МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №2

«Настройка сетевой системы ОС Linux»

по дисциплине

«Администрирование систем и сетей»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Кочешков А. А.\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнаков К. М.

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_19-ВМ\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2023

Цель работы

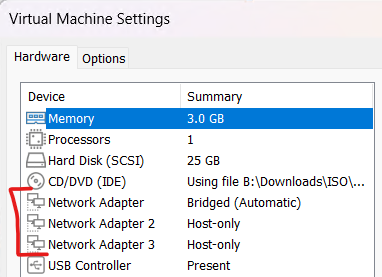
Изучение установки и конфигурирования сетевого интерфейса, маршрутизации и контроля сетевых связей в ОС Linux

Ход работы

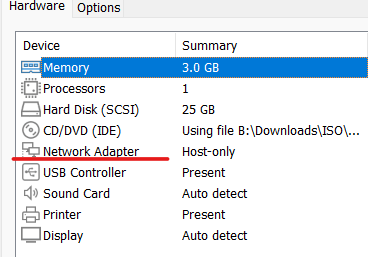
Конфигурирование сетевого интерфейса

В работе используются три виртуальные машины на основе ОС Astra Linux Орел с именами: astra, astra2-poly и astra3-poly.

Для начала работы необходимо одному узлы добавить два новых сетевых интерфейса, в итоге получим: Bridged и 2 Host-only.



На двух других установим по одному сетевому интерфейсу Host-only:



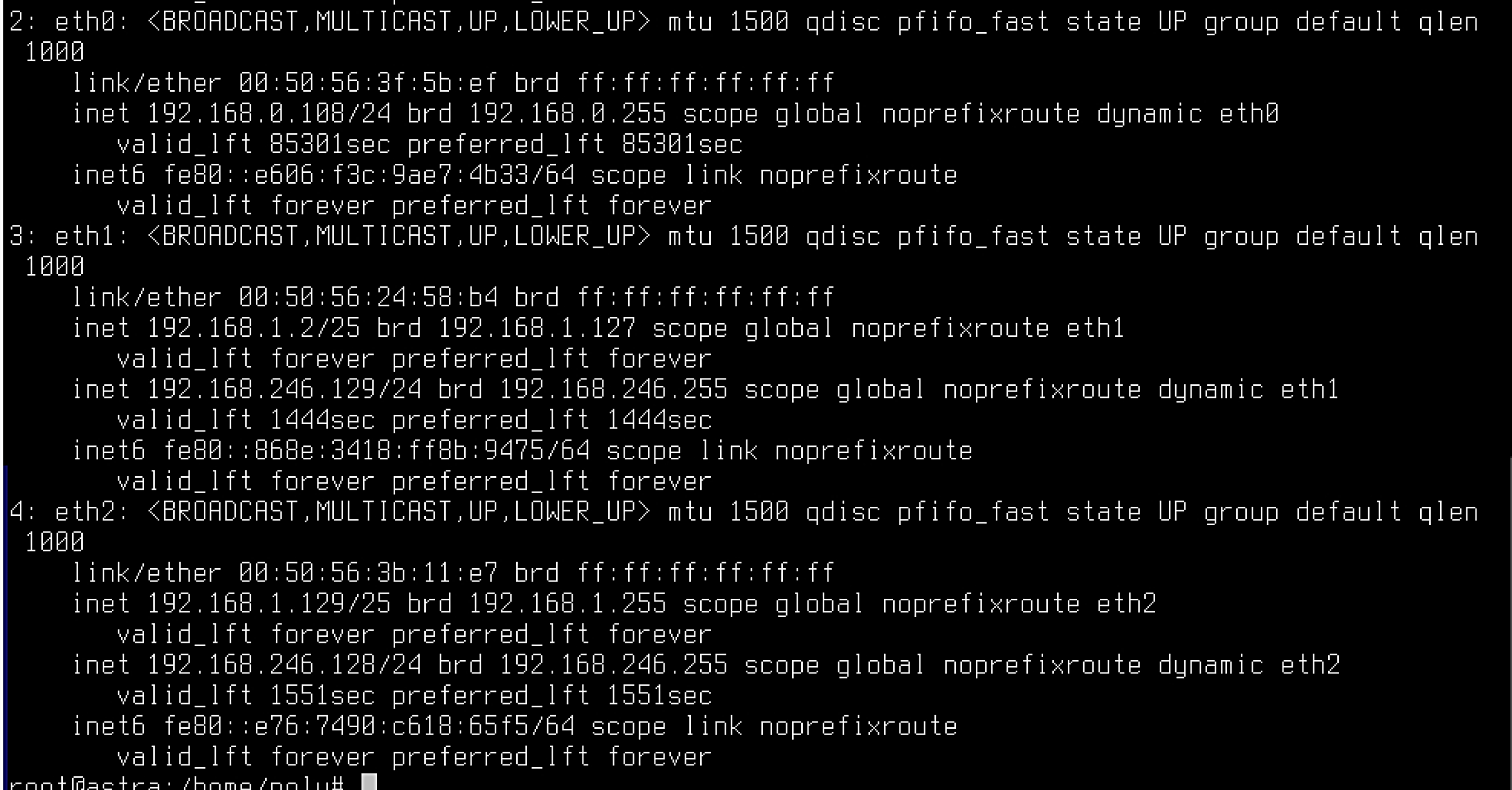
С помощью команды ifconfig зададим статический IP-адрес и маску для виртуальной машины astra на сетевом интерфейсе eth0, который подключен как сетевой мост и служит для связи с основной машиной, а также имеет доступ в интернет. Ip = 192.168.0.108, Netmask = 255.255.255.0, Broadcast = 192.168.0.255.



Настроим интерфейс eth1, который будет служить для связи с виртуальной машиной astra2-poly. Ip = 192.168.1.2, Netmask = 255.255.255.128, Broadcast = 192.168.1.127.



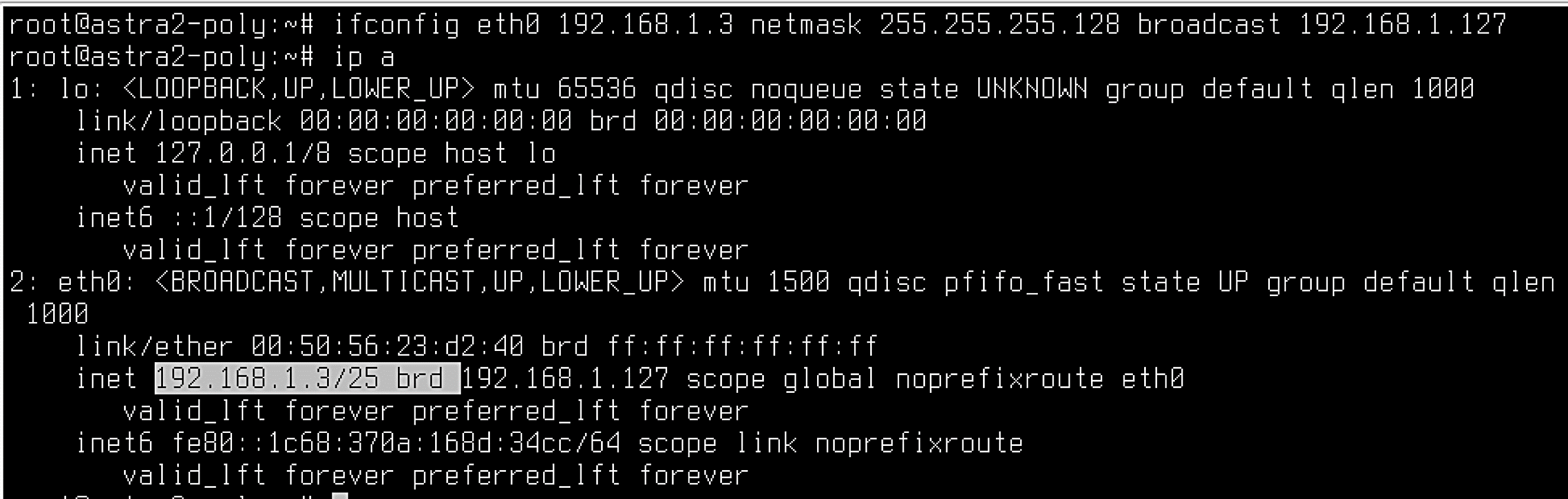
Выведем настройки сети для astra, увидим, что изменения успешно применились для всех сетевых интерфейсов:



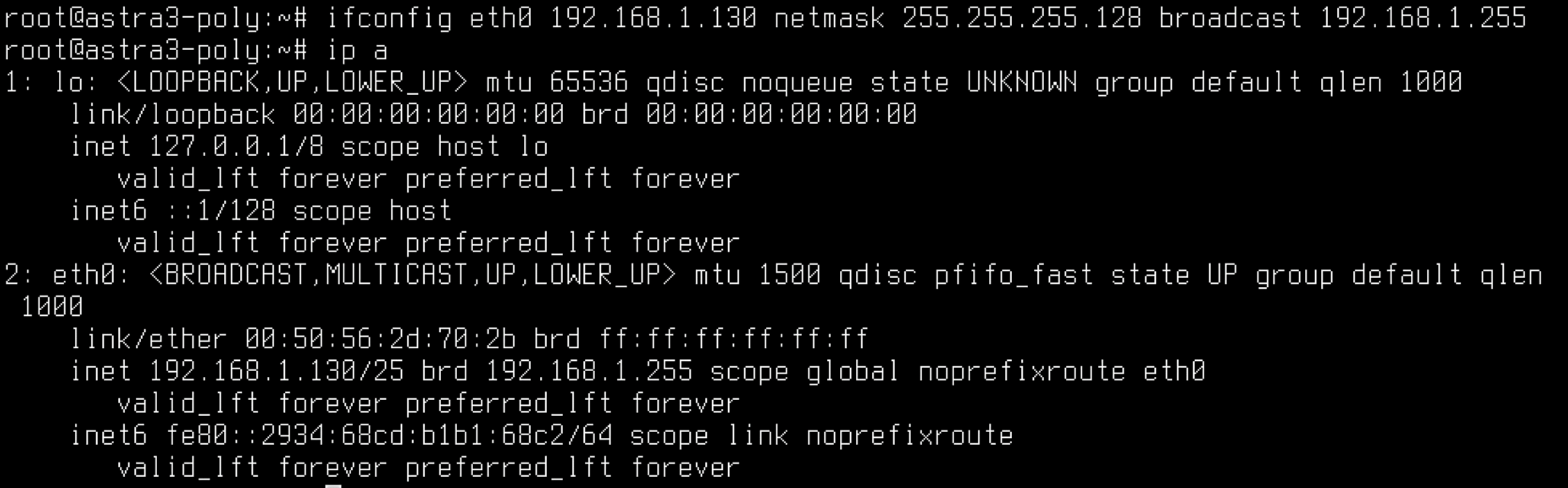
Для сетевого интерфейса eth2 настроим доступ для виртуальной машины astra3-poly. Ip = 192.168.1.129, Netmask = 255.255.255.128, Broadcast = 192.168.1.255.



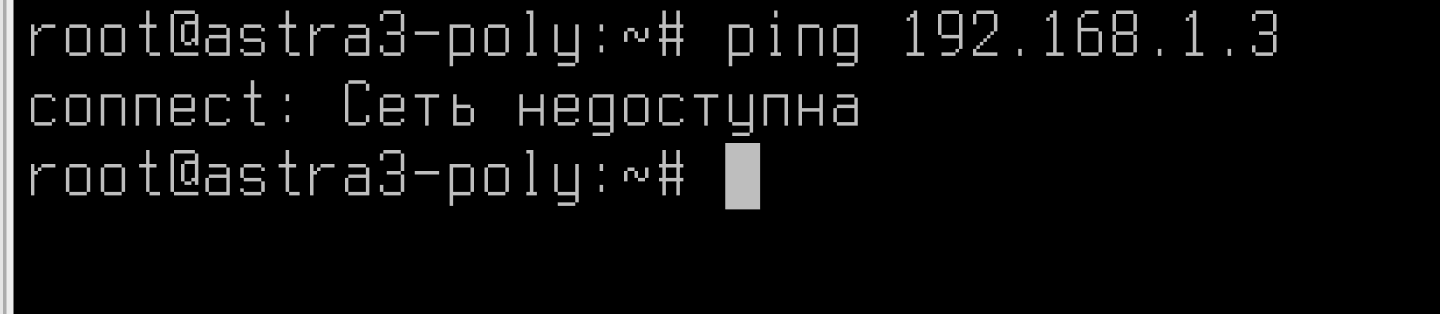
На второй машине astra2-poly настроим сетевой интерфейс eth0, которая будет входить в сеть 192.168.1.0/25, brd = 192.168.1.127 и Ip = 192.168.1.3.



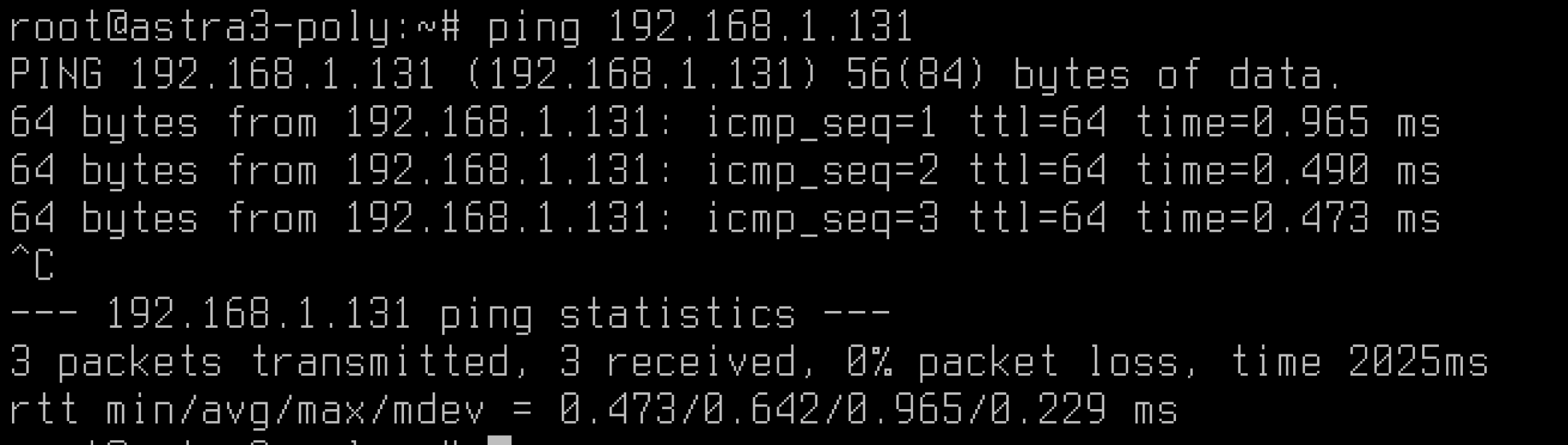
На третьей машине установим интерфейс eth0 со следующими параметрами: Ip = 192.168.1.130/25, brd = 192.168.1.255.



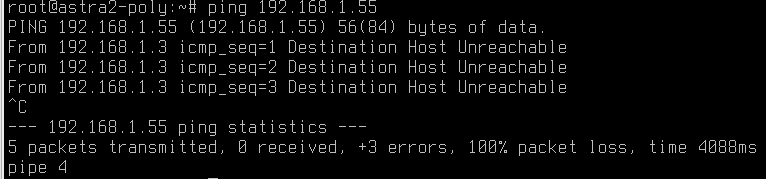
Попробуем с третьей машины проверить подключение ко второй.



Получили сообщение, что сеть не доступна, проверим подключение между третьей и первой, которая настроена с тремя сетевыми интерфейсами.

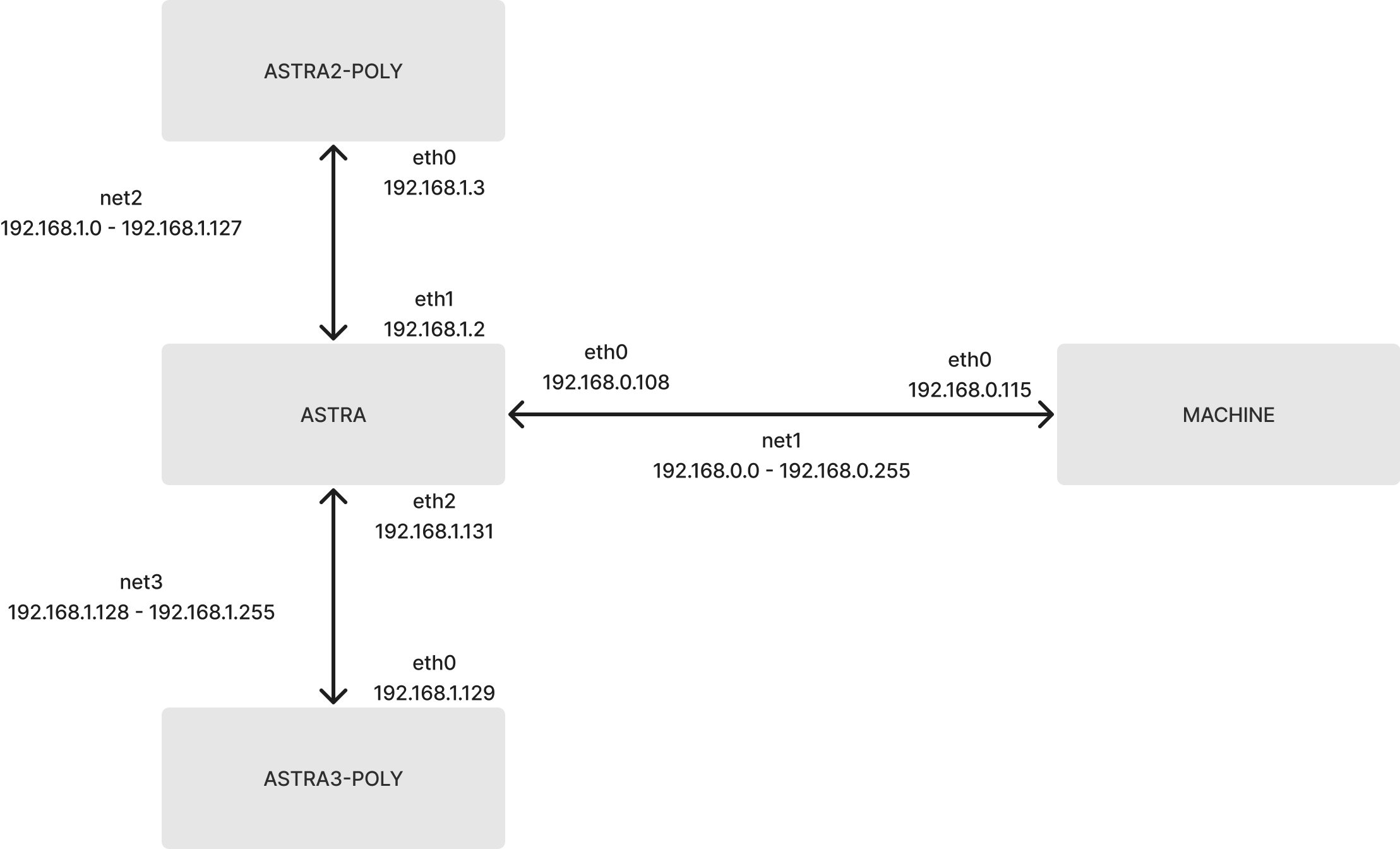


Попробуем со второй машины установить соединение по адресу, которого не существует в сети.



Получили сообщение, что хост не доступен в этой сети.

Составим схему сети после проделанных настроек.



Управление пространством сетевых имен

В сетях UNIX используется система именования узлов DNS, которая предоставляет полное доменное имя FQDN для каждого компьютера, состоящее из локального имени хоста и имени домена. В локальной сети можно использовать сокращенное локальное имя. Для преобразования сетевого имени хоста в IP-адрес должен быть настроен механизм разрешения (трансляции).

В локальной сети для разрешения сетевого имени хоста в IP-адрес могут быть использованы следующие решения:

* Файл /etc/hosts: это локальный текстовый файл, в котором можно определить связь между именем хоста и его IP-адресом. Этот файл расположен на каждом компьютере в локальной сети и позволяет решать имена хостов без использования DNS.
* DNS-сервер: это сервер, который обрабатывает DNS-запросы и возвращает соответствующие IP-адреса. Он используется для разрешения сетевых имён хостов в IP-адреса в локальной и глобальной сети. В локальной сети обычно устанавливается DNS-сервер, который хранит информацию о всех компьютерах в локальной сети.
* NetBIOS-имена: это метод именования компьютеров в сети, который используется в Microsoft Windows. Это позволяет обращаться к компьютеру по его имени NetBIOS вместо IP-адреса. Для разрешения NetBIOS-имён в IP-адрес используется служба WINS (Windows Internet Name Service), которая работает на сервере.
* mDNS (Multicast DNS): это протокол, позволяющий компьютерам в локальной сети автоматически обнаруживать друг друга и разрешать имена без использования централизованного DNS-сервера. Компьютеры в локальной сети отправляют широковещательные запросы для разрешения имен хостов.

Для нашей задачи воспользуемся первым вариантом и рассмотрим файлы hosts и hostname, которые расположены в директории /etc/.

В Linux для конфигурации компьютера в сети используется несколько файлов. Некоторые из них:

/etc/hosts - содержит таблицу соответствия между IP-адресами и именами хостов в локальной сети. Этот файл используется для быстрого разрешения имен хостов без обращения к DNS-серверу.

/etc/hostname - содержит имя хоста (hostname) системы, которое используется при загрузке системы и для локальной сети.

/etc/resolv.conf - определяет параметры настройки DNS-сервера, который используется для разрешения имен хостов в IP-адреса.

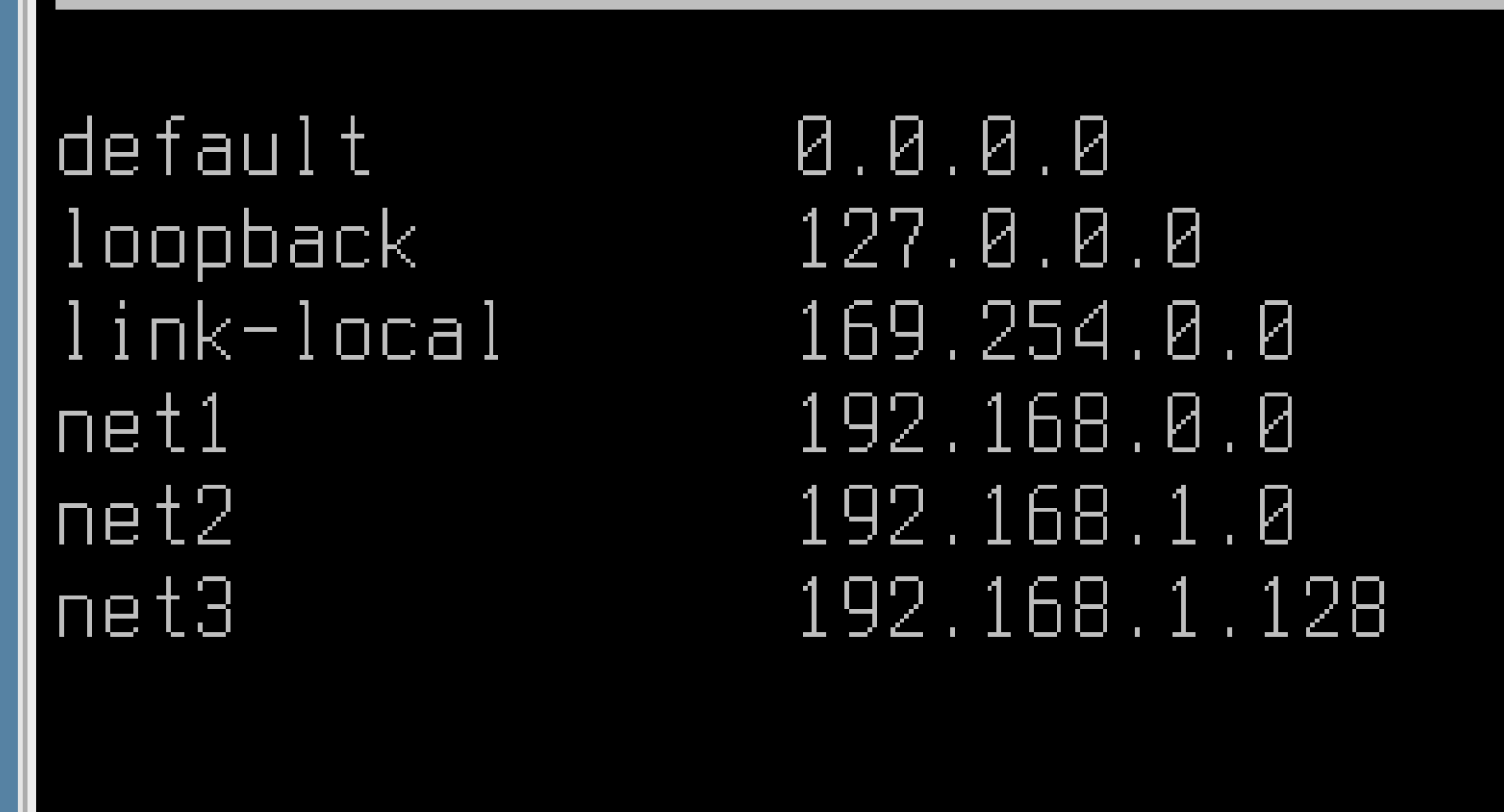
/etc/hosts.allow и /etc/hosts.deny - содержат список правил доступа для различных сетевых служб, таких как SSH, FTP, SMTP и т.д. Они позволяют настраивать доступ к сетевым сервисам на основе адреса IP или имени хоста.

/etc/host.conf - определяет порядок разрешения имен хостов в системе. Он содержит параметры, такие как "order", "multi", "spoof" и другие, которые контролируют использование локальных таблиц и DNS-серверов при разрешении имен хостов.

/etc/resolv.conf - содержит информацию о DNS-серверах, которые используются для разрешения имен в IP-адреса в локальной сети. Первый указанный DNS-сервер будет использован для разрешения имен в IP-адреса, если он не отвечает, то будет использован второй указанный DNS-сервер.

/etc/networks в Linux используется для определения сетей и связанных с ними атрибутов, таких как имя сети и IP-диапазон. Этот файл используется в различных сетевых утилитах, включая команду route, для определения маршрутов и сопоставления имен сетей с IP-адресами. Он может быть использован для определения различных сетевых параметров в многопользовательской среде.

Пропишем настройки во всех узлах.

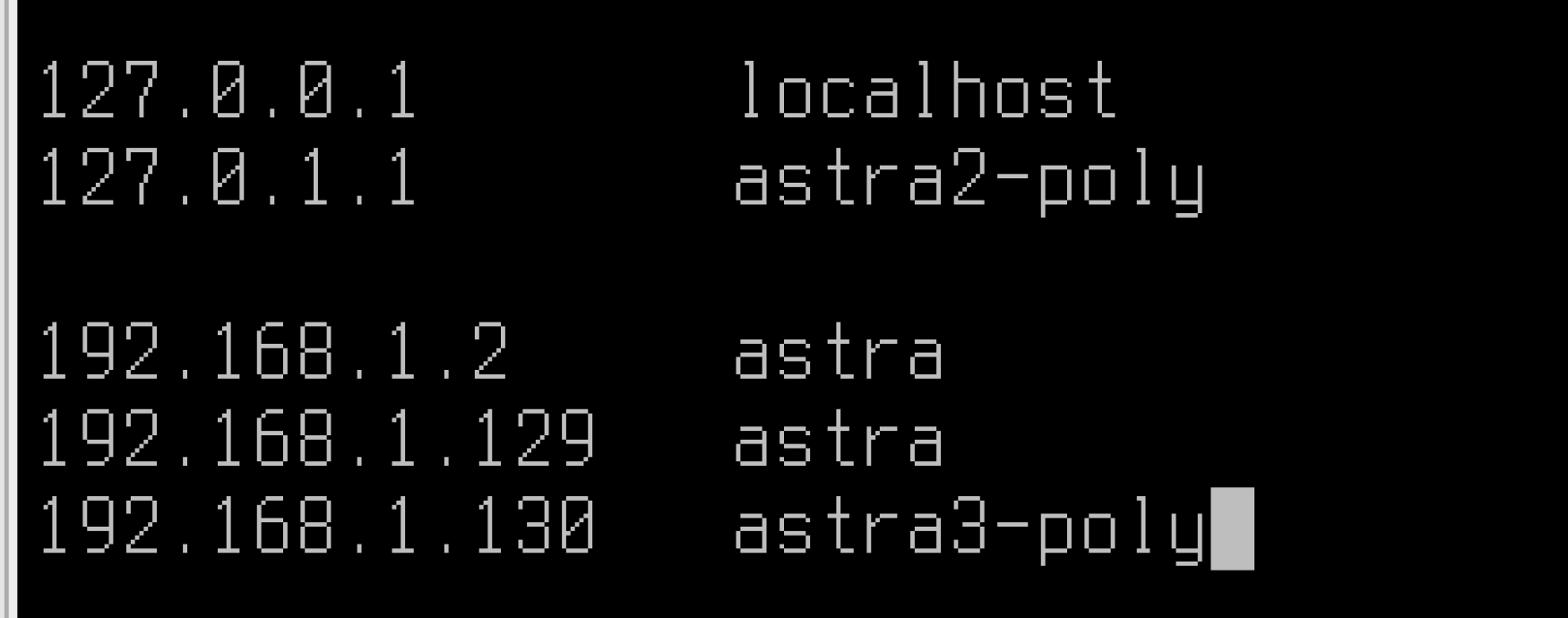


В файл /etc/hosts добавим имя и IP адреса “соседних” узлов.

ASTRA



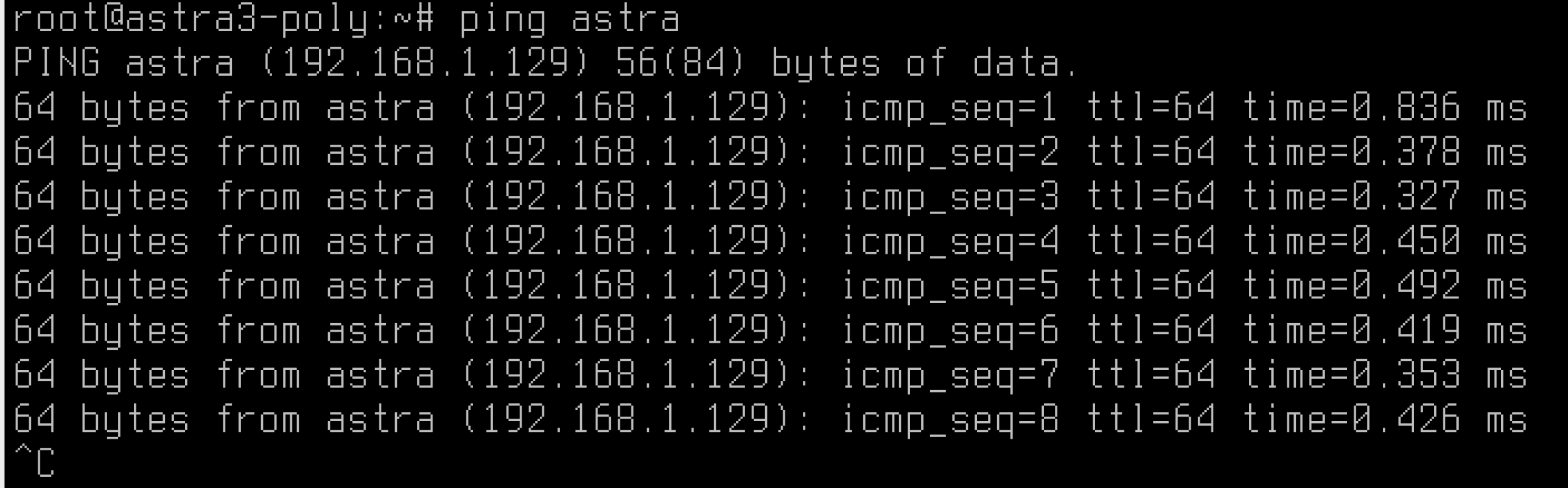
ASTRA2-POLY

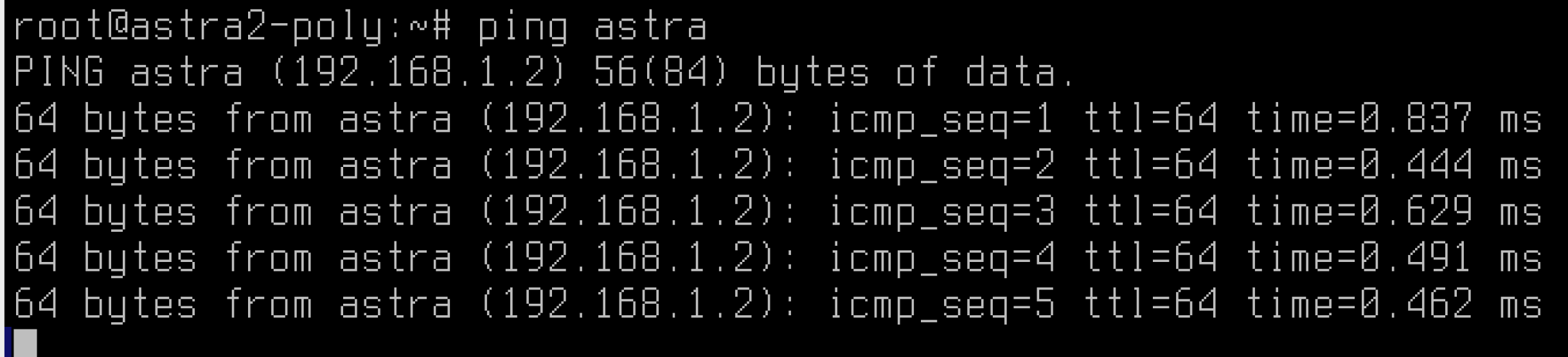


ASTRA3-POLY



Проверим доступ к хостам по их имени.





Доступ между узлами, находящимися в одной сети, успешно установлено.

Формирование подсетей и маршрутизация.

Команда route в Linux используется для просмотра и управления таблицами маршрутизации IP-пакетов в системе. Таблица маршрутизации содержит информацию о маршрутах, которые используются для пересылки IP-пакетов между различными сетевыми интерфейсами и хостами в сети.

Команда route без аргументов выводит текущую таблицу маршрутизации. С помощью различных опций, таких как add, del и change, можно добавлять, удалять и изменять маршруты в таблице маршрутизации. Например, с помощью команды route add можно добавить новый маршрут в таблицу маршрутизации, указав IP-адрес целевой сети, маску подсети и шлюз по умолчанию для доставки пакетов до этой сети.

Кроме того, команда route позволяет настраивать множество других параметров, связанных с маршрутизацией, таких как максимальное количество переходов до целевого хоста (таймауты), предпочтительный интерфейс для исходящих пакетов и т.д.

Посмотрим таблицу маршрутизации на узле ASTRA.

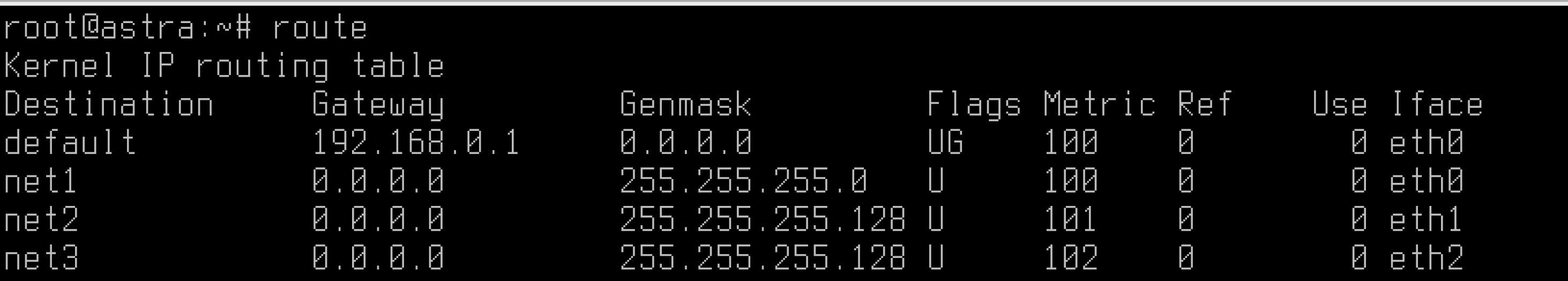
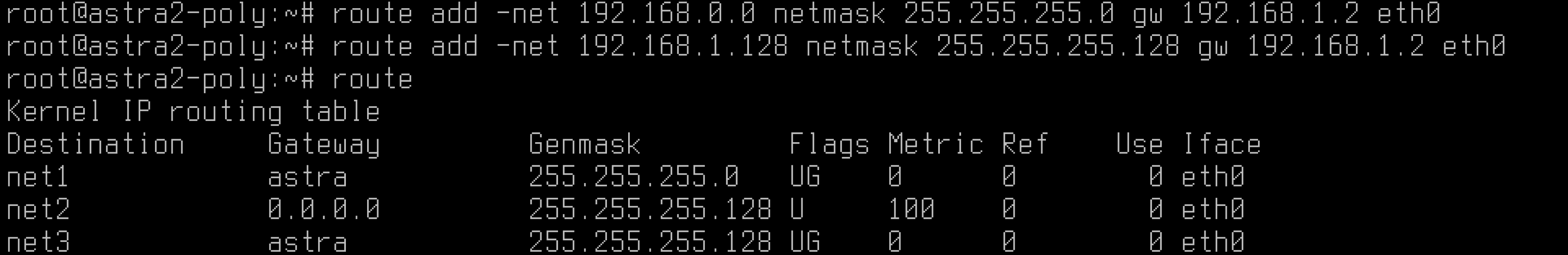


Таблица маршрутизации машины ASTRA2-POLY.

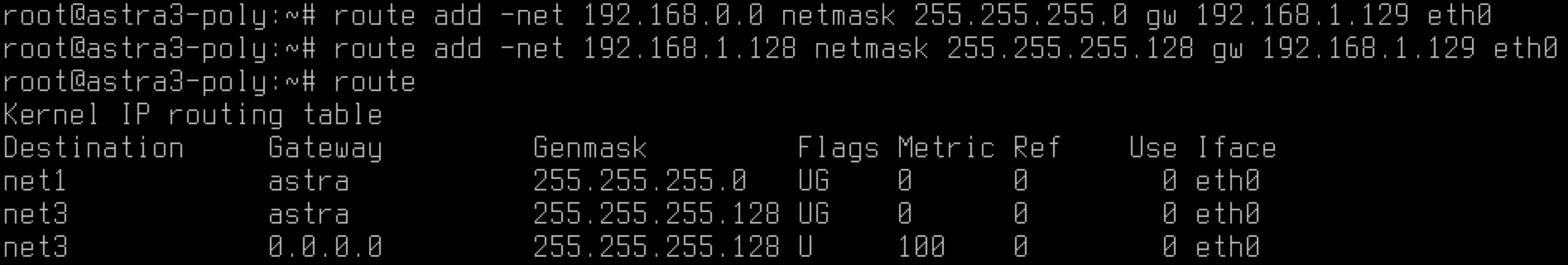


Добавим маршруты к сетям net1 и net3 и посмотрим результат.

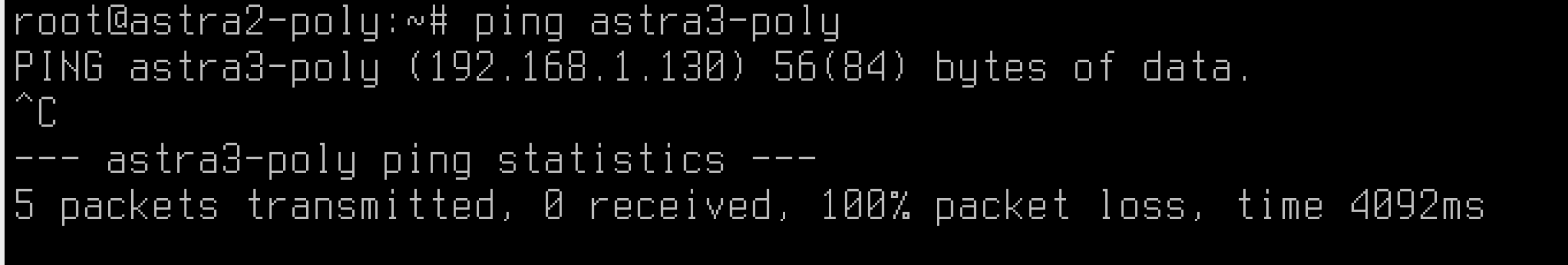


В столбце Flags значение U обозначает, что он включен, а G – используется шлюз.

Аналогично настроим таблицу маршрутизации для машины ASTRA3-POLY.



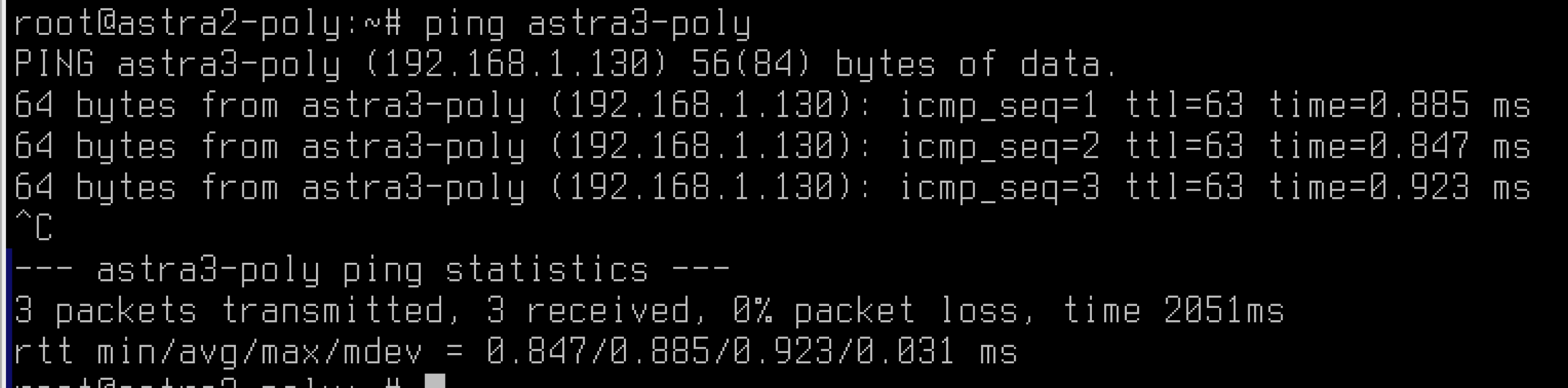
При попытке установить соединение между net2 и net3 возникнет “зависание” команды ping, это происходит из-за того, что пакеты не доходят до узла. Пример попытки установки соединения между третьей и второй машиной.

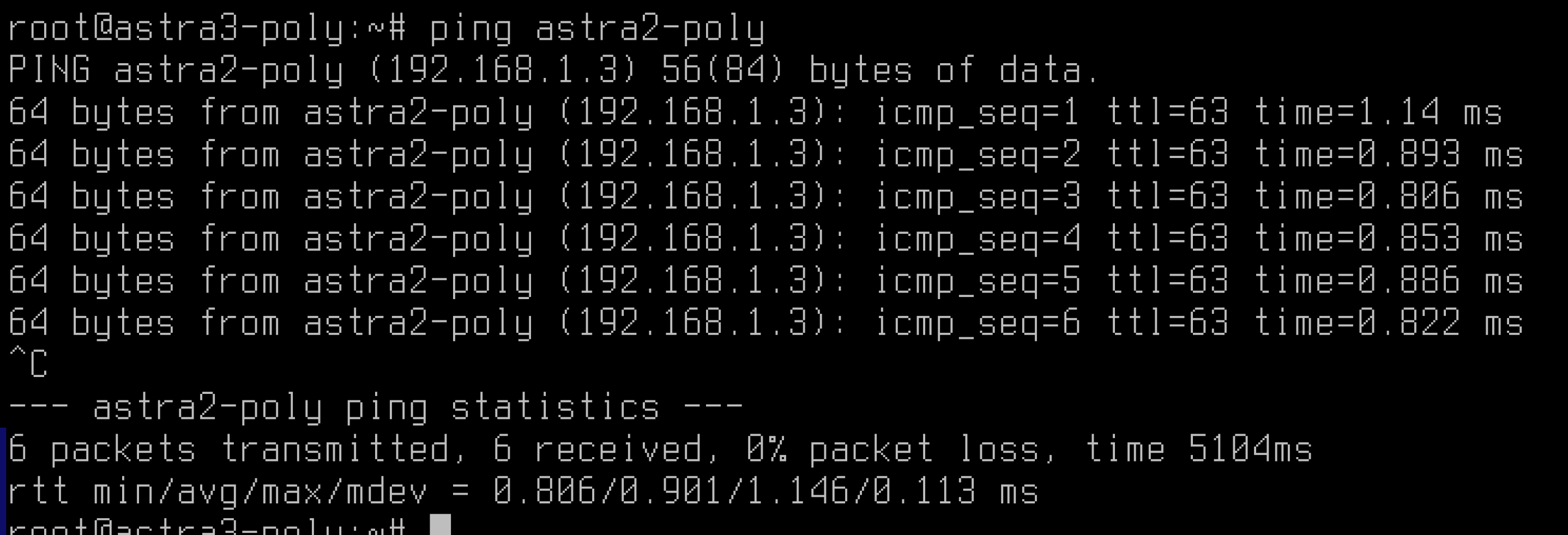


Это происходит потому, что на машине ASTRA отключена пересылка пакетов между сетевыми интерфейсами. Для того чтобы включить его в файл /proc/sys/net/ipv4/ip\_forward необходимо прописать значение 1.



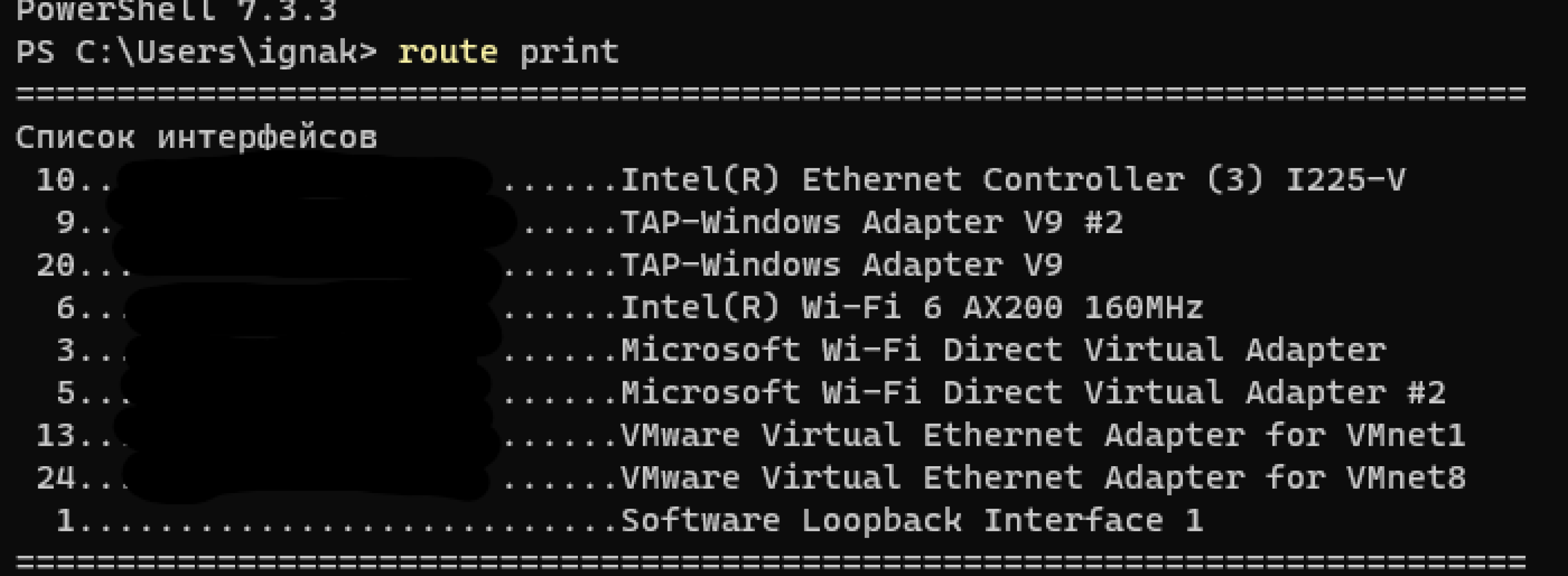
Попробуем еще раз установить соединение между ASTRA2-POLY и ASTRA3-POLY.



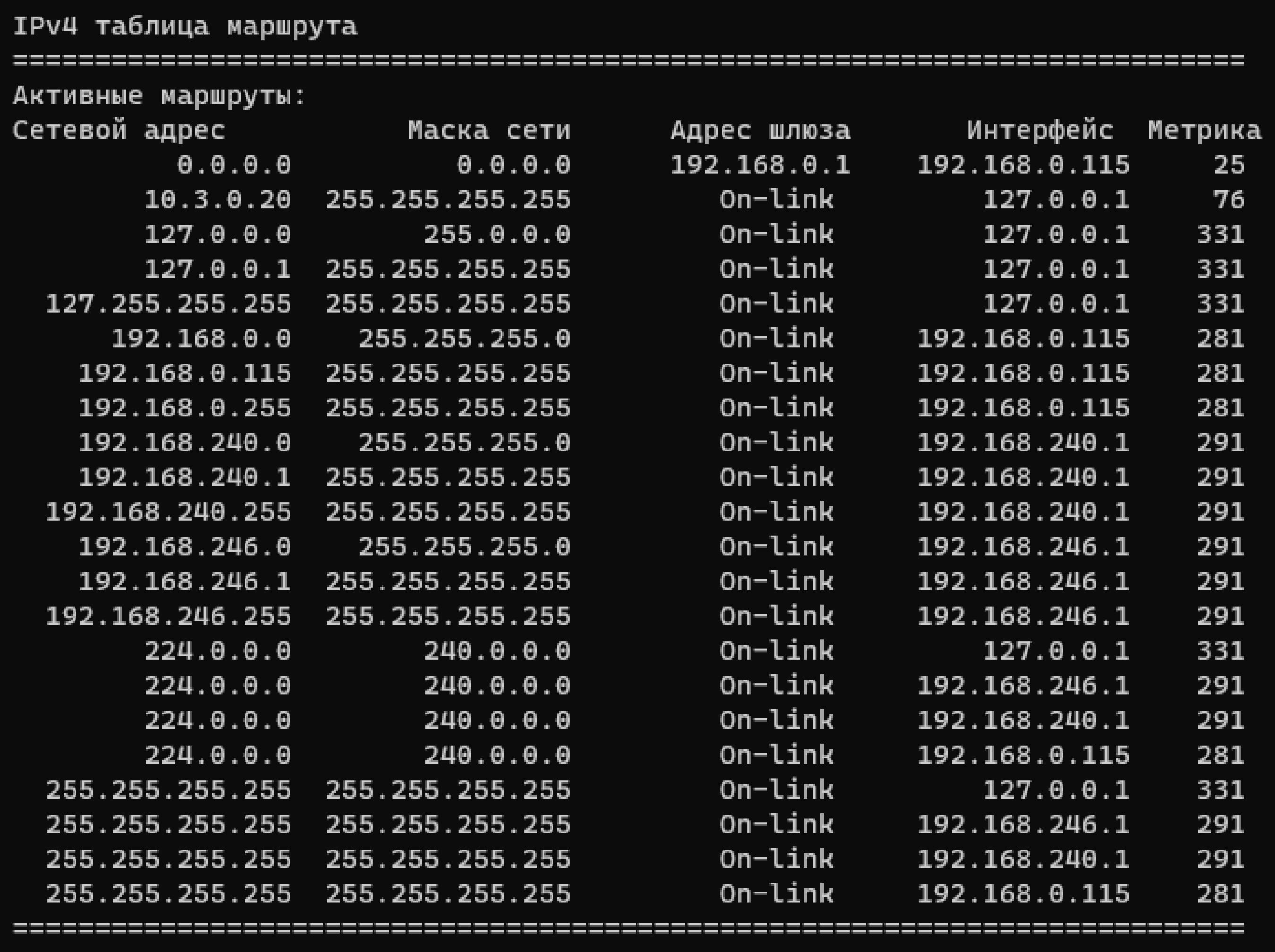


Соединение прошло успешно, узлы “видят” друг друга. Как это работает? Сначала пакет проходит через шлюз ASTRA, затем пакет идет в другую сеть, то есть ASTRA3-POLY -> ASTRA -> ASTRA2-POLY и обратно.

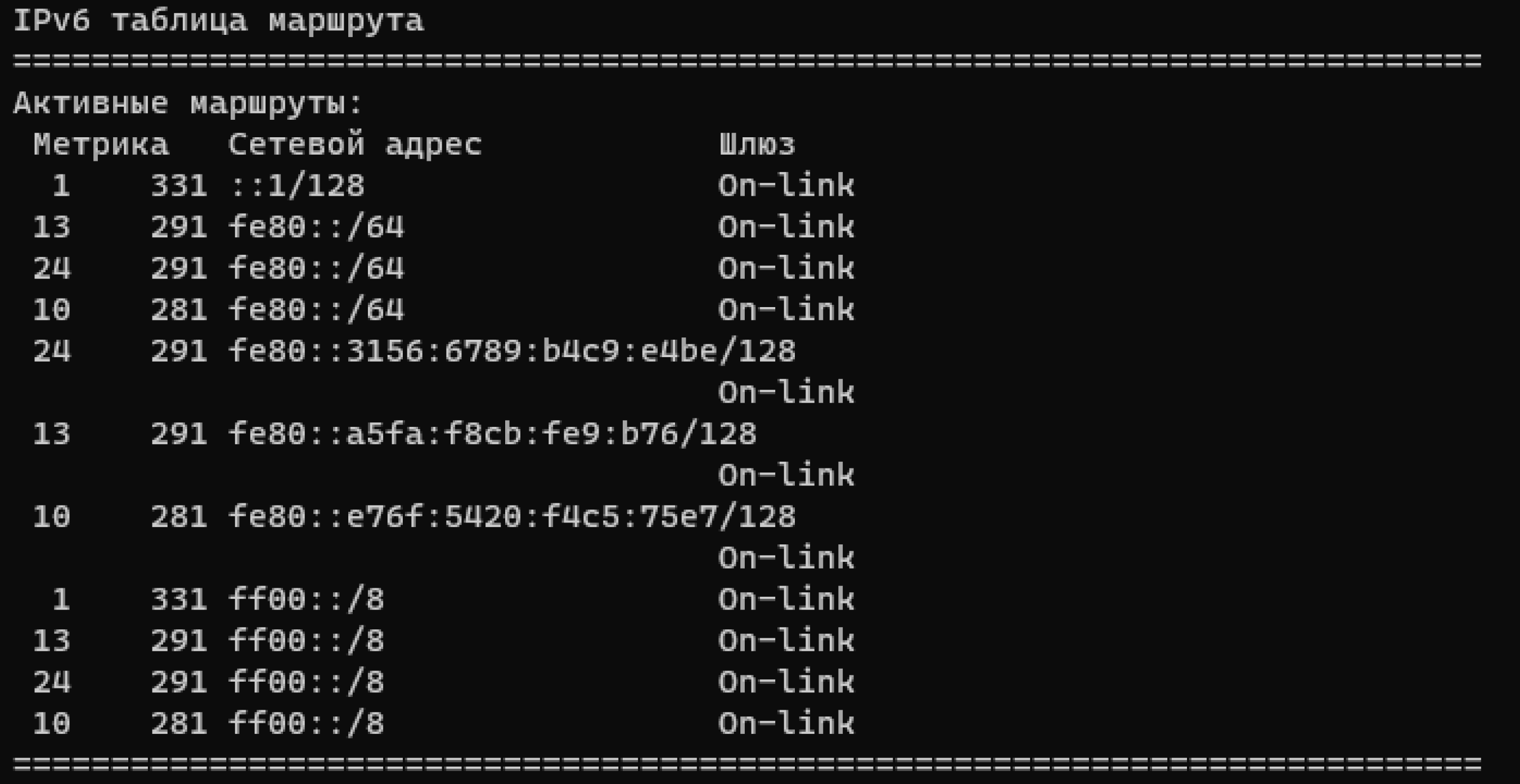
Рассмотрим таблицу маршрутизации на домашней системе Windows с помощью команды route print. В которой отображается ID, MAC адрес и название интерфейса.



Далее в вывод этой команды отображается таблица маршрута IPv4 сети, в “Метирика” хранится числовое значение флагов.

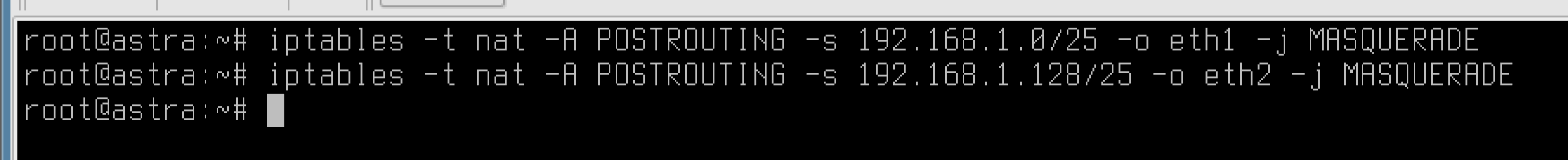


При настройке IPv6 маршрутизации будет так же отображено в выводе этой команды.



Реализовать настройку IP-сетей по внешним правилам для использования шлюза во внешней сети и Internet

Выполним следующую команду для трансляции адресов NAT на машине ASTRA.

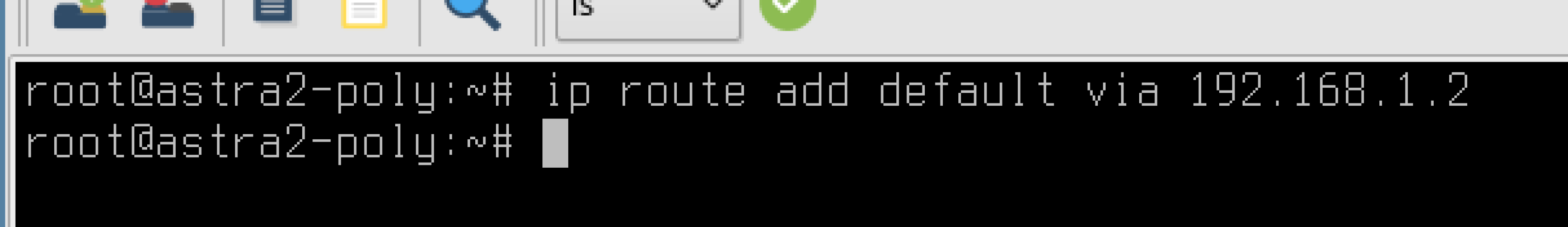


Эта команда добавляет правило в таблицу NAT (-t nat), цепочку POSTROUTING (-A POSTROUTING), которая будет применяться к пакетам, исходящим из подсети 192.168.1.0/25 (-s 192.168.1.0/25), и направляемым на интерфейс eth1 (-o eth1).

Действие, которое применяется к таким пакетам - MASQUERADE (-j MASQUERADE), которое заменяет исходный адрес и порт на адрес и порт, соответствующий выходному интерфейсу eth1. Это необходимо, чтобы пакеты могли правильно проходить через NAT-шлюз и выходить в интернет или другую сеть с адресом, соответствующим интерфейсу eth1.

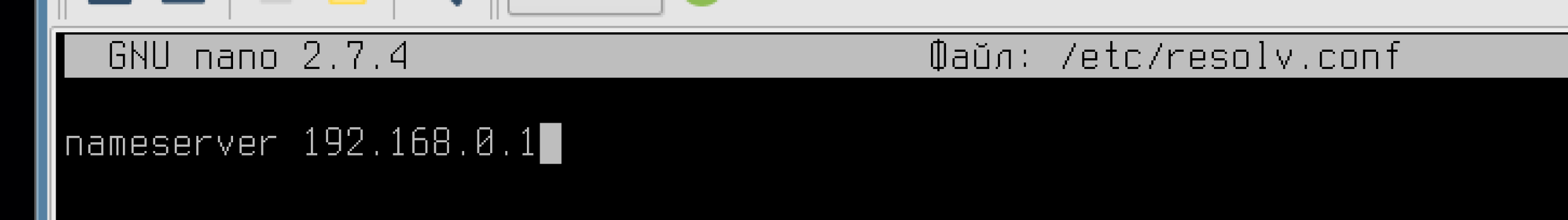
Таким образом, данная команда настраивает правило NAT для маршрутизатора, который используется для переадресации пакетов, проходящих через сеть, и позволяет подключенным устройствам в подсети 192.168.1.0/25 получать доступ к другим сетям через интерфейс eth1.

Далее нужно настроить шлюз по умолчанию на других машинах.

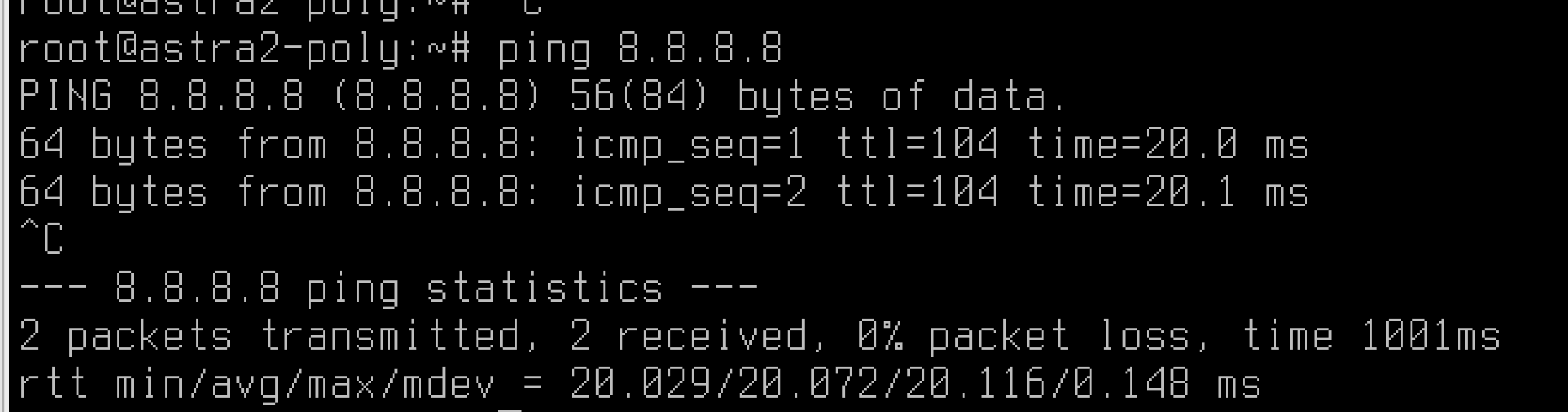




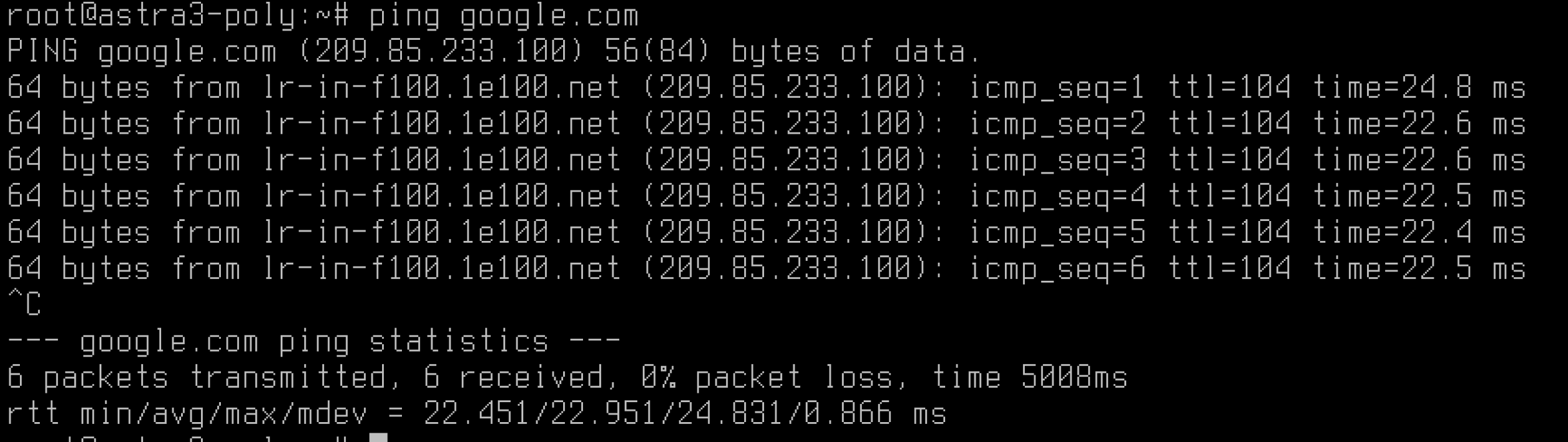
Теперь необходимо настроить DNS сервер в конфигурационном файле всех машин. В моем случае это 192.168.0.1 – IP адрес роутера.



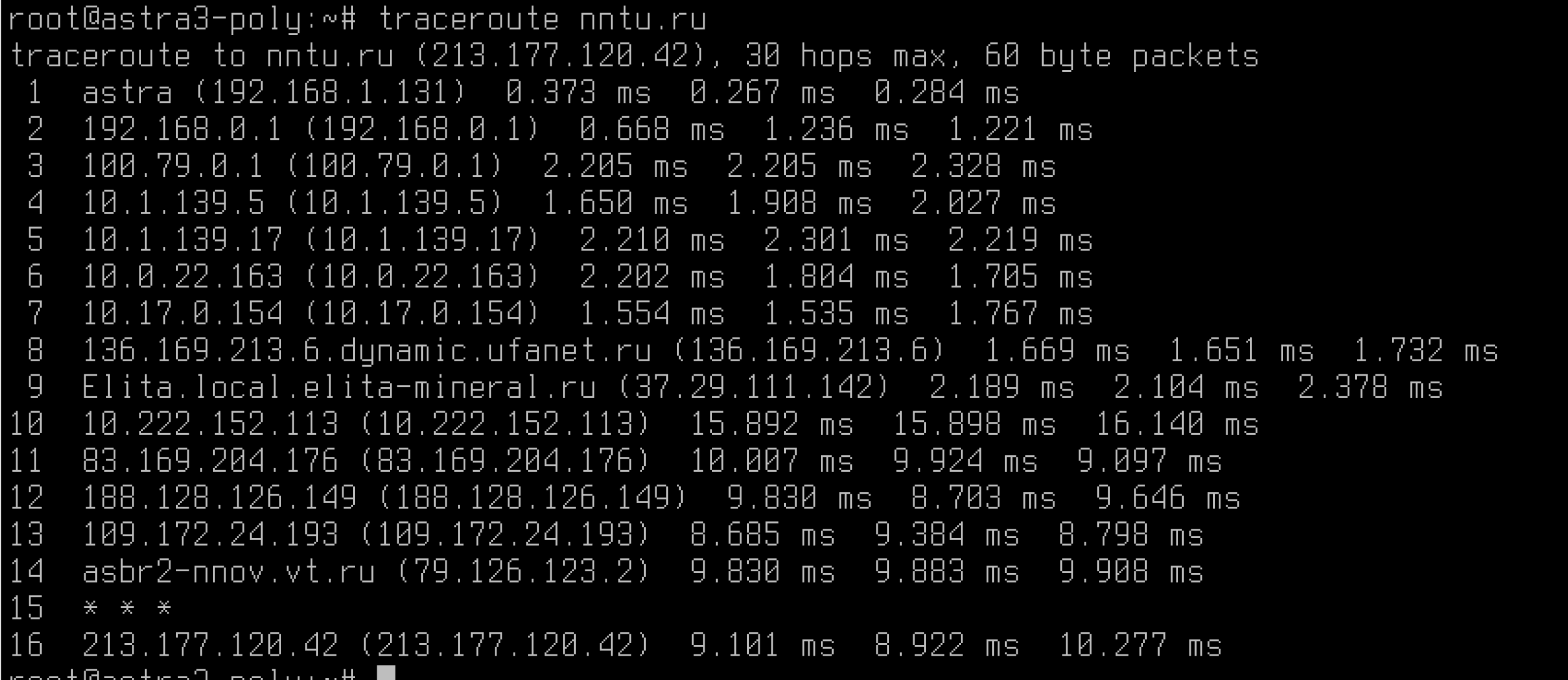
Попробуем установить соединение с DNS сервером Google. Как видим доступ в интернет есть.



Попробуем установить соединение по имени домена.

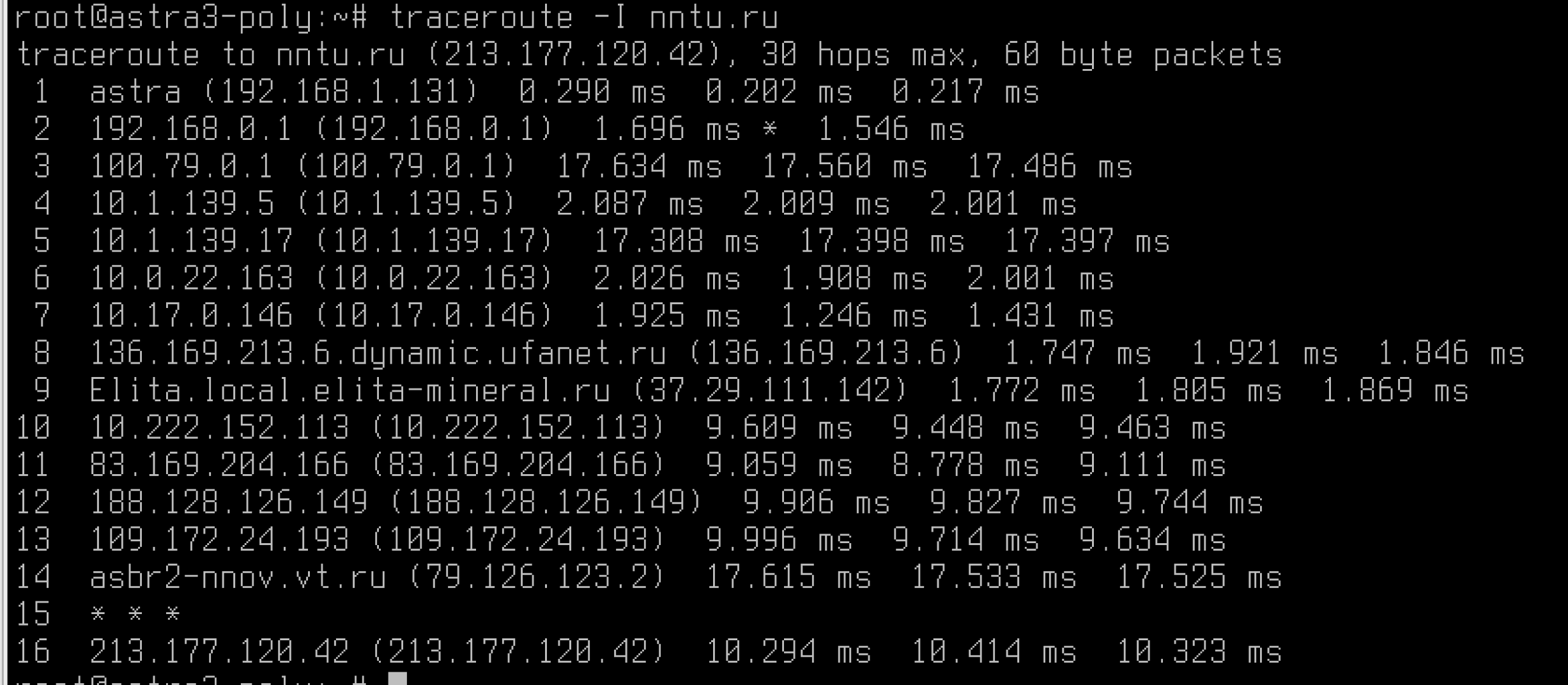


Для того чтобы узнать куда идет пакет во время установки соединения выполним команду traceroute, который по умолчанию пытается установить TCP-соединение.



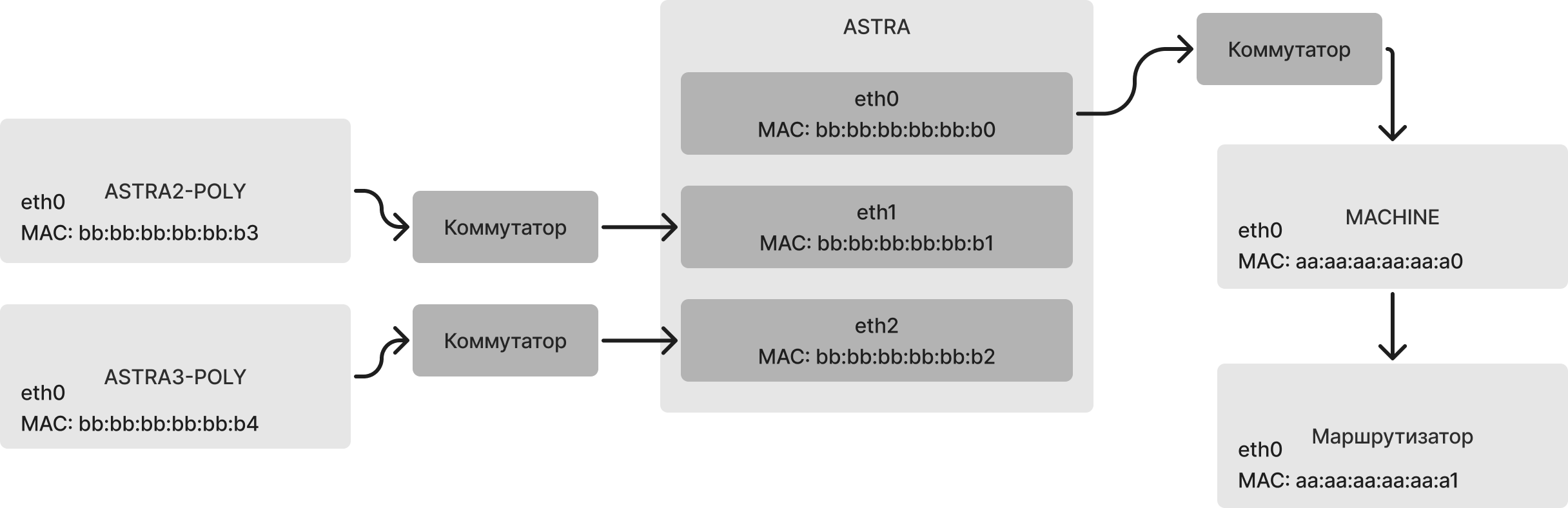
Видим, что пакет идет сначала на узел ASTRA, затем на роутер, а затем в интернет. Для достижения ресурса потребовалось 16 шагов.

Попробуем выполнить соединение по протоколу iCMP, для этого вызовем команду traceroute с флагом -I (заглавная i).

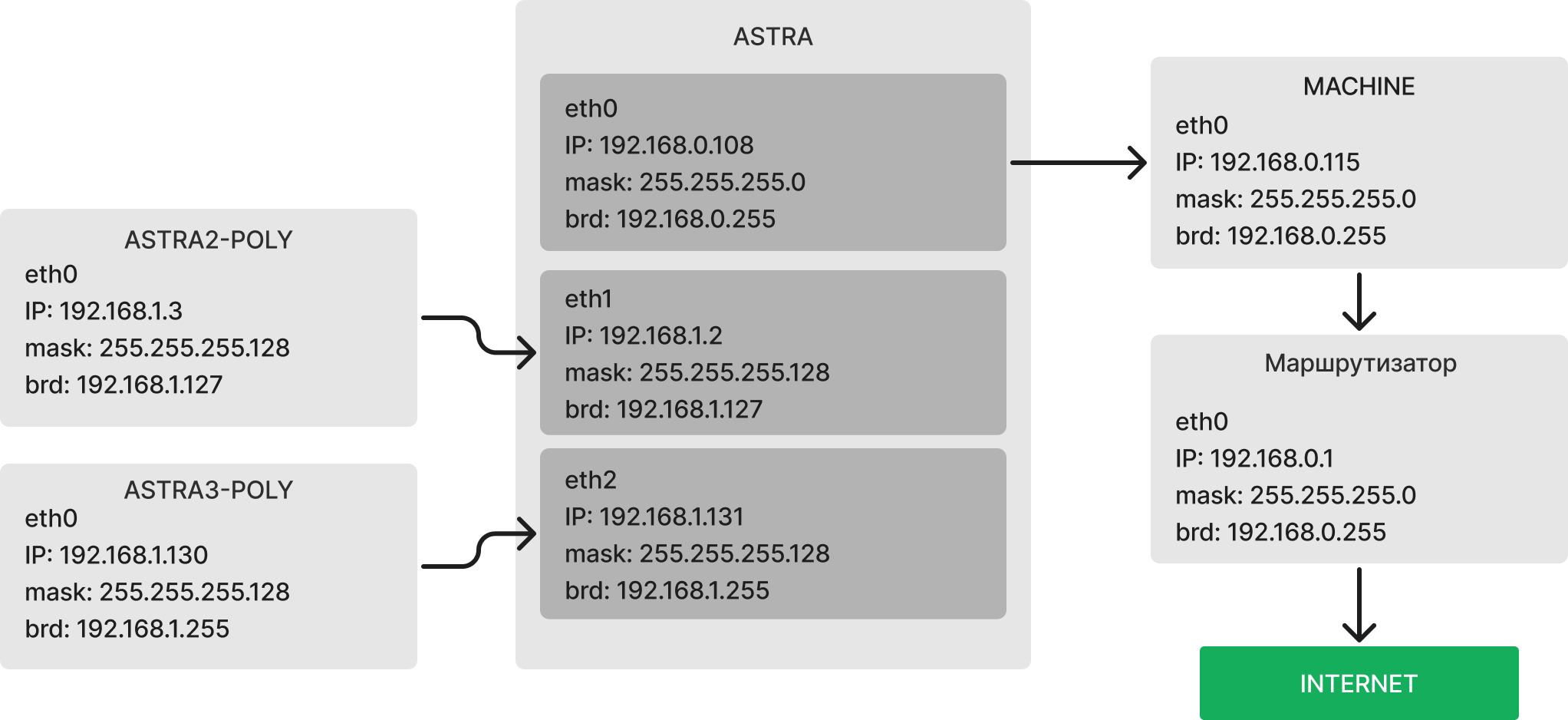


Нарисуем схему сети на канальном и сетевом уровне, для безопасности фактические MAC-адреса не были отображены.

Канальный.



Сетевой.

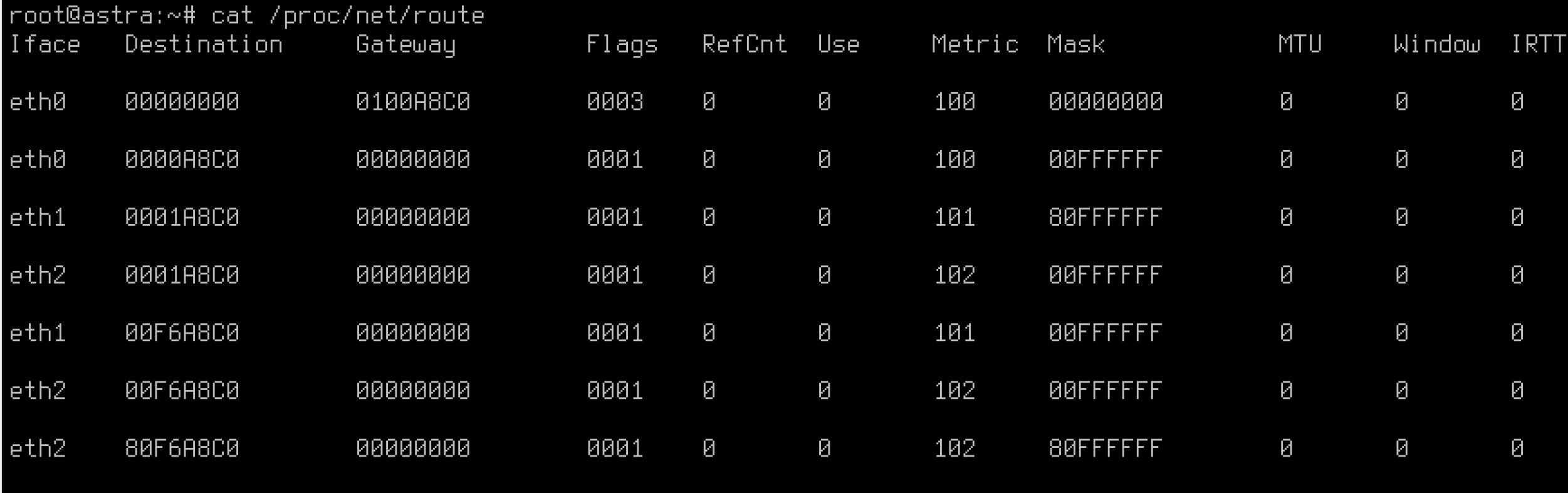


Контроль за сетью, получение статистики и другой сетевой информации.

Файл /proc/net/route является интерфейсом для доступа к таблице маршрутизации ядра Linux. Он содержит информацию о сетевых интерфейсах, маршрутах и шлюзах по умолчанию. Каждая строка в файле представляет собой запись таблицы маршрутизации для определенного сетевого интерфейса.

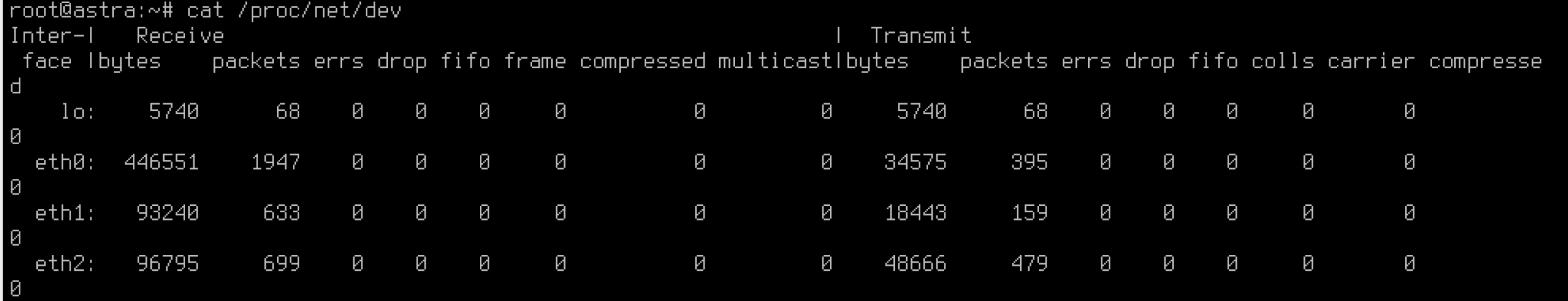
Структура записи, следующая:

* Имя сетевого интерфейса;
* Значение маркера;
* IP-адрес шлюза;
* IP-адрес сети;
* Флаги;
* Счетчики.



Файл /proc/net/dev в Linux предоставляет информацию о сетевых интерфейсах и статистику сетевого трафика.

В этом файле каждый сетевой интерфейс на сервере отображается в отдельной строке, в которой перечислены различные параметры, такие как количество принятых и отправленных пакетов, количество ошибок, количество отброшенных пакетов и т. д. Кроме того, эти параметры могут быть разделены на два столбца: "rx" для полученных пакетов и "tx" для отправленных пакетов.



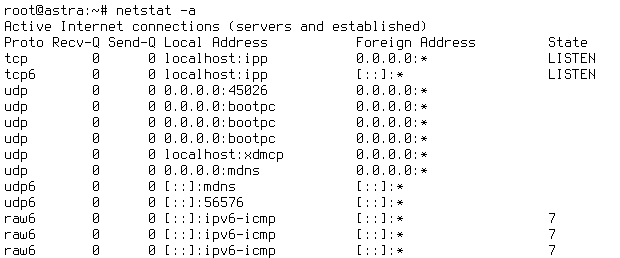
Используем команду arp для получения кэш-таблицы протокола.

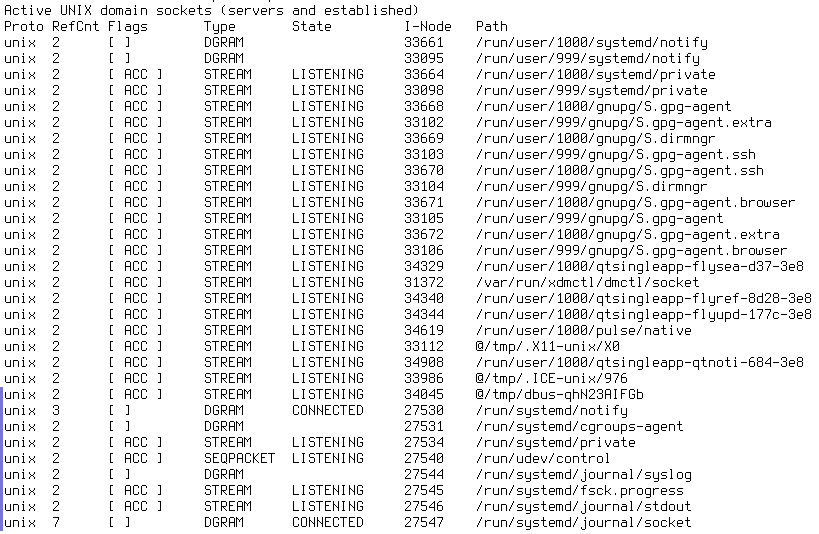


Команда netstat в Linux используется для отображения различной информации о сетевых соединениях, маршрутизации и сетевых интерфейсах на компьютере.

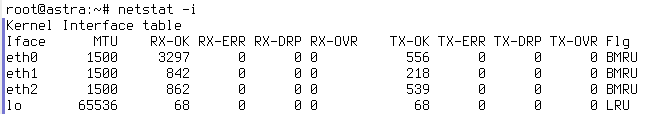
Она может принимать следующие аргументы:

* -a - отображение всех соединений (включая неактивные);

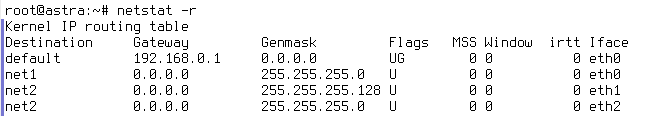




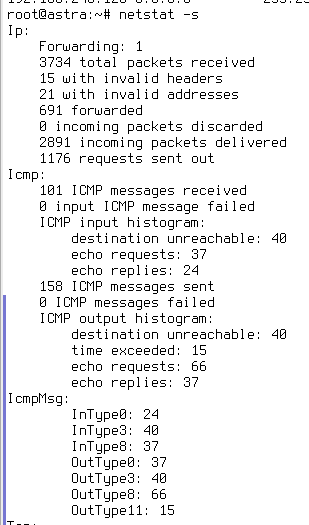
* -i - отображает статистику по интерфейсам (количество успешно принятых пакетов, количество ошибок, количество потерянных пакетов и количество потерянных пакетов из-за переполнения)



* -n - отображение портов в числовом формате (без преобразования в имена сервисов);
* -p - отображение идентификатора процесса, управляющего соединением;
* -r - отображение таблицы маршрутизации;

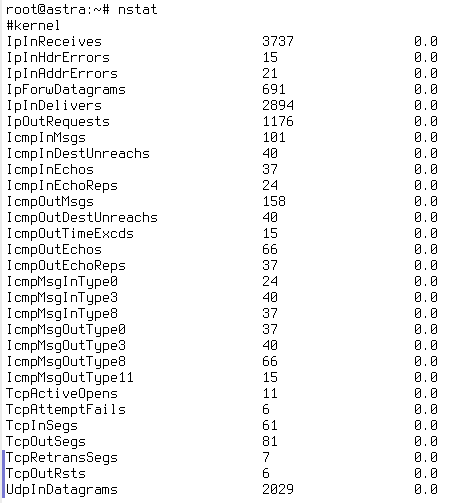


* -s - отображение статистики сетевых протоколов;

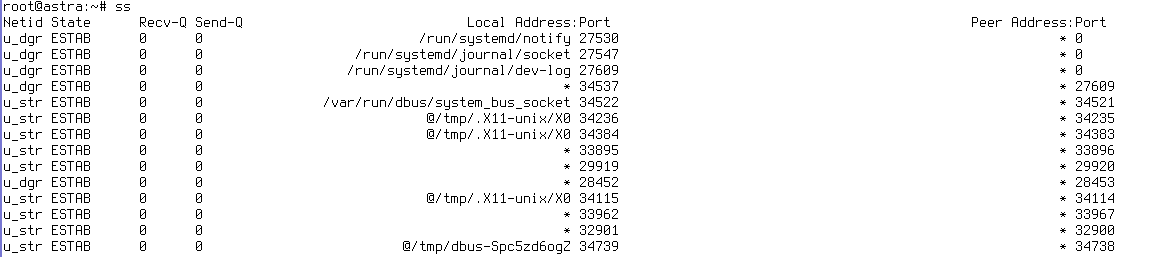


* -t - отображение только TCP-соединений, применятся с командами выше;
* -u - отображение только UDP-соединений, применятся с командами выше.

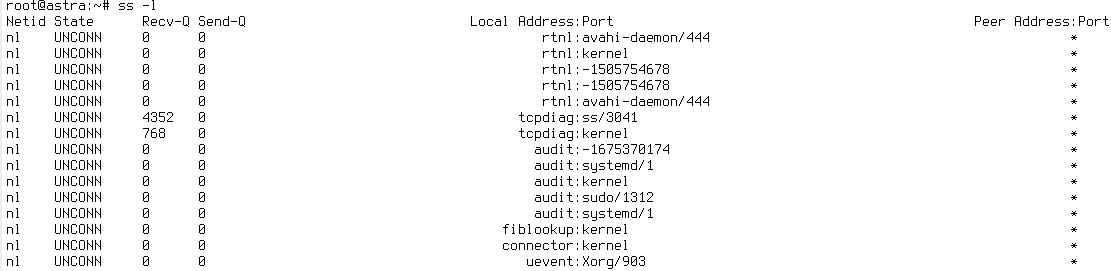
Команда nstat также предоставляет статистику сетевых подключений, но использует протокол SNMP для сбора этой информации. Эта команда позволяет получить более детальную статистику по протоколам и типам ошибок:



Команда ss (Socket Statistics) является альтернативой команде netstat и используется для отображения информации о сетевых сокетах (открытых соединениях) на Linux системах.



Конечно, в любых ситуациях команда в простейшем виде не так уж и полезна. Вывести список сокетов, ожидающих соединений.



Весьма удобная возможность команды ss заключается в том, что она может ***фильтровать вывод***, используя состояния TCP (или состояния жизненного цикла соединения). Благодаря использованию состояний облегчается фильтрация вывода ss. А именно, здесь доступны все стандартные состояния TCP:

1. established
2. syn-sent
3. syn-recv
4. fin-wait-1
5. fin-wait-2
6. time-wait
7. closed
8. close-wait
9. last-ack
10. listening
11. closing

Кроме того, ss распознаёт следующие идентификаторы состояний:

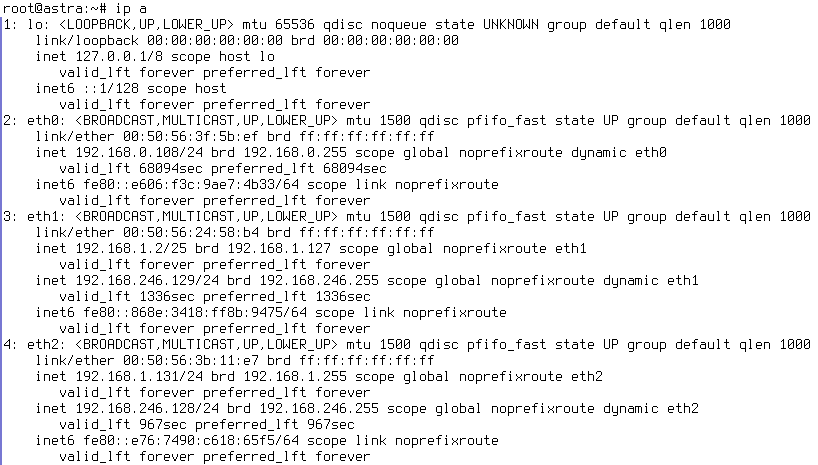
1. all (все вышеперечисленные состояния)
2. connected (все состояния, кроме ожидающих соединения и закрытых)
3. synchronized (все состояния, соответствующие установленным соединениям, за исключением syn-sent)
4. bucket (состояния, представляющие собой минисокеты, например — time-wait и syn-recv)
5. big (всё кроме того, что соответствует идентификатору bucket)

iproute2 – это набор утилит Linux, используемых для управления и настройки сетевых интерфейсов, IP-адресов, таблиц маршрутизации и других параметров, связанных с сетью. Он представляет собой более мощную и гибкую альтернативу старым утилитам ifconfig, route и arp.

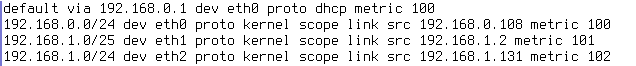


Некоторые из основных утилит, предоставляемых iproute2, включают:

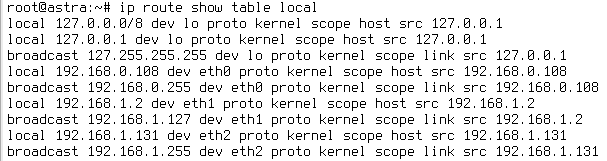
* ip: используется для отображения и работы с сетевыми интерфейсами, IP-адресами и таблицами маршрутизации.



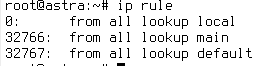
Для настройки роутинга служит команда ip route. Выполненная без параметров, она покажет список текущих правил маршрутизации



Изначально предопределены таблицы *local*, *main* и *default*. В таблицу *local* ядро заносит записи для локальных IP адресов (чтобы трафик на эти IP-адреса оставался локальным и не пытался уходить во внешнюю сеть), а также для бродкастов. Таблица main является основной, и именно она используется, если в команде не указано какую таблицу использовать (т.е. выше мы видели именно таблицу *main*). Таблица *default* изначально пуста.



Правила, по которым ядро выбирает в какую таблицу отправлять пакеты:



Число в начале строки – идентификатор правила, from all – условие, означает пакеты с любых адресов, lookup указывает в какую таблицу направлять пакет. Если пакет попадает под несколько правил, то он проходит их все по порядку возрастания идентификатора. Конечно, если пакет подпадет под какую-либо запись маршрутизации, то последующие записи маршрутизации и последующие правила он уже проходить не будет.

Возможные условия:

1. from – мы уже рассматривали выше, это проверка отправителя пакета.
2. to – получатель пакета.
3. iif – имя интерфейса, на который пришел пакет.
4. oif – имя интерфейса, с которого уходит пакет. Это условие действует только для пакетов, исходящих из локальных сокетов, привязанных к конкретному интерфейсу.
5. tos – значение поля TOS IP-пакета.
6. fwmark – проверка значения FWMARK пакета. Это условие дает потрясающую гибкость правил. При помощи правил iptables можно отфильтровать пакеты по огромному количеству признаков и установить определенные значения FWMARK. А затем эти значения учитывать при роутинге.

* bridge (мост): используется для настройки мостов Ethernet.
* tc: используется для настройки параметров управления трафиком и качества обслуживания.

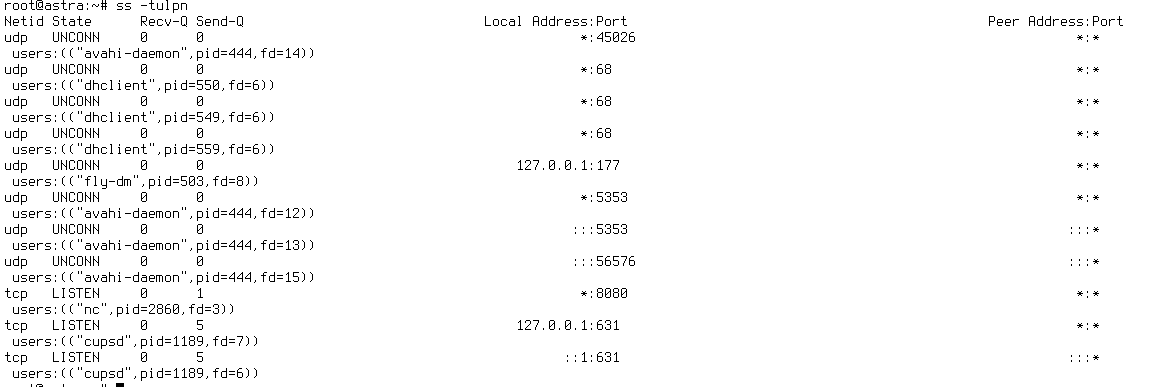
Выбор команды для вывода статистики зависит от ситуации, степени знакомства с каждой из них и возможности установки дополнительных пакетов. Команда netstat является стандартной и может помочь при разных вопросах. Когда необходимо подробнее поработать со статистикой протоколов – nstat. Если есть необходимость получить подробную информацию о портах – ss.

Пример: проверить используемые порты.

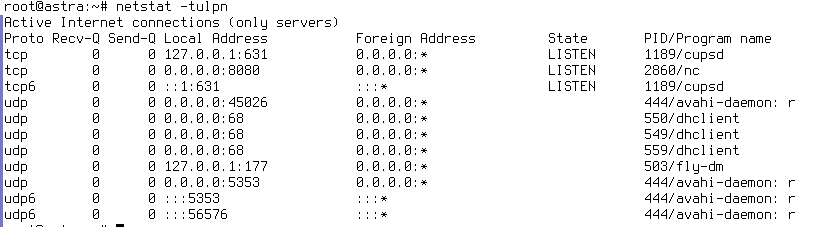
Создадим еще одну терминальную сессию, в которой запустим прослушивание порта 8080 по TCP.



Socket Statistic:



Netstat:



Вывод

В ходе выполнения лабораторной работы, были настроены сетевые интерфейсы на трех узлах на ОС Astra. Особенность в том, что два узла были клиентами, а один выступал в роли маршрутизатора. Были настроены таблицы маршрутизации так, что клиенты, находящиеся в разных сетях, могли устанавливать соединение друг с другом, а так же иметь доступ в интернет.

Рассмотрена настройка сетевых интерфейсов и таблиц маршрутизации с помощью команд: ifconfig, route, iptables.

Так же настроены следующие конфигурационные файлы: /etc/hostname, /etc/hosts, /etc/resolv.conf, networks, ip\_forward и другие.