

**Московский государственный технический  
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Сети и телекоммуникации»**

**Отчет по лабораторной работе №6  
«Изучение принципов работы утилит для исследования и  
мониторинга состояния сети»  
Вариант №3**

Выполнил:  
студент группы ИУ5-51Б  
Бирюкова Екатерина  
Подпись и дата:

Проверил:  
  
Подпись и дата:

## Цель работы

Получение базовых навыков по использованию основных сетевых утилит, применяемых для исследования и мониторинга состояния сети.

Изучение принципов их работы.

## Задание:

1. Просмотреть параметры сетевого интерфейса лабораторного ПК, воспользовавшись утилитой `ifconfig`. Выведенную на экран информацию сохранить для отчёта. Для того чтобы сразу записать вывод команды в файл, можно воспользоваться утилитой `tee`
2. С помощью утилиты `ping` проверить состояние связи с узлами, заданными в табл. 1.3. Результаты выполнения сохранить для отчета. Количество отправляемых до каждого узла «эхо-запросов» следует ограничить 4-5 пакетами. По результатам проверки состояния связи заполнить таблицу, приведенную ниже.

Доменное имя	IP-адрес	Страна	Число потерянных запросов	Среднее время прохождения запроса	TTL
...	...	...	...	...	...

3. При помощи утилиты `tracert` произвести трассировку узлов, заданных в табл. 1.3. Результаты протоколировать в файл. По результатам трассировки составить графики времени прохождения маршрутизаторов для каждого узла (для 3 пакетов), указать наиболее узкие места в сети.

№ варианта	Исследуемые узлы
3	<code>www.industry.su</code> <code>www.yandex.ru</code> <code>www.oracle.com</code>

4. Получить маршрут прохождения пакетов до одного из заданных в варианте узлов при помощи утилиты `ping`. Результаты протоколировать в файл. Для выполнения этого задания необходимо последовательно посылать «эхо-запросы» на искомый узел, последовательно увеличивая параметр TTL на 1. Начиная с TTL = 1 и заканчивая TTL, на котором будет достигнут

искомый узел. Количество отправляемых на каждом шаге пакетов следует ограничить 2-3.

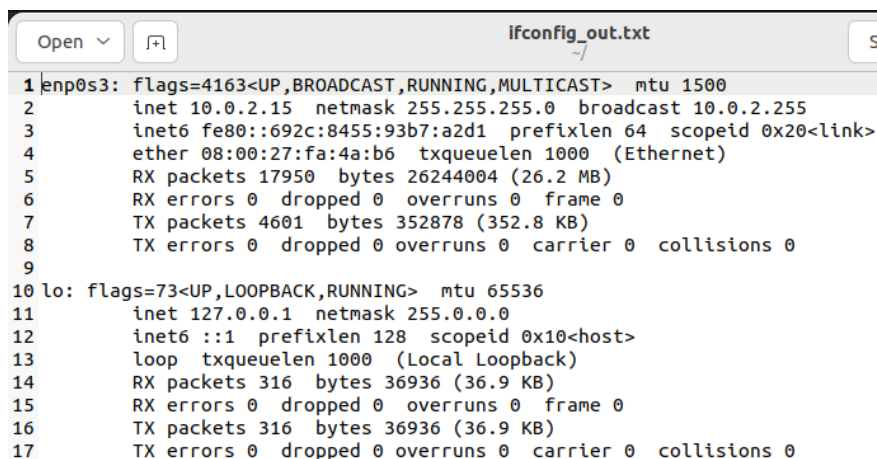
5. Определить маршрут прохождения пакетов до узла, выбранного в предыдущем пункте при помощи утилиты mtr. Результаты протоколировать в файл.
6. Провести сравнение результатов определения маршрута, полученных с помощью утилиты traceroute, ping, mtr.
7. Построить графическую карту трассировки одновременно ко всем заданным в табл. 1.3 узлам при помощи программы tracemap.

### Ход лабораторной работы:

1. Просмотреть параметры сетевого интерфейса лабораторного ПК, воспользовавшись утилитой ifconfig. Выведенную на экран информацию сохранить для отчёта. Для того чтобы сразу записать вывод команды в файл, можно воспользоваться утилитой tee.

```
stud51@ubuntu18:~$ ifconfig | tee ifconfig_out.txt
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::692c:8455:93b7:a2d1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:fa:4a:b6 txqueuelen 1000 (Ethernet)
    RX packets 17950 bytes 26244004 (26.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4601 bytes 352878 (352.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 316 bytes 36936 (36.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 316 bytes 36936 (36.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



```
Open  [+] ifconfig_out.txt
1 enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
2     inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
3     inet6 fe80::692c:8455:93b7:a2d1 prefixlen 64 scopeid 0x20<link>
4     ether 08:00:27:fa:4a:b6 txqueuelen 1000 (Ethernet)
5     RX packets 17950 bytes 26244004 (26.2 MB)
6     RX errors 0 dropped 0 overruns 0 frame 0
7     TX packets 4601 bytes 352878 (352.8 KB)
8     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
9
10 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
11     inet 127.0.0.1 netmask 255.0.0.0
12     inet6 ::1 prefixlen 128 scopeid 0x10<host>
13     loop txqueuelen 1000 (Local Loopback)
14     RX packets 316 bytes 36936 (36.9 KB)
15     RX errors 0 dropped 0 overruns 0 frame 0
16     TX packets 316 bytes 36936 (36.9 KB)
17     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. С помощью утилиты ping проверить состояние связи с узлами, заданными в табл. 1.3. Результаты выполнения сохранить для отчета. Количество отправляемых до каждого узла «эхо-запросов» следует ограничить 4-5 пакетами. По результатам проверки состояния связи заполнить таблицу, приведенную ниже.

```
stud51@ubuntu18:~$ ping -c 5 www.industry.su
PING www.industry.su (31.31.205.163) 56(84) bytes of data.
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=1 ttl=56 time=10.0 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=2 ttl=56 time=8.67 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=3 ttl=56 time=5.78 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=4 ttl=56 time=7.10 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=5 ttl=56 time=12.8 ms

--- www.industry.su ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 5.783/8.880/12.819/2.435 ms
```

```
stud51@ubuntu18:~$ ping -c 5 www.yandex.ru
PING www.yandex.ru (77.88.44.55) 56(84) bytes of data.
64 bytes from yandex.ru (77.88.44.55): icmp_seq=1 ttl=57 time=343 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=2 ttl=57 time=8.30 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=3 ttl=57 time=9.28 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=4 ttl=57 time=10.5 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=5 ttl=57 time=7.64 ms

--- www.yandex.ru ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 7.636/75.707/342.823/133.561 ms
```

```
stud51@ubuntu18:~$ ping -c 5 www.oracle.com
PING e2581.dscx.akamaiedge.net (92.122.109.102) 56(84) bytes of data.
64 bytes from a92-122-109-102.deploy.static.akamaitechnologies.com (92.122.109.102): icmp_seq=1 ttl=57 time=23.5 ms
64 bytes from a92-122-109-102.deploy.static.akamaitechnologies.com (92.122.109.102): icmp_seq=2 ttl=57 time=20.2 ms
64 bytes from a92-122-109-102.deploy.static.akamaitechnologies.com (92.122.109.102): icmp_seq=3 ttl=57 time=22.6 ms
64 bytes from a92-122-109-102.deploy.static.akamaitechnologies.com (92.122.109.102): icmp_seq=4 ttl=57 time=21.1 ms
64 bytes from a92-122-109-102.deploy.static.akamaitechnologies.com (92.122.109.102): icmp_seq=5 ttl=57 time=23.8 ms

--- e2581.dscx.akamaiedge.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4013ms
rtt min/avg/max/mdev = 20.239/22.231/23.806/1.381 ms
```

Таблица результатов:

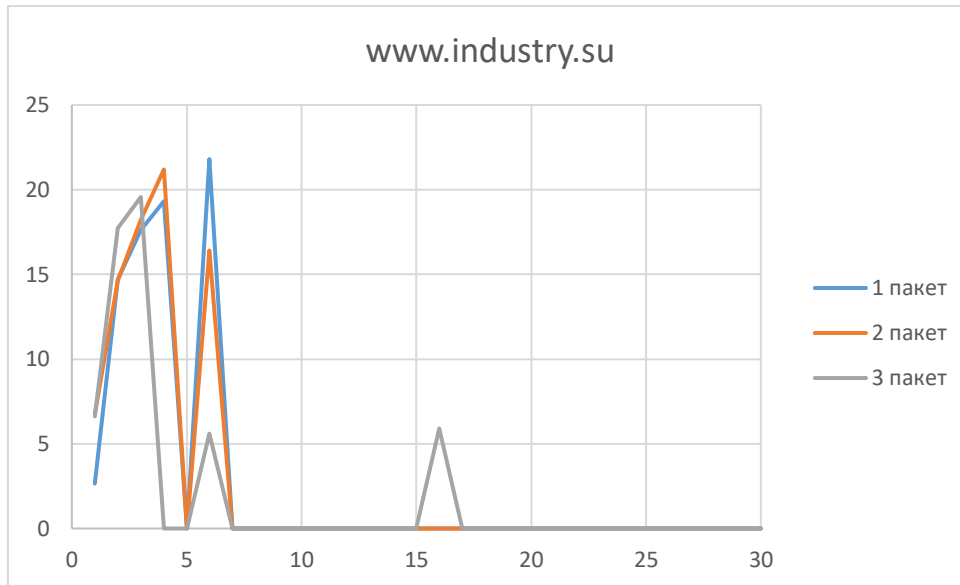
Доменное имя	IP-адрес	Страна	Число потерянных запросов	Среднее время прохождения запроса	TTL
www.industry.su	31.31.205.163	Россия	0	8.880	56
www.yandex.ru	77.88.44.55	Россия	0	75.707	57
www.oracle.com	92.122.109.102	Латвия	0	22.231	57

3. При помощи утилиты traceroute произвести трассировку узлов, заданных в табл. 1.3. Результаты протоколировать в файл. По результатам трассировки

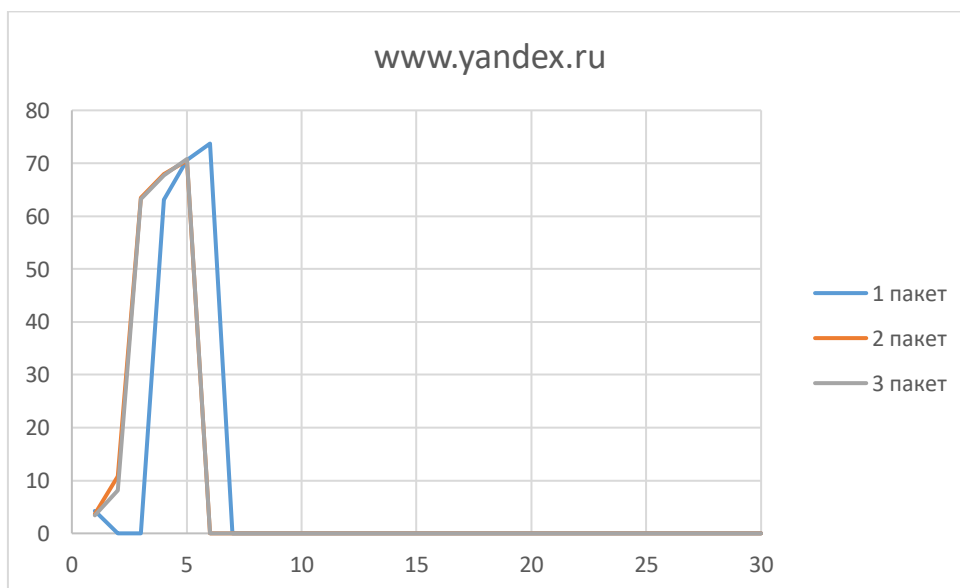
составить графики времени прохождения маршрутизаторов для каждого узла (для 3 пакетов), указать наиболее узкие места в сети.

```
stud51@ubuntu18:~$ traceroute -m 30 www.industry.su > traceroute_industry.txt
stud51@ubuntu18:~$ traceroute -m 30 www.yandex.ru > traceroute_yandex.txt
stud51@ubuntu18:~$ traceroute -m 30 www.oracle.com > traceroute_oracle.txt
```

Графики времени прохождения маршрутизаторов:

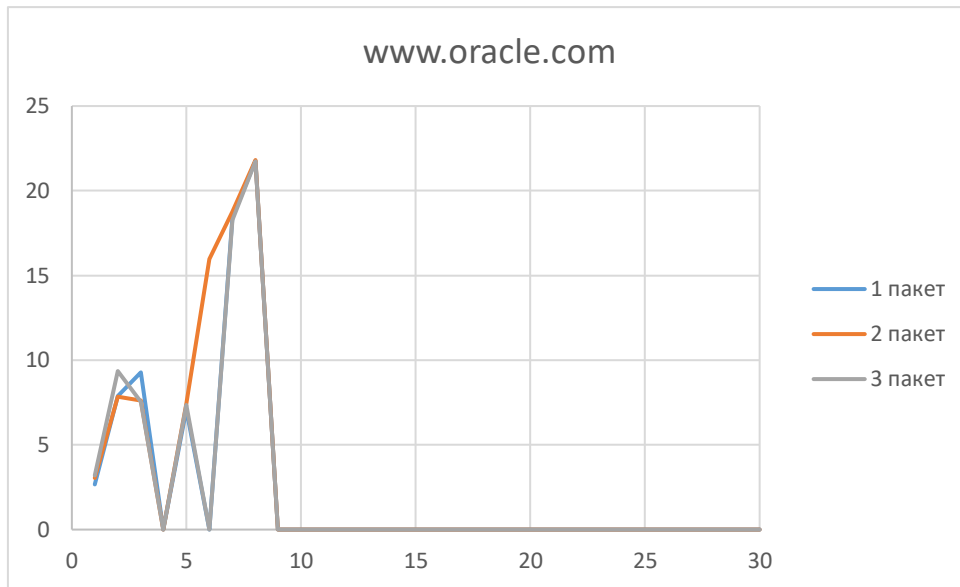


Из анализа видно, что наибольшая средняя задержка наблюдается на хопе 3. Также, хоп 6, хотя и имеет меньшую среднюю задержку, но показывает разброс значений (от 5.616 ms до 21.803 ms), что может говорить о нестабильности.



Из анализа видно, что наибольшая средняя задержка наблюдается на хопе 5. Также стоит отметить нестабильность хопов 2 и 3, которые показывают

задержку, как минимум для одного из пакетов, а также рост задержки с 0 до 60 мс.



Из анализа видно, что наибольшая средняя задержка наблюдается на хопе 8. Также стоит отметить хоп 6, который имеет задержку только для второго пакета (15.961 ms).

4. Получить маршрут прохождения пакетов до одного из заданных в варианте узлов при помощи утилиты ping. Результаты протоколировать в файл.

```
1 #! /bin/bash
2
3 target="www.yandex.ru"
4 max_ttl=30
5 logfile="ping_trace_yandex.txt"
6
7 > "$logfile"
8 found=0
9 for (( ttl=1; ttl<=max_ttl; ttl++ )); do
10     echo "Check TTL: $ttl" >> "$logfile"
11     ping -c 3 -t "$ttl" "$target" >> "$logfile"
12
13     if grep -q "bytes from" <(ping -c 1 -t "$ttl" "$target"); then
14         echo "Target node is reached at ttl: $ttl" >> "$logfile"
15         found=1
16         break
17     fi
18     echo "-----" >> "$logfile"
19 done
20
21 if [ $found -eq 0 ]; then
22     echo "Target node wasn't reached after $max_ttl tries." >> "$logfile"
23 fi
24
25 echo "Results are saved in $logfile"
```

```
stud51@ubuntu18:~$ ./ping_trace.sh
Results are saved in ping_trace_yandex.txt
```

Файлы результата:

ping\_trace\_yandex.txt

```
1 Check TTL: 1
2 PING www.yandex.ru (77.88.44.55) 56(84) bytes of data.
3 From _gateway (192.168.0.1) icmp_seq=1 Time to live exceeded
4 From _gateway (192.168.0.1) icmp_seq=2 Time to live exceeded
5 From _gateway (192.168.0.1) icmp_seq=3 Time to live exceeded
6
7 --- www.yandex.ru ping statistics ---
8 3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2004ms
9
10 -----
11 Check TTL: 2
12 PING www.yandex.ru (5.255.255.77) 56(84) bytes of data.
13 From 192.168.213.1 (192.168.213.1) icmp_seq=1 Time to live exceeded
14 From 192.168.213.1 (192.168.213.1) icmp_seq=2 Time to live exceeded
15 From 192.168.213.1 (192.168.213.1) icmp_seq=3 Time to live exceeded
16
17 --- www.yandex.ru ping statistics ---
18 3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2001ms
19
20 -----
21 Check TTL: 3
22 PING www.yandex.ru (77.88.44.55) 56(84) bytes of data.
23 From rkn.trancom.ru (172.19.250.201) icmp_seq=1 Time to live exceeded
24 From rkn.trancom.ru (172.19.250.201) icmp_seq=2 Time to live exceeded
25 From rkn.trancom.ru (172.19.250.201) icmp_seq=3 Time to live exceeded
26
27 --- www.yandex.ru ping statistics ---
28 3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2005ms
29
30 -----
31 Check TTL: 4
32 PING www.yandex.ru (5.255.255.77) 56(84) bytes of data.
33 From gw1.trancom.ru (172.19.250.5) icmp_seq=3 Time to live exceeded
34
35 --- www.yandex.ru ping statistics ---
36 3 packets transmitted, 0 received, +1 errors, 100% packet loss, time 2025ms
37
38 -----
39 Check TTL: 5
```

5. Определить маршрут прохождения пакетов до узла, выбранного в предыдущем пункте при помощи утилиты mtr. Результаты протоколировать в файл.

```
stud51@ubuntu18:~$ mtr --report-cycle=1 --report www.yandex.ru > mtr_output_yandex.txt
stud51@ubuntu18:~$
```

mtr\_output\_yandex.txt

Open ▾

⌕

mtr\_output\_yandex.txt

~/

1 Start: 2024-12-17T21:44:04+0300

2 HOST: ubuntu18

3 1. | -- \_gateway 0.0% 1 19.1 19.1 19.1 19.1 0.0

4 2. | -- 192.168.213.1 0.0% 1 5.7 5.7 5.7 5.7 0.0

5 3. | -- rkn.trancom.ru 0.0% 1 9.2 9.2 9.2 9.2 0.0

6 4. | -- ??? 100.0 1 0.0 0.0 0.0 0.0 0.0

7 5. | -- yandex.msk.piter-ix.net 0.0% 1 6.4 6.4 6.4 6.4 0.0

8 6. | -- 10.3.2.1 0.0% 1 10.3 10.3 10.3 10.3 0.0

9 7. | -- ??? 100.0 1 0.0 0.0 0.0 0.0 0.0

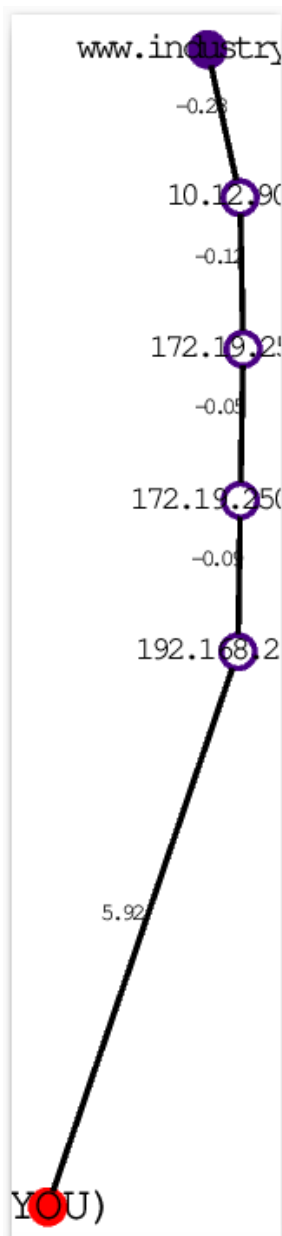
10 8. | -- yandex.ru 0.0% 1 205.7 205.7 205.7 205.7 0.0

6. Построить графическую карту трассировки одновременно ко всем заданным в табл. 1.3 узлам при помощи программы tracemap.

```
stud51@ubuntu18:~/lab1$ (echo www.yandex.ru) | perl tracemap.pl
readline() on closed filehandle PREFIXES at tracemap.pl line 44.
Tracing path to www.yandex.ru.....Done [last 6.746, total 41.477]
stud51@ubuntu18:~/lab1$ mcedit tracemap.dot

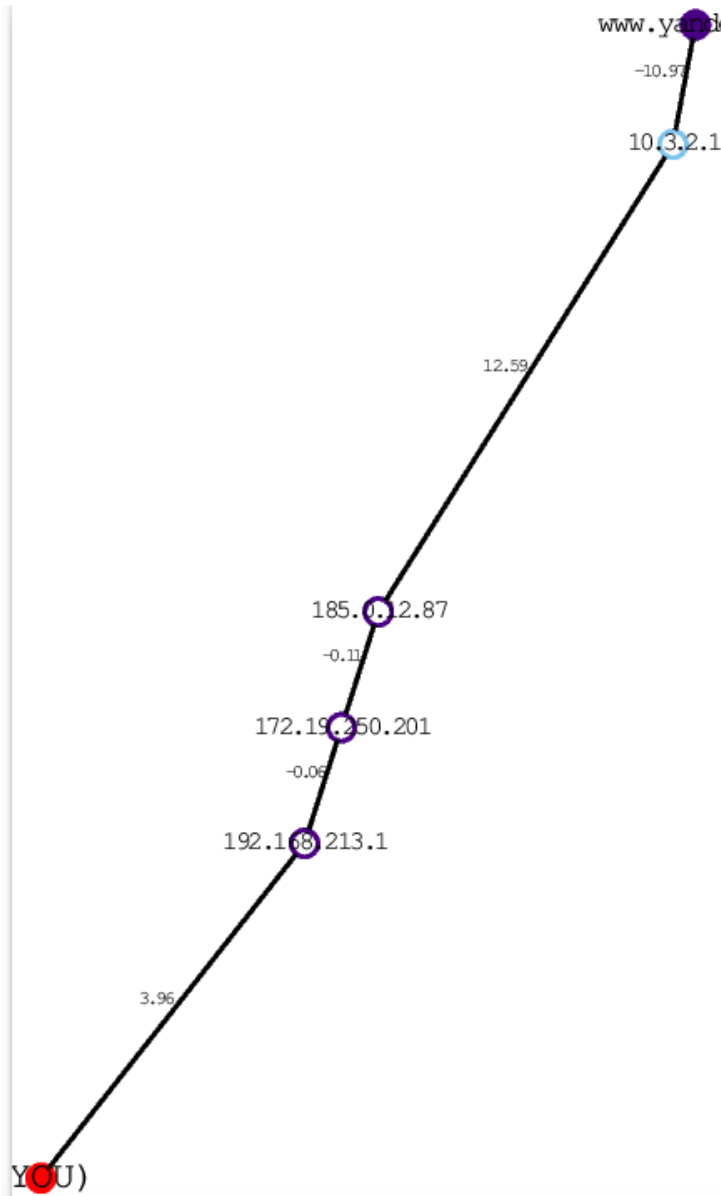
stud51@ubuntu18:~/lab1$ neato -Tps tracemap.dot -o trace_yandex.ps
Warning: node 'hop1', graph 'G' size too small for label
Warning: node 'hop2', graph 'G' size too small for label
Warning: node 'hop3', graph 'G' size too small for label
Warning: node 'hop4', graph 'G' size too small for label
Warning: node 'hop5', graph 'G' size too small for label
Warning: node 'hop6', graph 'G' size too small for label
```

При команде mcedit редактировать строку 5, добавить скобку «]» вместо «,».  
www.industry.su

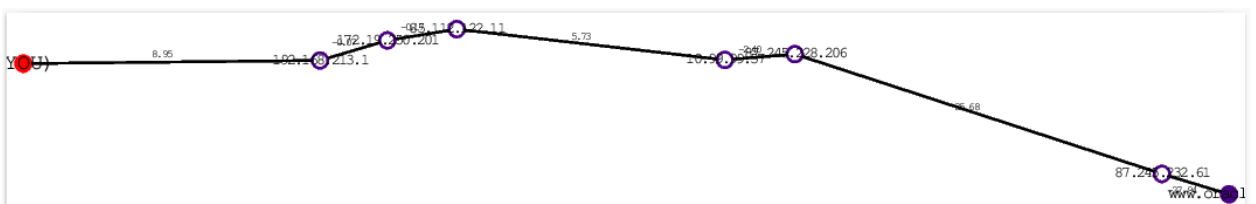




www.yandex.ru



www.oracle.com



### Контрольные вопросы:

#### 1. Утилита ping. Назначение и принцип работы.

ping - утилита командной строки, используемая для проверки доступности хоста в сети и определения времени отклика.

Принцип работы: ping отправляет последовательность ICMP-пакетов Echo Request к указанному хосту. Получив ICMP-пакет Echo Reply, утилита выводит информацию о времени прохождения запроса и ответа. Если хост недоступен, ping выводит сообщение об этом.

Назначение:

- Проверка доступности хоста.
- Определение задержки (ping-time) в пути к хосту.
- Диагностика сетевых проблем.

2. **Утилита traceroute. Назначение и принцип работы.**

traceroute (или tracert на Windows) - утилита для определения маршрута, который проходит пакет данных от отправителя до получателя.

Принцип работы: traceroute отправляет ICMP-пакеты (Echo Request), но с постепенно увеличивающимся TTL (Time To Live). Каждый маршрутизатор, через который проходит пакет, уменьшает TTL на 1. Когда TTL достигает нуля, маршрутизатор отправляет ICMP-пакет Time Exceeded обратно отправителю. На основании информации о хосте, который отправил ICMP-пакет Time Exceeded, и времени его отправки, traceroute строит карту маршрута.

Назначение:

- Определение маршрута пакета данных.
- Выявление “узких мест” в сети (например, медленных маршрутизаторов).

- Диагностика сетевых проблем.

3. **Утилита mtr. Назначение и принцип работы.**

mtr - утилита, сочетающая функциональность ping и traceroute. Она одновременно отправляет ICMP-пакеты ping и отслеживает traceroute маршрут.

Принцип работы: mtr отправляет пакет ping и собирает данные об ответном времени и маршрутизации, используя информацию об отправке и

получении пакетов. Она показывает эти данные в удобном табличном формате.

Назначение:

- Одновременное измерение ping-времени и трассировка маршрута.
- Более детальный анализ сетевых проблем, чем ping или traceroute по отдельности.
- Отслеживание изменений в задержке и маршрутизации в течение времени.

4. **Механизм TTL. Назначение и принцип работы.**

TTL (Time To Live) - поле в заголовке IP-пакета, определяющее максимальное количество маршрутизаторов, через которые может пройти пакет.

Принцип работы: Каждому маршрутизатору, через который проходит пакет, разрешено уменьшать значение TTL на 1. Когда TTL достигает 0, маршрутизатор отправляет ICMP-пакет “Time Exceeded” обратно отправителю пакета. Это предотвращает циклическое прохождение пакетов и позволяет избежать перегрузки сети.

Назначение:

- Предотвращение заикливания пакетов.
- Предотвращение перегрузки сети.
- Определение маршрута пакета.
- Диагностика сетевых проблем.

5. **Протокол ICMP.**

ICMP (Internet Control Message Protocol) - протокол сетевого уровня (IP), используемый для передачи сообщений об ошибках и управляющих сообщений в IP-сетях. Он не используется для передачи данных приложений.

6. **Формат пакета ICMP.**

Формат пакета ICMP включает:

- Тип (Type): Код, указывающий тип сообщения (например, Echo Request, Echo Reply, Time Exceeded).
- Код (Code): Дополнительная информация о типе сообщения (подтип).
- Дополнительные поля: В зависимости от типа сообщения, могут быть дополнительные поля, такие как идентификатор и последовательный номер для пакетов Echo.

## 7. **Виды пакетов ICMP.**

Существует множество типов ICMP-пакетов, но наиболее распространённые:

- Echo Request/Reply: Используются для проверки доступности хоста и измерения времени отклика.
- Time Exceeded: Отправляется маршрутизатором, когда TTL пакета достигает нуля.
- Destination Unreachable: Указывает, что пакет не может быть доставлен по указанному адресу.
- Parameter Problem: Указывает, что в заголовке IP-пакета обнаружена ошибка.
- Redirect Message: Перенаправляет хосту пакет на другой маршрутизатор.
- Source Quench: (менее распространён) Устаревшая функция, которая уже не используется.
- Router Advertisement/Router Solicitation: Используется в IPv6 для настройки параметров маршрутизации.

Полный список можно найти в стандарте RFC 792.

## Вот основные данные, которые выводит команда **mtr**:

Mtr (MyTraceroute) -- это служебная компьютерная программа, предназначенная для определения маршрутов следования данных в сетях ТСР/IP. Она постоянно показывает сведения о маршруте, потерях, минимальной и максимальной задержке. При помощи *mtr* можно узнавать, где в сети происходят потери, задержки и обрывается связь.

При запуске *mtr* начинается исследование сетевого соединения между локальной машиной и хостом, заданным пользователем. После того, как она определит адрес каждого транзитного узла на пути следования пакетов, она отправляет каждому из этих узлов последовательности запросов ICMP ЕСНО, пытаясь определить качество канала связи с каждым из них. По окончании исследования выдается статистика по каждому исследованному узлу.

Open ▾		mtr_output_yandex.txt	
1 Start: 2024-12-17T21:44:04+0300			
2 HOST: ubuntu18	Loss%	Snt	Last Avg Best Wrst StDev
3 1.   -- _gateway	0.0%	1	19.1 19.1 19.1 19.1 0.0
4 2.   -- 192.168.213.1	0.0%	1	5.7 5.7 5.7 5.7 0.0
5 3.   -- rkn.trancom.ru	0.0%	1	9.2 9.2 9.2 9.2 0.0
6 4.   -- ???	100.0	1	0.0 0.0 0.0 0.0 0.0
7 5.   -- yandex.msk.piter-ix.net	0.0%	1	6.4 6.4 6.4 6.4 0.0
8 6.   -- 10.3.2.1	0.0%	1	10.3 10.3 10.3 10.3 0.0
9 7.   -- ???	100.0	1	0.0 0.0 0.0 0.0 0.0
10 8.   -- yandex.ru	0.0%	1	205.7 205.7 205.7 205.7 0.0

- HOST: - Имя хоста или IP-адрес назначения, для которого вы запускаете mtr.
- Loss%: - Процент потерянных пакетов на протяжении всего маршрута.
- Snt: - Общее количество отправленных пакетов.
- Last: - Задержка (latency) в миллисекундах последнего ответа.
- Avg: - Средняя задержка в миллисекундах для всех ответов.
- Best: - Минимальная задержка в миллисекундах для всех ответов.
- Wrst: - Максимальная задержка в миллисекундах для всех ответов.
- StDev: - Стандартное отклонение задержки. Показывает, насколько меняется задержка.

## Вот основные данные, которые выводит команда **ping**:

Утилита ping предназначена для тестирования сетей, управления сетями и измерения производительности. Из-за нагрузок, которые она создает в сети, не всегда разумно использовать ping в рабочее время или в автоматических сценариях.

Она отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT - Round Trip Time) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов, т. е. косвенно определять загруженность на каналах передачи данных и промежуточных устройствах.

```
stud51@ubuntu18:~$ ping -c 5 www.industry.su
PING www.industry.su (31.31.205.163) 56(84) bytes of data.
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=1 ttl=56 time=10.0 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=2 ttl=56 time=8.67 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=3 ttl=56 time=5.78 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=4 ttl=56 time=7.10 ms
64 bytes from ns1.domainparking.int.reg.ru (31.31.205.163): icmp_seq=5 ttl=56 time=12.8 ms

--- www.industry.su ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 5.783/8.880/12.819/2.435 ms
```

### 1. Заголовок (Первая строка):

- PING <hostname> (<ip\_address>) <packet\_size> bytes of data.
- PING: Сообщает, что это вывод команды ping.
- <hostname>: Имя хоста, к которому выполняется ping (например, google.com).
- <ip\_address>: IP-адрес хоста, к которому выполняется ping (например, 172.217.160.142).
- <packet\_size>: Размер ICMP-пакета в байтах (обычно 32 или 56 байт).

### 2. Ответы от хоста (Строки ответов):

Для каждого отправленного ICMP-запроса вы увидите строку ответа, если хост отвечает. Строка ответа имеет следующий формат:

- <packet\_size> bytes from <hostname> (<ip\_address>):  
icmp\_seq=<sequence\_number> ttl=<ttl\_value> time=<time\_in\_milliseconds> ms
- <packet\_size>: Размер полученного ICMP-пакета в байтах.
- from <hostname> (<ip\_address>): Имя хоста и IP-адрес, от которого получен ответ.
- icmp\_seq=<sequence\_number>: Порядковый номер ICMP-пакета.
- ttl=<ttl\_value>: Значение TTL (Time To Live) в пакете ответа.
- time=<time\_in\_milliseconds> ms: Время, которое потребовалось для прохождения пакета туда и обратно (round-trip time) в миллисекундах. Это значение показывает задержку.

### 3. Статистика (Последние строки, после остановки ping):

После завершения команды ping (обычно по Ctrl+C) выводится статистика:

- --- <hostname> ping statistics ---: Заголовок статистики.
- <number> packets transmitted, <number> received, <packet\_loss\_percentage>% packet loss, time <time\_in\_milliseconds>ms
  - <number> packets transmitted: Общее количество отправленных пакетов.
  - <number> received: Общее количество полученных ответов.
  - <packet\_loss\_percentage>% packet loss: Процент потерянных пакетов.
  - time <time\_in\_milliseconds>ms: Общее время, которое потребовалось для выполнения ping.
- rtt min/avg/max/mdev =  
<min\_time>/<avg\_time>/<max\_time>/<standard\_deviation> ms
  - rtt: Round-trip time (время прохождения туда и обратно).
  - min\_time: Минимальная задержка в миллисекундах.
  - avg\_time: Средняя задержка в миллисекундах.
  - max\_time: Максимальная задержка в миллисекундах.
  - standard\_deviation: Стандартное отклонение задержек (показывает, насколько сильно задержки отклоняются от среднего).

### **Вот основные данные, которые выводит команда traceroute:**

Программа traceroute предназначена для определения маршрутов следования данных в сетях TCP/IP. Она основана на использовании протоколов ICMP и UDP, а также механизма TTL (Time to Live).

Traceroute выполняет отправку данных указанному узлу сети, при этом отображая сведения о всех промежуточных маршрутизаторах, через которые прошли данные на пути к целевому узлу. В случае проблем при доставке данных до какого-либо узла программа позволяет определить, на каком именно участке сети возникли неполадки. Программа работает только в направлении от источника пакетов и является весьма грубым инструментом для выявления неполадок в сети. В силу особенностей работы протоколов маршрутизации в сети Интернет, обратные маршруты часто не совпадают с прямыми, причем это справедливо для всех промежуточных узлов.

```

1 | traceroute to www.yandex.ru (77.88.55.88), 30 hops max, 60 byte packets
2 | 1 _gateway (192.168.0.1) 4.203 ms 3.643 ms 3.395 ms
3 | 2 * 192.168.213.1 (192.168.213.1) 10.768 ms 8.163 ms
4 | 3 * rkn.trancom.ru (172.19.250.201) 63.436 ms 63.263 ms
5 | 4 gw1.trancom.ru (172.19.250.5) 63.062 ms 67.930 ms 67.721 ms
6 | 5 yandex.msk.piter-ix.net (185.0.12.87) 70.578 ms 70.490 ms 70.797 ms
7 | 6 sas-32z7-ae2.yndx.net (87.250.239.81) 73.719 ms * *
8 | 7 * * *
9 | 8 * * *
10 | 9 * * *
11 | 10 * * *
12 | 11 * * *
13 | 12 * * *
14 | 13 * * *
15 | 14 * * *
16 | 15 * * *
17 | 16 * * *
18 | 17 * * *
19 | 18 * * *
20 | 19 * * *
21 | 20 * * *
22 | 21 * * *
23 | 22 * * *
24 | 23 * * *
25 | 24 * * *
26 | 25 * * *
27 | 26 * * *
28 | 27 * * *
29 | 28 * * *
30 | 29 * * *
31 | 30 * * *

```

## 1. Заголовок:

- traceroute to <hostname> (<ip\_address>), <max\_hops> hops max, <packet\_size> byte packets (Linux/macOS)
  - traceroute to: Сообщает, что это вывод команды traceroute.
  - <hostname>: Имя хоста назначения (например, google.com).
  - <ip\_address>: IP-адрес хоста назначения (например, 172.217.160.142).
  - <max\_hops>: Максимальное количество хопов, которые traceroute будет отслеживать.
  - <packet\_size>: Размер отправляемых пакетов в байтах.
- Tracing route to <hostname> [<ip\_address>] (Windows)
  - Tracing route to: Сообщает, что это вывод команды tracert.
  - <hostname>: Имя хоста назначения.
  - [<ip\_address>]: IP-адрес хоста назначения.

## 2. Информация о хопх:

Для каждого хопа на пути к целевому хосту выводится одна строка. В зависимости от операционной системы (Linux/macOS или Windows), формат вывода может немного отличаться, но основная информация остается одинаковой:

- Номер хопа (Hop Number):
  - <hop\_number>: Порядковый номер хопа на маршруте (начиная с 1).



- Имя хоста или IP-адрес:
  - <hostname>: Имя хоста промежуточного маршрутизатора (если доступно DNS). Если имя хоста не определено, то будет выведен только IP-адрес.
  - (<ip\_address>): IP-адрес промежуточного маршрутизатора.
- Задержки (Latency):
  - <time1> ms <time2> ms <time3> ms (Linux/macOS): Три значения задержки в миллисекундах для трех попыток отправки пакета на данный хоп.
  - <time>ms (Windows): Одно значение задержки в миллисекундах для каждой попытки отправки пакета на данный хоп. Может быть несколько значений для каждого хопа, в зависимости от настроек.
- Специальные символы:
  - \* (звездочка): Означает, что на данном хопе нет ответа на пакет. Это может быть связано с тем, что маршрутизатор не отправляет ICMP-ответы, или пакет был потерян.
  - ! (восклицательный знак) (Linux/macOS): Может быть использован для указания ошибки (например, !H, !N и др.).