

Лабораторна робота №6

з дисципліни «мережеві технології»

«Маршрутизація в мережах TCP/IP»

Виконав: студент групи ІП-73мп
Олександр Ковальчук

Контрольні запитання:

1. З якою метою в OSPF вводиться поняття область мережі?

Зі збільшенням розмірності мережі обчислювальна складність протоколу OSPF швидко зростає. Для подолання цього недоліку в протоколі OSPF вводиться поняття області мережі. Маршрутизатори, що належать деякій області, будують граф зв'язків тільки для цієї області, що скорочує розмірність мережі. Між областями інформація про зв'язки не передається, а прикордонні для областей маршрутизатори обмінюються тільки інформацією про адреси мереж, наявних в кожній з областей, і відстанню від прикордонного маршрутизатора до кожної мережі. При передачі пакетів між областями вибирається один з прикордонних маршрутизаторів області, а саме той, у якого відстань до потрібної мережі менше.

2. Особливості архітектури пакета Quagga.

Кожен протокол маршрутизації обслуговується окремим демоном, з подальшим формуванням таблиць маршрутизації. Одночасно працювати можуть кілька різних демонів разом із керуючим демоном zebra. ripd демон обробляє протокол RIP, в той час як ospfd - демон, який підтримує OSPFv2. bgpd підтримує протокол BGP-4. Демон zebra служить для формування таблиці маршрутизації і перерозподілу маршрутів між різними протоколами.

3. Які показники застосовуються для оцінки маршрутів в алгоритмах маршрутизації?

Довжина маршруту, надійність, затримка, пропускна здатність, навантаження, вартість зв'язку

4. Що показує значення змінної net.inet.ip.forwarding у FreeBSD?

Змінна net.inet.ip.forwarding вказує, чи увімкнена передача трафіку між інтерфейсами. Якщо значення змінної 1, то пересилка трафіку увімкнена, якщо 0 — то ні.

5. Сутність алгоритмів маршрутизації стану зв'язків.

Алгоритми стану зв'язків забезпечують кожен маршрутизатор інформацією, достатньою для побудови точного графа зв'язків мережі. Всі маршрутизатори працюють на підставі однакових графів, що робить процес маршрутизації стійкішим до змін конфігурації.

«Широкомовна» розсилка (тобто передача пакета всім безпосереднім сусідам маршрутизатора) використовується тут тільки при змінах стану зв'язків, що відбувається в надійних мережах не так часто. Вершинами графа є як маршрутизатори, так і мережі, які ними об'єднані. Поширювана по мережі інформація складається з опису зв'язків різних типів: маршрутизатор - маршрутизатор, маршрутизатор - мережа.

Щоб зрозуміти, в якому стані знаходяться лінії зв'язку, підключені до його портів, маршрутизатор періодично обмінюється короткими пакетами HELLO зі своїми найближчими сусідами.

6. Який параметр за замовчуванням використовується в якості метрики в OSPF?

Протокол OSPF за замовчуванням використовує метрику, що враховує пропускну здатність каналів зв'язку

7. Основні відмінності протоколів RIPv1 і RIPv2.

Існують дві версії протоколу RIP: RIPv1 і RIPv2. Версія 2 має деякі удосконалення, як то: можливість маршрутизації мереж по моделі CIDR (крім адреси мережі передається і маска), підтримка мультікастинга, можливість використання аутентифікації RIP-повідомлень і ін

8. Яка максимальна відстань між мережами допустима в протоколі RIP?

15

9. Класифікація алгоритмів маршрутизації.

- Однокрокові. Маршрутизація виконується за розподіленою схемою: кожен маршрутизатор бере на себе відповідальність за вибір тільки одного кроку маршруту. Остаточний маршрут складається в результаті роботи всіх маршрутизаторів, через які проходить даний пакет. Поділяються на:

- Алгоритми статичної маршрутизації. Всі записи в таблиці маршрутизації – статичні.
- Алгоритми простої маршрутизації. Таблиця маршрутизації або не використовується, або будується без участі протоколів маршрутизації.
 - Випадкова
 - Лавинна (відправляється усім, крім джерела)
 - Прецедентна (таблиця маршрутизації будується шляхом аналізу адресних полів вхідних пакетів)
- Алгоритми динамічної маршрутизації. Вони забезпечують автоматичне оновлення таблиць маршрутизації при змінах конфігурації мережі.
- Багатокрокові. Вузол, який відправляє пакет в мережу, повинен прописати повний маршрут його прямування через всі проміжні маршрутизатори. Зрозуміло, що при використанні багатокрокової маршрутизації немає необхідності будувати і аналізувати таблиці маршрутизації. Це прискорює проходження пакету по мережі, розвантажує маршрутизатори, але при цьому дуже велике навантаження лягає на кінцеві вузли

10. Сутність дистанційно-векторних алгоритмів маршрутизації.

В алгоритмах дистанційно-векторного типу кожен маршрутизатор періодично і широкомовно розсилає по мережі вектор, компонентами якого є відстані від даного маршрутизатора до всіх відомих йому мереж. При отриманні вектора від сусіда маршрутизатор нарощує відстані до зазначених в векторі мереж на відстань до даного сусіда. Отримавши вектор від сусіднього маршрутизатора, кожен маршрутизатор додає до нього інформацію про відомі йому інші мережі, про які він дізнався безпосередньо (якщо вони підключені до його портів) або з аналогічних оголошень інших маршрутизаторів, а потім знову розсилає нове значення вектора по мережі. Зрештою, кожен маршрутизатор дізнається інформацію про всі наявні мережі і про відстань до них через сусідні маршрутизатори.

11. Які вимоги пред'являються до алгоритмів динамічної маршрутизації?

До адаптивних алгоритмів маршрутизації пред'являються дуже важливі вимоги. По-перше, вони повинні забезпечувати, якщо не оптимальність, то хоча б раціональність маршруту. По-друге, алгоритми повинні бути досить простими, щоб при їх реалізації не витрачалося занадто багато мережевих ресурсів, зокрема вони не повинні вимагати занадто великого обсягу обчислень або породжувати інтенсивний службовий трафік. І, нарешті, алгоритми маршрутизації повинні мати властивість збіжності, тобто завжди приводити до однозначного результату за прийнятний час.

Конфігурування роутерів R1 (mary), R2 (shanee), R3 (liza)

R1 (mary)

```
#!/bin/sh
set -e

# =====
# Configure network interfaces
# =====
# Please note that all operations are done on em1 interface. This is due em0
# interface is used only to have internet access inside the VM
ifconfig em1 inet 10.18.1.1/24

# network between routers R1(IF1) and R2(IF2)
ifconfig vlan5 create
ifconfig vlan5 inet 10.18.51.65/26 vlan 5 vlandev em1
# network between routers R1(IF2) and R3(IF1)
ifconfig vlan7 create
ifconfig vlan7 inet 10.18.51.1/26 vlan 7 vlandev em1
# subnet with data R1(IF3)
ifconfig vlan103 create
ifconfig vlan103 inet 192.168.3.3/24 vlan 103 vlandev em1

# =====
# Install quagga
# =====
ASSUME_ALWAYS_YES=yes pkg install quagga

# =====
# Create RIP configuration
# =====
mkdir -p /usr/local/etc/quagga
mkdir -p /var/log/quagga
chown -R quagga:quagga /var/log/quagga

cat >/usr/local/etc/quagga/ripd.conf <<EOF
hostname mary.anxolerd.net
password zebra

debug rip events
debug rip packet

router rip
version 2

network vlan5
network vlan7
network vlan103

log file /var/log/quagga/ripd.log
log stdout
EOF

cat >/usr/local/etc/quagga/zebra.conf <<EOF
```

```

hostname mary.anxolerd.net
password zebra
enable password zebra

interface vlan6
    multicast
interface vlan7
    multicast
interface vlan103
    multicast

log file /var/log/quagga/zebra.log
EOF

# =====
# Test run
# =====
zebra & ripd
killall -9 zebra

# =====
# Run all the things on startup
# =====
cat >/etc/rc.conf <<EOF
#!/bin/sh
ifconfig_em0="DHCP"
sshd_enable="YES"
hostname="mary"

ifconfig_em1="inet 10.0.1.1 netmask 255.255.255.0"
cloned_interfaces="vlan5 vlan7 vlan103"
ifconfig_vlan5="inet 10.18.51.65/26 vlan 5 vlandev em1"
ifconfig_vlan7="inet 10.18.51.1/26 vlan 7 vlandev em1"
ifconfig_vlan103="inet 192.168.3.3/24 vlan 103 vlandev em1"

quagga_enable="YES"
quagga_daemons="zebra ripd"
EOF

```

R2 (shanee)

```

#!/bin/sh
set -e

# =====
# Configure network interfaces
# =====
# Please note that all operations are done on em1 interface. This is due em0
# interface is used only to have internet access inside the VM
ifconfig em1 inet 10.18.1.2/24

# network between routers R2(IF2) and R1(IF1)
ifconfig_vlan5 create
ifconfig_vlan5 inet 10.18.51.66/26 vlan 5 vlandev em1
# network between routers R2(IF1) and R3(IF2)
ifconfig_vlan6 create
ifconfig_vlan6 inet 10.18.51.129/26 vlan 6 vlandev em1
# subnet with data R2(IF3)

```

```
ifconfig vlan203 create
ifconfig vlan203 inet 192.168.6.3/24 vlan 203 em1
```

```
# =====
# Install quagga
# =====
ASSUME_ALWAYS_YES=yes pkg install quagga
```

```
# =====
# Create RIP configuration
# =====
mkdir -p /usr/local/etc/quagga
mkdir -p /var/log/quagga
chown -R quagga:quagga /var/log/quagga
```

```
cat >/usr/local/etc/quagga/ripd.conf <<EOF
hostname shanee.anxolerd.net
password zebra
```

```
debug rip events
debug rip packet
```

```
router rip
version 2
```

```
network vlan5
network vlan6
network vlan203
```

```
log file /var/log/quagga/ripd.log
log stdout
EOF
```

```
cat >/usr/local/etc/quagga/zebra.conf <<EOF
hostname shanee.anxolerd.net
password zebra
enable password zebra
```

```
interface vlan5
multicast
```

```
interface vlan6
multicast
```

```
interface vlan203
multicast
```

```
log file /var/log/quagga/zebra.log
EOF
```

```
# =====
# Test run
# =====
zebra & ripd
killall -9 zebra
```

```
# =====
# Run all the things on startup
```



```
# =====
cat >/etc/rc.conf <<EOF
#!/bin/sh
ifconfig_em0="DHCP"
sshd_enable="YES"
hostname="shanee"

ifconfig_em1="inet 10.0.1.2 netmask 255.255.255.0"
cloned_interfaces="vlan5 vlan6 vlan203"
ifconfig_vlan5="inet 10.18.51.66/26 vlan 5 vlandev em1"
ifconfig_vlan6="inet 10.18.51.129/26 vlan 6 vlandev em1"
ifconfig_vlan203="inet 192.168.6.3/24 vlan 203 vlandev em1"

quagga_enable="YES"
quagga_daemons="zebra ripd"
EOF
```

R3 (liza)

```
#!/bin/sh
set -e

# =====
# Configure network interfaces
# =====
# Please note that all operations are done on em1 interface. This is due em0
# interface is used only to have internet access inside the VM
ifconfig em1 inet 10.18.1.3/24

# network between routers R1(IF1) and R3(IF2)
ifconfig_vlan7 create
ifconfig_vlan7 inet 10.18.51.2/26 vlan 7 vlandev em1
# network between routers R3(IF2) and R2(IF1)
ifconfig_vlan6 create
ifconfig_vlan6 inet 10.18.51.130/26 vlan 6 vlandev em1
# subnet with data R2(IF3)
ifconfig_vlan303 create
ifconfig_vlan303 inet 192.168.9.3/24 vlan 303 vlandev em1

# =====
# Install quagga
# =====
ASSUME_ALWAYS_YES=yes pkg install quagga

# =====
# Create RIP configuration
# =====
mkdir -p /usr/local/etc/quagga
mkdir -p /var/log/quagga
chown -R quagga:quagga /var/log/quagga

cat >/usr/local/etc/quagga/ripd.conf <<EOF
hostname liza.anxolerd.net
password zebra

debug rip events
debug rip packet
```

```

router rip
version 2

network vlan6
network vlan7
network vlan303

log file /var/log/quagga/ripd.log
log stdout
EOF

cat >/usr/local/etc/quagga/zebra.conf <<EOF
hostname liza.anxolerd.net
password zebra
enable password zebra

interface vlan5
    multicast

interface vlan7
    multicast

interface vlan303
    multicast

log file /var/log/quagga/zebra.log
EOF

# =====
# Test run
# =====
zebra & ripd
killall -9 zebra

# =====
# Run all the things on startup
# =====
cat >/etc/rc.conf <<EOF
#!/bin/sh
ifconfig_em0="DHCP"
sshd_enable="YES"
hostname="liza"

ifconfig_em1="inet 10.0.1.3 netmask 255.255.255.0"
cloned_interfaces="vlan6 vlan7 vlan103"
ifconfig_vlan6="inet 10.18.51.130/26 vlan 6 vlandev em1"
ifconfig_vlan7="inet 10.18.51.2/26 vlan 7 vlandev em1"
ifconfig_vlan303="inet 192.168.9.3/24 vlan 303 vlandev em1"

quagga_enable="YES"
quagga_daemons="zebra ripd"
EOF

```

```

2017/12/10 21:01:25 RIP: update routes to neighbor 10.18.51.66
2017/12/10 21:01:25 RIP: rip_send_packet 10.18.51.65 > 10.18.51.66 (vlan5)
2017/12/10 21:01:25 RIP: SEND to 10.18.51.66.520
2017/12/10 21:01:25 RIP: SEND RESPONSE version 2 packet size 44
2017/12/10 21:01:25 RIP: 10.18.51.0/26 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:25 RIP: 192.168.3.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:25 RIP: RECV packet from 10.18.51.66 port 520 on vlan5
2017/12/10 21:01:25 RIP: RECV RESPONSE version 2 packet size 44
2017/12/10 21:01:25 RIP: 10.18.51.128/26 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:25 RIP: 192.168.6.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:28 RIP: RECV packet from 10.18.51.66 port 520 on vlan5
2017/12/10 21:01:28 RIP: RECV REQUEST version 2 packet size 24
2017/12/10 21:01:28 RIP: 0.0.0.0/0 -> 0.0.0.0 family 0 tag 0 metric 16
2017/12/10 21:01:28 RIP: update routes to neighbor 10.18.51.66
2017/12/10 21:01:28 RIP: rip_send_packet 10.18.51.65 > 10.18.51.66 (vlan5)
2017/12/10 21:01:28 RIP: SEND to 10.18.51.66.520
2017/12/10 21:01:28 RIP: SEND RESPONSE version 2 packet size 44
2017/12/10 21:01:28 RIP: 10.18.51.0/26 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:28 RIP: 192.168.3.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:29 RIP: RECV packet from 10.18.51.66 port 520 on vlan5
2017/12/10 21:01:29 RIP: RECV RESPONSE version 2 packet size 44
2017/12/10 21:01:29 RIP: 10.18.51.128/26 -> 0.0.0.0 family 2 tag 0 metric 1
2017/12/10 21:01:29 RIP: 192.168.6.0/24 -> 0.0.0.0 family 2 tag 0 metric 1

```

```

# netstat -r
Routing tables

Internet:
Destination      Gateway          Flags           Netif Expire
default          10.0.2.2        UGS             em0
10.0.1.0/24      link#2          U               em1
10.0.1.2         link#2          UHS             lo0
10.0.2.0/24      link#1          U               em0
10.0.2.15        link#1          UHS             lo0
10.18.51.0/26    10.18.51.65    UG1             vlan5
10.18.51.64/26   link#4          U               vlan5
10.18.51.66      link#4          UHS             lo0
10.18.51.128/26  link#5          U               vlan6
10.18.51.129     link#5          UHS             lo0
localhost        link#3          UH              lo0
192.168.3.0/24   10.18.51.65    UG1             vlan5
192.168.6.0/24   link#6          U               vlan203
192.168.6.3      link#6          UHS             lo0

```