Министерство науки и высшего образования

Российской федерации

Федеральное государственное бюджетное

Образовательное учреждение высшего образования

«Новосибирский Государственный Технический Университет»

Кафедра теоретической и прикладной информатики

Лабораторная работа №5

«Анализ функционирования и диагностика IP-сетей»

Факультет: прикладной математики и информатики

Группа: ПМИ-12

Бригада: 1

Студенты: Михайловский М.А.

Швадченко А.В.

Преподаватели: Кобылянский В.Г.

Филиппова Е.В.

Новосибирск, 2023

1. Цель работы:

Приобретение практических навыков работы с сетевыми командами операци-онных систем Windows и Linux, предназначенными для анализа и диагностики сетей TCP/IP, а также со средствами интерактивной диагностики сетей.

1. Содержание работы

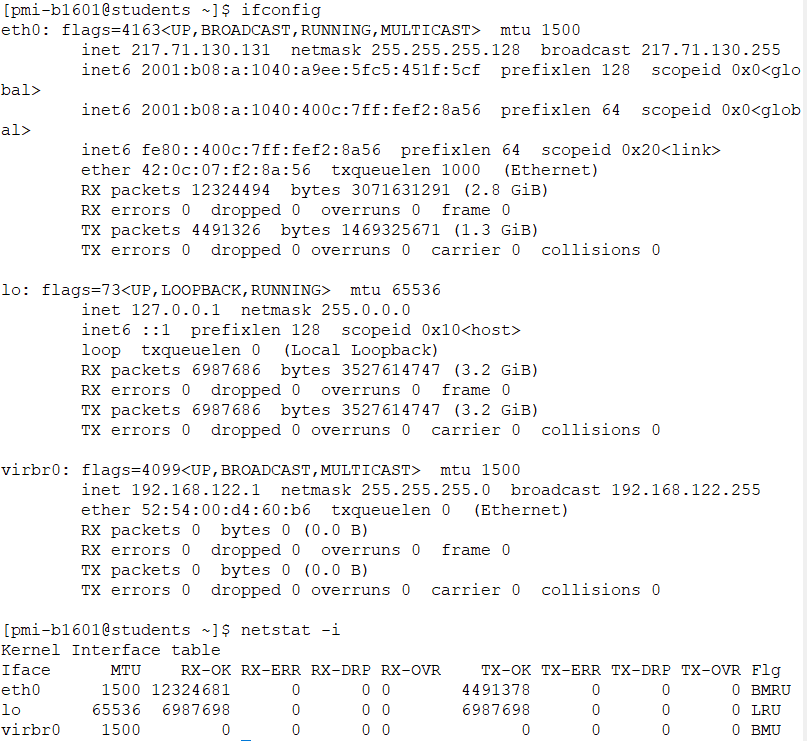
Выполнение лабораторной работы проводится в два этапа. На первом этапе необходимо изучить основные способы диагностики IP-сетей, на втором – спроектировать и реализовать программу, выполняющую основные функции утилит **ping** и **tracert**.

1. Выполнение работы:
   1. Подключиться с помощью клиента Putty к серверу **fpm2.ami.nstu.ru** и с помощью команды **uname** получить полную информацию об установленной операционной системе и аппаратной платформе, полученный результат включить в отчет.





* 1. Получить статистику по сетевым интерфейсам РК и сервера fpm2.ami.nstu.ru, пояснить результаты.



Всего имеется 3 сетевых интерфейса: eth0 (физический адаптер, используется для выхода в сеть Интернет), lo (loopback-интерфейс, используется только на данном устройстве) и virbr0 (интерфейс виртуального моста, предназначен для виртуальных машин). В первой строке информации о каждом сетевом интерфейсе (а также в столбце Flg таблицы, полученной командой netstat -i) указаны флаги.

Флаг статуса интерфейса UP (DOWN) говорит о том, что интерфейс в данный момент (не) инициализирован.

Флаг BROADCAST говорит о поддержке интерфейсом широковещательной рассылки IPv4.

Флаг статуса передачи RUNNING говорит о том, что интерфейс в данный момент передает пакеты.

Флаг MULTICAST указывает на поддержку мультивещания (групповой рассылки).

Рассмотрим теперь характеристики сетевых интерфейсов:

* mtu - максимальный размер передаваемых пакетов в октетах (байтах)
* inet - IPv4-адрес, назначенный интерфейсу
* inet6 - IPv6-адрес, назначенный интерфейсу
* netmask - маска IPv4 подсети
* broadcast - широковещательный IPv4-адрес
* prefixlen - длина префикса IPv6 подсети
* scopeid - IPv6 scope, указывающий в какой части сети адрес действителен
* ether - MAC-адрес интерфейса
* txqueuelen - размер буфера передачи (отправка данных осуществляется при заполнении буфера до указанного значения)
* RX packets и TX packets - количество полученных и переданных пакетов/байтов
* соответственно
* RX errors и TX errors - статистика по полученным и переданным пакетам
* соответственно; dropped (DRP) - количество пропавших пакетов, overruns (OVR) количество ошибок из-за превышения скорости.

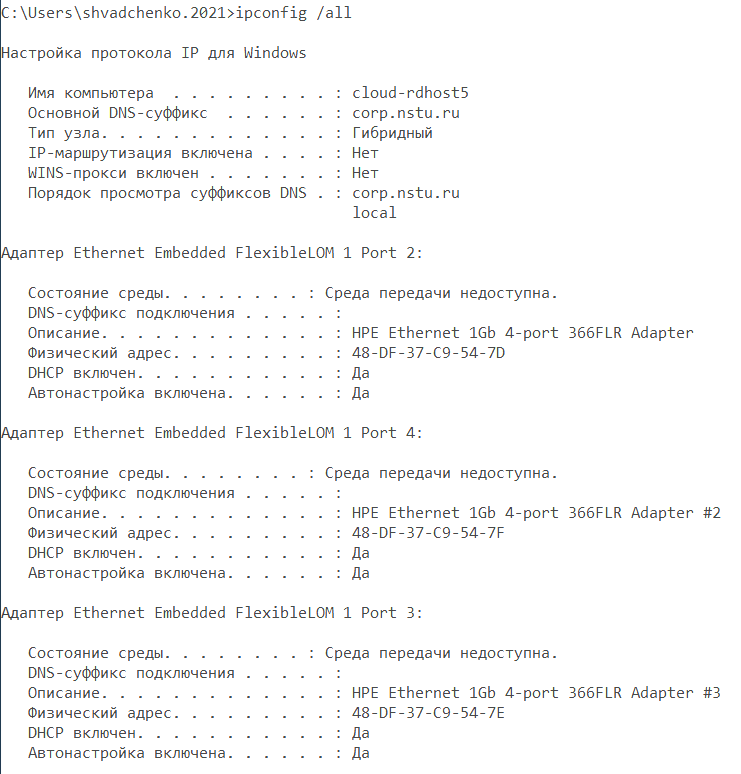
Заметим, что IPv6-адреса используют не маски, а префиксы. Они выполняют ту же функцию, что и маски - отделяют адрес подсети от адреса хоста (интерфейса). Например, длина префикса /64 указывает на то, что первые 64 бита IPv6-адреса относятся к адресу сети, а оставшиеся 64 - к адресу хоста (интерфейса).

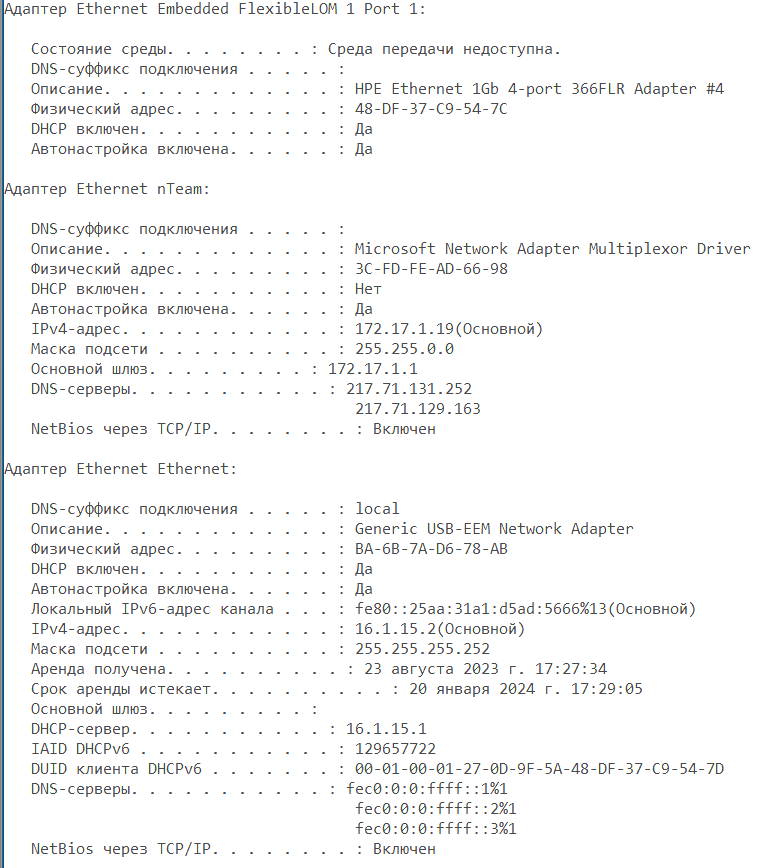
Также в IPv6 нет широковещательного адреса (он был заменен на ANYCAST).

Обратим также внимание на характеристики loopback-интерфейса. Любые сообщения,

посылаемые на этот канал принимаются им же самим. В связи с этим, значения RX и TX совпадают, а буфер передачи имеет нулевой размер.

РК





Работающие с сетью Интернет интерфейсы можно определить по наличию у них IPv4-адреса и маски подсети. Все интерфейсы имеют MAC-адрес, информацию о состоянии и автонастройке протокола DHCP, а также информацию о DHCPv6: IAID (интерфейс

клиентской системы) и DUID (идентификатор клиентской системы, используется клиентом для получения IP-адреса от DHCPv6-сервера). Также имеется информация о системе доменных имен DNS: DNS-серверы, DNS-суффиксы подключения и порядок их просмотра.

Получим статистику Ethernet:

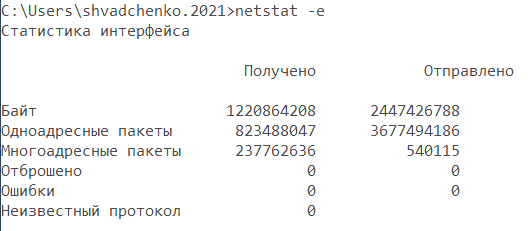


Таблица показывает количество переданных и полученных байтов и пакетов, а также их типы и информацию об ошибках при передаче/получении.

* 1. Просмотреть содержимое DNS-кэша, пояснить характеристики записей, очистить кэш.

Система доменных имен используется для получения информации о доменах. Кэш DNS

содержит информацию обо всех последних посещениях (и попытках посещения) веб-сайтов, а также другие IP-адреса сайтов. Это позволяет быстрее загружать ранее посещенные страницы.

Первые несколько записей DNS-кэша

На скриншоте видим, что кэш состоит из записей, имеющих имена, - это домены, с которыми устанавливалось соединение. Тип записи - это формат и назначение записи: наиболее популярными являются A (1) - соответствие IPv4 адреса доменному имени, AAAA (28) - IPv6 и CNAME (5) - привязка псевдонима (например, www.example.com и example.com). На скриншоте видны только записи типа

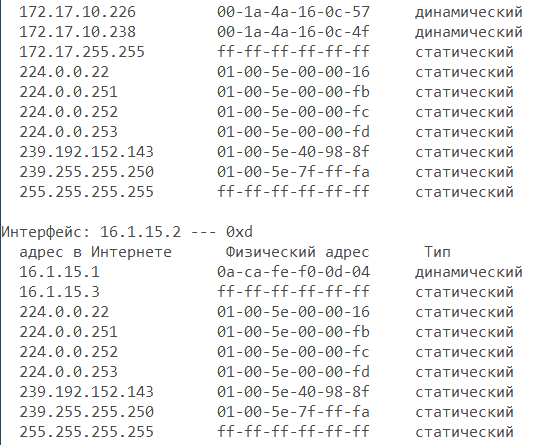
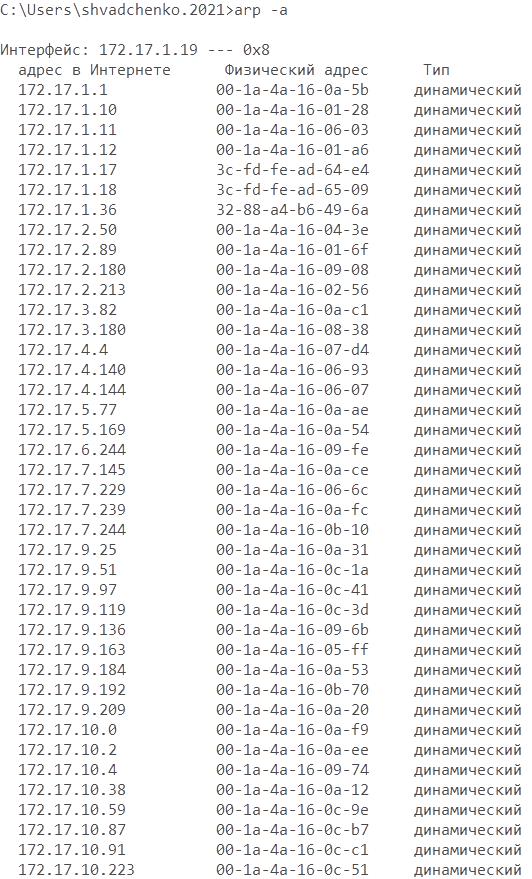
A, поэтому для них выводится A-запись (IPv4-адрес домена), а длина данных соответствует длине IPv4-адреса. Очистка DNS-кэша производится командой ipconfig /flushdns, но у нас нет для этого прав.



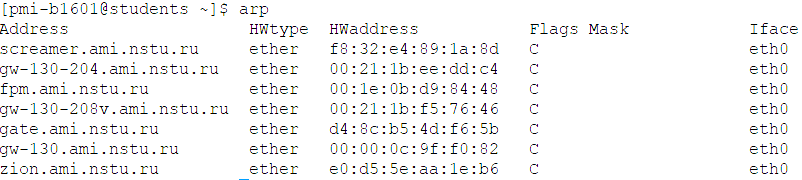
* 1. Просмотреть содержимое ARP-таблицы, пояснить характеристики записей, выполнить добавление и удаление статических записей. ARP-таблица показывает соответствия IP-адресов с MAC-адресами, а также их тип.

Добавление и удаление записей невозможно из-за отсутствия прав.

РК



fpm2.ami.nstu.ru

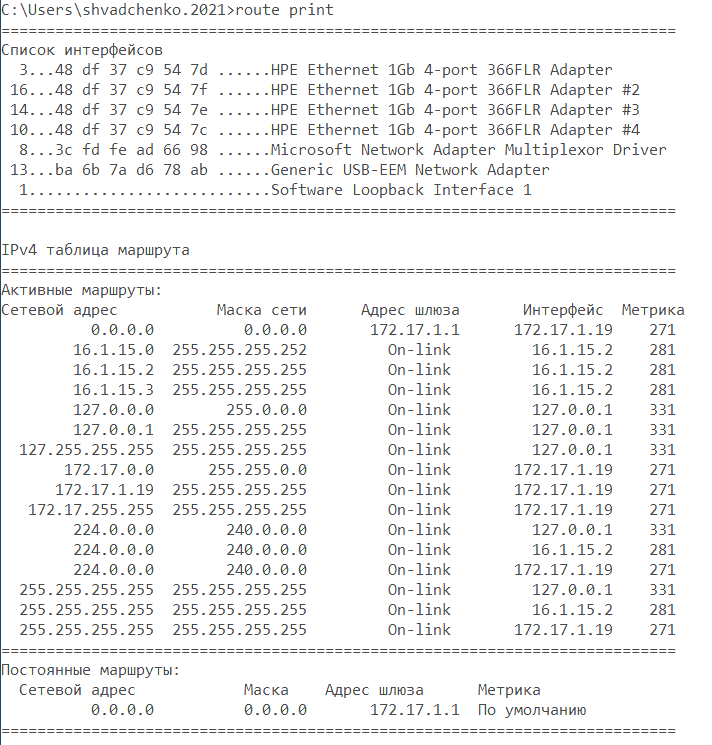


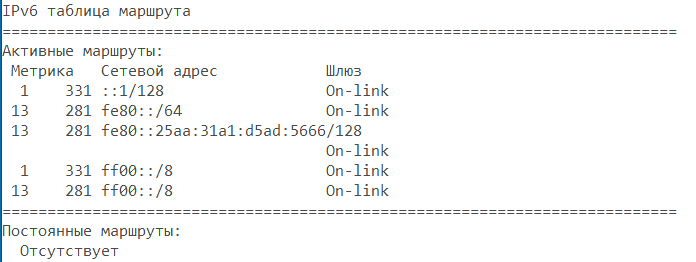
В столбце Flags Mask указан тип записей - динамический, в столбце Iface указан сетевой

интерфейс, который использует эти узлы.

* 1. Просмотреть содержимое таблицы маршрутизации, пояснить характеристики записей.

РК:



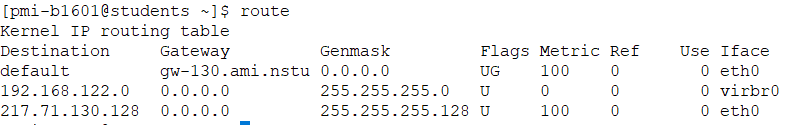


В таблице представлены сетевые адреса различных узлов сети, с помощью которых маршрутизатор определяет, куда отправлять IP-пакет. Столбец Адрес шлюза задает IP-адрес следующего ближайшего маршрутизатора (таким образом пакет продвигается к сети назначения).

Значение On-link говорит о том, что адрес доступен напрямую (не требуются доп. маршрутизаторы).

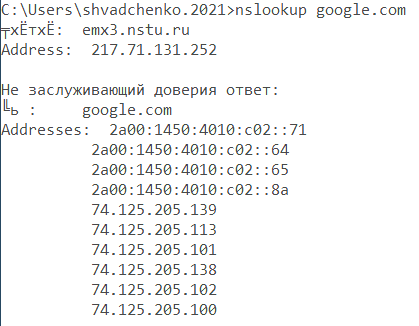
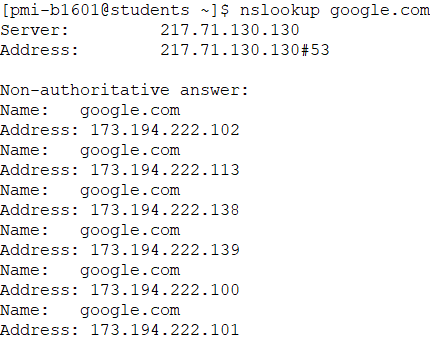
Метрика используется для выбора маршрута (чем меньше, тем маршрут предпочтительнее).

Столбец Интерфейс содержит IP-адрес выходного сетевого интерфейса данного маршрутизатора.



Здесь столбец Destination является аналогом столбца Сетевой адрес, Gateway – аналогом столбца Интерфейс, Metric и Genmask - метрика и маска соответственно.

* 1. В командном режиме на РК и на сервере определить IP-адреса поисковых систем в соответствии с вариантом из таблицы 5.1, пояснить результаты.

РК fpm2.ami.nstu.ru

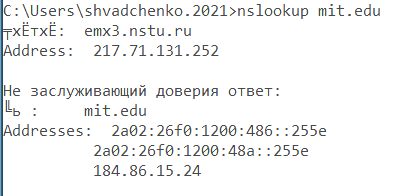
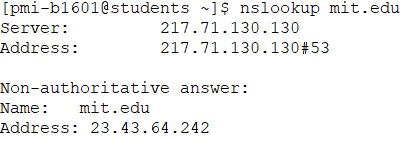
На РК видим IP-адрес 217.71.131.252 - это адрес DNS-сервера emx3.nstu.ru. Поскольку домен google.ru является поисковой системой, для распределения нагрузки он использует несколько IP-адресов, указанных в поле Addresses.

На fpm2.ami.nstu.ru используемый DNS-сервер - 217.71.130.130.

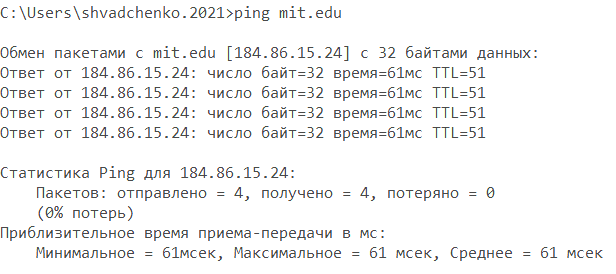
* 1. В командном режиме на РК и на сервере определить IP-адрес узлов сети в соответствии с номером варианта, указанного в таблице 5.2, выполнить его пингование и трассировку.

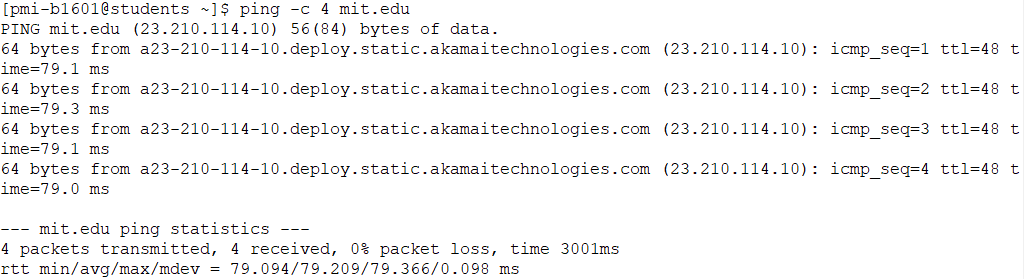
mit.edu

Определение IP адреса на РК на fpm2.ami.nstu.ru

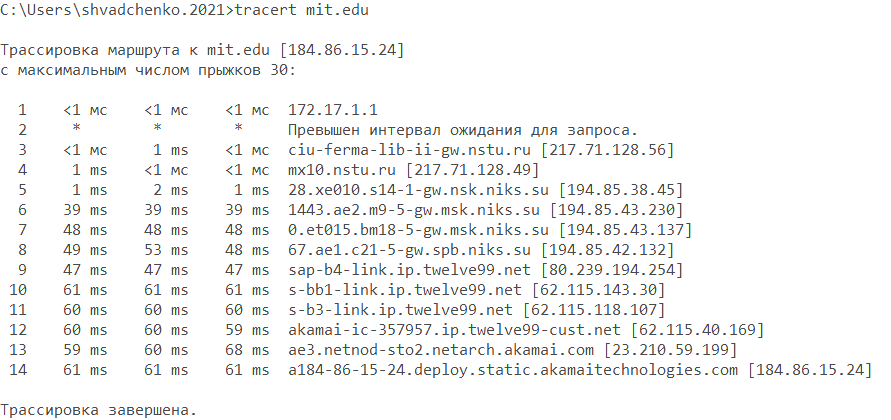
 

Пингование на РК

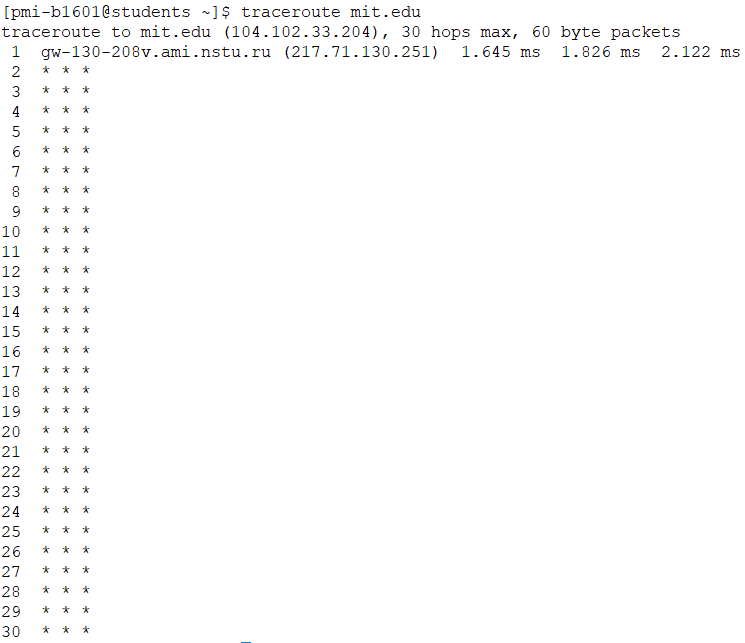
   
 Пингование на fpm2.ami.nstu.ru



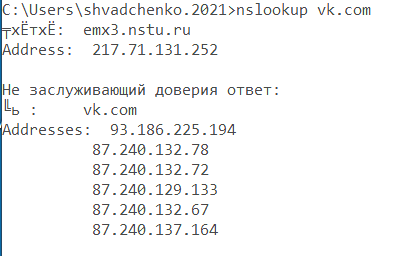
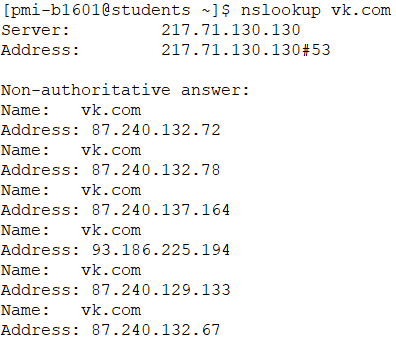
Трассировка на РК



Трассировка на fpm2.ami.nstu.ru

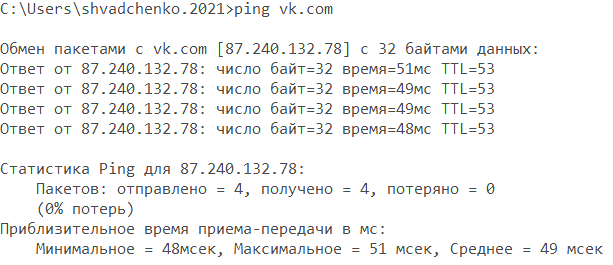
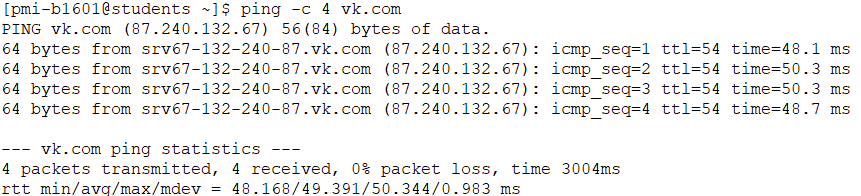


vk.com

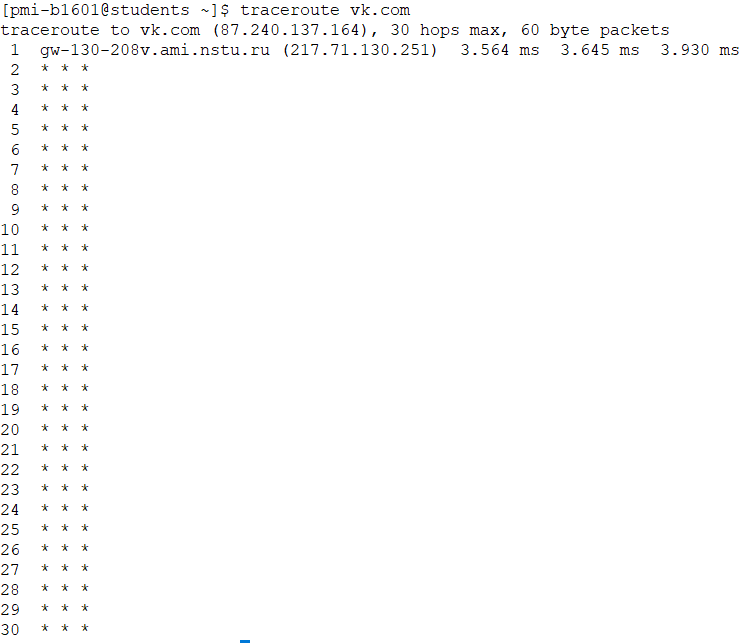
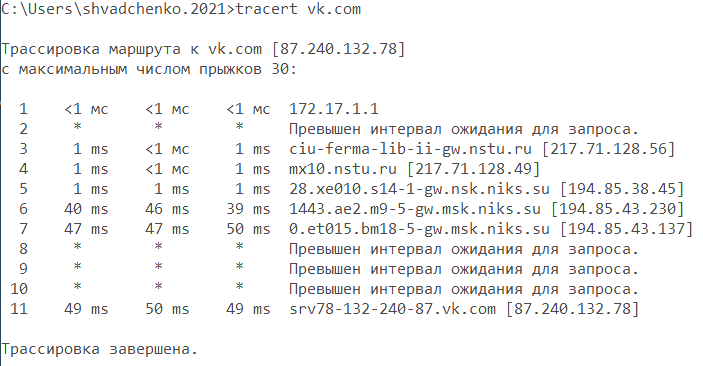
 

Определение IP на РК на fpm2.ami.nstu.ru

Пингование на РК на fpm2.ami.nstu.ru

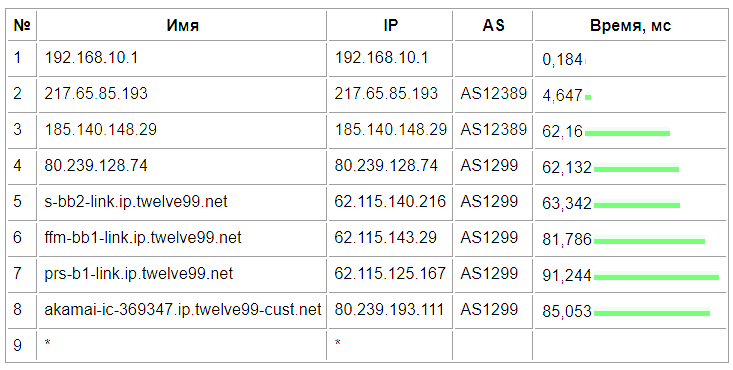
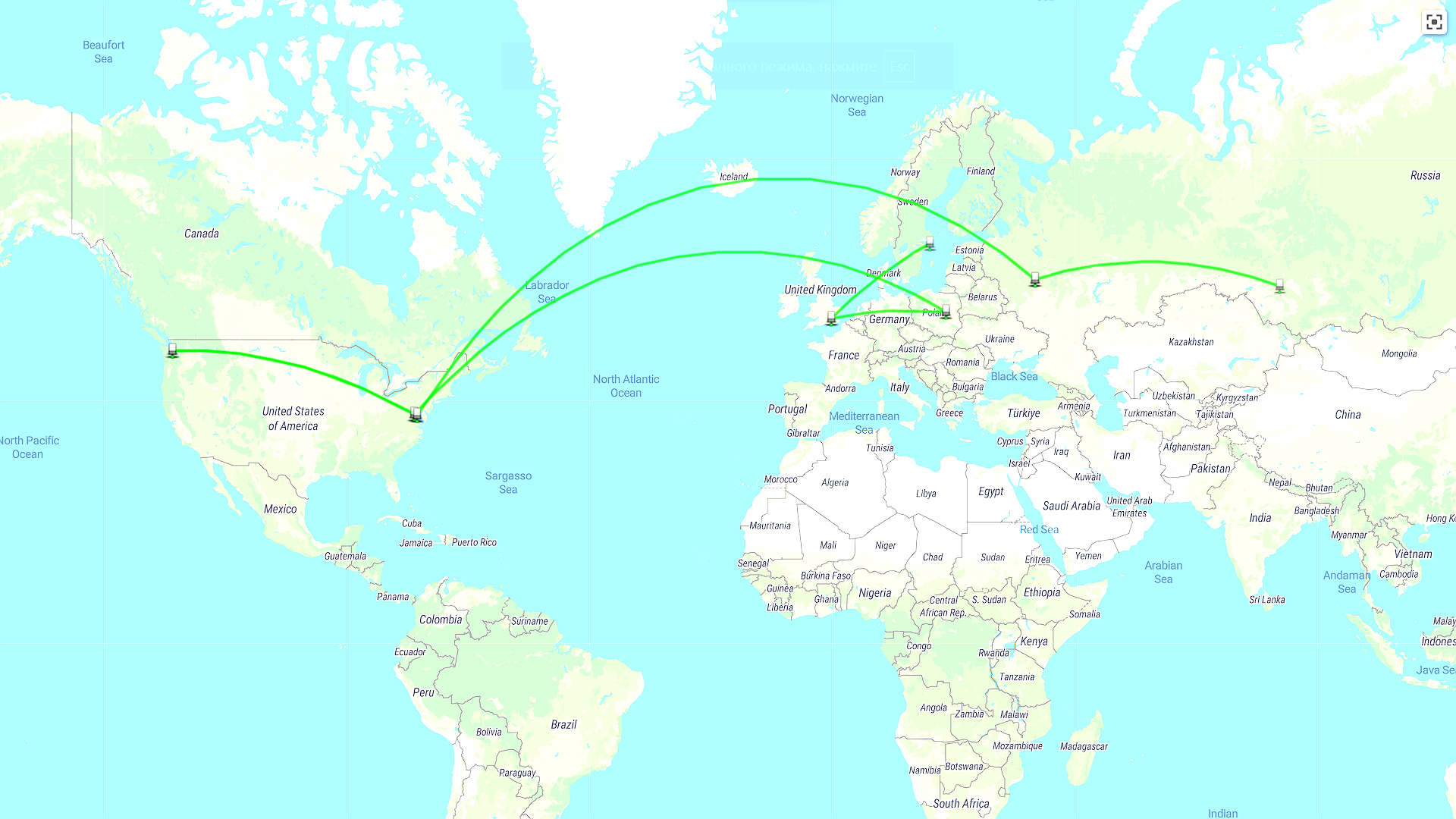
Трассировка на РК на fpm2.ami.nstu.ru



Исходя из результатов можно сделать вывод, что доступ к данным сайтам заблокирован на сервере, в то время как на РК доступ есть и на трассировке видно каждый конечный пункт.

* 1. С помощью интерактивных сетевых сервисов (например, **ping-admin.ru**) выполнить трассировку, определить местонахождение и владельца узла сети в соответствии с номером варианта, указанного в таблице 5.2. Результат трассировки в виде скриншота географической карты представить в отчете и выполнить его анализ. Начальный пункт трассировки – г. Новосибирск.

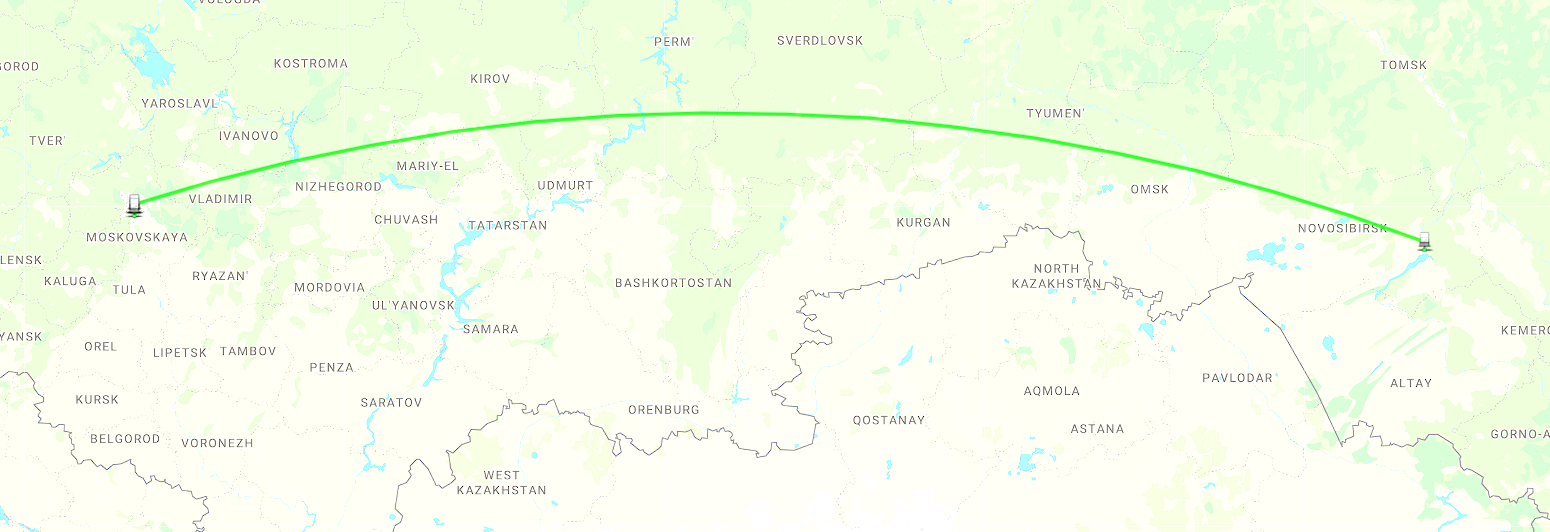
mit.edu

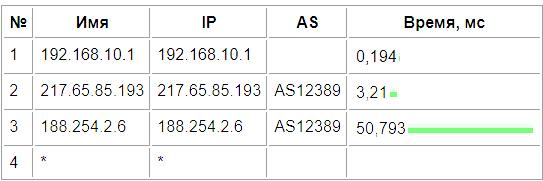


С помощью сервиса whois определили местоположение и владельца узла



vk.com





Судя по трассировке владелец находится в Москве.

С помощью сервиса whois определили владельца узла



2 часть

Реализовать Windows-приложение, которое будет выполнять основные функции одной из утилит мониторинга сети в соответствии с таблицей 5.3. Результат работы функций должен быть идентичен результату работы программ **ping.**

Текст программы

main.cpp

#include <iostream>

#include <string>

#include "Network.h"

#pragma comment(lib, "iphlpapi.lib")

#pragma comment(lib, "ws2\_32.lib")

using namespace std;

int main()

{

InitLibrary();

SOCKET sock = InitSock();

string hostname;

cout << "Enter hostname: ";

cin >> hostname;

string host;

IPAddr ipaddr;

ConvertHostName(hostname, host, ipaddr);

cout

<< "\n\nPinging with " << hostname << " [" << host << "] ";

MakeRequest(ipaddr, host);

}

Network.h

#pragma once

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#include <winsock2.h>

#include <ws2tcpip.h>

#include <iphlpapi.h>

#include <icmpapi.h>

#include <iostream>

#include <string>

using namespace std;

const int BufferSize = 4;

const int requestCount = 4;

const char TTL = 64;

void InitLibrary();

SOCKET InitSock();

bool ConvertHostName(string& hostname, string& host, IPAddr& ipaddr);

void MakeRequest(IPAddr ipaddr, string& host);

network.cpp

#include "Network.h"

void InitLibrary()

{

WSAData wsaData;

WSAStartup(MAKEWORD(2, 2), &wsaData);

}

SOCKET InitSock()

{

SOCKET sock = socket(AF\_INET, SOCK\_RAW, IPPROTO\_ICMP);

setsockopt(sock, IPPROTO\_IP, IP\_TTL, &TTL, sizeof(TTL));

return sock;

}

bool ConvertHostName(string& hostname, string& host, IPAddr& ipaddr)

{

hostent\* host\_entity = gethostbyname(hostname.c\_str());

if (host\_entity == nullptr)

return false;

in\_addr\* addr = (in\_addr\*)(host\_entity->h\_addr\_list[0]);

host = inet\_ntoa(\*addr);

inet\_pton(AF\_INET, host.c\_str(), &ipaddr);

}

void MakeRequest(IPAddr ipaddr, string& host)

{

int min = 1000, average = 0, max = 0, sent = 0, received = 0; char\* buffer = new char[BufferSize];

for (int i = 0; i < BufferSize; i++)

buffer[i] = rand();

HANDLE hIcmpFile = IcmpCreateFile();

int ReplySize = sizeof(ICMP\_ECHO\_REPLY) + BufferSize;

char\* ReplyBuffer = new char[ReplySize];

for (int i = 0; i < requestCount; ++i)

{

int result = IcmpSendEcho(hIcmpFile, ipaddr, buffer, BufferSize, NULL, ReplyBuffer, ReplySize, 1000);

PICMP\_ECHO\_REPLY pEchoReply = (PICMP\_ECHO\_REPLY)ReplyBuffer;

sent++;

if (result != 0)

{

received++;

}

else

cout << endl << "ERROR!";

}

delete[] ReplyBuffer;

delete[] buffer;

cout << endl;

cout

<< "Ping statistics for " << host << ":\n" << "\tPackets: Sent = " << sent

<< ","

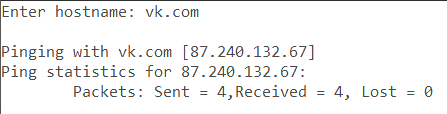
<< "Received = " << received << ", " << "Lost = "

<< sent - received << endl;

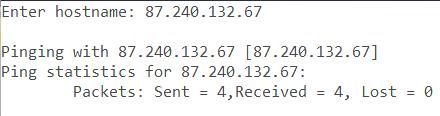
}

Тестирование

Ввод имени хоста



Ввод IP-адреса



Ввод несуществующего адреса

