

## Базовый набор команд IP (iproute2)

ip	object	command / description
ip	addr	Управление IP-адресами на интерфейсах
ip	link	Управление интерфейсами
ip	route	Управление таблицей маршрутизации
ip	neigh	Управление arp cache
ip	netns	Управление сетевыми namespace

Большинство объектов имеют похожий набор действий:

show	Показать список объектов
add/del	Добавить/удалить объекты

## IP-адреса и интерфейсы (ip addr)

ip addr [show]	Показать IP-адреса на всех интерфейсах
ip addr add <a href="#">192.168.1.1/24</a> dev <a href="#">eth0</a>	Назначить адрес <a href="#">192.168.1.1</a> на интерфейс <a href="#">eth0</a> и добавить в таблицу маршрутизации маршрут до <a href="#">192.168.1.0/24</a> , ведущий через <a href="#">eth0</a>

## Маршрутизация (ip route)

ip route [show]	Показать таблицу маршрутизации
ip route add <a href="#">192.168.1.0/24</a> via <a href="#">172.0.0.1</a>	Добавить маршрут к подсети <a href="#">192.168.1.0/24</a> через узел <a href="#">172.0.0.1</a>
ip route del <a href="#">192.168.1.0/24</a> via <a href="#">172.0.0.1</a>	Удалить маршрут к подсети <a href="#">192.168.1.0/24</a>
ip route get <a href="#">1.1.1.1</a>	Показать маршрут из таблицы маршрутизации, который будет выбран при отправке пакета на адрес <a href="#">1.1.1.1</a>

## Прослушивание трафика (tcpdump)

tcpdump -nnl -i <a href="#">eth0</a>	Показывать пакеты, идущие через интерфейс <a href="#">eth0</a>
tcpdump -nnl -e -i <a href="#">eth0</a>	Аналогично предыдущей, но с отображением L2 заголовков (mac)

## Сетевые namespaces (ip netns)

ip netns [ls]	Показать список namespace
ip netns exec <a href="#">ns_name</a> <CMD>	Запустить команду CMD внутри netns с именем ns_name
ip netns attach <a href="#">ns_name</a> <PID>	Дать имя ns_name для netns, которая создана для процесса с идентификатором PID

## Пример работы с ip netns exec

ip netns exec <NS>	ip a[ddr]	Показать IP-адреса внутри netns NS
	ip r[oute]	Показать маршруты внутри netns NS
	ping ...	Запуск ping внутри netns NS
	tcpdump ...	Запуск tcpdump внутри netns NS

## Работа с «мостами» (bridge + ip neigh)

ip neigh [show]	Показать ARP-таблицу записей
ip neigh del <a href="#">10.0.0.1</a> dev <a href="#">eth0</a>	Удалить запись для адреса <a href="#">10.0.0.1</a> из ARP-таблицы записей
bridge fdb	Показать forwarding-таблицу для всех бриджей
bridge link show	Показать все интерфейсы всех бриджей

## Алгоритм отладки сетевых проблем

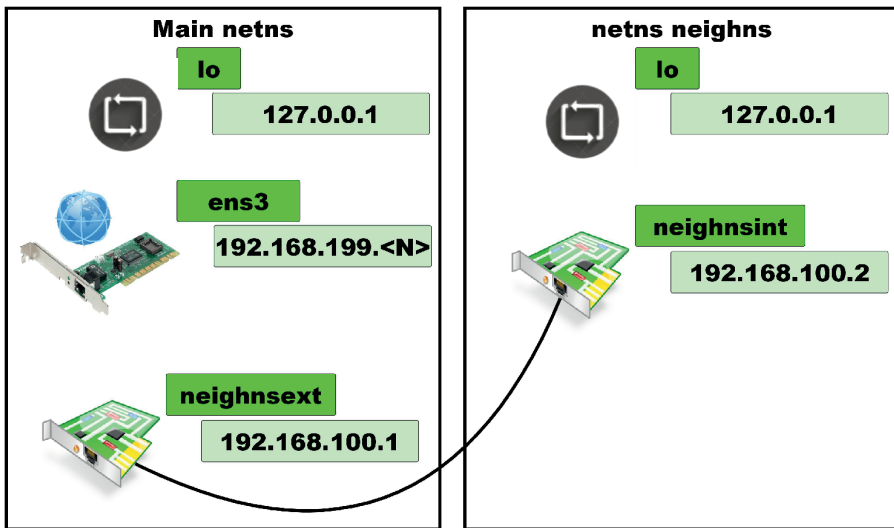
- 1 Изучите схему сети, в которой вы хотите отлаживать проблему
- 2 Последовательно убедитесь, что каждый элемент на схеме присутствует в реальной системе, которую вы хотите отлаживать, и имеет такие же параметры (в пределах вашей сферы влияния)
- 3 Запустите генерацию трафика из точки А в точку Б
- 4 Последовательно, с помощью tcpdump, проверяйте наличие трафика на каждом интерфейсе, через который должен пройти пакет
- 5 Не забывайте проверять трафик, идущий в обратную сторону
- 6 При расхождении реальной ситуации от желаемой начинайте поиск проблемы с самого нижнего уровня (L1→L2→L3→...)



## Лабораторная работа № 1: Знакомство

Задача: ping 192.168.100.2

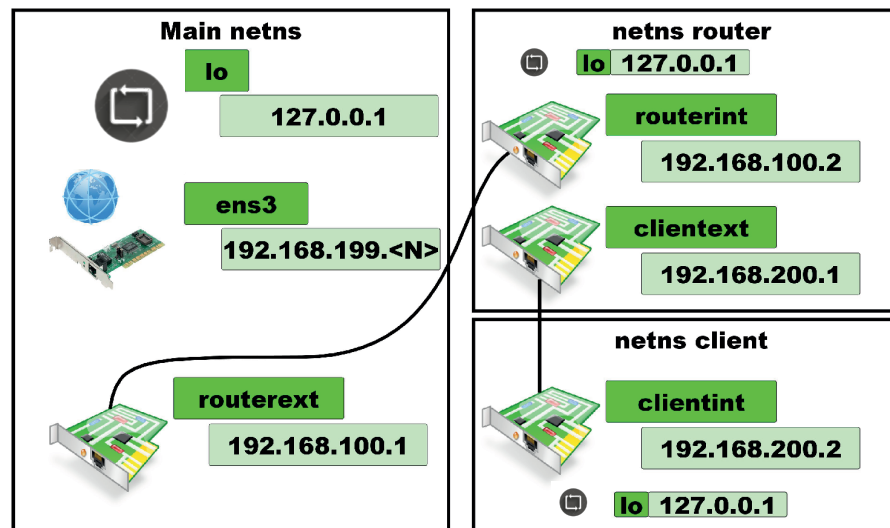
Полезные утилиты: `ip addr` `ip netns` `ping` `tcpdump`



## Лабораторная работа №2: Маршрутизация

Задача: ping 192.168.200.2

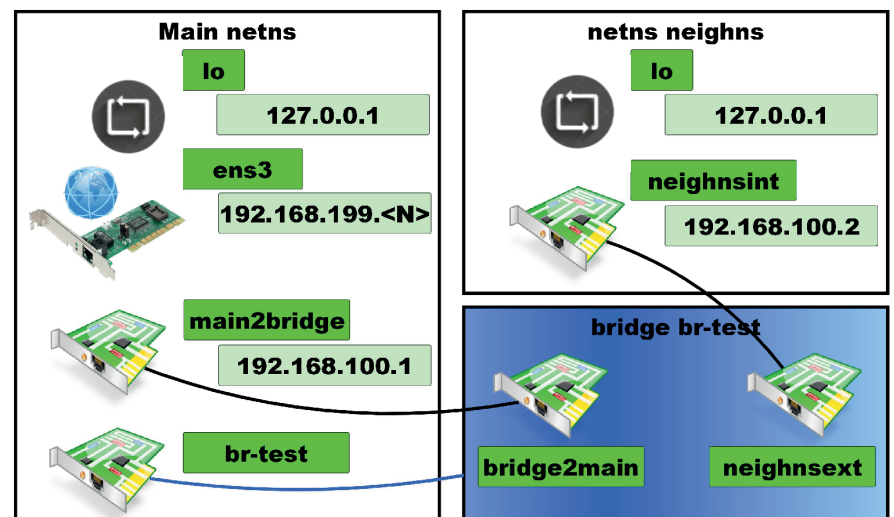
Полезные утилиты: `ip addr` `ip netns` `ping` `tcpdump`



## Лабораторная работа № 3: Работа с ARP и L2

Задача: ping 192.168.100.2

Полезные утилиты: `ip addr` `ip netns` `ip neigh` `bridge` `ping` `tcpdump -e`



## Лабораторная работа № 4: Сеть в контейнерах

Задача: ping 172.18.0.22

Полезные утилиты: `ip addr` `ip netns` `ip neigh` `bridge` `ping` `tcpdump -e`

