Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ
к лабораторной работе №1
на тему

МАРШРУТИРИЗАЦИЯ В ЛОКАЛЬНОЙ СЕТИ

Выполнил: студент гр.253504
Фроленко К.Ю.
Проверил: ассистент кафедры информатики
Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

1 ФОРМУЛИРОВКА ЗАДАЧИ

В рамках данной работы необходимо создать виртуальную локальную сеть, обеспечив возможность взаимодействия между устройствами посредством статической *IP*-адресации. В качестве среды для эмуляции сети предлагается использовать контейнеризированный подход, позволяющий гибко управлять конфигурацией и параметрами сети без использования физического оборудования.

Предполагается развертывание нескольких виртуальных узлов, каждый из которых должен быть оснащен уникальным статическим IP-адресом в заданном диапазоне. Следует настроить механизмы маршрутизации и обеспечить корректное функционирование сетевого взаимодействия. Для проверки работоспособности сети требуется выполнить тестирование передачи данных между устройствами, используя стандартные сетевые инструменты, а также убедиться в их доступности из основной операционной системы.

Особое внимание следует уделить корректности сетевых настроек, а также проверке маршрутов передачи данных между узлами. В ходе эксперимента необходимо зафиксировать полученные результаты и проанализировать возможные проблемы, связанные с конфигурацией сети, их причины и способы устранения.

2 ХОД РАБОТЫ

Создадим локальную сеть с фиксированной *IP*-адресацией, что позволит обеспечить стабильное взаимодействие между виртуальными устройствами. Для этого необходимо задать определенный диапазон *IP*-адресов и убедиться, что сеть была успешно создана. После выполнения этой операции можно проверить список доступных сетей, чтобы убедиться в наличии новой сети и отсутствии ненужных записей. Результаты данной проверки представлены на Рисунке 1.

```
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
87c26dbe5191   bridge    bridge    local
e3342f556add   host      host      local
db99add6bf6a   none      null      local
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker network create --subnet=192.168.1.0/24 Custom_network
75763f49a37fb0c164b9fefed04c53d15319faf09351839160c8163fb5579b5c
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker network ls
NETWORK ID     NAME             DRIVER    SCOPE
75763f49a37f   Custom_network   bridge    local
87c26dbe5191   bridge           bridge    local
e3342f556add   host             host      local
db99add6bf6a   none             null      local
```

Рисунок 1 – Создание сети

После создания сети необходимо развернуть два контейнера с операционной системой *Ubuntu*, которые будут взаимодействовать в данной среде. Каждому контейнеру назначается статический *IP*-адрес в пределах ранее заданного диапазона, что обеспечивает их уникальную идентификацию внутри сети. Проверка успешного назначения *IP*-адресов выполняется путем просмотра сетевых интерфейсов запущенных контейнеров, как показано на Рисунке 2.

```
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker run -dit --name ubuntu_server1 --network Custom_network --ip 192.168.1.100 ubuntu bash
8d5a56854d57bc12fda4b6d35e360c89477c1d3a2d73b3c8a88ad6bba0014ed6
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker run -dit --name ubuntu_server2 --network Custom_network --ip 192.168.1.101 ubuntu bash
e643f08fd69d30ee832765bb4c1762589d5297cc82e8b086179103cc094519f1
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED              STATUS              PORTS     NAMES
e643f08fd69d   ubuntu    "bash"    58 seconds ago       Up 58 seconds               ubuntu_server2
8d5a56854d57   ubuntu    "bash"    About a minute ago   Up About a minute           ubuntu_server1
```

Рисунок 2 – Создание контейнеров в созданной сети

После создания контейнеров необходимо выполнить их первоначальную настройку, установив утилиты, необходимые для работы с сетью. Для этого осуществляется вход в каждый контейнер, после чего устанавливаются инструменты, такие как *iproute2* для управления сетевыми интерфейсами и *iputils-ping* для проверки соединений. Установка данных пакетов позволяет в дальнейшем выполнить диагностику сети и убедиться в правильности конфигурации. Данный этап представлен на Рисунке 3.

```
PS C:\Workspace\BSUIR-Labs\ISOB\Lab1> docker exec -it ubuntu_server1 bash
root@8d5a56854d57:/# apt update && apt install -y iproute2 iputils-ping net-tools traceroute
```

Рисунок 3 – Настройка контейнеров

После установки необходимых утилит в контейнерах можно приступить к проверке сетевых параметров. Для этого необходимо выполнить просмотр доступных сетевых интерфейсов, определить назначенные *IP*-адреса и убедиться в корректности сетевой конфигурации. Данный этап позволяет подтвердить, что контейнеры действительно подключены к созданной сети и имеют статические *IP*-адреса, соответствующие заданному диапазону. Также необходимо проверить маршрут по умолчанию и убедиться, что каждый контейнер имеет доступ к сети через назначенный шлюз. Результаты данной проверки представлены на Рисунке 4.

```
root@8d5a56854d57:/# ip addr show eth0
48: eth0@if49: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:a8:01:64 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.1.100/24 brd 192.168.1.255 scope global eth0
       valid_lft forever preferred_lft forever
root@8d5a56854d57:/# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 02:42:c0:a8:01:64  txqueuelen 0  (Ethernet)
        RX packets 21979  bytes 32001822 (32.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7513  bytes 502850 (502.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 4 – Проверка правильности создания контейнера

После подтверждения правильности сетевых настроек выполняется тестирование взаимодействия контейнеров. Для этого проверяется возможность передачи данных между ними с использованием стандартных сетевых инструментов. В первую очередь осуществляется проверка доступности контейнеров с помощью *ICMP*-запросов *ping*, что позволяет убедиться, что пакеты успешно достигают целевого контейнера. Затем выполняется трассировка маршрута *traceroute*, которая показывает путь прохождения пакетов внутри созданной сети. Данный этап подтверждает корректную работу сети и возможность взаимодействия между контейнерами. Результаты проверки представлены на Рисунке 5.

```
root@8d5a56854d57:/# ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.168 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.162 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=0.139 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=64 time=0.139 ms
^C
--- 192.168.1.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4140ms
rtt min/avg/max/mdev = 0.139/0.152/0.168/0.011 ms
root@8d5a56854d57:/# traceroute 192.168.1.101
traceroute to 192.168.1.101 (192.168.1.101), 30 hops max, 60 byte packets
 1  ubuntu_server2.Custom_network (192.168.1.101)  0.337 ms  0.274 ms  0.254 ms
```

Рисунок 5 – Проверка отправки запросов и нахождение пути

ЗАКЛЮЧЕНИЕ

В ходе данной работы была создана виртуальная локальная сеть с фиксированной IP-адресацией, что позволило обеспечить стабильное взаимодействие между виртуальными устройствами. Для реализации данной задачи использовался контейнеризированный подход, благодаря которому удалось гибко управлять параметрами сети и конфигурацией узлов без использования физического оборудования.

На первом этапе была развернута виртуальная сеть, назначены диапазоны *IP*-адресов и проверено её успешное создание. Далее были созданы два контейнера с операционной системой *Ubuntu*, которым вручную назначены статические *IP*-адреса в пределах заданной сети. После этого была проведена их первоначальная настройка, включающая установку необходимых сетевых утилит, что позволило в дальнейшем выполнить диагностику сети.

Проверка сетевых параметров контейнеров показала, что каждому из них корректно присвоены статические *IP*-адреса, и они находятся в пределах созданной сети. Дополнительно была проведена диагностика сетевого взаимодействия, включающая проверку доступности контейнеров с помощью *ICMP*-запросов *ping* и трассировку маршрута *traceroute*. В результате было подтверждено, что контейнеры успешно обмениваются данными, что свидетельствует о корректной настройке сети.

Таким образом, поставленная задача была успешно выполнена. Виртуальная сеть с фиксированной IP-адресацией создана, сетевые параметры контейнеров проверены, а их взаимодействие протестировано с использованием стандартных инструментов диагностики. Полученные результаты подтверждают корректную работу сети и возможность передачи данных между виртуальными узлами.