Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет Программной инженерии и компьютерной техники

**Лабораторная работа №4**

по дисциплине «Сети ЭВМ и телекоммуникации»

**Выполнил**

Ореховский А.,

группа P3317

**Преподаватель**

Манаев Н.Ю.

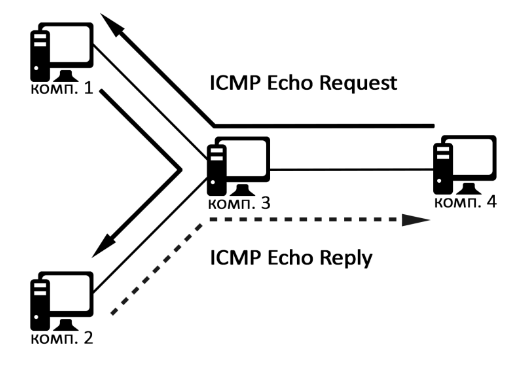
Санкт-Петербург

2020

# Цель

Цель работы состоит в изучении основных методов настройки маршрутизируемых компьютерных сетей на примере сети, состоящей из компьютеров под управлением ОС Linux.

# Вариант



# Этап I. Настройка маршрутизации.

Для начала, чтобы иметь возможность передавать транзитные пакеты, я прописал на каждом компьютере

> sysctl -w net.ipv4.ip\_forward=1

## Определения

Я определил ip-адреса в каждой подсети:

1. Для компьютера 1 определил:
   1. Является узлом подсети **s1\_s3**
   2. Имеет ip-адрес **5.10.13.1/30** на интерфейсе **eth0**
2. Для компьютера 2 определил:
   1. Является узлом подсети **s2\_s3**
   2. Имеет ip-адрес **5.10.23.1/30** на интерфейсе **eth0**
3. Для компьютера 3 определил:
   1. Является узлом подсети **s1\_s2, s2\_s3, s3\_s4**
   2. Имеет ip-адрес **5.10.13.2/30** на интерфейсе **eth0**
   3. Имеет ip-адрес **5.10.23.2/30** на интерфейсе **eth1**
   4. Имеет ip-адрес **5.10.43.2/30** на интерфейсе **eth2**
4. Для компьютера 2 определил:
   1. Является узлом подсети **s4\_s3**
   2. Имеет ip-адрес **5.10.43.1/30** на интерфейсе **eth0**

## Команды

### Компьютер 1

> ip a add 5.10.13.1/30 dev eth0

> ip ro add default via 5.10.13.2/30

### Компьютер 2

> ip a add 5.10.23.1/30 dev eth0

> ip ro add default via 5.10.23.2/30

### Компьютер 3

# Поднял интерфейсы

> ip link set eth0 up

> ip link set eth1 up

> ip link set eth2 up

# Определил ip-адреса

> ip a add 5.10.13.2/30 dev eth0

> ip a add 5.10.23.2/30 dev eth1

> ip a add 5.10.43.2/30 dev eth2

# Добавил правила для каждого компьютера в разные таблицы

> ip ro add default via 5.10.13.1 table 4

> ip ro add default via 5.10.23.1 table 1

> ip ro add default via 5.10.43.1 table 2

# Определил к какой таблице обращаться в зависимости от источника

> ip rule add prio 1 from 5.10.43.1 table 4

> ip rule add prio 2 from 5.10.13.1 table 1

> ip rule add prio 3 from 5.10.23.1 table 2

### Компьютер 4

> ip a add 5.10.43.1/30 dev eth0

> ip ro add default via 5.10.43.2/30

## Результат

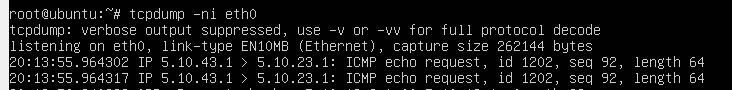
В данной конфигурации пакеты терялись перед тем, как достигнуть источника. Например:

Попробуем «пингануть» 2 комп. с 4-го:

*комп. 4*

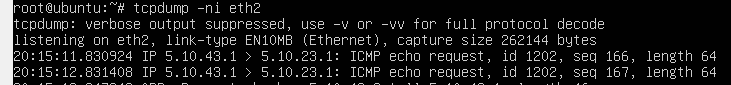


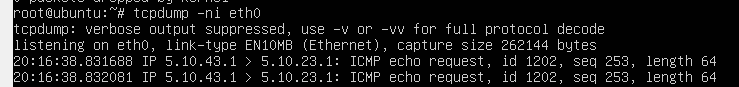
Видно что комп.4 не может пингануть второй. Разбираемся и смотрим на поток пакетов в комп.1: *комп.1*



Видны запросы с 4 комп. Посмотрим на интерфейс eth2 и eth0 у комп. 3

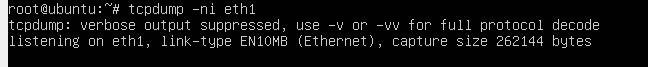
*комп. 3*



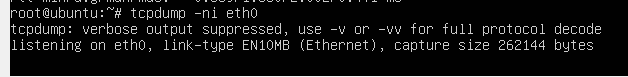


Комп. 3 видит запросы с комп.4 и проброшенные через комп. 1. Проверим eth1 у комп. 3 и eth0 у комп. 2:

*комп. 3*



*комп. 2*



Видно, что пакеты теряются не достигнув конечного узла. То же самое происходи при попытке пингануться к соседу против часовой. Например с комп.4 к комп. 1.

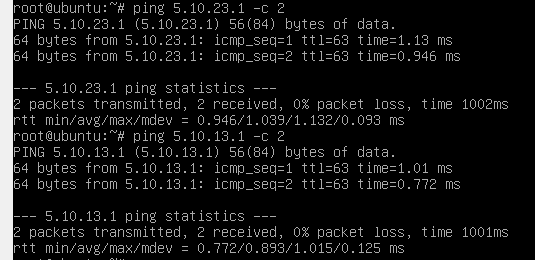
Данная настройка сети была осуществлена мной несколько раз, с использованием разных длин масок (я использовал и 24, и 30 бит).

В чем состоит проблема?

После долго поиска проблемы, я понял, что это происходит потому что на виртуальных машинах после того как я прописал команды отключения rp\_filter, их необходимо было перезапускать. При перезапуске же самой системы linux будь то при помощи команды sudo reboot или при обычном отключении и включении у меня по необъяснимым обстоятельствам сбрасываются настройки сети. Исходя из данной проблемы, я выполнял следующие задания без «хитрой» маршрутизации.

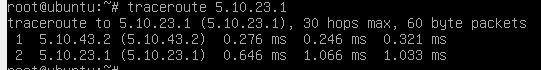
После очистки таблиц маршрутизации командами ip ro flush table <tableNO>, все компьютеры стали «пинговать» друг друга.

*комп. 4*



При этом, ожидаемо, маршрутизация «против часовой стрелки» отсутствует.

*комп. 4*



## Утилита nc

Комп. 4 в роли сервера:



Комп. 2 в роли клиента:



# Этап II. Фильтрация

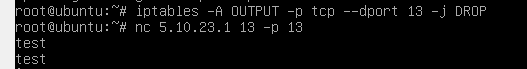
* Запретить передачу только тех пакетов, которые отправлены на TCP-порт, заданный в настройках утилиты nc.

*комп. 4*

> iptables -A OUTPUT -p tcp --dport 13 -j DROP

**Запрещенный пакет**

*комп. 4*



*комп. 2*



**Не запрещенный пакет**

*комп. 4*



*комп. 2*



* Запретить приём только тех пакетов, которые отправлены с UDP-порта утилиты nc.

*комп. 2*

> iptables -A INPUT -p udp -j DROP

**Запрещенный пакет**

*комп. 4*

****

*комп. 2*

****

**Не запрещенный пакет**

*комп. 4*

****

*комп. 2*

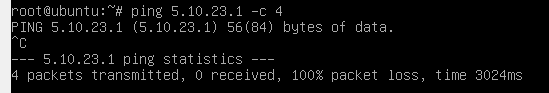
****

* Запретить передачу только тех пакетов, которые отправлены с IP-адреса компьютера А.

*комп. 3*

> iptables -A FORWARD -s 5.10.43.1 -j DROP

*комп. 4*

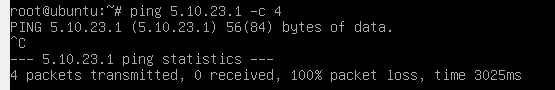


* Запретить приём только тех пакетов, которые отправлены на IP-адрес компьютера Б.

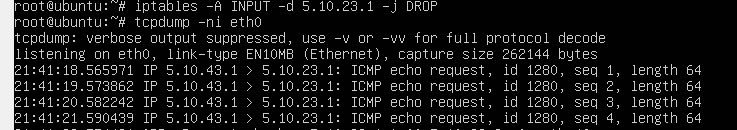
*комп. 2*

> iptables -A INPUT -d 5.10.23.1 -j DROP

*комп. 4*



*комп. 2*



* Запретить приём и передачу ICMP-пакетов, размер которых превышает 1000 байт, а поле TTL при этом меньше 10.

*комп. 2*

> iptables -A INPUT -p icmp -m ttl --ttl-lt 10 -m length -–length 101:65535 -j DROP

*комп. 4*

