NOVOSIBIRSK STATE UNIVERSITY
Department of Computer Science
Group 19213

CONNECTION CHECK

LAB 1 REPORT

Patrushev Boris
Supervisor: Ruslan V. Akhpashev

Novosibirsk State University
*THE REAL SCIENCE

February 2021

List of Contents

1 Introduction

What is the Internet? In fact, the *Internet* is the interaction of computers (data transfer) with each other. The *server* is also a computer, only a little more powerful, with a wider range of tasks. Accordingly, any computer can become a server, if the necessary equipment (network) is available.

In our work we will use command **ping**, which is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network, in other words it is command, which sending echo request packets to the target host and waiting for an echo reply. The command-line options of the **ping** utility and its output vary between the numerous implementations. Options may include *the size of the payload, count of tests, limits for the number of network hops* (TTL) that probes traverse, interval between the requests and time to wait for a response. The time between sending a request and receiving a response **(Round Trip Time(RTT))** allows you to determine two-way delays along the route and the frequency of packet loss, that is, indirectly determine the congestion on data transmission channels and intermediate devices.

But besides that, we are interested to know how the packet got from the computer to a certain server, that is, the *route*. The **tracert** command is responsible for this. Or more formally, **tracert** sends data to the specified host, while displaying information about all intermediate routers through which the data passed on the way to the target host. In case of problems during data delivery to any node, the program allows you to determine on which part of the network the problems have arisen.

Tasks

1. Choose 3 different sites you visit most often
2. Send an echo request to each site
3. Analyze the effect of the size of the sent packet (ping packet size) on the latency and display the graphs of the dependencies
4. Find routes from computer to server
5. Calculate the approximate distance to each routing point based on link delays
6. Create a diagram of the packet's routes in the form of a graph, where the weight coefficients of the arcs are equal to the values of the delays
7. To make a report

2 Working process

2.1 Choosing websites

So, I decide to choose these 3 different websites:

1. **nsu.ru**
2. **vk.com**
3. **yummyanime.club**

2.2 nsu.ru

2.2.1 Sending echo-request

Using *ping* command to send echo-request:

```
C:\Users\KazukiNagasa>ping nsu.ru

Pinging nsu.ru [84.237.50.25] with 32 bytes of data:
Reply from 84.237.50.25: bytes=32 time=46ms TTL=51
Reply from 84.237.50.25: bytes=32 time=5ms TTL=51
Reply from 84.237.50.25: bytes=32 time=5ms TTL=51
Reply from 84.237.50.25: bytes=32 time=5ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 46ms, Average = 15ms
```

Echo request to a node named nsu.ru[84.237.50.25] with default parameters - the number of packets is 4, the length of the data array = 32 bytes and we can see that the average time between sending a request and receiving a response, that is, RTT is 15 milliseconds

2.2.2 Dependence of the size of the sent packet on the delay time

Again we will use ***ping*** command, but now we will also add a flag ***-l $size***, which is responsible for the size of the data field in bytes of the sent request.**$size** - size in bytes(in my work I chose these values {64,128,256,512,1024,2048,4096,7000}):

```
C:\Users\KazukiNagasa>ping -l 64 nsu.ru

Pinging nsu.ru [84.237.50.25] with 64 bytes of data:
Reply from 84.237.50.25: bytes=64 time=6ms TTL=51
Reply from 84.237.50.25: bytes=64 time=6ms TTL=51
Reply from 84.237.50.25: bytes=64 time=7ms TTL=51
Reply from 84.237.50.25: bytes=64 time=8ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 8ms, Average = 6ms

C:\Users\KazukiNagasa>ping -l 128 nsu.ru

Pinging nsu.ru [84.237.50.25] with 128 bytes of data:
Reply from 84.237.50.25: bytes=128 time=67ms TTL=51
```

```
Reply from 84.237.50.25: bytes=128 time=5ms TTL=51
Reply from 84.237.50.25: bytes=128 time=7ms TTL=51
Reply from 84.237.50.25: bytes=128 time=6ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 67ms, Average = 21ms

C:\Users\KazukiNagasa>ping -l 256 nsu.ru

Pinging nsu.ru [84.237.50.25] with 256 bytes of data:
Reply from 84.237.50.25: bytes=256 time=5ms TTL=51
Reply from 84.237.50.25: bytes=256 time=6ms TTL=51
Reply from 84.237.50.25: bytes=256 time=5ms TTL=51
Reply from 84.237.50.25: bytes=256 time=12ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 12ms, Average = 7ms

C:\Users\KazukiNagasa>ping -l 512 nsu.ru

Pinging nsu.ru [84.237.50.25] with 512 bytes of data:
Reply from 84.237.50.25: bytes=512 time=5ms TTL=51
Reply from 84.237.50.25: bytes=512 time=6ms TTL=51
Reply from 84.237.50.25: bytes=512 time=6ms TTL=51
Reply from 84.237.50.25: bytes=512 time=8ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 8ms, Average = 6ms

C:\Users\KazukiNagasa>ping -l 1024 nsu.ru

Pinging nsu.ru [84.237.50.25] with 1024 bytes of data:
Reply from 84.237.50.25: bytes=1024 time=6ms TTL=51
Reply from 84.237.50.25: bytes=1024 time=6ms TTL=51
Reply from 84.237.50.25: bytes=1024 time=11ms TTL=51
Reply from 84.237.50.25: bytes=1024 time=9ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 11ms, Average = 8ms

C:\Users\KazukiNagasa>ping -l 2048 nsu.ru

Pinging nsu.ru [84.237.50.25] with 2048 bytes of data:
Reply from 84.237.50.25: bytes=2048 time=20ms TTL=51
Reply from 84.237.50.25: bytes=2048 time=11ms TTL=51
Reply from 84.237.50.25: bytes=2048 time=9ms TTL=51
Reply from 84.237.50.25: bytes=2048 time=9ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

```
    Minimum = 9ms, Maximum = 20ms, Average = 12ms

C:\Users\KazukiNagasa>ping -l 4096 nsu.ru

Pinging nsu.ru [84.237.50.25] with 4096 bytes of data:
Reply from 84.237.50.25: bytes=4096 time=10ms TTL=51
Reply from 84.237.50.25: bytes=4096 time=13ms TTL=51
Reply from 84.237.50.25: bytes=4096 time=11ms TTL=51
Reply from 84.237.50.25: bytes=4096 time=9ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 13ms, Average = 10ms

C:\Users\KazukiNagasa>ping -l 7000 nsu.ru

Pinging nsu.ru [84.237.50.25] with 7000 bytes of data:
Reply from 84.237.50.25: bytes=7000 time=12ms TTL=51
Reply from 84.237.50.25: bytes=7000 time=12ms TTL=51
Reply from 84.237.50.25: bytes=7000 time=58ms TTL=51
Reply from 84.237.50.25: bytes=7000 time=13ms TTL=51

Ping statistics for 84.237.50.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 58ms, Average = 23ms
```

From all these data, we take the average *RTT* values for each *size* and enter it into a table to make it easier to plot and analyze:

x (size of packet), bytes	64	128	256	512	1024	2048	4096	7000
y (RTT), ms	6	21	7	6	8	12	10	23

Figure 1: table for nsu.ru

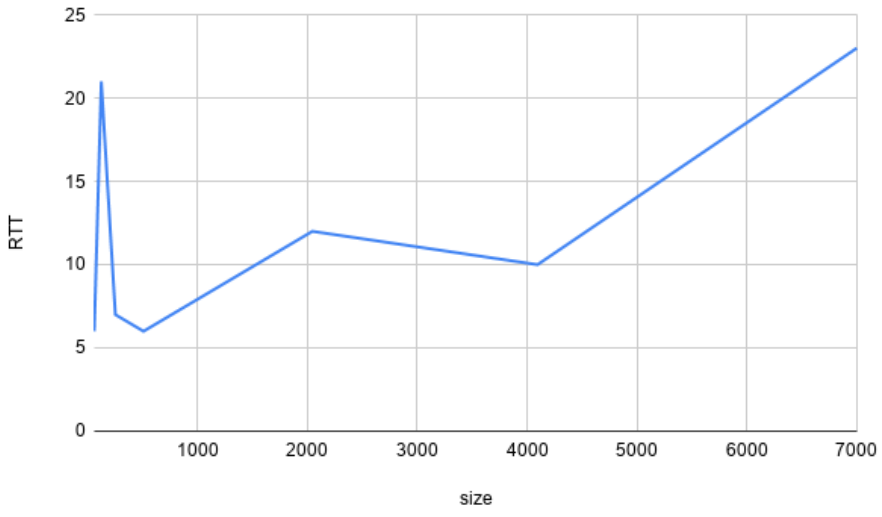


Figure 2: Dependency graph for nsu.ru

It can be concluded from the results of the data (graph) that the waiting time grows with the packet size, and therefore, it can be assumed that this is a linear growth, but it can also be seen that the waiting time is influenced not only by the packet size.

2.2.3 Routes from my computer to servers

Using *tracert* command to find route from my PC to server:

```
C:\Users\KazukiNagasa>tracert nsu.ru

Tracing route to nsu.ru [84.237.50.25]
over a maximum of 30 hops:

  1    <1 ms     2 ms    <1 ms  Dlink-Router.Dlink [192.168.0.1]
  2     2 ms     1 ms     1 ms  10.70.16.1
  3     4 ms     2 ms     4 ms  10.254.254.144
  4     3 ms     2 ms     3 ms  10.254.254.145
  5    22 ms     3 ms     2 ms  10.254.254.151
  6     2 ms     2 ms     2 ms  78.40.80.56
  7     3 ms    19 ms     2 ms  bhm-cr01-ae21.10.nsk.mts-internet.net [212.188.23.69]
  8     6 ms     3 ms     3 ms  212.188.61.10
  9     6 ms     4 ms     4 ms  79.104.246.107
 10     4 ms     4 ms     3 ms  Avantel-gw.nsk.gldn.net [212.46.198.134]
 11     4 ms     4 ms     6 ms  host-95-170-130-190.avantel.ru [95.170.130.190]
 12     5 ms     7 ms     4 ms  host20.49.237.84.nsu.ru [84.237.49.20]
 13     4 ms     4 ms     4 ms  84.237.49.164
 14     6 ms     8 ms     6 ms  host25.50.237.84.nsu.ru [84.237.50.25]

Trace complete.
```

So, we can see that the route goes through different nodes and then reaches the server we need

2.2.4 Approximate distance to each routing point based on delays

RTT (datagram round trip time) is certainly impacted by physical distance. So, RTT = 2 * Distance / Speed of propagation Also, we know that the speed of propagation would be = the speed of light for an optical circuit (3 $\cdot 10^8 m/s$) :

$$RTT = \frac{2distance}{c} \Rightarrow distance = \frac{c \cdot RTT}{2} \tag{1}$$

$$1ms \Rightarrow distance = \frac{3 \cdot 10^8 \frac{m}{s} \cdot 1ms}{2} = 150km \tag{2}$$

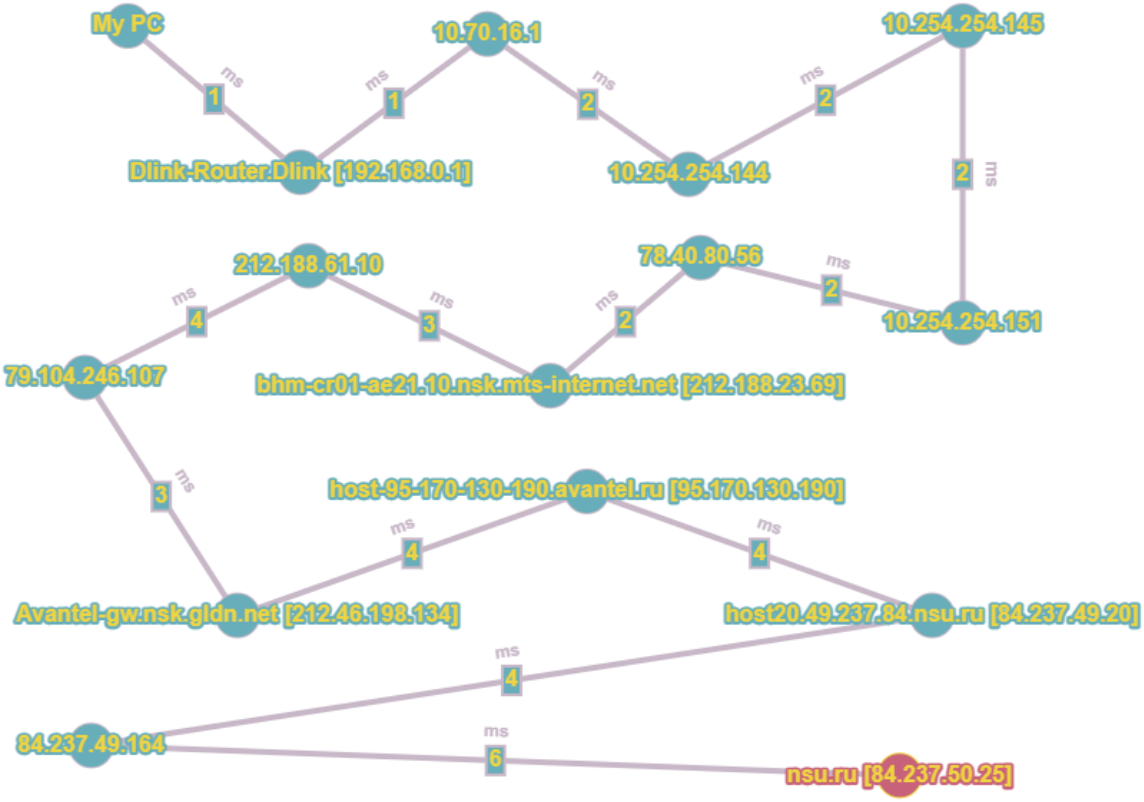
So, we can calculate the distances to each routing point:

```
Tracing route to nsu.ru [84.237.50.25]
over a maximum of 30 hops:

  1    <1 ms/<150.0 km     2 ms/300.0 km     <1 ms/<150.0 km  Dlink-Router.Dlink
    [192.168.0.1]
  2     2 ms/300.0 km      1 ms/150.0 km      1 ms/150.0 km   10.70.16.1
  3     4 ms/600.0 km      2 ms/300.0 km      4 ms/600.0 km   10.254.254.144
  4     3 ms/450.0 km      2 ms/300.0 km      3 ms/450.0 km   10.254.254.145
```

```
  5    22 ms/3300.0 km      3 ms/450.0 km      2 ms/300.0 km   10.254.254.151
  6     2 ms/300.0 km      2 ms/300.0 km      2 ms/300.0 km   78.40.80.56
  7     3 ms/450.0 km     19 ms/2850.0 km      2 ms/300.0 km   bhm-cr01-ae21.10.nsk.mts
    -internet.net [212.188.23.69]
  8     6 ms/900.0 km      3 ms/450.0 km      3 ms/450.0 km   212.188.61.10
  9     6 ms/900.0 km      4 ms/600.0 km      4 ms/600.0 km   79.104.246.107
 10     4 ms/600.0 km      4 ms/600.0 km      3 ms/450.0 km   Avantel-gw.nsk.gldn.net
    [212.46.198.134]
 11     4 ms/600.0 km      4 ms/600.0 km      6 ms/900.0 km   host-95-170-130-190.
    avantel.ru [95.170.130.190]
 12     5 ms/750.0 km      7 ms/1050.0 km      4 ms/600.0 km   host20.49.237.84.nsu.ru
    [84.237.49.20]
 13     4 ms/600.0 km      4 ms/600.0 km      4 ms/600.0 km   84.237.49.164
 14     6 ms/900.0 km      8 ms/1200.0 km      6 ms/900.0 km   host25.50.237.84.nsu.ru
    [84.237.50.25]

Trace complete.
```

2.2.5 Package route scheme(graph)

From the received data using *tracert*, we can build an undirected graph of the route from my computer to the server we need:



2.3 vk.com

2.3.1 Sending echo-request

Using *ping* command to send echo-request:

```
C:\Users\KazukiNagasa>ping vk.com

```

```
Pinging vk.com [87.240.190.72] with 32 bytes of data:
Reply from 87.240.190.72: bytes=32 time=63ms TTL=48
Reply from 87.240.190.72: bytes=32 time=63ms TTL=48
Reply from 87.240.190.72: bytes=32 time=63ms TTL=48
Reply from 87.240.190.72: bytes=32 time=67ms TTL=48

Ping statistics for 87.240.190.72:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 63ms, Maximum = 67ms, Average = 64ms
```

Echo request to a node named vk.com[87.240.190.72] with default parameters - the number of packets is 4, the length of the data array = 32 bytes and we can see that the average time between sending a request and receiving a response, that is, RTT is 64 milliseconds

2.3.2 Dependence of the size of the sent packet on the delay time

Again we will use ***ping*** command, but now we will also add a flag ***-l $size***, which is responsible for the size of the data field in bytes of the sent request.**$size** - size in bytes(in my work I chose these values {64,128,256,512,1024,2048,4096,7000}):

```
C:\Users\KazukiNagasa>ping -l 64 vk.com

Pinging vk.com [87.240.190.78] with 64 bytes of data:
Reply from 87.240.190.78: bytes=64 time=65ms TTL=48
Reply from 87.240.190.78: bytes=64 time=63ms TTL=48
Reply from 87.240.190.78: bytes=64 time=63ms TTL=48
Reply from 87.240.190.78: bytes=64 time=65ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 63ms, Maximum = 65ms, Average = 64ms

C:\Users\KazukiNagasa>ping -l 128 vk.com

Pinging vk.com [87.240.190.78] with 128 bytes of data:
Reply from 87.240.190.78: bytes=128 time=67ms TTL=48
Reply from 87.240.190.78: bytes=128 time=63ms TTL=48
Reply from 87.240.190.78: bytes=128 time=64ms TTL=48
Reply from 87.240.190.78: bytes=128 time=63ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 63ms, Maximum = 67ms, Average = 64ms

C:\Users\KazukiNagasa>ping -l 256 vk.com

Pinging vk.com [87.240.190.78] with 256 bytes of data:
Reply from 87.240.190.78: bytes=256 time=140ms TTL=48
Reply from 87.240.190.78: bytes=256 time=63ms TTL=48
Reply from 87.240.190.78: bytes=256 time=65ms TTL=48
Reply from 87.240.190.78: bytes=256 time=64ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

```
    Minimum = 63ms, Maximum = 140ms, Average = 83ms

C:\Users\KazukiNagasa>ping -l 512 vk.com

Pinging vk.com [87.240.190.78] with 512 bytes of data:
Reply from 87.240.190.78: bytes=512 time=64ms TTL=48
Reply from 87.240.190.78: bytes=512 time=63ms TTL=48
Reply from 87.240.190.78: bytes=512 time=64ms TTL=48
Reply from 87.240.190.78: bytes=512 time=64ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 63ms, Maximum = 64ms, Average = 63ms

C:\Users\KazukiNagasa>ping -l 1024 vk.com

Pinging vk.com [87.240.190.78] with 1024 bytes of data:
Reply from 87.240.190.78: bytes=1024 time=64ms TTL=48
Reply from 87.240.190.78: bytes=1024 time=64ms TTL=48
Reply from 87.240.190.78: bytes=1024 time=64ms TTL=48
Reply from 87.240.190.78: bytes=1024 time=65ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 64ms, Maximum = 65ms, Average = 64ms

C:\Users\KazukiNagasa>ping -l 2048 vk.com

Pinging vk.com [87.240.190.78] with 2048 bytes of data:
Reply from 87.240.190.78: bytes=2048 time=69ms TTL=48
Reply from 87.240.190.78: bytes=2048 time=67ms TTL=48
Reply from 87.240.190.78: bytes=2048 time=65ms TTL=48
Reply from 87.240.190.78: bytes=2048 time=65ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 65ms, Maximum = 69ms, Average = 66ms

C:\Users\KazukiNagasa>ping -l 4096 vk.com

Pinging vk.com [87.240.190.78] with 4096 bytes of data:
Reply from 87.240.190.78: bytes=4096 time=66ms TTL=48
Reply from 87.240.190.78: bytes=4096 time=69ms TTL=48
Reply from 87.240.190.78: bytes=4096 time=66ms TTL=48
Reply from 87.240.190.78: bytes=4096 time=68ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 66ms, Maximum = 69ms, Average = 67ms

C:\Users\KazukiNagasa>ping -l 7000 vk.com

Pinging vk.com [87.240.190.78] with 7000 bytes of data:
Reply from 87.240.190.78: bytes=7000 time=74ms TTL=48
Reply from 87.240.190.78: bytes=7000 time=69ms TTL=48
```

```
Reply from 87.240.190.78: bytes=7000 time=69ms TTL=48
Reply from 87.240.190.78: bytes=7000 time=69ms TTL=48

Ping statistics for 87.240.190.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 69ms, Maximum = 74ms, Average = 70ms
```

From all these data, we take the average *RTT* values for each *size* and enter it into a table to make it easier to plot and analyze:

x (size of packet), bytes	64	128	256	512	1024	2048	4096	7000
y (RTT), ms	64	64	83	63	63	66	67	70

Figure 3: table for vk.com

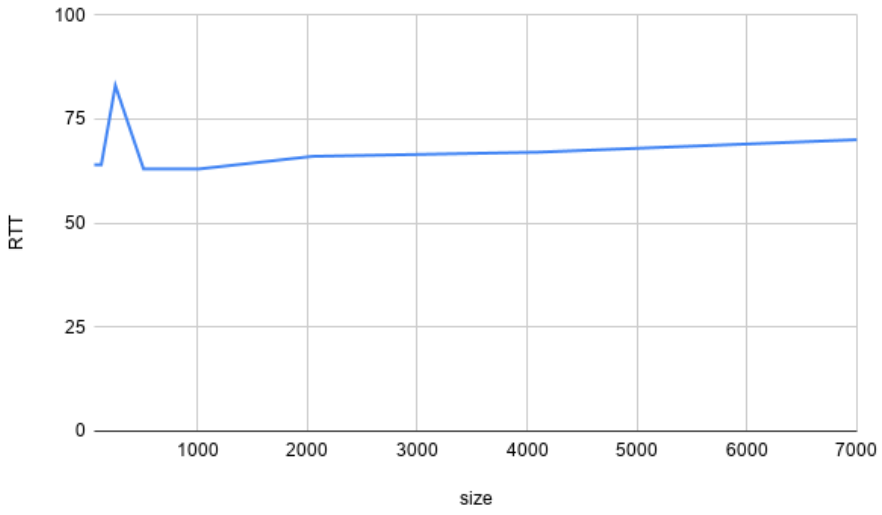


Figure 4: Dependency graph for vk.com

It can be concluded from the results of the data (graph) that the waiting time grows with the packet size, and therefore, it can be assumed that this is a linear growth, but it can also be seen that the waiting time is influenced not only by the packet size.

2.3.3 Routes from my computer to servers

Using *tracert* command to find route from my PC to server:

```
C:\Users\KazukiNagasa>tracert vk.com

Tracing route to vk.com [87.240.190.78]
over a maximum of 30 hops:

  1     1 ms    <1 ms    <1 ms  Dlink-Router.Dlink [192.168.0.1]
  2     1 ms     2 ms     1 ms  10.70.16.1
  3     3 ms     3 ms     2 ms  10.254.254.144
  4     2 ms     4 ms     1 ms  10.254.254.145
  5     4 ms     2 ms     3 ms  10.254.254.151
  6     2 ms     2 ms     3 ms  78.40.80.56
```

```
 7     3 ms     2 ms     3 ms  bhm-cr01-ae21.10.nsk.mts-internet.net [212.188.23.69]
 8     2 ms     2 ms     2 ms  bhm-VAS-IN.nsk.mts-internet.net [195.34.50.10]
 9     3 ms     2 ms     2 ms  bhm-cr02-ae13.0.nsk.mts-internet.net [212.188.2.185]
10     3 ms     3 ms     2 ms  bhm-cr01-ae1.10.nsk.mts-internet.net [195.34.50.14]
11     *        *       81 ms  bhm-cr03-ae8.54.nsk.mts-internet.net [212.188.28.227]
12     *        *        *     Request timed out.
13     *        *       63 ms  che-cr02-ae10.63.sam.mts-internet.net [212.188.42.129]
14    62 ms    63 ms    61 ms  a197-cr04-be1.63.msk.mts-internet.net [212.188.29.25]
15    63 ms     *        *     a433-cr02-be2.77.msk.mts-internet.net [212.188.28.150]
16    60 ms    61 ms    67 ms  mmon-cr01-be1.78.spb.mts-internet.net [212.188.2.53]
17    62 ms    61 ms    62 ms  212.188.18.195
18    62 ms    65 ms    62 ms  srv208-191-240-87.vk.com [87.240.191.208]
19     *        *        *     Request timed out.
20    65 ms    65 ms    64 ms  srv78-190-240-87.vk.com [87.240.190.78]

Trace complete.
```

So, we can see that the route goes through different nodes and then reaches the server we need

2.3.4 Approximate distance to each routing point based on delays

RTT (datagram round trip time) is certainly impacted by physical distance. So, RTT = 2 * Distance / Speed of propagation Also, we know that the speed of propagation would be = the speed of light for an optical circuit (3 $\cdot 10^8 m/s$) :

$$RTT = \frac{2distance}{c} \Rightarrow distance = \frac{c \cdot RTT}{2} \tag{3}$$

$$1ms \Rightarrow distance = \frac{3 \cdot 10^8 \frac{m}{s} \cdot 1ms}{2} = 150km \tag{4}$$

So, we can calculate the distances to each routing point:

```
C:\Users\KazukiNagasa>tracert vk.com

Tracing route to vk.com [87.240.190.78]
over a maximum of 30 hops:

 1     1 ms/150.0 km     <1 ms/<150.0 km    <1 ms/<150.0 km  Dlink-Router.Dlink
   [192.168.0.1]
 2     1 ms/150.0 km      2 ms/300.0 km      1 ms/150.0 km   10.70.16.1
 3     3 ms/450.0 km      3 ms/450.0 km      2 ms/300.0 km   10.254.254.144
 4     2 ms/300.0 km      4 ms/600.0 km      1 ms/150.0 km   10.254.254.145
 5     4 ms/600.0 km      2 ms/300.0 km      3 ms/450.0 km   10.254.254.151
 6     2 ms/300.0 km      2 ms/300.0 km      3 ms/450.0 km   78.40.80.56
 7     3 ms/450.0 km      2 ms/300.0 km      3 ms/450.0 km   bhm-cr01-ae21.10.nsk.mts-
   internet.net [212.188.23.69]
 8     2 ms/300.0 km      2 ms/300.0 km      2 ms/300.0 km   bhm-VAS-IN.nsk.mts-
   internet.net [195.34.50.10]
 9     3 ms/450.0 km      2 ms/300.0 km      2 ms/300.0 km   bhm-cr02-ae13.0.nsk.mts-
   internet.net [212.188.2.185]
10     3 ms/450.0 km      3 ms/450.0 km      2 ms/300.0 km   bhm-cr01-ae1.10.nsk.mts-
   internet.net [195.34.50.14]
11     *        *       81 ms/12150.0 km   bhm-cr03-ae8.54.nsk.mts-internet.net
   [212.188.28.227]
12     *        *        *     Request timed out.
```

```
 13     *        *       63 ms/9450.0 km   che-cr02-ae10.63.sam.mts-internet.net
    [212.188.42.129]
 14    62 ms/9300.0 km     63 ms/9450.0 km     61 ms/9150.0 km   a197-cr04-be1.63.msk.
    mts-internet.net [212.188.29.25]
 15    63 ms/9450.0 km      *        *     a433-cr02-be2.77.msk.mts-internet.net
    [212.188.28.150]
 16    60 ms/9000.0 km     61 ms/9150.0 km     67 ms/10050.0 km   mmon-cr01-be1.78.spb.
    mts-internet.net [212.188.2.53]
 17    62 ms/9300.0 km     61 ms/9150.0 km     62 ms/9300.0 km   212.188.18.195
 18    62 ms/9300.0 km     65 ms/9750.0 km     62 ms/9300.0 km   srv208-191-240-87.vk.
    com [87.240.191.208]
 19     *        *        *     Request timed out.
 20    65 ms/9750.0 km     65 ms/9750.0 km     64 ms/9600.0 km   srv78-190-240-87.vk.
    com [87.240.190.78]

Trace complete.
```

2.3.5 Package route scheme(graph)

From the received data using *tracert*, we can build an undirected graph of the route from my computer to the server we need:

2.4 yummyanime.club

2.4.1 Sending echo-request

Using *ping* command to send echo-request:

```
C:\Users\KazukiNagasa>ping yummyanime.club

Pinging yummyanime.club [104.26.3.13] with 32 bytes of data:
```

```
Reply from 104.26.3.13: bytes=32 time=76ms TTL=48
Reply from 104.26.3.13: bytes=32 time=76ms TTL=48
Reply from 104.26.3.13: bytes=32 time=76ms TTL=48
Reply from 104.26.3.13: bytes=32 time=87ms TTL=48

Ping statistics for 104.26.3.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 76ms, Maximum = 87ms, Average = 78ms
```

Echo request to a node named yummyanime.club [104.26.3.13] with default parameters - the number of packets is 4, the length of the data array = 32 bytes and we can see that the average time between sending a request and receiving a response, that is, RTT is 78 milliseconds

2.4.2 Dependence of the size of the sent packet on the delay time

Again we will use ***ping*** command, but now we will also add a flag ***-l $size***, which is responsible for the size of the data field in bytes of the sent request.**$size** - size in bytes(in my work I chose these values {64,128,256,512,1024,1600}):

```
C:\Users\KazukiNagasa>ping -l 64 yummyanime.club

Pinging yummyanime.club [104.26.2.13] with 64 bytes of data:
Reply from 104.26.2.13: bytes=64 time=76ms TTL=48
Reply from 104.26.2.13: bytes=64 time=75ms TTL=48
Reply from 104.26.2.13: bytes=64 time=74ms TTL=48
Reply from 104.26.2.13: bytes=64 time=74ms TTL=48

Ping statistics for 104.26.2.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 74ms, Maximum = 76ms, Average = 74ms

C:\Users\KazukiNagasa>ping -l 128 yummyanime.club

Pinging yummyanime.club [104.26.2.13] with 128 bytes of data:
Reply from 104.26.2.13: bytes=128 time=76ms TTL=48
Reply from 104.26.2.13: bytes=128 time=75ms TTL=48
Reply from 104.26.2.13: bytes=128 time=75ms TTL=48
Reply from 104.26.2.13: bytes=128 time=81ms TTL=48

Ping statistics for 104.26.2.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 75ms, Maximum = 81ms, Average = 76ms

C:\Users\KazukiNagasa>ping -l 256 yummyanime.club

Pinging yummyanime.club [104.26.2.13] with 256 bytes of data:
Reply from 104.26.2.13: bytes=256 time=423ms TTL=48
Reply from 104.26.2.13: bytes=256 time=74ms TTL=48
Reply from 104.26.2.13: bytes=256 time=74ms TTL=48
Reply from 104.26.2.13: bytes=256 time=75ms TTL=48

Ping statistics for 104.26.2.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 74ms, Maximum = 423ms, Average = 161ms
```

```
39
40 C:\Users\KazukiNagasa>ping -l 512 yummyanime.club
41
42 Pinging yummyanime.club [104.26.2.13] with 512 bytes of data:
43 Reply from 104.26.2.13: bytes=512 time=74ms TTL=48
44 Reply from 104.26.2.13: bytes=512 time=75ms TTL=48
45 Reply from 104.26.2.13: bytes=512 time=75ms TTL=48
46 Reply from 104.26.2.13: bytes=512 time=75ms TTL=48
47
48 Ping statistics for 104.26.2.13:
49     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
50 Approximate round trip times in milli-seconds:
51     Minimum = 74ms, Maximum = 75ms, Average = 74ms
52
53 C:\Users\KazukiNagasa>ping -l 1024 yummyanime.club
54
55 Pinging yummyanime.club [104.26.2.13] with 1024 bytes of data:
56 Reply from 104.26.2.13: bytes=1024 time=75ms TTL=48
57 Reply from 104.26.2.13: bytes=1024 time=78ms TTL=48
58 Reply from 104.26.2.13: bytes=1024 time=75ms TTL=48
59 Reply from 104.26.2.13: bytes=1024 time=75ms TTL=48
60
61 Ping statistics for 104.26.2.13:
62     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
63 Approximate round trip times in milli-seconds:
64     Minimum = 75ms, Maximum = 78ms, Average = 75ms
65
66 C:\Users\KazukiNagasa>ping -l 1600 yummyanime.club
67
68 Pinging yummyanime.club [104.26.2.13] with 1600 bytes of data:
69 Reply from 104.26.2.13: bytes=1600 time=78ms TTL=48
70 Reply from 104.26.2.13: bytes=1600 time=79ms TTL=48
71 Reply from 104.26.2.13: bytes=1600 time=77ms TTL=48
72 Request timed out.
73
74 Ping statistics for 104.26.2.13:
75     Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
76 Approximate round trip times in milli-seconds:
77     Minimum = 77ms, Maximum = 79ms, Average = 78ms
```

From all these data, we take the average *RTT* values for each *size* and enter it into a table to make it easier to plot and analyze:

x (size of packet), bytes	64	128	256	512	1024	1600
y (RTT), ms	74	76	161	74	75	78

Figure 5: table for yummyanime.club

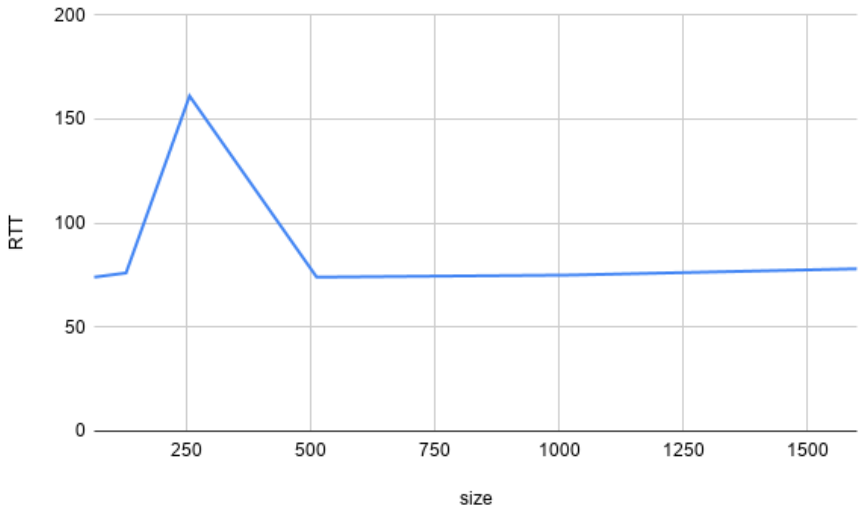


Figure 6: Dependency graph for yummyanime.club

It can be concluded from the results of the data (graph) that the waiting time grows with the packet size, and therefore, it can be assumed that this is a linear growth, but it can also be seen that the waiting time is influenced not only by the packet size.

2.4.3 Routes from my computer to servers

Using *tracert* command to find route from my PC to server:

```
C:\Users\KazukiNagasa>tracert yummyanime.club

Tracing route to yummyanime.club [104.26.3.13]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  Dlink-Router.Dlink [192.168.0.1]
  2     1 ms     2 ms     1 ms  10.70.16.1
  3     6 ms     2 ms     1 ms  10.254.254.145
  4     3 ms     4 ms     5 ms  10.254.254.151
  5     2 ms     2 ms     3 ms  78.40.80.56
  6     7 ms     3 ms     3 ms  bhm-cr01-ae21.10.nsk.mts-internet.net [212.188.23.69]
  7    78 ms    77 ms    77 ms  bhm-cr03-ae8.54.nsk.mts-internet.net [212.188.28.227]
  8     *        *        *     Request timed out.
  9    82 ms     *       80 ms  che-cr02-ae10.63.sam.mts-internet.net [212.188.42.129]
 10    53 ms    57 ms    52 ms  a197-cr04-be1.63.msk.mts-internet.net [212.188.29.25]
 11    53 ms    53 ms    53 ms  mag9-cr01-be10.77.msk.mts-internet.net [195.34.50.74]
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14    76 ms    75 ms    76 ms  bro-cr01-be7.135.stk.mts-internet.net [195.34.50.146]
 15    76 ms    78 ms    76 ms  netnod-ix-ge-b-sth-1500.cloudflare.com [194.68.128.246]
 16    76 ms    76 ms    75 ms  104.26.3.13

Trace complete.
```

So, we can see that the route goes through different nodes and then reaches the server we need

2.4.4 Approximate distance to each routing point based on delays

RTT (datagram round trip time) is certainly impacted by physical distance. So, RTT = 2 * Distance / Speed of propagation Also, we know that the speed of propagation would be = the speed of light for an optical circuit (3 $\cdot 10^8 m/s$) :

$$RTT = \frac{2distance}{c} \Rightarrow distance = \frac{c \cdot RTT}{2} \tag{5}$$

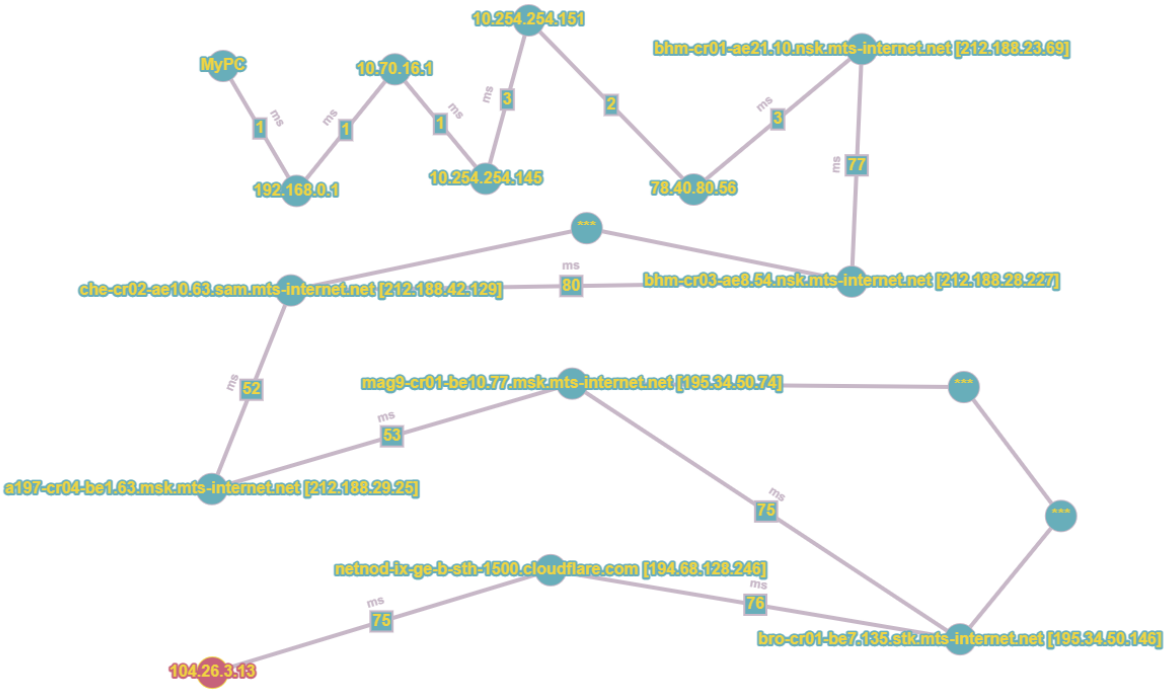
$$1ms \Rightarrow distance = \frac{3 \cdot 10^8 \frac{m}{s} \cdot 1ms}{2} = 150km \tag{6}$$

So, we can calculate the distances to each routing point:

```
C:\Users\KazukiNagasa>tracert yummyanime.club

Tracing route to yummyanime.club [104.26.3.13]
over a maximum of 30 hops:

  1    <1 ms/<150.0 km    <1 ms/<150.0 km    <1 ms/<150.0 km  Dlink-Router.Dlink
    [192.168.0.1]
  2     1 ms/150.0 km      2 ms/300.0 km      1 ms/150.0 km   10.70.16.1
  3     6 ms/900.0 km      2 ms/300.0 km      1 ms/150.0 km   10.254.254.145
  4     3 ms/450.0 km      4 ms/600.0 km      5 ms/750.0 km   10.254.254.151
  5     2 ms/300.0 km      2 ms/300.0 km      3 ms/450.0 km   78.40.80.56
  6     7 ms/1050.0 km      3 ms/450.0 km      3 ms/450.0 km   bhm-cr01-ae21.10.nsk.mts
    -internet.net [212.188.23.69]
  7    78 ms/11700.0 km     77 ms/11550.0 km     77 ms/11550.0 km   bhm-cr03-ae8.54.nsk
    .mts-internet.net [212.188.28.227]
  8     *        *        *     Request timed out.
  9    82 ms/12300.0 km      *       80 ms/12000.0 km   che-cr02-ae10.63.sam.mts-
    internet.net [212.188.42.129]
 10    53 ms/7950.0 km     57 ms/8550.0 km     52 ms/7800.0 km   a197-cr04-be1.63.msk.
    mts-internet.net [212.188.29.25]
 11    53 ms/7950.0 km     53 ms/7950.0 km     53 ms/7950.0 km   mag9-cr01-be10.77.msk.
    mts-internet.net [195.34.50.74]
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14    76 ms/11400.0 km     75 ms/11250.0 km     76 ms/11400.0 km   bro-cr01-be7.135.
    stk.mts-internet.net [195.34.50.146]
 15    76 ms/11400.0 km     78 ms/11700.0 km     76 ms/11400.0 km   netnod-ix-ge-b-sth
    -1500.cloudflare.com [194.68.128.246]
 16    76 ms/11400.0 km     76 ms/11400.0 km     75 ms/11250.0 km   104.26.3.13

Trace complete.
```

2.4.5 Package route scheme(graph)

From the received data using *tracert*, we can build an undirected graph of the route from my computer to the server we need:


10.254.254.151
bhm-cr01-ae21.10.nsk.mts-internet.net [212.188.23.69]
MyPC
10.70.16.1
ms
1
ms
1
ms
1
ms
3
2
ms
3
ms
77
192.168.0.1
10.254.254.145
78.40.80.56
***
che-cr02-ae10.63.sam.mts-internet.net [212.188.42.129]
ms
80
bhm-cr03-ae8.54.nsk.mts-internet.net [212.188.28.227]
ms
52
mag9-cr01-be10.77.msk.mts-internet.net [195.34.50.74]
***
ms
53
a197-cr04-be1.63.msk.mts-internet.net [212.188.29.25]
ms
75
***
netnod-ix-ge-b-sth-1500.cloudflare.com [194.68.128.246]
ms
75
ms
76
bre-cr01-be7.135.stk.mts-internet.net [195.34.50.146]
104.26.3.13