



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**Методические указания к лабораторной работе № 6 по дисциплине
Сети и телекоммуникации
Для студентов 3-го курса кафедры ИУ5**

Разработали:
Старший преподаватель
Антонов А.И.
Старший преподаватель
Лосева С.С.

Москва, 2025 г.

Оглавление

Лабораторная работа 6. Изучение принципов работы утилит для исследования и мониторинга состояния сети	3
1.1. Цель работы	3
1.2. Задачи	3
1.3. Теоретический материал	3
1.3.1. Утилиты <i>ifconfig</i> и <i>ipconfig</i>	3
1.3.2. Утилита <i>ping</i>	4
1.3.3. Утилита <i>tracert</i>	6
1.3.4. Утилита <i>mtr</i>	8
1.3.5. Утилита <i>tracert</i>	9
1.4. Порядок выполнения лабораторной работы	11
Содержание отчета	14
Контрольные вопросы	15

Лабораторная работа 6.

Изучение принципов работы утилит для исследования и мониторинга состояния сети

1.1. Цель работы

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

1.2. Задачи

Проверить доступность заданных узлов в сети Интернет и различными способами проверить маршрут прохождения пакетов до этих узлов.

1.3. Теоретический материал

Существует огромный спектр различных утилит, предназначенных для работы с сетью и сетевыми интерфейсами. Одни из них используются при настройке сетевых интерфейсов компьютера. Другие – для анализа текущего состояния локальных и глобальных компьютерных сетей. Третьи – для мониторинга состояния компьютерных сетей. Сложность этих утилит также разнится. Используются как узкоспециализированные утилиты, так и программы-«комбайны», решающие сразу набор задач. Выбор используемого программного обеспечения зависит от стоящих задач. В данной работе предлагается рассмотреть работу ряда широко используемых узкоспециализированных сетевых утилит.

1.3.1. Утилиты *ifconfig* и *ipconfig*

Утилита *ifconfig* предназначена для просмотра сетевых интерфейсов ПК и для проверки их состояния. Утилита считается устаревшей, но тем не менее до сих пор широко используется. Ей на замену пришла утилита *ip*, входящая в набор программ *iproute2*.

Для настройки сетевого интерфейса утилитой *ifconfig* необходимо иметь права суперпользователя. Просмотр настроек и состояния сетевых интерфейсов возможен от лица обычного пользователя.

Команда IPCONFIG служит для управления сетевыми интерфейсами и отображения всех текущих параметров сети TCP/IP, а также обновления параметров DHCP и DNS в операционных системах Windows. При вызове команды ipconfig без параметров выводится только IP-адрес, маска подсети и основной шлюз для каждого сетевого адаптера. А для проверки соединений в TCP/IP сетях используется команда Ping.

1.3.2. Утилита 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.

Полное отсутствие ICMP-ответов может также означать, что удаленный узел (или какой-либо из промежуточных маршрутизаторов) блокирует ICMP Echo-Reply или игнорирует ICMP Echo-Request.

Название происходит от английского названия звука импульса, издаваемого сонаром при отражении импульса от объекта.

Также есть несколько альтернативных толкований:

- PING - акроним «Packet InterNet Grouper (Grouper)»;
- Ping – часть названия игры пинг-понг. Это толкование подразумевает, что компьютеры обмениваются сигналами аналогично тому, как игроки в пинг-понг отбивают друг другу мяч.

Утилита ping запускается в терминале. Формат запуска команды (для Linux):

```
user@user-pc:~$ ping address
```

Пример

```

user@user-pc:~$ ping 8.8.8.8 -c 4
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=24.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=20.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=21.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=21.0 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 20.513/21.990/24.828/1.682 ms
    
```

Опции/флаг	Описание
-b	Разрешить использование широковещательного адреса в качестве целевого
-c количество	Остановить работу после передачи заданного количества пакетов ECHOREQUEST. Если задано <i>ограничение на время работы</i> (-w), программа будет ждать указанное количество ответных пакетов ECHOREPLY в указанный временной интервал
-i интервал	Интервал в секундах между отправкой пакетов. По умолчанию между отправкой пакетов делается пауза в 1 с, либо, в случае лавинообразного режима, отправка производится без пауз. Задавать значения меньше 0,2 может только суперпользователь
-L	Подавлять циклические петли для широковещательных пакетов. Этот ключ применяется только если в качестве целевого адреса указан широковещательный
-n	Только цифровой вывод. Не расшифровывать имена (символьный вид) адресов
-q	Выводить только начальные и итоговые данные (не выводить информацию об отдельных запросах)
-s размер-пакета	Размер пакетов для пересылки. По умолчанию - 56, что соответствует размеру 64 байта после добавления 8 байтов заголовка ICMP

Опции/флаг	Описание
-t TTL	Время актуальности пакета IP (TTL - Time to Live)
-wограничение на время работы	Время, по истечении которого <i>ping</i> завершит свою работу независимо от количества посланных и принятых пакетов. При указании этого параметра время ожидания для одного пакета игнорируется, и работа может быть завершена ранее указанного срока только в случае получения информации об ошибке (т. е. уведомления о том, что ответных пакетов точно не будет)
-W время ожидания ответа	Время ожидания (в секундах) ответного пакета. Принимается во внимание только если не было принято ни одного ответа. В противном случае программа ожидает получения. Время актуальности пакета IP (TTL - Time to Live) двух ответов

1.3.3. Утилита *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. Поэтому, ICMP ответ от каждого промежуточного узла может идти своим собственным маршрутом, затеряться или прийти с большой задержкой, хотя в реальности с

пакетами, которые адресованы конечному узлу, этого не происходит. Кроме того, на промежуточных маршрутизаторах часто стоит ограничение числа ответов ICMP в единицу времени, что приводит к появлению ложных потерь.

Утилита *traceroute* входит в поставку большинства современных сетевых операционных систем. В системах Microsoft Windows эта программа носит название *tracert*, а в системах GNU/Linux, Cisco IOS и Mac OS – *traceroute*.

Для определения промежуточных маршрутизаторов *traceroute* отправляет серию (обычно три) UDP датаграмм целевому узлу, при этом каждый раз увеличивая на 1 значение поля TTL («время жизни») в заголовке IP-пакета. Это поле указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно сообщение ICMP «время жизни истекло» (тип 11), указывающее на невозможность доставки данных. *Traceroute* фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем *traceroute* повторяет отправку серии пакетов, но уже с TTL, равным 2, что позволяет первому маршрутизатору пропустить их дальше.

Процесс повторяется до тех пор, пока при определенном значении TTL пакет не достигнет целевого узла. При получении ответа от этого узла процесс трассировки считается завершенным.

На конечном хосте IP-пакет с TTL = 1 не отбрасывается и должен быть передан приложению. Достижение пункта назначения определяется следующим образом: отсылаемые *traceroute* пакеты содержат датаграмму UDP с таким номером порта адресата (превышающим 30 000), что он заведомо не используется на адресуемом хосте. В пункте назначения модуль UDP, получая подобные дейтаграммы, возвращает сообщения ICMP «порт недоступен» (тип 3, код 3). Таким образом, чтобы узнать о завершении работы,

программе *traceroute* достаточно обнаружить, что поступило сообщение ICMP об ошибке этого типа.

Формат запуска команды:

```
user@user-pc:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  1.435 ms  1.391 ms  1.370 ms
 2 10.32.0.1 (10.32.0.1)  4.393 ms  4.374 ms  4.354 ms
 3 h153.net32.bmstu.ru (195.19.32.153)  4.334 ms  4.312 ms  4.288 ms
 4 194.190.254.105 (194.190.254.105)  11.970 ms  11.945 ms  16.286 ms
 5 403.et011.m9-5-gw.msk.niks.su (194.85.42.66)  5.886 ms  5.870 ms  5.856 ms
 6 72.14.210.102 (72.14.210.102)  5.841 ms  5.454 ms  5.400 ms
 7 * * *
 8 108.170.250.129 (108.170.250.129)  5.036 ms  108.170.227.88 (108.170.227.88)  4.656 ms  72.14.233.90 (72.14.233.90)  5.004 ms
 9 108.170.250.51 (108.170.250.51)  4.989 ms  108.170.250.99 (108.170.250.99)  5.082 ms  4.838 ms
10 142.251.238.82 (142.251.238.82)  22.548 ms  22.445 ms  142.250.239.64 (142.250.239.64)  18.086 ms
11 142.251.237.146 (142.251.237.146)  22.413 ms  27.380 ms  142.250.235.62 (142.250.235.62)  29.551 ms
12 216.239.49.107 (216.239.49.107)  27.316 ms  142.250.56.221 (142.250.56.221)  31.505 ms  216.239.58.67 (216.239.58.67)  23.565 ms
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 dns.google (8.8.8.8)  22.407 ms  22.348 ms  22.326 ms
```

Можно видеть время достижения каждого из промежуточных узлов для каждого из трех отправляемых пакетов.

Более подробную информацию об использовании утилиты можно получить из официальной [man-страницы](#).

1.3.4. Утилита *mtr*

Mtr (MyTraceroute) -- это служебная компьютерная программа, предназначенная для определения маршрутов следования данных в сетях TCP/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 ECHO, пытаясь определить качество канала связи с каждым из них. По окончании исследования выдается статистика по каждому исследованному узлу.

Формат запуска команды:

```
user@user-pc:~$ mtr 8.8.8.8
```

```

Host
1. _gateway
2. 10.32.0.1
3. h153.net32.bmstu.ru
4. 194.190.254.105
5. 403.et011.m9-5-gw.msk.niks.su
6. 72.14.210.102
7. 108.170.250.129
8. 108.170.250.130
9. 142.250.238.214
10. 142.250.235.62
11. 72.14.236.73
12. (waiting for reply)
13. (waiting for reply)
14. (waiting for reply)
15. (waiting for reply)
16. (waiting for reply)
17. (waiting for reply)
18. (waiting for reply)
19. (waiting for reply)
20. (waiting for reply)
21. dns.google

Packets
Loss% Snt Last Avg Best Wrst StDev
0.0% 4 1.3 2.7 1.3 5.1 1.8
0.0% 4 3.8 4.6 3.0 8.2 2.4
66.7% 4 7.2 7.2 7.2 7.2 0.0
33.3% 4 2.7 6.0 2.7 9.3 4.7
33.3% 4 3.8 29.1 3.8 54.5 35.9
33.3% 4 4.2 8.3 4.2 12.3 5.7
33.3% 4 9.9 7.4 4.9 9.9 3.6
33.3% 4 3.7 4.4 3.7 5.2 1.0
33.3% 4 21.0 22.8 21.0 24.6 2.5
33.3% 4 23.3 26.5 23.3 29.6 4.4
33.3% 4 20.9 21.9 20.9 22.9 1.4

0.0% 3 18.2 20.2 18.2 21.6 1.8

```

Опция/флаг	Описание
-r	Переводит <i>mtr</i> в режим <i>report</i> . В этом режиме <i>mtr</i> обрабатывает количество циклов, заданное флагом -c, затем выводит статистику и завершает работу
-c число	Задаёт количество циклов опроса маршрута. Каждый цикл длится 1 с
-i число	Задаёт время в секундах на один цикл опроса. По умолчанию - 1 с

1.3.5. Утилита *tracemap*

Tracemap – программа, позволяющая выполнить трассировку пути на несколько хостов сразу и представить полученные данные в виде графической карты. Она написана Игорем Шубиным на языке Perl. В лабораторной работе используется измененная версия программы, не требующая прав суперпользователя. (больше информации по ссылке <http://xgu.ru/wiki/tracemap>)

Программу *tracemap* необходимо скачать. Для этого можно воспользоваться ссылкой <http://xgu.ru/downloads/tracemap.pl>

Также необходимо установить *grafviz* и *libnet-ip-perl*

Программа будет сохранена в текущем каталоге.

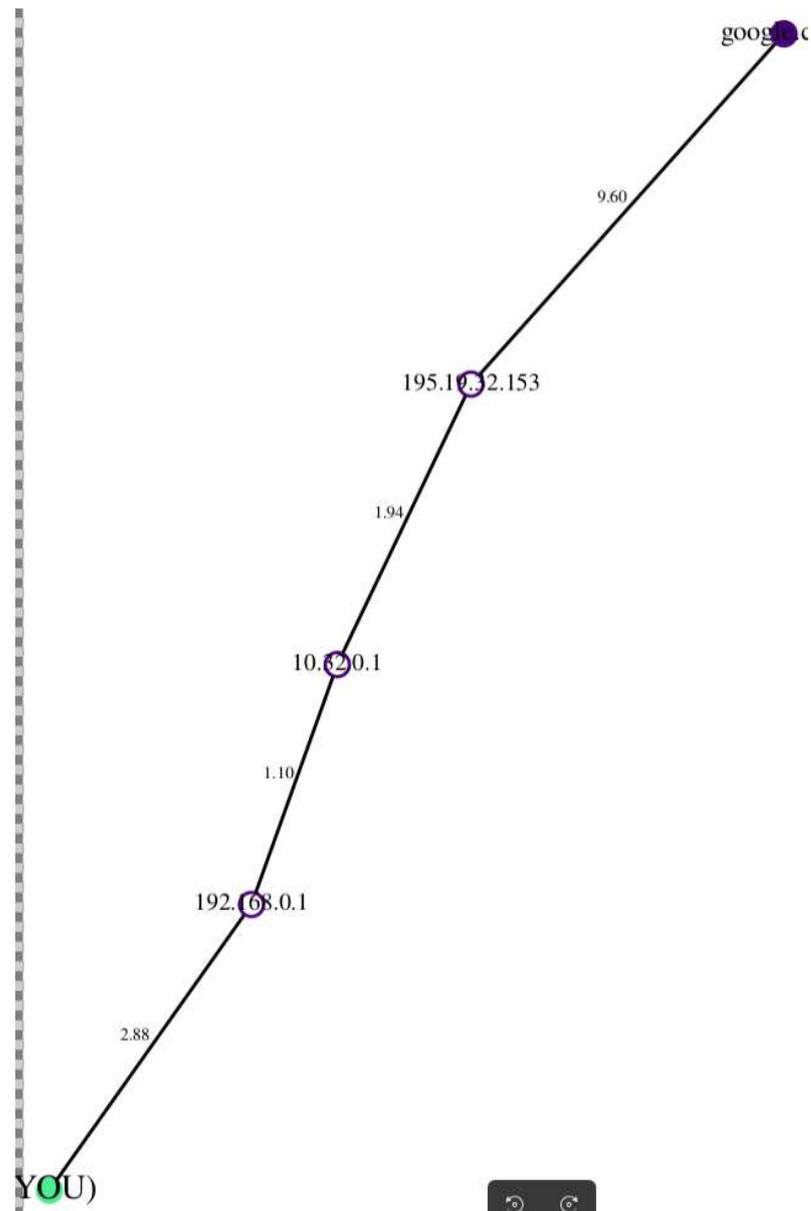
Альтернативным инструментом может быть создание персональной веб-версии утилиты. Описание и реализация есть в проекте на Github <https://github.com/edebu/tracemap>

Формат запуска команды:

```
user@user-pc:~$ echo google.com | perl tracemap.pl  
readline() on closed filehandle PREFIXES at tracemap.pl line 44.  
Tracing path to google.com....Done [last 30.023, total 75.919]
```

В результате выполнения программы в текущем каталоге появятся файлы *tracemap.svg* и *tracemap.png* с графической картой путей.

Пример: запуск утилиты для построения карты маршрутов до хостов



1.4. Порядок выполнения лабораторной работы

1. Просмотреть параметры сетевого интерфейса ПК с которого выполняется ЛР, воспользовавшись утилитой *ipconfig* и/или *ifconfig (ip)*. Выведенную на экран информацию сохранить для отчёта.

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
1	elibrary.ru cyberleninka.ru openknowledgemaps.org
2	russianfood.com eda.ru povar.ru
3	teatrium.ru bolshoi.ru mospuppets.ru
4	radiolibrary.ru cxem.net radioskot.ru
5	liquipedia.net cybersport.ru escharts.com
6	twitch.tv goodgame.ru trovo.live
7	velocityk.ru rollingmoto.ru pitbikemarket.ru
8	planetarium-moscow.ru nebo-nsk.ru astronomy.ru

№ варианта	Исследуемые узлы
16	flreport.ru silkwayrally.com rallycrossarena.ru
17	sunlight.net sokolov.ru miuz.ru
18	sportmaster.ru decsports.ru adidas-ru.com
19	zvetnoe.ru leonardo.ru lemanapro.ru
20	afisha.yandex.ru redkassa.ru kassir.ru
21	anilight.online animeheaven.me yummy-anime.ru
22	jeep.com suzuki-motor.ru landrover-avilon.ru
23	honda.ru ru.yamaha.com harley-davidsons.ru

№ варианта	Исследуемые узлы
9	tattooclassic.ru tattoo.ru empiretattoo.ru
10	asna.ru cloikk.ru 366.ru
11	uralaz.ru kamaz.ru avtogaz.ru
12	podrygka.ru goldapple.ru rivegauche.ru
13	hobbygames.ru edinorog.shop lavkaigr.ru
14	mnogo-cvetov.ru klumba.moscow moscvettorg.com
15	litres.ru ukazka.ru labirint.ru

№ варианта	Исследуемые узлы
24	mvideo.ru dns-shop.ru eldorado.ru
25	vkusnoitochka.ru vkusvill.ru rostics.ru
26	wildberries.ru ozon.ru lamoda.ru
27	rbc.ru lenta.ru gazeta.ru
28	kinopoisk.ru nb.lordfilm17.ru premier.one
29	bmstu.ru mipt.ru imash.ru
30	ejeminutka.ru zaycafe.ru bugcafe.ru

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.

Провести сравнение результатов определения маршрута, полученных с помощью утилиты *traceroute*, *ping* и *mtr*.

Важно: Утилита *mtr* должна быть запущена в режиме *report* с ограничением количества циклов опроса маршрута.

Дополнительное задание

Построить графическую карту трассировки одновременно ко всем заданным в табл. 1.3 узлам при помощи программы *tracemap* или веб-приложения *tracemap*.

Содержание отчета

а) листинг параметров сетевого интерфейса ПК с которого выполняется ЛР;

б) листинг результатов, полученных при работе с утилитой *ping*, плюс таблица с результатами исследований при использовании утилиты *ping*;

в) листинг результатов, полученных при работе с утилитой *tracert*, плюс графики времени прохождения шлюзов (для 3 пакетов) с указанием наиболее узких мест в сети;

г) листинг результатов, полученных при определении маршрута прохождения пакетов утилитой *ping*;

д) листинг результатов, полученных при работе с утилитой *mtr*, и привести сравнение результатов определения маршрута, полученных с помощью, утилита *tracert*, *ping* и *mtr*.

Важно: перед каждым листингом результатов работы должна быть написана соответствующая команда.

е) **Дополнительно.** Карта трассировки, полученная при помощи *tracemap*.

Контрольные вопросы

1. Утилита *ping*. Назначение и принцип работы.
2. Утилита *tracert*. Назначение и принцип работы
3. Утилита *mtr*. Назначение и принцип работы.
4. Механизм TTL. Назначение и принцип работы.
5. Протокол ICMP.
6. Формат пакета ICMP.
7. Виды пакетов ICMP.