МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

# Отчет

по лабораторной работе №3



по дисциплине «Сети и телекоммуникации»

8 вариант

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гай В.Е.

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ширшов А.А.

19-В-1

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

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

Нижний Новгород

2021

**Задание на работу**

Для экспериментов использовать схему из первой лабораторной работы. Все ip-адреса (или маски) необходимо поменять так, чтобы адрес сети у всех компьютеров был один. Все действия должны быть выполнены в симуляторе сетей CORE.

**Часть 1. Формирование запроса и получение ответа**

1. Начать захват пакетов при помощи WireShark.

2. Сформировать кадр ARP-запроса с помощью утилиты PackETH и отправить его в сеть (компьютеры выбрать самостоятельно).

3. Убедиться, что был получен кадр ARP-ответа, соответствующий посланному запросу. Захваченные пакеты сохранить для отчета. Вывести arp таблицу (команда «arp»).

4. Прекратить захват пакетов.

**Часть 2. ARP-спуфинг**

1. Выделить на схеме и обозначить три компьютера: A, B, Сервер.

2. Подготовить кадр ARP-ответа, направляемый Сервером хосту А с помощью программы PackETH. Кадр должен быть составлен так, чтобы MAC-адресу Сервера соответствовал IP-адрес хоста В. Вывести arp таблицу на хосте А. Отправить сформированный пакет от Сервера хосту А.

Для запуска packEth в консоли выполните команду «xhost +», в консоли узла «DISPLAY=:0 packeth» или «DISPLAY=:0 packeth --sync».

netcat (англ. net сеть + cat) — утилита Unix, позволяющая устанавливать соединения TCP и UDP, принимать оттуда данные и передавать их.

Организация чата между узлами с помощью netcat:

1. На первом узле (192.168.1.100): $ nc -lp 9000

2. На втором узле: $ nc 192.168.1.100 9000

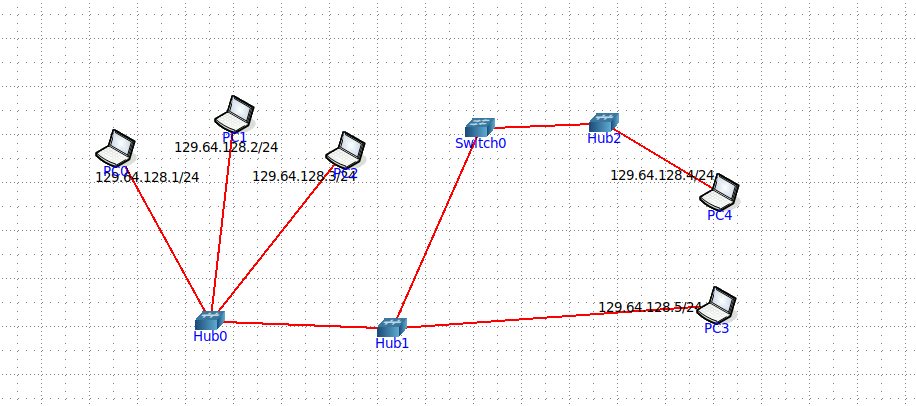
3. Начать захват пакетов при помощи WireShark на Сервере.

4. Попытаться установить соединение между хостом А и хостом В с помощью программы netcat (А отправляет сообщения В). Убедиться, что запросы от хоста A, направленные хосту В поступают на Сервер.

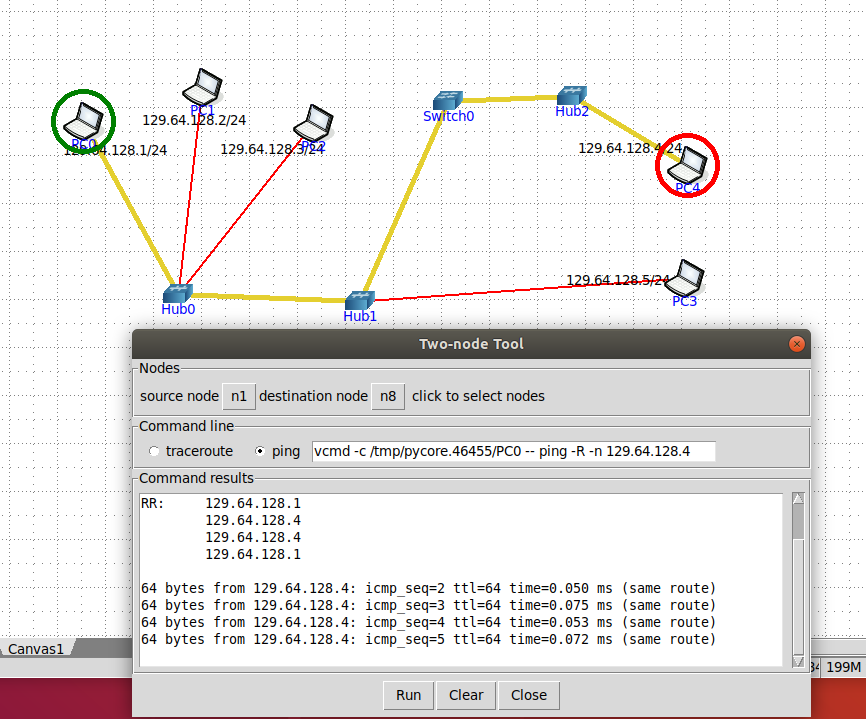
5. Прекратить захват пакетов.

6. Сохранить для отчета отправленный кадр ARP-ответа и несколько перехваченных пакетов, переданных на Сервер, arp таблицу хоста А.

**Структура сети**

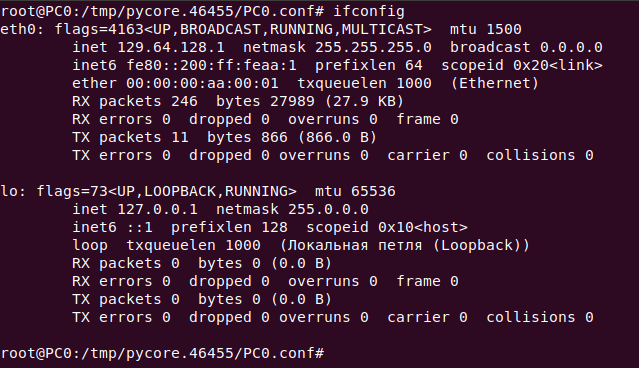


Все компьютеры в одной сети, пинг проходит:

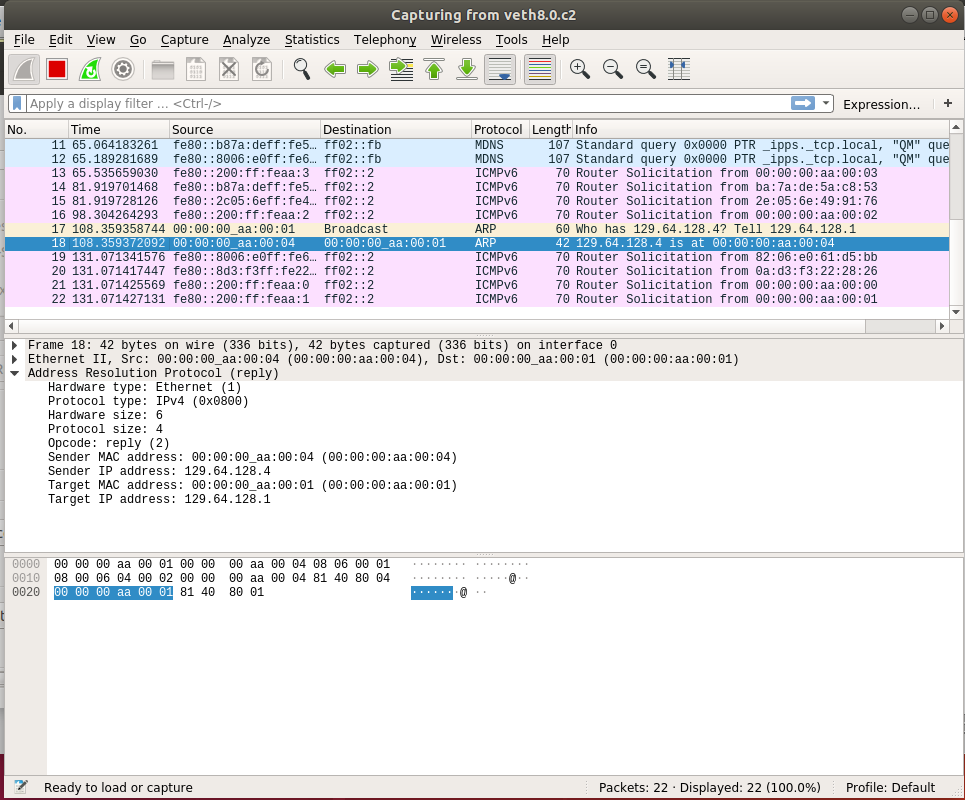
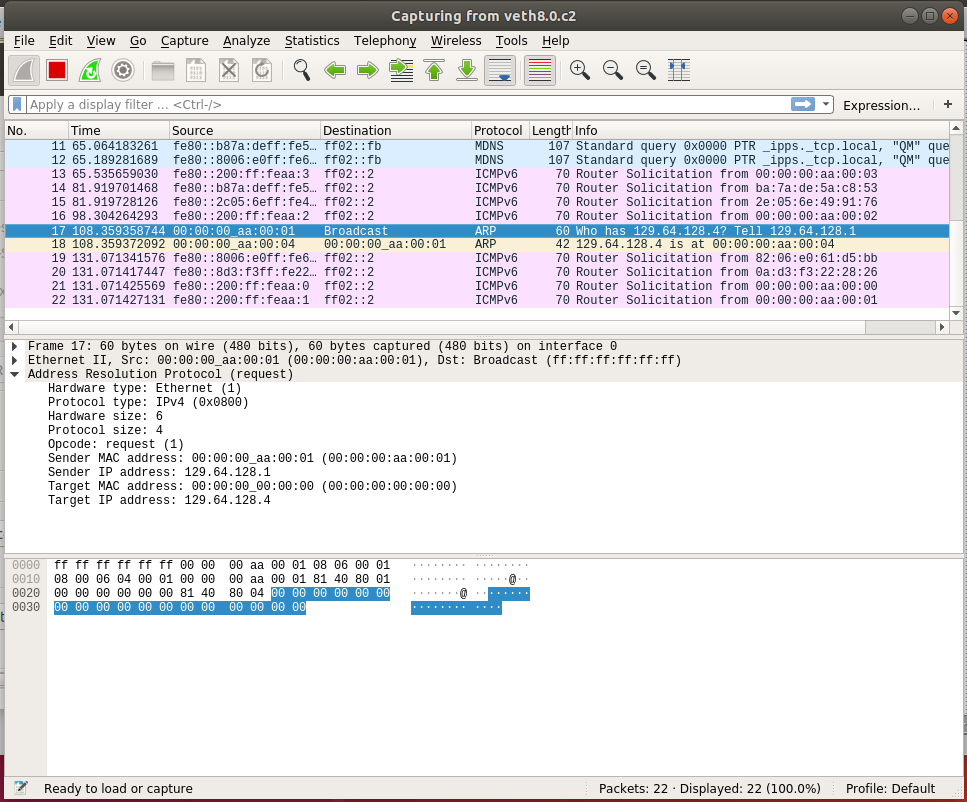
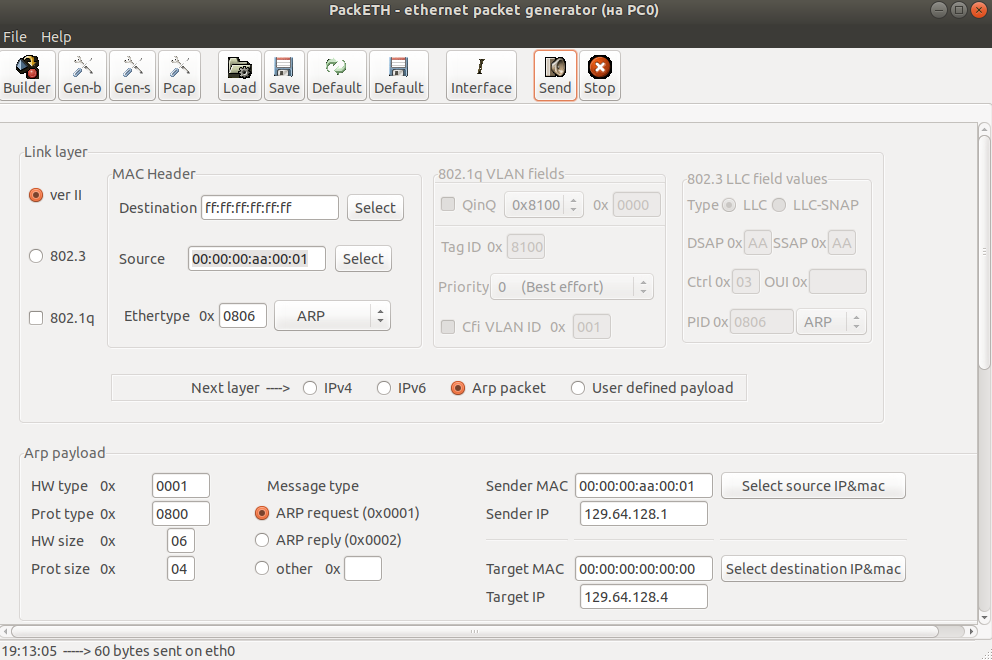


**Часть 1. Формирование запроса и получение ответа**

Были выбраны компьютеры PC0 и PC4

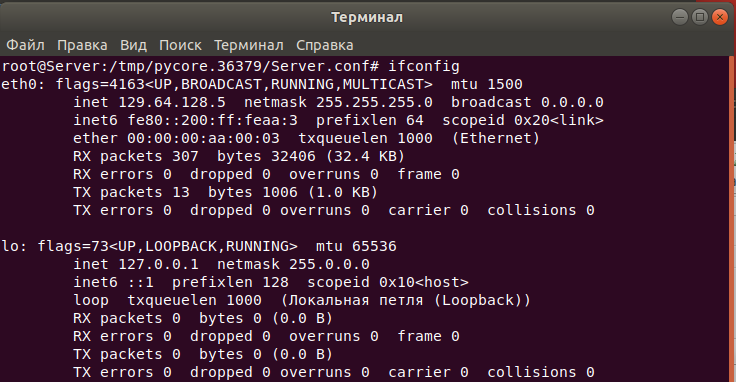
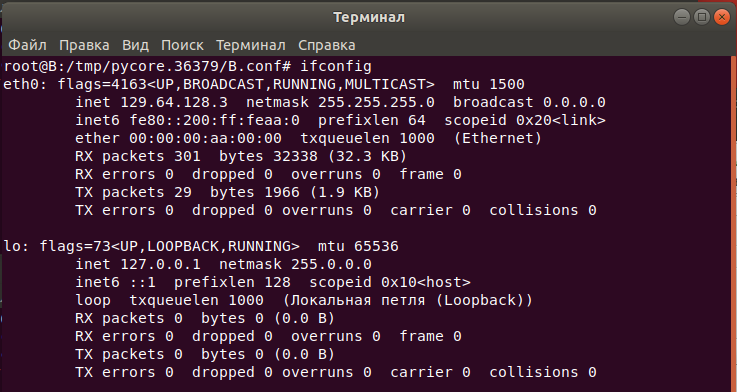
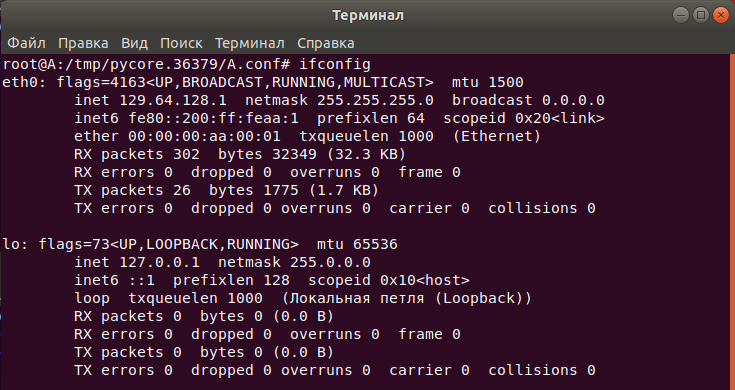
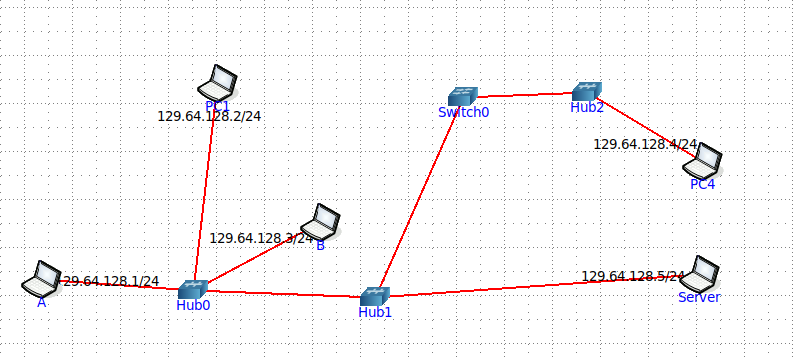


Запускаем на PC0 packeth, на PC4 Wireshark



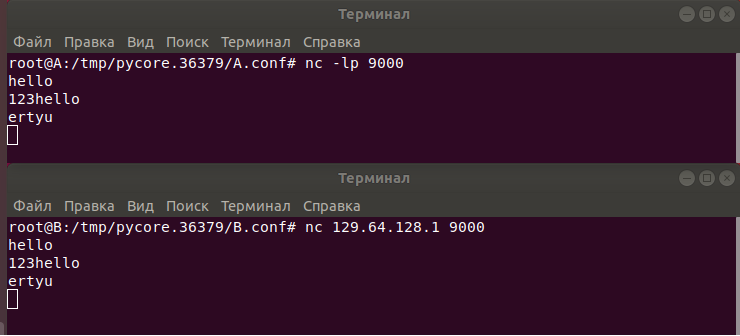
**Часть 2. ARP-спуфинг**

В качестве компьютеров A, B, Server были выбраны PC0, PC2 и PC3, обозначены на схеме:

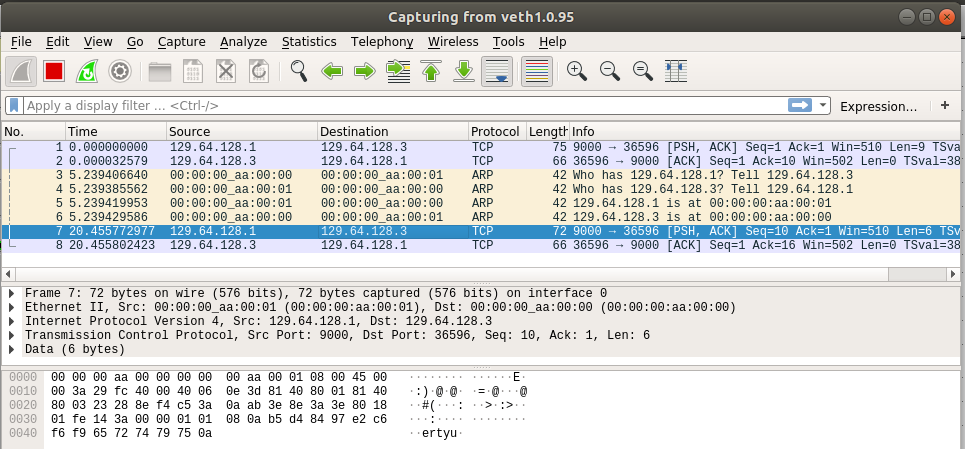


На узле A выполняем nc -lp 9000, на узле B подключаемся nc 129.64.128.1 9000

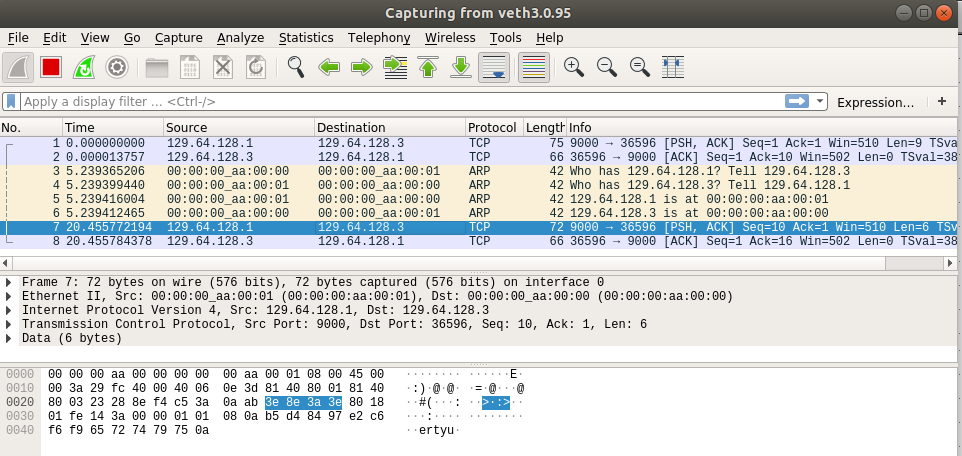
Видно, что оба получают сообщения:



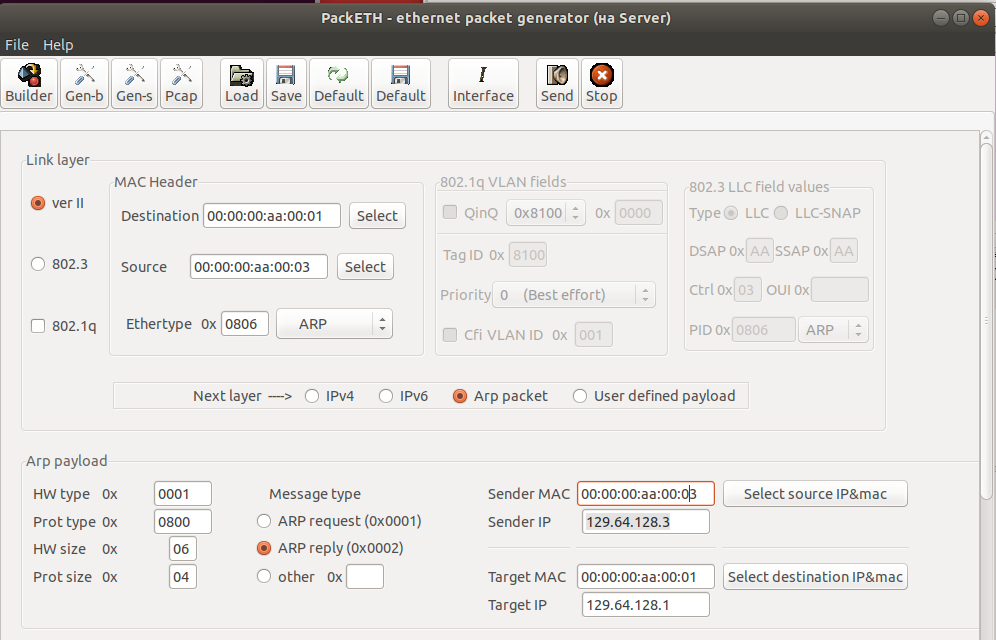
Wireshark на A:



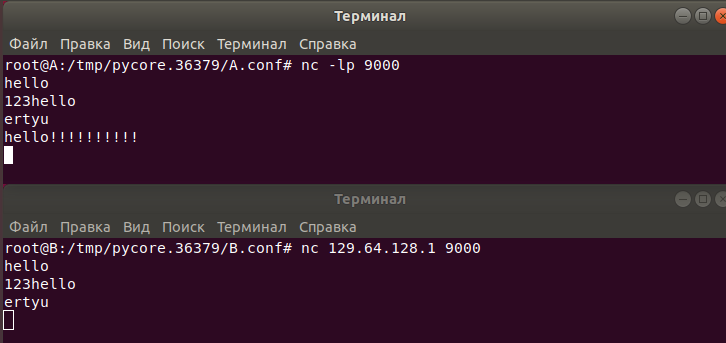
Wireshark на B:

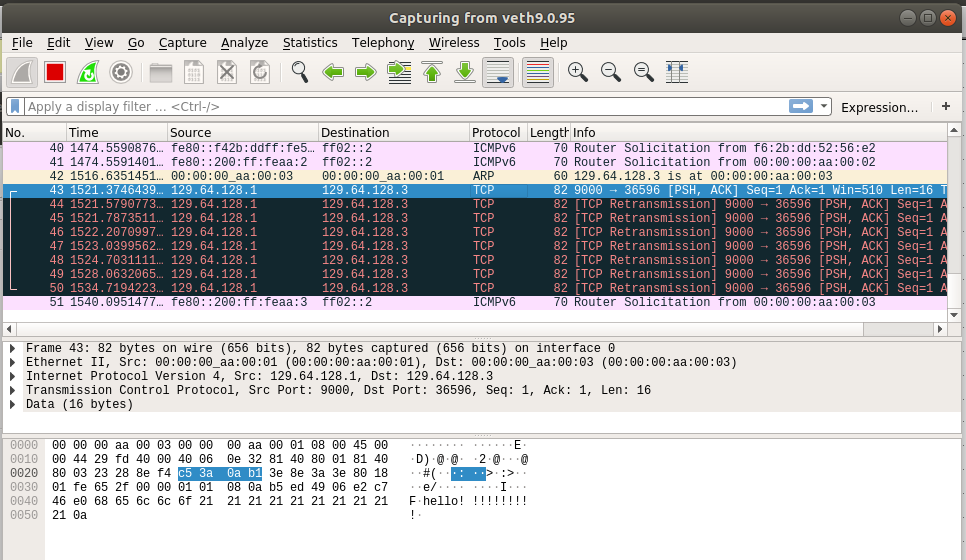


Теперь Server совершит ARP-спуфинг, он представится A узлом B и станет пропускать через себя пакеты, откроем на сервере packeth и сформируем ARP ответ:

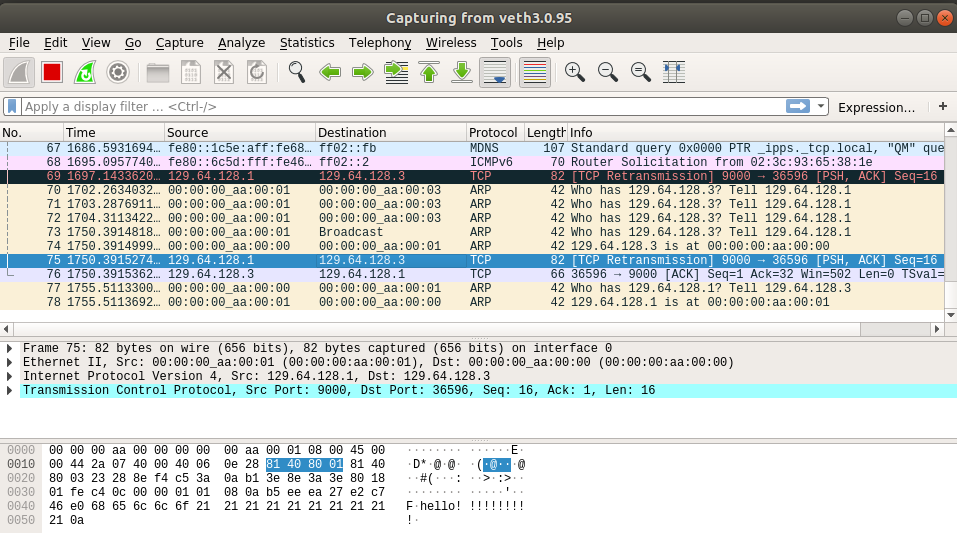


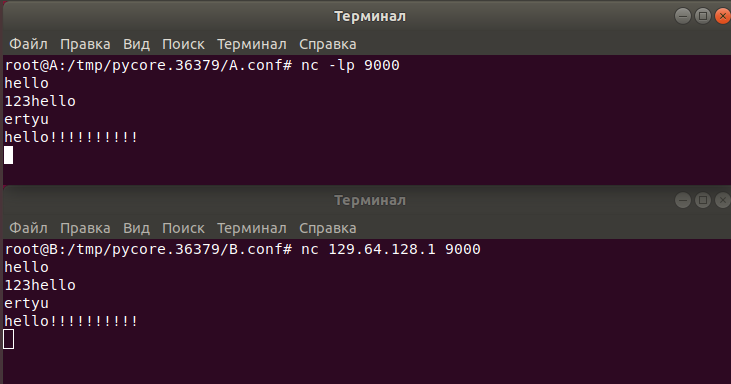
Теперь, когда компьютер A отправит сообщение, то его не получит B, но получит Server:





Через некоторое время пакеты все-таки дойдут до B, т.к. произойдет широковещательный запрос на получение mac адреса B:

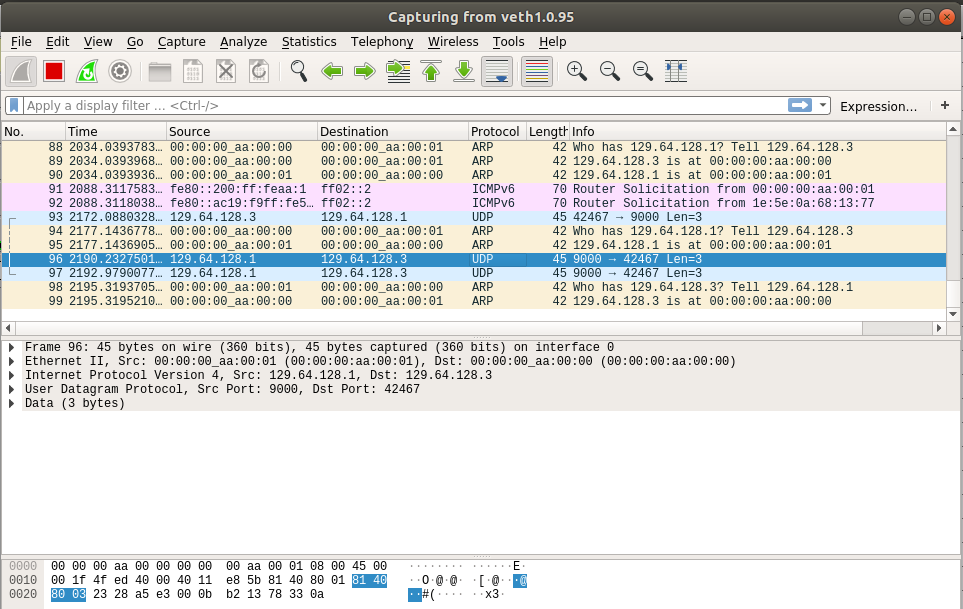
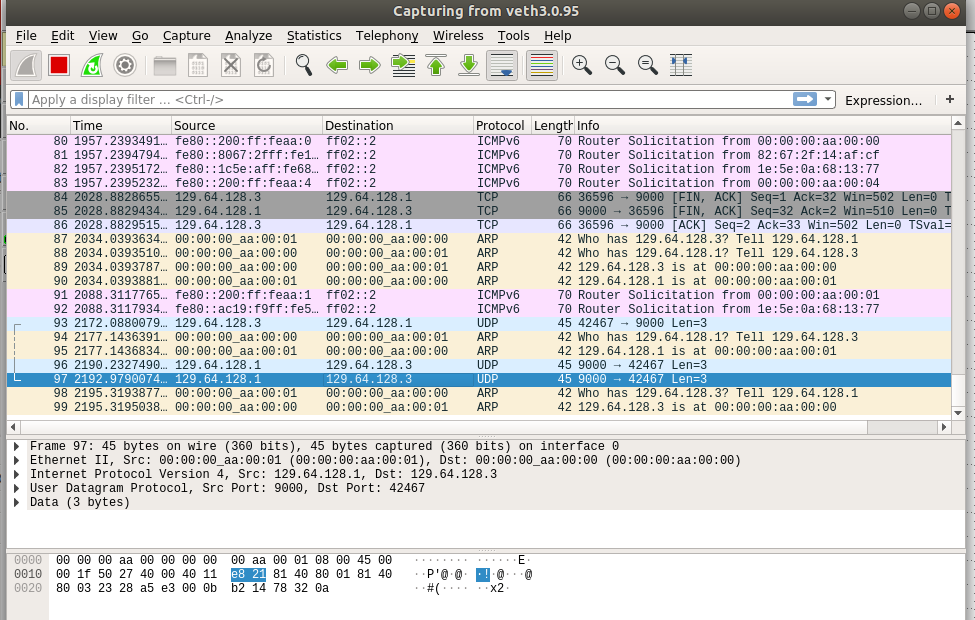
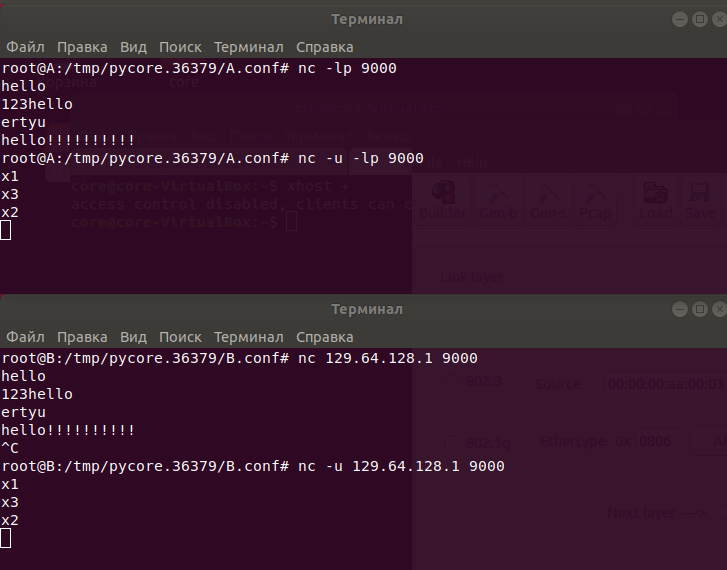




Теперь сделаем то же самое, но по протоколу UDP:

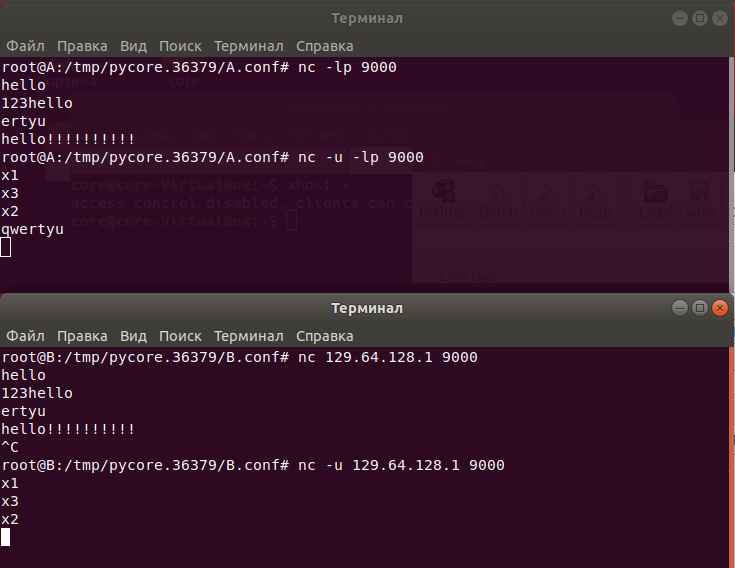
Он не требует никакого подтверждения, а просто отправляет пакеты.

Видно, что пакеты доходят:



Теперь, как и в прошлом примере, сервер совершит ARP-спуфинг:

Узел A снова отправит сообщение, а вот узел B его уже никогда не получит:



А вот сервер получает сообщение:

