

Лабораторная работа №5

Дисциплина: Сетевые технологии

Комягин Андрей Николаевич

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
2.1 Простейшая модель сети на базе коммутатора	6
2.1.1 Анализ трафика с помощью Wireshark	8
2.2 Моделирование простейшей сети на базе маршрутизатора FRR	12
2.3 Моделирование простейшей сети на базе маршрутизатора VyOS	16
3 Выводы	19
Список литературы	20

Список иллюстраций

2.1 Размещение устройств	6
2.2 Задание IP-адреса для PC1	7
2.3 Задание IP-адреса для PC2	7
2.4 Проверка работоспособности	8
2.5 Остановка узлов	8
2.6 Запуск анализа. Протокол ARP	9
2.7 ARP. Информация	9
2.8 Опции команды ping	10
2.9 ICMP. Запрос	10
2.10 ICMP. Wireshark	10
2.11 UDP. Запрос	11
2.12 UDP. Wireshark	11
2.13 TCP. Запрос	11
2.14 TCP. Wireshark	12
2.15 Размещение устройств	12
2.16 Настройка IP-адресации PC1	13
2.17 Настройка IP-адресации FRR - вход и hostname	13
2.18 Настройка IP-адресации FRR - интерфейс	14
2.19 Проверка конфигурации и IP-адресации	14
2.20 Проверка подключения	15
2.21 Wireshark	15
2.22 Размещение объектов	16
2.23 IP-адресация для интерфейса узла PC1	17
2.24 Вход	17
2.25 Настройка	18
2.26 Проверка подключения	18
2.27 Wireshark	18

Список таблиц

1 Цель работы

Построение простейших моделей сети на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, анализ трафика посредством Wireshark.

2 Выполнение лабораторной работы

Запустим GNS3 VM и GNS3. Создадим новый проект.

2.1 Простейшая модель сети на базе коммутатора

В рабочей области GNS3 разместим коммутатор Ethernet и два VPCS. Изменим название устройства, включив в имя устройства имя учётной записи. Коммутатору присвоим название `msk-ankomyagin-sw-01`. Соединим VPCS с коммутатором. Отобразим обозначение интерфейсов соединения (рис. 2.1).

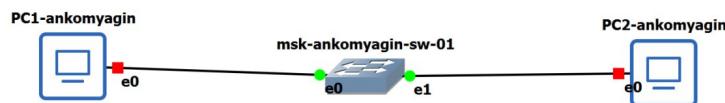
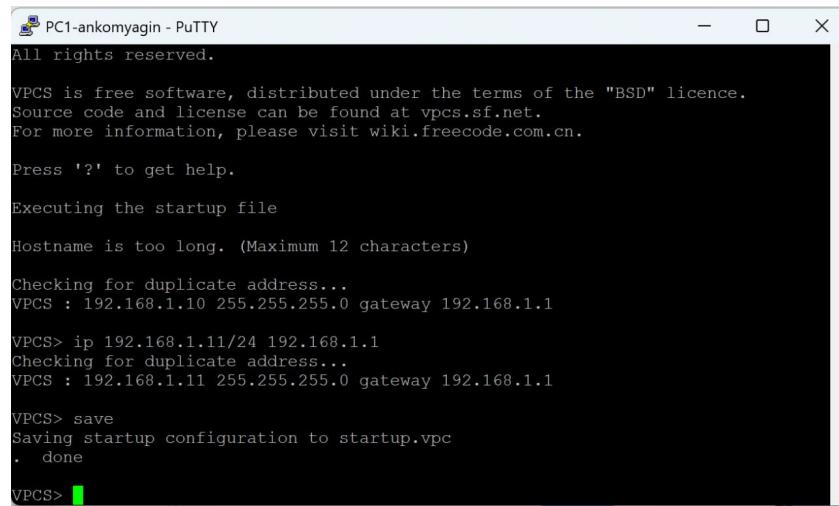


Рис. 2.1: Размещение устройств

Зададим IP-адреса VPCS. Для PC1-ankomyagin зададим IP-адрес 192.168.1.11/24, для PC2-ankomyagin - 192.168.1.12/24. Просмотрим синтаксис и введем команды, сохранив изменения (рис. 2.2 и рис. fig. 2.3).



```
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Hostname is too long. (Maximum 12 characters)

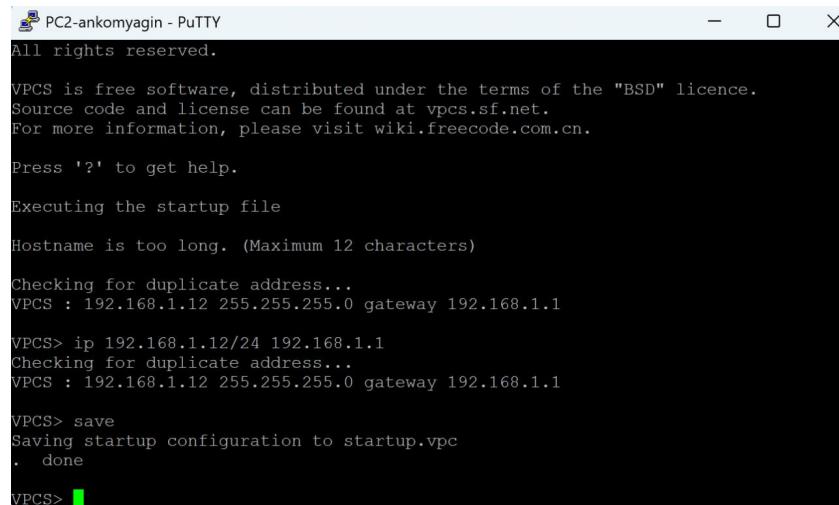
Checking for duplicate address...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

VPCS> ip 192.168.1.11/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS>
```

Рис. 2.2: Задание IP-адреса для PC1



```
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Hostname is too long. (Maximum 12 characters)

Checking for duplicate address...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

VPCS> ip 192.168.1.12/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS>
```

Рис. 2.3: Задание IP-адреса для PC2

Командой ping 192.168.1.11 с PC2-ankomyagin проверяем работоспособность соединения и останавливаем все узлы (рис. 2.4 и рис. 2.5).

```
PC2-ankomyagin - PuTTY
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Hostname is too long. (Maximum 12 characters)

Checking for duplicate address...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

VPCS> ip 192.168.1.12/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.12 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.565 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=5.860 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.852 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=2.554 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=3.719 ms

VPCS>
```

Рис. 2.4: Проверка работоспособности



Рис. 2.5: Остановка узлов

2.1.1 Анализ трафика с помощью Wireshark

Перейдем к анализу трафика в GNS3 посредством Wireshark.

Запускаем анализ трафика, стартируем все узлы и видим информацию по протоколу ARP - были успешно захвачены широковещательные ARP-сообщения, которые устройства рассылают при запуске для объявления своих адресов (рис. 2.6 и рис. 2.7).

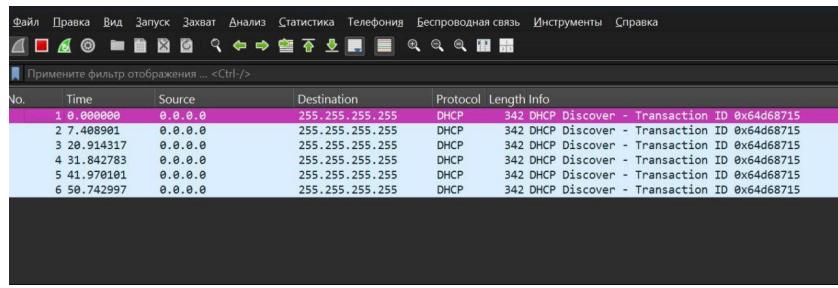


Рис. 2.6: Запуск анализа. Протокол ARP

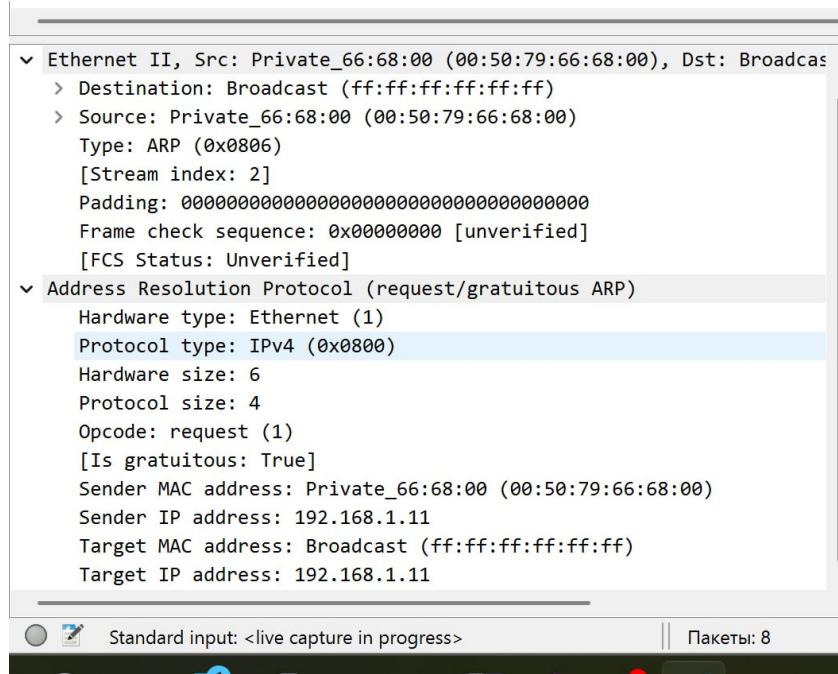


Рис. 2.7: ARP. Информация

В терминале PC-2 посмотрим информацию по опциям команды ping, введя ping /? (рис. 2.8). Затем сделаем эхо-запрос в ICMP-моде к узлу PC-1 (рис. 2.9) и посмотрим окно Wireshark (рис. 2.10).

```
VPCS> ping /?

ping HOST [OPTION ...]
  Ping the network HOST. HOST can be an ip address or name
  Options:
    -1          ICMP mode, default
    -2          UDP mode
    -3          TCP mode
    -c count   Packet count, default 5
    -D          Set the Don't Fragment bit
    -f FLAG    Tcp header FLAG |C|E|U|A|P|R|S|F|
                bits |7 6 5 4 3 2 1 0|
    -i ms      Wait ms milliseconds between sending each packet
    -l size    Data size
    -P protocol Use IP protocol in ping packets
                  1 - ICMP (default), 17 - UDP, 6 - TCP
    -p port    Destination port
```

Рис. 2.8: Опции команды ping

```
VPCS> ping 192.168.1.11 -1

84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=2.125 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=4.885 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=2.261 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=4.543 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=1.992 ms

VPCS>
```

Рис. 2.9: ICMP. Запрос

21	299.401942	Private	66:68:01	Broadcast	ARP	64 Who has 192.168.1.11? Tell 192.168.1.12
22	299.402877	Private	66:68:00	Private	66:68:01	ARP 64 192.168.1.11 is at 00:58:79:66:68:00
23	299.404766	192.168.1.12		192.168.1.11	ICMP	98 Echo (ping) request id=0xab05, seq=1/256, ttl=64 (reply in 24)
24	299.406487	192.168.1.11		192.168.1.12	ICMP	98 Echo (ping) reply id=0xab05, seq=1/256, ttl=64 (request in 23)
25	300.410141	192.168.1.12		192.168.1.11	ICMP	98 Echo (ping) request id=0xacd5, seq=2/512, ttl=64 (reply in 26)
26	300.413061	192.168.1.11		192.168.1.12	ICMP	98 Echo (ping) reply id=0xacd5, seq=2/512, ttl=64 (request in 25)
27	301.417029	192.168.1.12		192.168.1.11	ICMP	98 Echo (ping) request id=0xadd5, seq=3/768, ttl=64 (reply in 28)
28	301.417640	192.168.1.11		192.168.1.12	ICMP	98 Echo (ping) reply id=0xadd5, seq=3/768, ttl=64 (request in 27)
29	302.421827	192.168.1.12		192.168.1.11	ICMP	98 Echo (ping) request id=0xaea5, seq=4/1024, ttl=64 (reply in 30)
30	302.423317	192.168.1.11		192.168.1.12	ICMP	98 Echo (ping) reply id=0xaea5, seq=4/1024, ttl=64 (request in 29)
31	303.428119	192.168.1.12		192.168.1.11	ICMP	98 Echo (ping) request id=0xafd5, seq=5/1280, ttl=64 (reply in 32)
32	303.429538	192.168.1.11		192.168.1.12	ICMP	98 Echo (ping) reply id=0xafd5, seq=5/1280, ttl=64 (request in 31)

Рис. 2.10: ICMP. Wireshark

Аналогично сделаем эхо-запросы в UDP-моде и в TCP-моде к узлу PC-1 (рис. 2.11 и рис. 2.12, рис. 2.13 и рис. 2.14). Были захвачены и проанализированы ICMP-пакеты, наглядно демонстрирующие процесс проверки связности с помощью утилиты ping. Виден полный цикл: предварительный ARP-запрос для определения MAC-адреса и последующий обмен эхо-запросами и ответами. Было успешно захвачено и проанализировано взаимодействие по протоколу TCP. Видны все этапы: “тройное рукопожатие” для установки соединения, передача данных с подтверждением и корректное завершение сеанса. Остановим захват пакетов.

```
VPCS> ping 192.168.1.11 -2
84 bytes from 192.168.1.11 udp_seq=1 ttl=64 time=1.285 ms
84 bytes from 192.168.1.11 udp_seq=2 ttl=64 time=4.175 ms
84 bytes from 192.168.1.11 udp_seq=3 ttl=64 time=3.538 ms
84 bytes from 192.168.1.11 udp_seq=4 ttl=64 time=3.742 ms
84 bytes from 192.168.1.11 udp_seq=5 ttl=64 time=3.516 ms
VPCS>
```

Рис. 2.11: UDP. Запрос

33	383.226204	192.168.1.12	192.168.1.11	ECHO	98 Request
34	383.227010	192.168.1.11	192.168.1.12	ECHO	98 Response
35	384.230252	192.168.1.12	192.168.1.11	ECHO	98 Request
36	384.232445	192.168.1.11	192.168.1.12	ECHO	98 Response
37	385.234518	192.168.1.12	192.168.1.11	ECHO	98 Request
38	385.236110	192.168.1.11	192.168.1.12	ECHO	98 Response
39	386.239953	192.168.1.12	192.168.1.11	ECHO	98 Request
40	386.241225	192.168.1.11	192.168.1.12	ECHO	98 Response
41	387.245618	192.168.1.12	192.168.1.11	ECHO	98 Request
42	387.247249	192.168.1.11	192.168.1.12	ECHO	98 Response

Рис. 2.12: UDP. Wireshark

```
VPCS> ping 192.168.1.11 -3
Connect 7@192.168.1.11 seq=1 ttl=64 time=2.101 ms
SendData 7@192.168.1.11 seq=1 ttl=64 time=1.441 ms
Close 7@192.168.1.11 seq=1 ttl=64 time=4.948 ms
Connect 7@192.168.1.11 seq=2 ttl=64 time=6.275 ms
SendData 7@192.168.1.11 seq=2 ttl=64 time=4.350 ms
Close 7@192.168.1.11 seq=2 ttl=64 time=7.061 ms
Connect 7@192.168.1.11 seq=3 ttl=64 time=6.294 ms
SendData 7@192.168.1.11 seq=3 ttl=64 time=1.584 ms
Close 7@192.168.1.11 seq=3 ttl=64 time=8.123 ms
Connect 7@192.168.1.11 seq=4 ttl=64 time=6.274 ms
SendData 7@192.168.1.11 seq=4 ttl=64 time=4.331 ms
Close 7@192.168.1.11 seq=4 ttl=64 time=5.504 ms
Connect 7@192.168.1.11 seq=5 ttl=64 time=4.592 ms
SendData 7@192.168.1.11 seq=5 ttl=64 time=3.086 ms
Close 7@192.168.1.11 seq=5 ttl=64 time=7.091 ms
VPCS>
```

Рис. 2.13: TCP. Запрос

43	416..689274	192.168.1.12	192.168.1.11	TCP	74	35934 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSecr=0
44	416..689667	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
45	416..611206	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSecr=0
46	416..612892	192.168.1.12	192.168.1.11	ECHO	122	Request
47	416..613053	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=57 Win=2920 Len=0
48	416..618926	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSecr=0
49	416..619143	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=58 Win=2920 Len=0
50	416..619485	192.168.1.11	192.168.1.12	TCP	66	35934 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSecr=0
51	416..624418	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSecr=0
52	417..624649	192.168.1.12	192.168.1.11	TCP	74	[TCP Port numbers reused] 35934 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1
53	417..626822	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
54	417..630139	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSecr=0
55	417..633457	192.168.1.12	192.168.1.11	ECHO	122	Request
56	417..635508	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=57 Win=2920 Len=0
57	417..642422	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSecr=0
58	417..643374	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=58 Win=2920 Len=0
59	417..643866	192.168.1.11	192.168.1.12	TCP	66	35934 → 7 [FIN, ACK] Seq=1 Ack=58 Win=2920 Len=0
60	417..649112	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSecr=0
61	418..652808	192.168.1.12	192.168.1.11	TCP	74	[TCP Port numbers reused] 35934 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1
62	418..655366	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
63	418..656238	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSecr=0
64	418..659727	192.168.1.12	192.168.1.11	ECHO	122	Request
65	418..660188	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=57 Win=2920 Len=0
66	418..666118	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSecr=0
67	418..668117	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=58 Win=2920 Len=0
68	418..668266	192.168.1.11	192.168.1.12	TCP	66	35934 → 7 [FIN, ACK] Seq=1 Ack=58 Win=2920 Len=0
69	418..673757	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSecr=0
70	419..677226	192.168.1.12	192.168.1.11	TCP	74	[TCP Port numbers reused] 35934 → 7 [SYN] Seq=0 Win=2920 Len=0 MSS=1
71	419..679414	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
72	419..681758	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSecr=0
73	419..686375	192.168.1.12	192.168.1.11	ECHO	122	Request
74	419..688551	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=57 Win=2920 Len=0
75	419..694872	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSecr=0
76	419..695201	192.168.1.11	192.168.1.12	TCP	54	7 → 35934 [ACK] Seq=1 Ack=58 Win=2920 Len=0
77	419..695236	192.168.1.11	192.168.1.12	TCP	66	35934 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSecr=0
78	419..698766	192.168.1.12	192.168.1.11	TCP	66	35934 → 7 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSecr=0

Рис. 2.14: TCP. Wireshark

2.2 Моделирование простейшей сети на базе маршрутизатора FRR

Далее перейдем к моделированию простейшей сети на базе маршрутизатора FRR в GNS3.

Создаем новый проект, разместив VPCS, коммутатор Ethernet и маршрутизатор FRR (рис. 2.15).

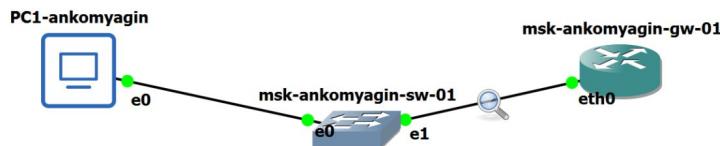
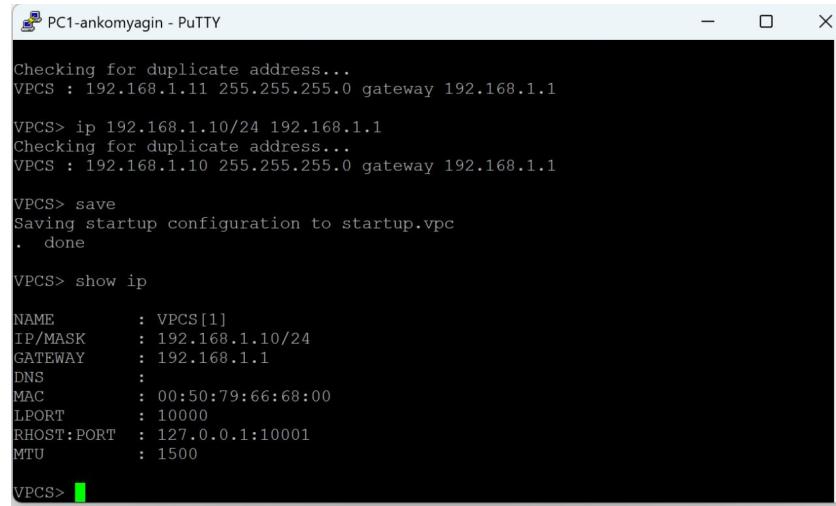


Рис. 2.15: Размещение устройств

Включим захват трафика на соединении между коммутатором и маршрутизатором. Запустим все устройства проекта. Откроем консоль всех устройств

проекта и настроим IP-адресацию для интерфейса узла PC1-ankomyagin: ip 192.168.1.10/24 192.168.1.1 (рис. 2.16).



```
Checking for duplicate address...
VPCS : 192.168.1.11 255.255.255.0 gateway 192.168.1.1

VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

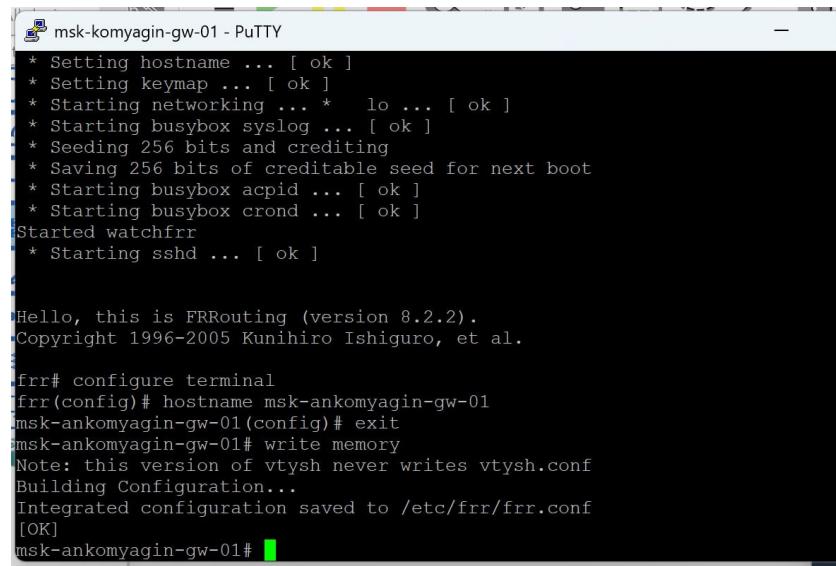
VPCS> show ip

NAME      : VPCS[1]
IP/MASK   : 192.168.1.10/24
GATEWAY   : 192.168.1.1
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 10000
RHOST:PORT: 127.0.0.1:10001
MTU       : 1500

VPCS>
```

Рис. 2.16: Настройка IP-адресации PC1

Настроим IP-адресацию для интерфейса локальной сети маршрутизатора, дав имя узла msk-ankomyagin-gw-01 (рис. 2.17 и рис. 2.18).

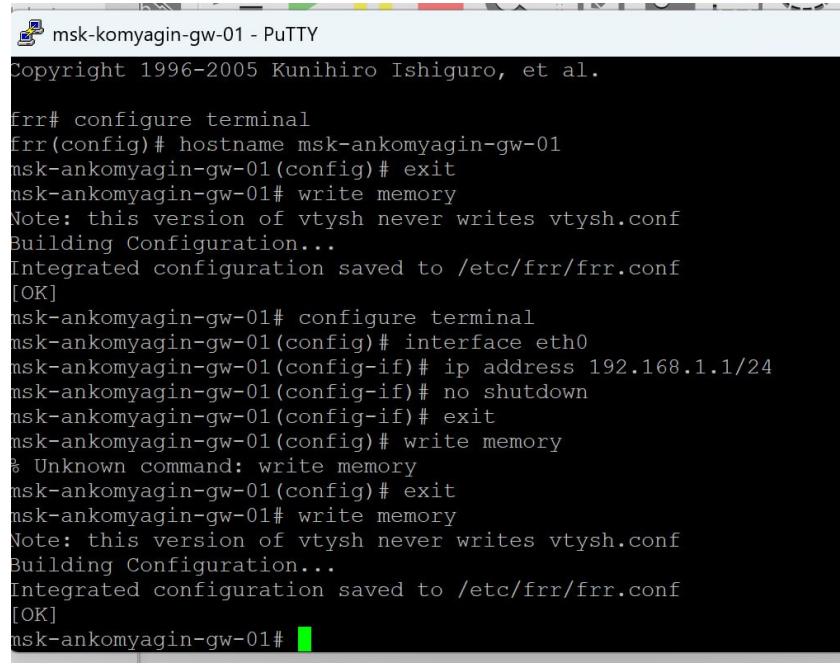


```
* Setting hostname ... [ ok ]
* Setting keymap ... [ ok ]
* Starting networking ... * lo ... [ ok ]
* Starting busybox syslog ... [ ok ]
* Seeding 256 bits and crediting
* Saving 256 bits of creditable seed for next boot
* Starting busybox acpid ... [ ok ]
* Starting busybox crond ... [ ok ]
Started watchfrr
* Starting sshd ... [ ok ]

Hello, this is FRRouting (version 8.2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr# configure terminal
frr(config)# hostname msk-ankomyagin-gw-01
msk-ankomyagin-gw-01(config)# exit
msk-ankomyagin-gw-01# write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-ankomyagin-gw-01#
```

Рис. 2.17: Настройка IP-адресации FRR - вход и hostname

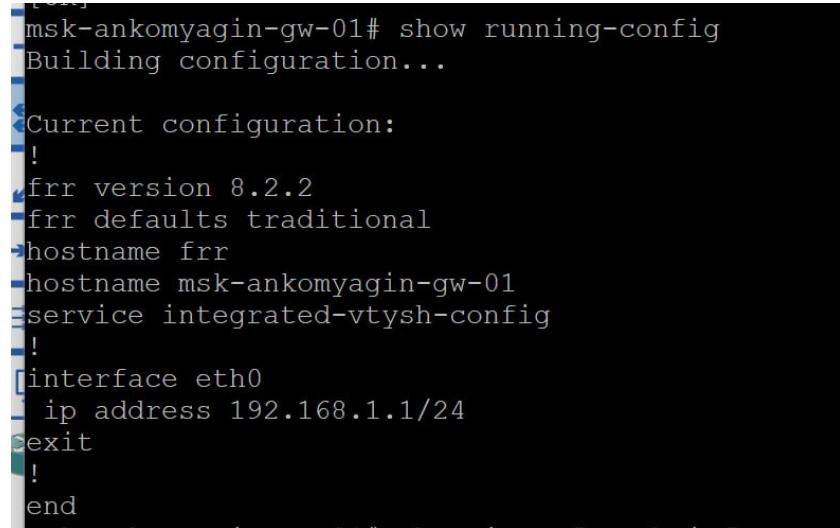


```
msk-komyagin-gw-01 - PuTTY
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr# configure terminal
frr(config)# hostname msk-ankomyagin-gw-01
msk-ankomyagin-gw-01(config)# exit
msk-ankomyagin-gw-01# write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-ankomyagin-gw-01# configure terminal
msk-ankomyagin-gw-01(config)# interface eth0
msk-ankomyagin-gw-01(config-if)# ip address 192.168.1.1/24
msk-ankomyagin-gw-01(config-if)# no shutdown
msk-ankomyagin-gw-01(config-if)# exit
msk-ankomyagin-gw-01(config)# write memory
% Unknown command: write memory
msk-ankomyagin-gw-01(config)# exit
msk-ankomyagin-gw-01# write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
msk-ankomyagin-gw-01#
```

Рис. 2.18: Настройка IP-адресации FRR - интерфейс

Проверим конфигурацию маршрутизатора с помощью `show running-config` и `show interface brief`, а также проверим подключение, выполнив `ping` с PC1 на шлюз 192.168.1.1 (рис. 2.19 и рис. 2.20).



```
msk-ankomyagin-gw-01# show running-config
Building configuration...

Current configuration:
!
frr version 8.2.2
frr defaults traditional
hostname frr
hostname msk-ankomyagin-gw-01
service integrated-vtysh-config
!
interface eth0
  ip address 192.168.1.1/24
exit
!
end
```

Рис. 2.19: Проверка конфигурации и IP-адресации

```

Press '?' to get help.

Executing the startup file

Hostname is too long. (Maximum 12 characters)

VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> ping 192.168.1.1

84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=11.249 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=4.823 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=2.991 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=3.522 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=2.823 ms

VPCS>

```

Рис. 2.20: Проверка подключения

Просмотрим информацию в окне Wireshark. В захвате видны ARP-запросы для определения MAC-адресов PC1 (192.168.1.10) и шлюза (192.168.1.1), а также последующий обмен ICMP-пакетами (Echo request/reply) (рис. 2.21).

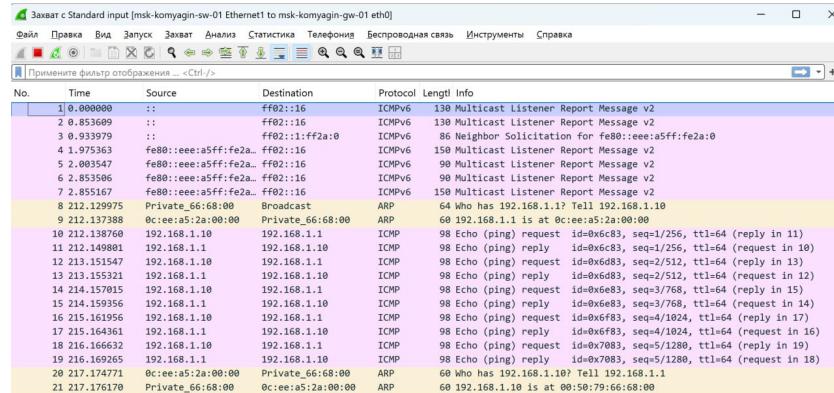


Рис. 2.21: Wireshark

2.3 Моделирование простейшей сети на базе маршрутизатора VyOS

Наконец, перейдем к моделированию простейшей сети на базе маршрутизатора VyOS в GNS3.

Создаем новый проект и размещаем объекты - VPCS, коммутатор Ethernet и маршрутизатор VyOS (рис. 2.22).



Рис. 2.22: Размещение объектов

Включаем захват трафика, запускаем все устройства проекта и настраиваем IP-адресацию для интерфейса узла PC1-ankomyagin: ip 192.168.1.10/24 192.168.1.1 (рис. 2.23).

```

VPCS> ip 192.168.1.10/24 192.168.1.1
Checking for duplicate address...
VPCS : 192.168.1.10 255.255.255.0 gateway 192.168.1.1

VPCS> save
Saving startup configuration to startup.vpc
. done

VPCS> show ip

NAME      : VPCS[1]
IP/MASK   : 192.168.1.10/24
GATEWAY   : 192.168.1.1
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 10004
RHOST:PORT: 127.0.0.1:10005
MTU       : 1500

VPCS>

```

Рис. 2.23: IP-адресация для интерфейса узла PC1

Далее настраиваем маршрутизатор VyOS. Вводим логин, пароль (рис. 2.24), после проверки установки в режиме конфигурирования изменяем имя устройства на `msk-ankomyagin-gw-01`, задаем ip-адрес `192.168.1.1/24`, просматриваем и применяем изменения (рис. 2.25).

```

[ 19.109236] systemd[1]: Created slice system-serial\x2dgetty.slice.
[ 19.116722] systemd[1]: Listening on initctl Compatibility Named Pipe.
[ 19.125560] systemd[1]: Listening on Journal Socket (/dev/log).
[ 19.249824] bridge: filtering via arp/ip/ip6tables is no longer available by
default. Update your scripts to load br_netfilter if you need this.
[ 19.322138] Bridge firewalls registered
[ 19.378706] mpls_gso: MPLS GSO support

Welcome to VyOS - vyos ttyS0

vyos login: vyos
Password:

Login incorrect
vyos login: vyos
Password:
Welcome to VyOS!

```

Рис. 2.24: Вход

```

[edit]
vyos@vyos# commit
Can't configure both static IPv4 and DHCP address on the same interface
[[interfaces ethernet eth0]] failed
Commit failed
[edit]
vyos@vyos# delete interfaces ethernet eth0 address dhcp
[edit]
vyos@vyos# set interfaces ethernet eth0 address 192.168.1.1/24
    Configuration path: [interfaces ethernet eth0 address 192.168.1.1/24] already
exists
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos#

```

Рис. 2.25: Настройка

Проверяем подключение. Узел PC1-ankomyagin должен успешно отправлять эхо-запросы на адрес маршрутизатора 192.168.1.1 (рис. 2.26).

```

VPCS> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=64 time=13.095 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=64 time=3.422 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=64 time=3.916 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=64 time=10.886 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=64 time=3.146 ms
VPCS>

```

Рис. 2.26: Проверка подключения

Просмотрим информацию в окне Wireshark, остановим захват и завершим работу (рис. 2.27).

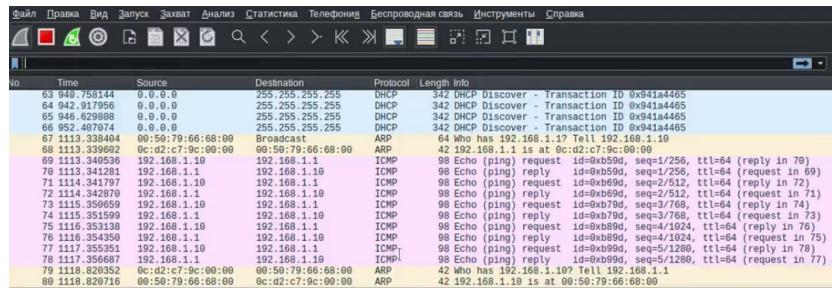


Рис. 2.27: Wireshark

3 Выводы

В ходе работы были получены навыки по построению простейших моделей сети на базе коммутатора и маршрутизаторов FRR и VyOS в GNS3, произведен анализ трафика (ARP, ICMP, UDP, TCP) посредством Wireshark.

Список литературы

ТУИС