

via[Адреса] - адреса переходу до наступного маршрутизатора

src[Адреса] - адреса джерела

nexthop- наступний перехід у випадку якщо використовується багатоканальна маршрутизація.

dev [Ім'я] - ім'я пристрою.

weight [Число] - вага маршруту, що визначається шириною каналу або якістю .

Маршрутизація може проводитися не тільки залежно від адреси одержувача, але і за адресою джерела, IP протоколу і транспортного протоколу.

За умовчанням в існують 3 правила:

Таблиця Local (ID 255) - спеціальна таблиця маршрутизації з найбільшим пріоритетом, яка містить таблиці для локальних і ширококомовних адрес.

Таблиця Main (ID 254) - звичайна таблиця маршрутизації.

Таблиця Default (ID 253) - порожня за умовчанням таблиця.

ipruleadd - додати нове правило.

Ipruledelete - видалити правило

Аргументи:

from - адреса джерело

to - адреса одержувача

iif[Ім'я] - ім'я інтерфейсу з якого буде отримано пакет

fwmark[Влучна] - мітка пакету, встановлювана firewall.

table[Ідентифікатор таблиці] - таблиця пов'язана з маршрутом. Ідентифікатором таблиці може бути номер або рядок з файлу /etc/iproute2/rt_tables

priority[Число]- пріоритет таблиці.

1.3. Загальні приклади використання команд

Розпочнемо з найпростіших команд.

Для отримання інформації про мережевий інтерфейс використовуємо команду:


```
# iplinklist
```

Для отримання інформації про стан інтерфейсу(наприклад eth3) використовуємо відповідно команду:

```
# ip link lsdev eth3
```

Виведемо статистику інтерфейсу eth3

```
# ip -s link lsdev eth3 (рис.1.1.)
```

Управління адресами на інтерфейсі

```
# ip address(рис.1.2.)
```

Виводимо інформацію про адресу:

```
# ip address show (рис 1.3.)
```

Для перегляду таблиць маршрутизації використовуємо команду:

```
# ip route list (рис.1.4.)
```

Для перегляду певної таблиці використовується та ж команда але з вказаною назвою таблиці:

```
# ip route list [table_name](на прикладі таблиці local) (рис.1.5.)
```

Утиліта ip route2 надає можливість створювати власні таблиці що буде описано пізніше.

Щоб додати маршрут до мережі використовуємо команду route add.

Наприклад додамо до мережі 10.0.0.0/8 через 193.222.8.75:

```
# ip route add 10.0.0.0/8 via 193.222.8.75
```

Додамо шлюз за умовчанням у разі використання двох каналів залежно від завантаження каналу:

```
# ip route add default scope global nexthop dev eth0
```

```
# nexthop dev eth0
```

Командою `iproutedelete` можна видалити створені маршрути:

```
#iproutedeldefaultscopeglobalnexthopdeveth0
```

```
#nexthopdeveth0
```

1.4. Використання маршрутизації на основі правил

Розглянемо приклади використання маршрутизації на основі правил.

Створимо нове правило. Хоча ми можемо працювати просто з числами, набагато простіше і зрозуміліше якщо ми визначимо назви наших таблиць у файлі `/etc/iproute2/rt_tables`.

```
# echo 200 User>> /etc/iproute2/rt_tables
```

```
# ipruleaddfrom 172.25.17.8 tableAndriy (рис.1.6.)
```

```
# iprulels (рис.1.7.)
```

Тепер нам потрібно лише згенерувати таблицю `Andriy` і очистити кеш маршрутів:

```
# iprouteadddefaultvia172.25.17.254deveth4tableAndriy
```

```
# iprouteflushcache(рис1.8.)
```

ВИСНОВКИ

Отже, підіб'ємо підсумок. Ми розглянули команди та способи їх застосування утиліти `iproute2`, також ми навели декілька прикладів керування маршрутизацією за допомогою цієї утиліти. Тому згідно з виконаною роботою можна зробити висновок, що для виконання статичної маршрутизації найкраще підходить `iproute2` для GNU / Linux. Маршрутизація дозволяє виконувати досить складні операції з передачі пакетів, при цьому можливо грамотно встановити політику доступу до певних підмереж і вузлів мережі. Одним з найбільш корисних у практичному плані інструментом оптимізації мережевих операцій є управління чергами пристроїв, але це уже інша тема. Для встановлення маршрутизатора не потрібно потужного комп'ютера, в деяких випадках досить floppy-дистрибутива Linux. Одним з таких дистрибутивів є орієнтований на маршрутизацію `linuxrouter`. Він побудований на базі ядра 2.2 і 2.0, що є прийнятним варіантом для побудови маршрутизатора. Якщо ж у вашій мережі декілька маршрутизаторів або структура мережі є непостійною, то кращим вибором є установка динамічного маршрутизатора, що має можливість автоматичного оновлення маршрутних таблиць.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Настройка маршрутизации за допомогою іproute. [Електронний ресурс]. - <http://nodeny.dp.ua/rozpodilennya-trafiku/nastrojka-marshrutyzaciji-za-dopomogoyu-iproute/>
2. Настройка маршрутизации с помощью іproute.[Електронний ресурс].- <http://www.nixp.ru/articles/%D0%9D%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-%D0%BC%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8-%D1%81-%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D1%8C%D1%8E-iproute.html>
3. Практическоеруководство по применениюіproute2 для управления трафиком.[Електронний ресурс].- <http://docstore.mik.ua/manuals/ru/LARTC/x213.html>

ДОДАТКИ

```

root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip -s link ls dev eth3
2: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether e0:cb:4e:d6:33:02 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped  overrun  mcast
    5981244   12093    0       6        0        0
    TX: bytes  packets  errors  dropped  carrier  collsns
    573257    5000     0       0        0        0
root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip -s link ls dev eth4
3: eth4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 1c:af:f7:6f:37:dd brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped  overrun  mcast
    24903     324      0       0        0        0
    TX: bytes  packets  errors  dropped  carrier  collsns
    41269     187      0       0        0        0

```

рис.1.1

```

root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av
root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether e0:cb:4e:d6:33:02 brd ff:ff:ff:ff:ff:ff
    inet 172.25.17.108/24 brd 172.25.17.255 scope global eth3
    inet6 fe80::e2cb:4eff:fed6:3302/64 scope link
        valid_lft forever preferred_lft forever
3: eth4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 1c:af:f7:6f:37:dd brd ff:ff:ff:ff:ff:ff
    inet6 fe80::1eaf:f7ff:fe6f:37dd/64 scope link
        valid_lft forever preferred_lft forever

```

Рис 1.2

```

root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip addr ls eth3
2: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether e0:cb:4e:d6:33:02 brd ff:ff:ff:ff:ff:ff
    inet 172.25.17.108/24 brd 172.25.17.255 scope global eth3
    inet6 fe80::e2cb:4eff:fed6:3302/64 scope link
        valid_lft forever preferred_lft forever

```

Рис 1.3

```

root@lw-417-08: /home/likewise-open/WFIZMAT/mandebura_av
mandebura_av@lw-417-08:~$ sudo su
[sudo] password for mandebura_av:
root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip route show
default via 172.25.17.254 dev eth3 proto static
default via 172.25.17.254 dev eth3 metric 100
169.254.0.0/16 dev eth3 scope link metric 1000
172.25.17.0/24 dev eth3 proto kernel scope link src 172.25.17.108

```

Рис 1.4

```

root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip route list table local
broadcast 127.0.0.0 dev lo proto kernel scope link src 127.0.0.1
local 127.0.0.0/8 dev lo proto kernel scope host src 127.0.0.1
local 127.0.0.1 dev lo proto kernel scope host src 127.0.0.1
broadcast 127.255.255.255 dev lo proto kernel scope link src 127.0.0.1
broadcast 172.25.17.0 dev eth3 proto kernel scope link src 172.25.17.108
broadcast 172.25.17.0 dev eth4 proto kernel scope link src 172.25.17.201
local 172.25.17.108 dev eth3 proto kernel scope host src 172.25.17.108
local 172.25.17.201 dev eth4 proto kernel scope host src 172.25.17.201
broadcast 172.25.17.255 dev eth3 proto kernel scope link src 172.25.17.108
broadcast 172.25.17.255 dev eth4 proto kernel scope link src 172.25.17.201
root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av#

```

Рис 1.5

```

/iproute2/rt_tables
root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip rule add from 172.25.17.9

```

Рис 1.6

```

root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip rule ls
0:      from all lookup local
32765:  from 172.25.17.9 lookup main
32766:  from all lookup main
32767:  from all lookup default

```

Рис 1.7

```

root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip route add default via 172.25.17.254 dev eth4 table Andriy
root@lw-417-08:/home/likewise-open/WFIZMAT/mandebura_av# ip route flush cache

```

Рис 1.8