Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет безопасности информационных технологий

Дисциплина:

«Компьютерные сети »

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

«Основы администрирования маршрутизируемых компьютерных сетей»

Выполнили:

Бардышев Артём Антонович,

студент группы N3346

_____
(подпись)

Проверил:

Ярошевский Дмитрий Сергеевич,

Ведущий инженер, ФБИТ

_____
(отметка о выполнении)

_____
(подпись)

Санкт-Петербург

2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

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

В процессе выполнения работы изучается сетевой уровень модели OSI. Производится базовая настройка связности в сети, управление таблицами маршрутизации и правилами трансляции сетевых адресов. При помощи утилиты tcpdump выполняются наблюдения за передачей трафика по каналам связи в маршрутизируемой компьютерной сети. Применение утилиты tcpdump позволяет непосредственно в терминале (это основной метод управления сетевым оборудованием) наблюдать проходящие через интерфейсы компьютера пакеты и изучить их внутреннюю структуру. В данной работе изучаются методы маршрутизации в сетях IPv4 и IPv6, а также широко распространенная в компьютерных сетях технология NAT.

1 ПОДГОТОВКА

1.1 Определение варианта

V1 = 1 + (N mod5), V2 = 6 + (N mod5)

где V1 и V2 – номера вариантов; N – сумма количества букв в фамилии и имени студента; mod – операция взятия остатка от деления.

ФИ: Бардышев Артём

Количество уникальных букв 10, значит N = 10, значит V1 = 1, V2 = 6.

1.2 Разбор задания

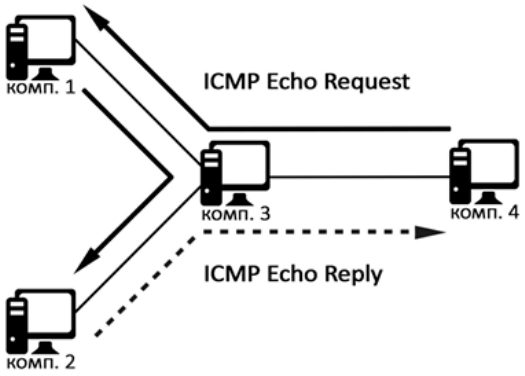


Рисунок 4.5 - Топология сети и схема прохождения трафика

На рисунке 4.5 изображена топология сети и требуемый путь прохождения сетевых пакетов. Необходимо так настроить хосты сети, чтобы ping-запрос (request) от компьютера 4 к компьютеру 2 шёл по пути, указанному сплошной стрелкой. При этом ping-ответ (reply) должен возвращаться другим путём: по пунктирной стрелке.

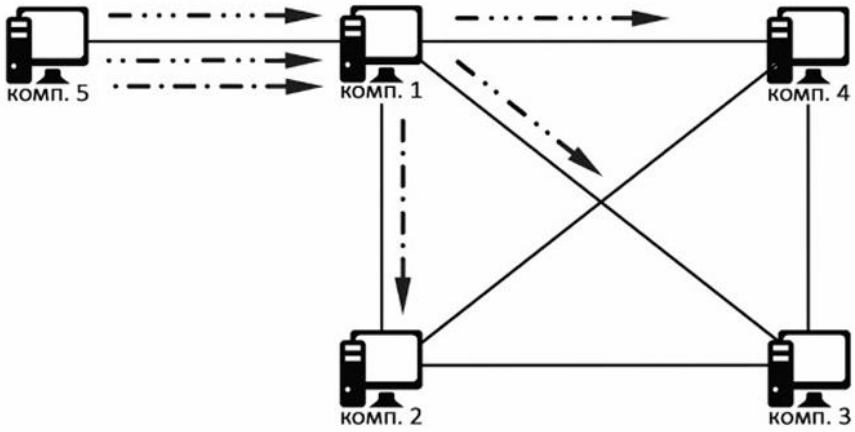


Рисунок 4.14 - Топология сети и схема прохождения трафика

На рисунке 4.14 изображена топология сети. С компьютера 5 посылается ICMP Echo Request на компьютер 3. Размер пакета и MTU сетевого интерфейса должны быть выбраны таким образом, чтобы пакет был фрагментирован на 3 фрагмента. При этом компьютер 1 получит 3 пакета, которые должны быть направлены по 3 различным каналам. После получения всех фрагментов, в компьютере 3 формируется ICMP Echo Reply, который должен быть разделен на 2 фрагмента. Два фрагмента должны пройти различный по длине обратный путь к компьютеру 5.

Для выполнения лабораторной работы был использован гипервизор 2 уровня VirtualBox. В качестве виртуального образа — Ubuntu Server 14.04.5.

2 ПОСТРОЕНИЕ СЕТИ

2.1.1 Определим адреса хостов:

№	IPv4	IPv6
1	7.8.1.2	::ffff:0708:0102
2	7.8.2.2	::ffff:0708:0202
3	7.8.1.1	::ffff:0708:0101
4	7.8.3.2	::ffff:0708:0302

Все скрипты выполняются в sudo режиме. Виртуальный жёсткий диск машины можно просто продублировать, чтобы не повторять процесс установки 4 раза (именно поэтому на скриншотах имя пользователя и компьютера везде одинаковое).

2.1.2 Формирование топологии, проверка утилитой ping.

Для настройки интерфейсов будем использовать скрипт form_topology.sh. Запускать в каждой виртуальной машине, передав в качестве аргумента её номер и режим адресации: IPV4 или IPV6.

```
function configure_eth() {
    echo "Настройка интерфейсов"
    ip "$net_param""$mask" a add "$1" dev enp0s3
}

function add_route_to() {
    echo "Добавление маршрута"
    ip "$net_param" ro add default via "$1"
}

function ping_to() {
    echo "Проверка утилитой ping"
    for i in "${1[@]}"
    do
      ping "${hosts[$i]}" -c 2
    done
}
```

```
if [ $# -lt 2 ]
  then
    echo "Необходимо задать номер хоста и режим адресации"
    exit
fi
# i-элемент == номеру хоста, а 0-вой оставим пустым для удобства обращения
case ${2} in
  "IPV4")
    hosts=("" "7.8.1.2" "7.8.2.2" "7.8.1.1" "7.8.3.2")
    net_param=""
    mask="/30"
    ;;
  "IPV6")
    hosts=("" "::ffff:0708:0102" "::ffff:0708:0202" "::ffff:0708:0101"
"::ffff:0708:0302")
    net_param="-6"
    mask="/126"
    ;;
  *)
    echo "Не выбран режим адресации"
    exit
    ;;
esac
sysctl -w net.ipv4.ip_forwarding = 1
sysctl -w net.ipv4.conf.all.rp_filter = 1
ip link set enp0s3 up
ip a flush enp0s3
case ${1} in
    "1")
        configure_eth "${hosts[1]}"
        add_route_to "${hosts[3]}"
        dest=(2 3 4)
        ping_to "${dest[@]}"
        ;;
    "2")
        configure_eth "${hosts[2]}"
        add_route_to "${hosts[3]}"
        dest=(1 3 4)
        ping_to "${dest[@]}"
        ;;
    "3")
        configure_eth "${hosts[3]}"
        add_route_to "${hosts[4]}"
        dest=(1 2 4)
        ping_to "${dest[@]}"
        ;;
    "4")
        configure_eth "${hosts[4]}"
        dest=(1 2 3)
        ping_to "${dest[@]}"
        ;;
    *)
        echo "Некорректно задан номер хоста"
        ;;
esac
```

*host*1 → *host*2

```
comp1@comp1:~$ ping 7.8.2.2 -c 3
PING 7.8.2.2 (7.8.2.2) : 56(84) bytes of data.
64 bytes from 7.8.2.2: icmp_seq=1 ttl=64 time=2.18 ms
64 bytes from 7.8.2.2: icmp_seq=2 ttl=64 time=3.22 ms
64 bytes from 7.8.2.2: icmp_seq=3 ttl=64 time=1.75 ms

--- 7.8.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 1.75/2.38/3.22/1.388 ms
comp1@comp1:~$
```

$host1 \rightarrow host3$

```
comp1@comp1:~$ ping 7.8.1.1 -c 3
PING 7.8.1.1 (7.8.1.1) : 56(84) bytes of data.
64 bytes from 7.8.1.1: icmp_seq=1 ttl=64 time=2.60 ms
64 bytes from 7.8.1.1: icmp_seq=2 ttl=64 time=2.45 ms
64 bytes from 7.8.1.1: icmp_seq=3 ttl=64 time=2.80 ms

--- 7.8.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3100ms
rtt min/avg/max/mdev = 2.45/2.61/2.80/0.145 ms
comp1@comp1:~$
```

$host2 \rightarrow host1$

```
comp1@comp1:~$ ping 7.8.1.2 -c 3
PING 7.8.1.2 (7.8.1.2) : 56(84) bytes of data.
64 bytes from 7.8.1.2: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 7.8.1.2: icmp_seq=2 ttl=64 time=2.64 ms
64 bytes from 7.8.1.2: icmp_seq=3 ttl=64 time=3.11 ms

--- 7.8.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 1.68/2.47/3.11/0.678 ms
comp1@comp1:~$ _
```

$host2 \rightarrow host3$

```
comp1@comp1:~$ ping 7.8.1.1 -c 3
PING 7.8.1.1 (7.8.1.1) : 56(84) bytes of data.
64 bytes from 7.8.1.1: icmp_seq=1 ttl=64 time=1.96 ms
64 bytes from 7.8.1.1: icmp_seq=2 ttl=64 time=3.69 ms
64 bytes from 7.8.1.1: icmp_seq=3 ttl=64 time=3.69 ms

--- 7.8.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2505ms
rtt min/avg/max/mdev = 1.96/3.11/3.69/1.101 ms
comp1@comp1:~$ _
```

*host*2 → *host*4

```
comp1@comp1:~$ ping 7.8.3.2 -c 3
PING 7.8.3.2 (7.8.3.2) : 56(84) bytes of data.
64 bytes from 7.8.3.2: icmp_seq=1 ttl=64 time=1.74 ms
64 bytes from 7.8.3.2: icmp_seq=2 ttl=64 time=3.08 ms
64 bytes from 7.8.3.2: icmp_seq=3 ttl=64 time=3.29 ms

--- 7.8.3.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.74/2.70/3.29/1.029 ms
comp1@comp1:~$ _
```

*host*3 → *host*1

```
comp1@comp1:~$ ping 7.8.1.2 -c 3
PING 7.8.1.2 (7.8.1.2) : 56(84) bytes of data.
64 bytes from 7.8.1.2: icmp_seq=1 ttl=64 time=1.81 ms
64 bytes from 7.8.1.2: icmp_seq=2 ttl=64 time=2.24 ms
64 bytes from 7.8.1.2: icmp_seq=3 ttl=64 time=2.69 ms

--- 7.8.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 1.81/2.24/3.69/0.360 ms
comp1@comp1:~$ _
```

*host*3 → *host*2

```
comp1@comp1:~$ ping 7.8.2.2 -c 3
PING 7.8.2.2 (7.8.2.2) : 56(84) bytes of data.
64 bytes from 7.8.2.2: icmp_seq=1 ttl=64 time=0.91 ms
64 bytes from 7.8.2.2: icmp_seq=2 ttl=64 time=2.19 ms
64 bytes from 7.8.2.2: icmp_seq=3 ttl=64 time=2.55 ms

--- 7.8.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2777ms
rtt min/avg/max/mdev = 0.91/1.88/2.55/0.824 ms
comp1@comp1:~$ _
```

*host*3 → *host*4

```
comp1@comp1:~$ ping 7.8.3.2 -c 3
PING 7.8.3.2 (7.8.3.2) : 56(84) bytes of data.
64 bytes from 7.8.3.2: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 7.8.3.2: icmp_seq=2 ttl=64 time=3.00 ms
64 bytes from 7.8.3.2: icmp_seq=3 ttl=64 time=2.69 ms

--- 7.8.3.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 1.38/2.35/3.00/0.702 ms
comp1@comp1:~$ _
```

*host*4 → *host*1

```
comp1@comp1:~$ ping 7.8.1.2 -c 3
PING 7.8.1.2 (7.8.1.2) : 56(84) bytes of data.
64 bytes from 7.8.1.2: icmp_seq=1 ttl=64 time=1.76 ms
64 bytes from 7.8.1.2: icmp_seq=2 ttl=64 time=3.86 ms
64 bytes from 7.8.1.2: icmp_seq=3 ttl=64 time=2.33 ms

--- 7.8.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1966ms
rtt min/avg/max/mdev = 1.76/2.65/3.86/1.600 ms
comp1@comp1:~$
```

*host*4 → *host*2

```
comp1@comp1:~$ ping 7.8.2.2 -c 3
PING 7.8.2.2 (7.8.2.2) : 56(84) bytes of data.
64 bytes from 7.8.2.2: icmp_seq=1 ttl=64 time=1.72 ms
64 bytes from 7.8.2.2: icmp_seq=2 ttl=64 time=1.73 ms
64 bytes from 7.8.2.2: icmp_seq=3 ttl=64 time=1.00 ms

--- 7.8.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1.420ms
rtt min/avg/max/mdev = 1.72/1.48/1.73/0.666 ms
comp1@comp1:~$ _
```

*host*4 → *host*3

```
comp1@comp1:~$ ping 7.8.1.1 -c 3
PING 7.8.1.1 (7.8.1.1) : 56(84) bytes of data.
64 bytes from 7.8.1.1: icmp_seq=1 ttl=64 time=3.20 ms
64 bytes from 7.8.1.1: icmp_seq=2 ttl=64 time=3.90 ms
64 bytes from 7.8.1.1: icmp_seq=3 ttl=64 time=2.74 ms

--- 7.8.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1.420ms
rtt min/avg/max/mdev = 2.74/3.28/3.90/0.472 ms
comp1@comp1:~$
```

3 ОТПРАВКА ПАКЕТОВ УТИЛИТОЙ NC

В качестве аргумента передаётся режим работы nc. Скрипт отправляет сообщение автоматически.

```
modes=(“CLIENT” “SERVER”)
case ${1} in
	“${modes[0]}”)
		nc 7.8.3.2 8090 << EOF
Artem Bardyshev
EOF
		;;
	“${modes[1]}”)
		nc -l 8090
		;;
	*)
	echo “Режим задан не корректно”
esac
```

Далее при вызове (в режиме клиента):

```
nc 7.8.3.2 8090
```

Приходит ответ:

```
Artem Bardyshev
```

Так же при вызове (в режиме сервера):

```
nc -l 8090
```

Приходит ответ:

```
Artem Bardyshev
```

4 СОЗДАНИЕ МЕЖСЕТЕВОГО ЭКРАНА

4.1 Блокировка *tcp*−пакетов (утилете не удаётся установить соединение)

```
iptables -F
echo "Запретить передачу только тех пакетов, которые отправлены на TCP-порт,
заданный в настройках утилиты nc."
iptables -A OUTPUT -j REJECT -o enp0s3 -p tcp --dport 8090
```

```
comp1@comp1:~$ nc 7.8.2.2
```

```
comp1@comp1:~$ nc -l 8090
```

4.2 Блокировка входящих *udp*−пакетов

```
echo "Запретить приём только тех пакетов, которые отправлены с UDP-порта
утилиты nc."
iptables -A INPUT -j REJECT -i enp0s3 -p udp -sport 8090
```

```
comp1@comp1:~$ nc -l 8090
```

4.3 Правила фильтрации

```
echo "Запретить передачу только тех пакетов, которые отправлены с IP-адреса
компьютера А."
iptables -A OUTPUT -j REJECT -o enp0s3 -s 7.8.1.2
```

```
comp1@comp1:~$ iptables -L
Chain INPUT (policy ACCEPT)
target   prot   opt   source     destination
REJECT   udp    --    anywhere   anywhere       udp spt:rplay reject-with icmp-port-unreachable
reachable

Chain FORWARD (policy ACCEPT)
target  prot   opt    source     destination

Chain OUTPUT (policy ACCEPT)
target  prot   opt    source     destination
REJECT  tcp    --     anywhere   anywhere       tcp dpt:8090 reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target  prot   opt    source     destination
REJECT  all    --     7.8.1.2    anywhere       reject-with icmp-port-unreachable
```

4.4 Пример приема и неприема пакета

```
echo "Запретить приём только тех пакетов, которые отправлены на IPадрес
компьютера Б."
iptables -A INPUT -j REJECT -i enp0s3 -d 7.8.3.2/30
```

```
comp1@comp1:~$ ping 7.8.3.2 -c 3
PING 7.8.3.2 (7.8.3.2) 56(84) bytes of data.
From 7.8.3.2 icmp_seq=1 Destination Port Unreachable
From 7.8.3.2 icmp_seq=2 Destination Port Unreachable
From 7.8.3.2 icmp_seq=3 Destination Port Unreachable

--- 7.8.3.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2005ms

comp1@comp1:~$ ping 7.8.1.1 -c 3
PING 7.8.1.1 (7.8.1.1) 56(84) bytes of data.
64 bytes from 7.8.1.1: icmp_seq=1 ttl=64 time=1.42ms
64 bytes from 7.8.1.1: icmp_seq=2 ttl=64 time=1.83ms
64 bytes from 7.8.1.1: icmp_seq=3 ttl=64 time=3.79ms

--- 7.8.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2015ms
rrt min/avg/max/mdev = 1.42/2.35/3.79/1.036 ms
```

4.5 Пример фильтрации при отправке пакетов с помощью *ping*

```
echo "Запретить приём и передачу ICMP-пакетов, размер которых превышает 1000
байт, а поле TTL при этом меньше 10."
iptables -A OUTPUT -o enp0s3 -p icmp -m length ! --length 0:1000 -m ttl --
ttl-lt 10 -j REJECT
comp1@comp1:~$ ping -c 3 -s 100 -t 7.8.2.2
PING 7.8.2.2 (7.8.2.2) 56(84) bytes of data.
From 7.8.2.2 icmp_seq=1 Destination Net Unreachable
From 7.8.2.2 icmp_seq=2 Destination Net Unreachable

--- 7.8.2.2 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 999ms
comp1@comp1:~$ _
```

5 ЗАДАНИЕ ПО ВАРИАНТУ (IPV4) (РИСУНОК 4.5)

Для настройки перенаправления используем скрипт forwarding.sh, который в качестве параметра принимает номер машины и тип адресации. Сложность задачи, что пакет от хоста 4, после попадания на хост 3, должен не сразу отправиться на хост2, а сначала обойти хост1. Для достижения этого, пакет будет передаваться через разные порты, чтобы мы могли определить, откуда он пришёл, и, следовательно, куда должен отправиться дальше.

Итоговая схема будет выглядеть так:

С 4 на 2:

1. Пакеты, отправленные с хоста4, или полученные извне, перенаправляются на хост3 на порт 55004.

2. С хоста3 с порта 55004 на хост1.

3. С хоста 1 на хост3 порт 55001.

4. С хоста3 порт 55001 на хост2.

С 2 на 4:

1. С хоста2 на хост3 порт 5002.

2. С хоста3 порт 5002 на хост4.

```
function forward_icmp_packets() {
  # 1-ый параметр: IP, куда направляется пакет
  # 2-ой параметр: IP, куда перенаправить пакет
  # 3-ой параметр: порт, куда перенаправить
  # 4-ой параметр: порт, откуда перенаправить
  case "$#" in
    "2")
      echo "Настройка переадресации"
      iptables -t nat -A PREROUTING -i enp0s3 -p icmp -d "$1" -j DNAT --to-
destination "$2"
      ;;
    "3")
      echo "Настройка переадресации с заданием порта на который отправлять
пакет"
      iptables -t nat -A PREROUTING -i enp0s3 -p icmp -d "$1" -j DNAT --to-
destination "$2":"$3"
      ;;
    "4")
      echo "Настройка переадресации с заданием порта на который отправлять
пакет и откуда он пришёл"
      iptables -t nat -A PREROUTING -i enp0s3 -p icmp -d "$1" --dport="$4" -j
DNAT --to-destination "$2":"$3"
      ;;
```

```
      echo "Неверно задано количество аргументов"
      exit
      ;;
  esac
  iptables -t filter -A FORWARD -p icmp -d "$2" -j ACCEPT
}

case ${2} in
  "IPV4")
    hosts=("" "7.8.1.2" "7.8.2.2" "7.8.1.1" "7.8.3.2")
    ;;
  "IPV6")
    hosts=("" "::ffff:0708:0102" "::ffff:0708:0202" "::ffff:0708:0101"
"::ffff:0708:0302")
    ;;
  *)
    echo "Не выбран режим адресации"
    exit
    ;;
esac
case ${1} in
  "1")
    iptables -t nat -F
    forward_icmp_packets_from_to "${hosts[1]}" "${hosts[3]}" 55001
    ;;
  "2")
    iptables -t nat -F
    forward_icmp_packets_from_to "${hosts[4]}" "${hosts[3]}" 55002
    ;;
  "3")
    iptables -t nat -F
    forward_icmp_packets_from_to "${hosts[3]}" "${hosts[1]}" 55001 55004
    forward_icmp_packets_from_to "${hosts[3]}" "${hosts[2]}" 55002 55001
    forward_icmp_packets_from_to "${hosts[3]}" "${hosts[4]}" 55004 55002
    ;;
  "4")
    iptables -t nat -F
    forward_icmp_packets_from_to "${hosts[2]}" "${hosts[3]}" 55004
    ;;
  *)
    echo "Узел задан некорректно"
    ;;
esac
```

Запрос доходит до цели:

```
comp1@comp1:~$ ping 7.8.2.2 -c 3
PING 7.8.2.2 (7.8.2.2) : 56(84) bytes of data.
64 bytes from 7.8.2.2: icmp_seq=1 ttl=64 time=2.00 ms
64 bytes from 7.8.2.2: icmp_seq=2 ttl=64 time=1.88 ms
64 bytes from 7.8.2.2: icmp_seq=3 ttl=64 time=1.76 ms

--- 7.8.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2.007ms
rtt min/avg/max/mdev = 1.76/1.88/2.00/0.808 ms
```

И обратно:

```
comp1@comp1:~$ ping 7.8.2.2 -c 3
PING 7.8.2.2 (7.8.2.2) : 56(84) bytes of data.
64 bytes from 7.8.2.2: icmp_seq=1 ttl=64 time=2.00 ms
64 bytes from 7.8.2.2: icmp_seq=2 ttl=64 time=1.88 ms
64 bytes from 7.8.2.2: icmp_seq=3 ttl=64 time=1.76 ms

--- 7.8.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2.007ms
rtt min/avg/max/mdev = 1.76/1.88/2.00/0.808 ms
```

Трассировка пути туда:

```
comp1@comp1:~$ traceroute 7.8.2.2
traceroute to 7.8.2.2 (7.8.2.2), 30 hops max, 60 byte packets
 1   7.8.1.1     (7.8.1.1)     2.718 ms     2.707 ms     2.696 ms
 2   7.8.1.2     (7.8.1.2)     2.901 ms     2.696 ms     2.505 ms
 3   7.8.1.1     (7.8.1.1)     2.621 ms     2.403 ms     2.701 ms
 4   7.8.2.2     (7.8.2.2)     2.310 ms !X  2.464 ms !X  2.632 ms !X
```

И обратно:

```
comp1@comp1:~$ traceroute 7.8.3.2
traceroute to 7.8.3.2 (7.8.3.2), 30 hops max, 60 byte packets
 1   7.8.1.1     (7.8.1.1)     2.311 ms     2.531 ms     3.001 ms
 2   7.8.3.2     (7.8.3.2)     2.542 ms !X  2.961 ms !X  2.112 ms
```

6 ЗАДАНИЕ ПО ВАРИАНТУ (IPV6) (РИСУНОК 4.5)

Для создания топологии запустить на каждом хосте скрипт из пункта 1, передав в качестве 2-го аргумента IPV6. Пример:

./form_topology 1 IPv6

Для настройки маршрутизации, нужно запустить скрипт из пункта 4 с 2-ом параметром IPV6 для каждого хоста, т.к. выполняемые команды отличаются исключительно форматом адреса. Пример:

./forwarding.sh 1 IPv6

Проверки:

```
comp1@comp1:~$ ping6 ::ffff:0708:0202 -c 1
PING ::ffff:0708:0202 (::ffff:0708:0202) 56(84) bytes of data.
64 bytes from ::ffff:0708:0202 icmp_seq=1 ttl=64 time=3.01 ms

--- ::ffff:0708:0202 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2.987ms
rrt min/avg/max/mdev = 3.01/3.01/3.01/1.132 ms
comp1@comp1:~$ ping6 ::ffff:0708:0302 -c 1
PING ::ffff:0708:0302 (::ffff:0708:0302) 56(84) bytes of data.
64 bytes from ::ffff:0708:0302 icmp_seq=1 ttl=64 time=2.05 ms

--- ::ffff:0708:0302 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2.444 ms
rrt min/avg/max/mdev = 2.05/2.05/2.05/1.230 ms
```

Трассировки:

```
comp1@comp1:~$ traceroute6 ::ffff:0708:0202
traceroute to ::ffff:0708:0202 (::ffff:0708:0202), 30 hops max, 60 byte packets
 1   ::ffff:0708:0101     (::ffff:0708:0101)     2.514 ms     2.564 ms     2.941 ms
 2   ::ffff:0708:0102     (::ffff:0708:0102)     2.754 ms     2.743 ms     2.234 ms
 3   ::ffff:0708:0101     (::ffff:0708:0101)     2.231 ms     2.451 ms     2.701 ms
 4   ::ffff:0708:0202     (::ffff:0708:0202)     2.321 ms !X  2.544 ms !X  2.135 ms !X
```

```
comp1@comp1:~$ traceroute6 ::ffff:0708:0302
traceroute to ::ffff:0708:0302 (::ffff:0708:0302), 30 hops max, 60 byte packets
 1   ::ffff:0708:0101     (::ffff:0708:0101)     2.121 ms     2.954 ms     2.432 ms
 2   ::ffff:0708:0302     (::ffff:0708:0302)     3.012 ms !X  3.111 ms !X  2.811 ms
```

7 ЗАДАНИЕ ПО ВАРИАНТУ (РИСУНОК 4.14)

Для выполнения задания пришлось развернуть ещё пять виртуальных машин pc1, pc2, pc3, pc4, pc5. Пришлось добавить дополнительный путь маршрутизации для pc1 (согласно схеме), который должен связывать с ним сразу 3 машины pc2, pc3 и p4.

Так же присваиваем новые адреса вновь созданным виртуальным машинам.

Для удобства работы сразу перейдем в root, чтобы каждый раз не использовать sudo, как мы это делали ранее.

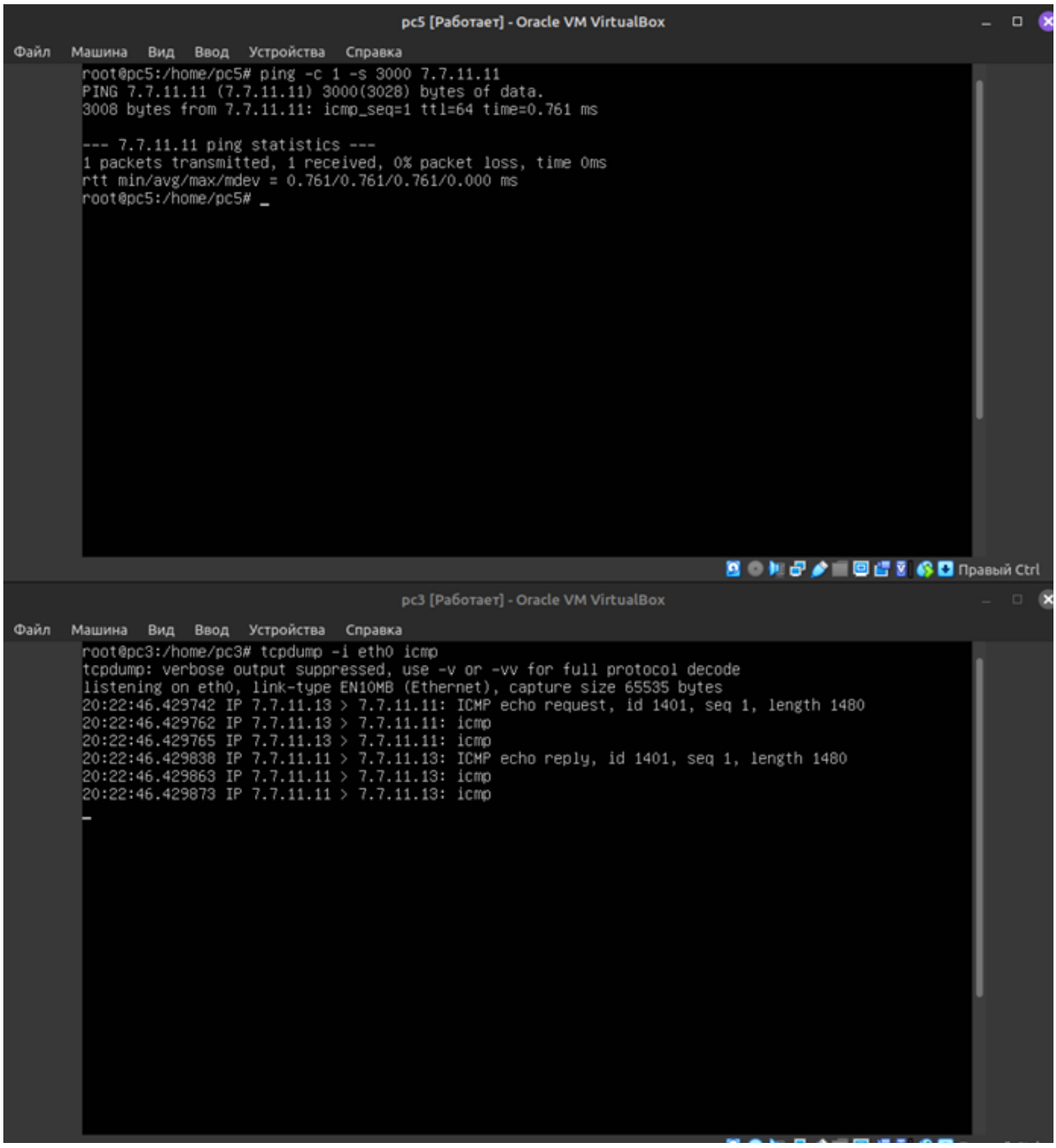
```
root@pc1:/home/pc1# ip route
7.7.11.8/29 dev eth1  proto kernel  scope link  src 7.7.11.9
7.7.11.8/29 dev eth0  proto kernel  scope link  src 7.7.11.9
7.7.11.8/29 dev eth2  proto kernel  scope link  src 7.7.11.9
7.7.11.11
        nexthop via 7.7.11.10  dev eth0 weight 1
        nexthop via 7.7.11.12  dev eth1 weight 1
        nexthop via 7.7.11.13  dev eth2 weight 1
root@pc1:/home/pc1# _
```

После чего для pc1 было настроено правило для рассылке пакетов на адреса 3 смежных машин:

```
root@pc1:/home/pc1# iptables -t mangle -A PREROUTING -s 7.7.11.13 -d 7.7.11.11 -p icmp -j MARK
-mark 1
root@pc1:/home/pc1# iptables -t mangle -A PREROUTING -s 7.7.11.13 -d 7.7.11.11 -p icmp -j MARK
-mark 2
root@pc1:/home/pc1# iptables -t mangle -A PREROUTING -s 7.7.11.13 -d 7.7.11.11 -p icmp -j MARK
-mark 3
root@pc1:/home/pc1# ip rule add fwmark 1 table 100
root@pc1:/home/pc1# ip rule add fwmark 2 table 101
root@pc1:/home/pc1# ip rule add fwmark 3 table 102
root@pc1:/home/pc1# ip route add 7.7.11.11 via 7.7.11.10 dev eth0 table 100
root@pc1:/home/pc1# ip route add 7.7.11.11 via 7.7.11.12 dev eth1 table 101
root@pc1:/home/pc1# ip route add 7.7.11.11 via 7.7.11.13 dev eth2 table 102
root@pc1:/home/pc1#
```

Отправив после этого с pc5 на pc3 пакет из 3000 байт, pc3 получает только пакет меньше 1500 байт, как и должно быть.

Отправка с pc5 на pc3:


pc5 [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
root@pc5:/home/pc5# ping -c 1 -s 3000 7.7.11.11
PING 7.7.11.11 (7.7.11.11) 3000(3028) bytes of data.
3008 bytes from 7.7.11.11: icmp_seq=1 ttl=64 time=0.761 ms
--- 7.7.11.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.761/0.761/0.761/0.000 ms
root@pc5:/home/pc5#
Правый Ctrl
pc3 [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
root@pc3:/home/pc3# tcpdump -i eth0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:22:46.429742 IP 7.7.11.13 > 7.7.11.11: ICMP echo request, id 1401, seq 1, length 1480
20:22:46.429762 IP 7.7.11.13 > 7.7.11.11: icmp
20:22:46.429765 IP 7.7.11.13 > 7.7.11.11: icmp
20:22:46.429838 IP 7.7.11.11 > 7.7.11.13: ICMP echo reply, id 1401, seq 1, length 1480
20:22:46.429863 IP 7.7.11.11 > 7.7.11.13: icmp
20:22:46.429873 IP 7.7.11.11 > 7.7.11.13: icmp

ЗАКЛЮЧЕНИЕ

В процессе выполнения были получены навыки управления вычислительной сетью из 4 хостов такие как: перенаправление трафика через нужные узлы, создание простейших правил фильтрации, настройка маршрутизации в локальных сетях. Основная сложность работы заключается в том, что предоставленные утилиты многофункциональны и сложны, разобраться в них непросто.

К тому же, сложно было догадаться, как при маршрутизации настроить правила хождения трафика, когда у нас есть «узел-перекрёсток», было решено использовать правило пересылки через порт.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Т. И. Алиев, В. В. Соснин, Д. Н. Шинкарук – Компьютерные сети и телекоммуникации: задания и тесты – СПБ: СПБГУ ИТМО, 2018. – 112 с.

2. Т. И. Алиев – Сети ЭВМ и телекоммуникации – СПБ: СПБГУ ИТМО, 2011 – 400 с.

3. Олифер Виктор, Олифер Наталья. Компьютерные сети. Принципы, технологии, протоколы: Юбилейное издание. — СПб.: Питер, 2020. — 1008 с.: ил. — (Серия «Учебник для вузов»)