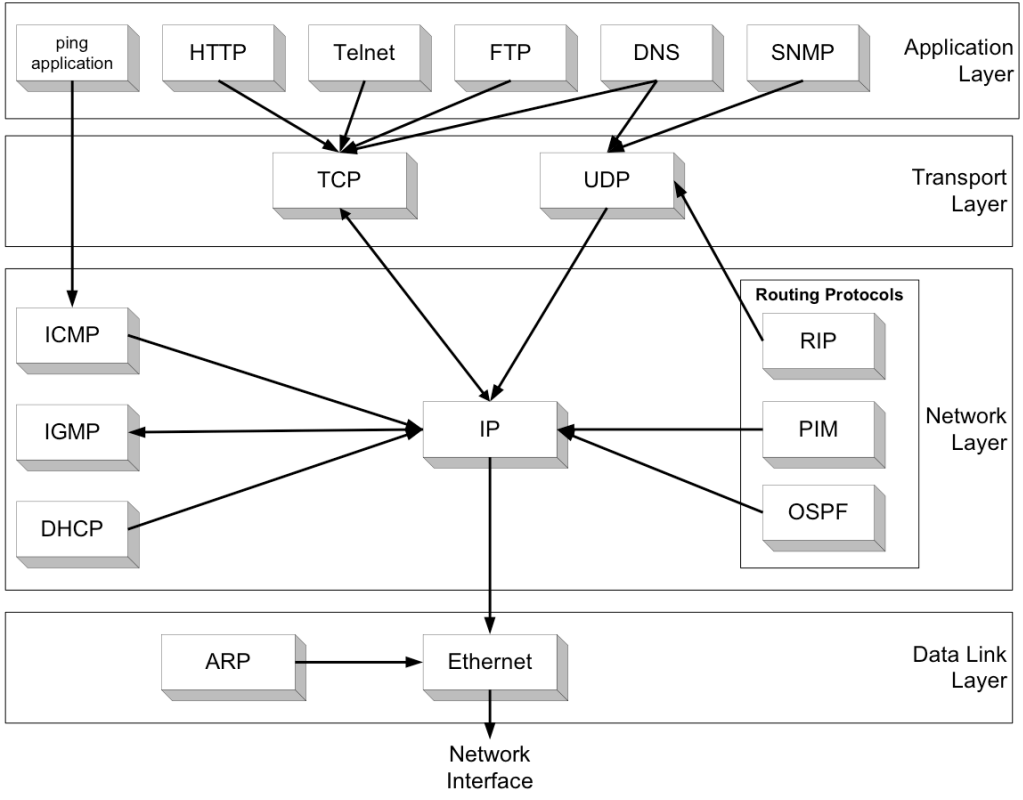
Протокол ICMP.



1. Назначение протокола ICMP.

Протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol) является обязательным стандартом TCP/IP, описанным в документе RFC 792. Используя ICMP, узлы и маршрутизаторы, связывающиеся по протоколу IP, могут сообщать об ошибках и обмениваться ограниченной управляющей информацией и сведениями о состоянии.

ICMP-сообщения обычно автоматически отправляются в следующих случаях:

- IP-дейтаграмма не может попасть к узлу назначения.
- IP-маршрутизатор не может перенаправлять дейтаграммы с текущей скоростью.
- IP-маршрутизатор перенаправляет узел-отправитель на другой, более выгодный маршрут к узлу назначения.

Протокол обмена управляющими сообщениями ICMP позволяет маршрутизатору сообщить первичному отправляющему узлу **об ошибках, с которыми маршрутизатор столкнулся** при передаче какого-либо IP-пакета от этого узла.

Управляющие сообщения ICMP **не могут направляться промежуточному маршрутизатору**, который участвовал в передаче пакета, с которым возникли проблемы, так как для такой посылки нет адресной информации - пакет несет в себе только адрес источника и адрес назначения, не фиксируя адреса промежуточных маршрутизаторов.

Протокол ICMP - это протокол поддерживающий сообщения об ошибках, а **не протокол коррекции ошибок**. Конечный узел может предпринять некоторые действия для того, чтобы ошибка больше не возникала, но эти действия протоколом ICMP не регламентируются.

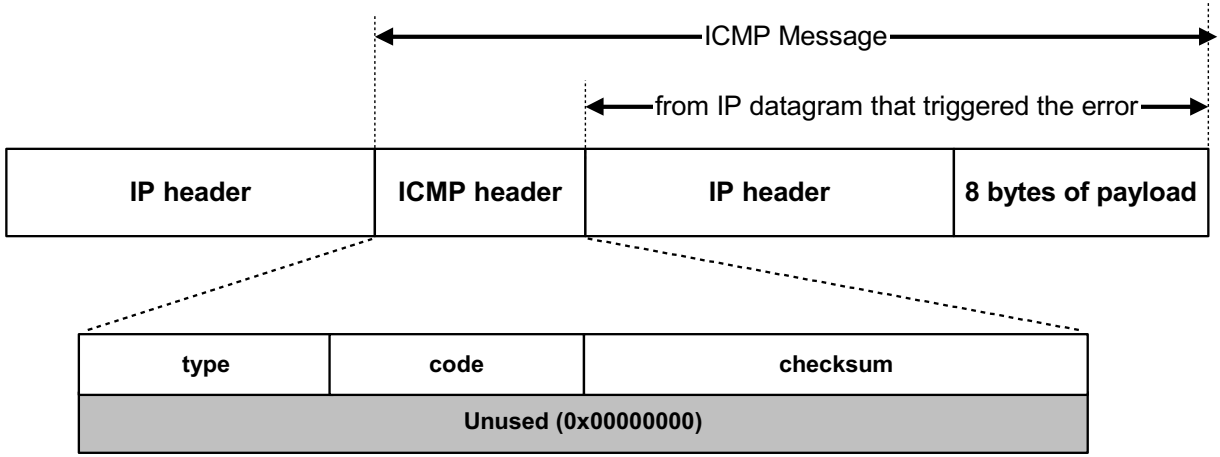
Каждое сообщение протокола ICMP передается по сети внутри пакета IP. Пакеты IP с сообщениями ICMP маршрутизируются точно так же, как и обычные пакеты, т.е. без приоритетов, поэтому они также могут теряться. ICMP сообщения об ошибках могут вызывать дополнительную загрузку маршрутизаторов.

Для того, чтобы не вызывать лавины сообщений, введены специальные ограничения. **Сообщение ICMP об ошибке никогда не генерируется в ответ на**:

1. ICMP сообщение об ошибке. Однако, ICMP сообщение об ошибке может быть сгенерировано в ответ на ICMP запрос.
2. Дейтаграмму, направляющуюся на широковещательный IP адрес или групповой IP адрес (адрес класса D).
3. Дейтаграмму, которая посылается широковещательным запросом на канальном уровне.
4. Фрагмент, который не является первым. (Мы опишем фрагментацию в разделе "Фрагментация IP" главы 11.)
5. Дейтаграмму, адрес источника которой не указывает на конкретный хост. Это означает, что адрес источника не может быть нулевым, loopback адресом, широковещательным или групповым адресом.

2. Форматы ICMP-пакетов.

ICMP-сообщения инкапсулируются и передаются в IP-дейтаграммах, как показано на следующем рисунке.



Особенностью протокола ICMP является функциональное разнообразие решаемых задач, а следовательно, и связанных с этим сообщений. Существует несколько типов сообщений ICMP. Каждый тип сообщения имеет свой формат, но, все они начинаются с общих трех полей – Type, Code, Checksum, затем идут различающиеся поля.

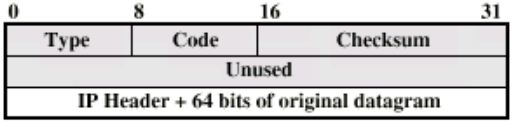
ICMP Header - заголовок ICMP-сообщения состоит из 8 байт:

- **TYPE** – 1-байтное целое число, обозначающего тип сообщения;
- **CODE** – 1-байтное поле кода, который конкретизирует назначение сообщения;
- **CHECKSUM** – 2-байтная контрольная сумма, считается для всего icmp-сообщения аналогично алгоритму IP Checksum;
- **Additional Information** – 4-байтное поле различающиеся для разных типов ICMP;

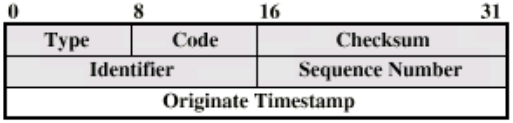
Data – поле данных переменной длины различающееся для разных типов ICMP. Для ICMP-сообщений информирующих об ошибках поле Data содержит:

- **IP Header** – IP-заголовок (от 20 до 60 байт) исходного сообщения вызвавшего ошибку.
- **IP Payload** – первые 8 байт данных IP-пакета, который вызвал ошибку. Это делается для того, чтобы узел-отправитель смог более точно проанализировать причину ошибки, так как все протоколы прикладного уровня стека TCP/IP содержат наиболее важную информацию для анализа в первых 64 битах своих сообщений (начало заголовков TCP и UDP).

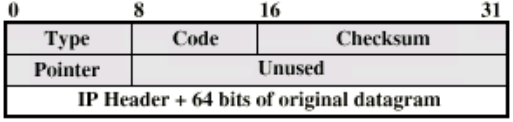
ICMP Message Formats



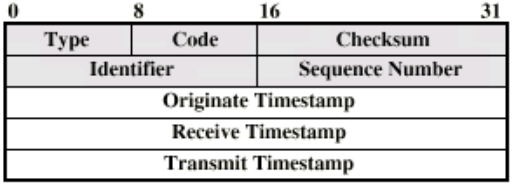
(a) Destination Unreachable; Time Exceeded; Source Quench



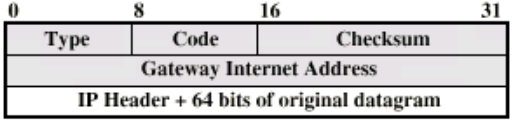
(e) Timestamp



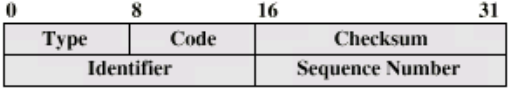
(b) Parameter Problem



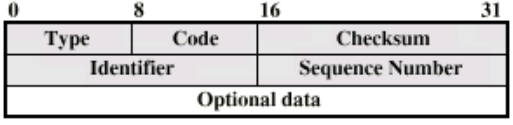
(f) Timestamp Reply



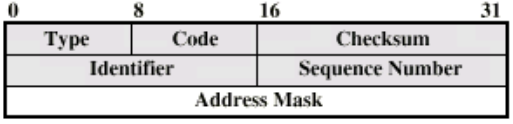
(c) Redirect



(g) Address Mask Request



(d) Echo, Echo Reply



(h) Address Mask Reply

2.1. Формат ICMP запроса и отклика о штампе времени.

ICMP запрос штампа времени позволяет системе запросить другую систему о текущем времени. Рекомендуемое значение, которое должно быть возвращено, это количество миллисекунд, которые прошли с полуночи в формате UTC, (Универсальное согласованное время - Coordinated Universal Time). ICMP предоставляет время с точностью до миллисекунд. Недостаток заключается в том, что сообщается только время, прошедшее с полуночи, - запрашивающая система должна знать текущую дату.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 13 or 14**								**Code = 0**								**Checksum**															
04	**Identifier**																**Sequence Number**															
08	**Sender Timestamp**																															
12	**Receive Timestamp**																															
16	**Transmit Timestamp**																															

Рис. Формат ICMP запроса и отклика о штампе времени (Timestamp).

Запрашивающий заполняет Sender Timestamp и отправляет запрос. Отвечающая система заполняет Receive Timestamp (приём), когда получает запрос, и Transmit Timestamp (передача), когда отправляет отклик.

Существование трёх полей позволяет отправителю более точно вычислить сдвиг времени, см. рисунок.

Рис. Взаимосвязь между значениями timestamp.

Существуют и другие способы получить время и дату, например, протокол сетевого времени, например, NTP - Network Time Protocol. NTP протокол использует определенную технику поддержания точности часов для группы систем в локальной или глобальной сети с точностью до миллисекунды. Для более подробного изучения функционирования протокола NTP, необходимо обратиться к RFC1305 [Mills 1992].

2.2. Формат ICMP запроса и отклика о маске подсети для адреса.

ICMP запрос маски адреса используется бездисковыми системами, чтобы получить маску подсети во время загрузки. Система посылает широковещательный ICMP запрос, это напоминает то, как бездисковые системы с использованием RARP получают свои IP адреса во время загрузки.

Поля идентификатора и номера последовательности в ICMP сообщении могут быть установлены по выбору отправителя, эти же значения будут возвращены в отклике. Именно таким образом отправитель идентифицирует отклик на свой запрос.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 17 or 18**								**Code = 0**								**Checksum**															
04	**Identifier**																**Sequence Number**															
08	**32-bit Subnet Mask**																															

Рис. ICMP запрос и отклик маски адреса.

Альтернативный метод для бездисковых систем получить маски своих подсетей и свой IP адрес - протокол BOOTP. Для более подробного изучения функционирования протокола BOOTP, необходимо обратиться к RFC.

2.3. Формат ICMP ошибки о необходимости фрагментации.

ICMP ошибка о причине недоступности узла генерируется, когда маршрутизатор принимает дейтаграмму, которую необходимо фрагментировать, но, в IP заголовке установлен запрещающий флаг DF (не фрагментировать). На рисунке показан формат такого ICMP. Биты 16-31 во втором 32-битном слове могут содержать MTU следующей пересылки, или быть установленными в 0, если маршрутизатор не поддерживает такой ICMP.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 3**								**Code = 4**								**Checksum**															
04	**Unused=0**																**MTU сети следующей пересылки**														
08-15	**IP Header**																															
16-n	**+ опции IP**																															
n+1-n+8	**+ первые 8 байт данных исходной IP дейтаграммы**																															

Рис. ICMP ошибка о недоступности, когда необходима фрагментация, но установлен бит DF.

Генерация и анализ такой ошибки может быть использованы для определения минимального MTU в маршруте до пункта назначения - что называется механизмом определения транспортного MTU (path MTU discovery).

3. Типы и коды ICMP.

Ниже показана таблица основных типов ICMP-сообщений. Всего можно определить 256 типов. Эти сообщения можно разделить на две группы:

- сообщения об ошибках;
- сообщения запрос-ответ.

Сообщения типа запрос-ответ всегда имеют код 0 и связаны в пары: (эхо-запрос и эхо-ответ), (запрос маски и ответ маски), (запрос времени и ответ времени). Отправитель сообщения-запроса всегда рассчитывает на получение соответствующего сообщения-ответа.

Сообщения, относящиеся к некоторому типу сообщений об ошибках, конкретизируются уточняющим кодом.

Всего можно определить 256 кодов в каждом типе об ошибке, но, использованное количество кодов для конкретного типа сообщения различается, используются не все возможные комбинации кода.

Таблица типов и кодов ICMP.

e – сообщение ошибка
? – сообщение запрос
a – сообщение ответ

Тип	Код	Описание	Запрос	Error
0	0	эхо-отклик (отклик-Ping)	a	
3		назначение недоступно:		
	0	сеть недоступна - network unreachable		e
	1	хост недоступен - host unreachable		e
	2	протокол недоступен - protocol unreachable		e
	3	порт недоступен - port unreachable		e
	4	необходима фрагментация, однако установлен бит “не фрагментировать”- fragmentation needed but don’t-fragment bit set		e
	5	не работает маршрутизация от источника - source route failed		e
	6	неизвестна сеть назначения - destination network unknown		e
	7	неизвестен хост назначения - destination host unknown		e
	8	хост источник изолирован - source host isolated		e
	9	сеть назначения закрыта администратором - destination network administrativrly prohibited		e
	10	хост назначения закрыт администратором - destination host administrativrly prohibited		e
	11	сеть недоступна для TOS - network unreachable for TOS		e
	12	хост недоступен для TOS - host unreachable for TOS		e
	13	связь административно закрыта путем фильтрации - communication administratively prohibited by filtering		e
	14	нарушено старшинство для хоста - host precedence violation		e

	15	старшинство разъединено - precedence cutoff in effect		e
4	0	подавление источника (элементарное управление потоком данных) - source quench		e
5		перенаправление - redirect		
	0	перенаправление в сеть - redirect for network		e
	1	перенаправление в хост - redirect for host		e
	2	перенаправление для типа сервиса и сети - redirect for type-of-service and network		e
	3	перенаправление для типа сервиса и хоста - redirect for type-of-service and host		e
8	0	эхо запрос - echo request (Ping запрос)	?	
9	0	объявление маршрутизатора - router advertisement	a	
10	0	запрос к маршрутизатору - router solicitation	?	
11		время истекло - time exceeded:		
	0	TTL стало равным 0 в процессе передачи - time-to-live equals 0 during transit (traceroute)		e
	1	время жизни стало равным 0 в процессе повторной сборки - time-to-live equals 0 during reassembly (дефрагментации на конечном узле)		e
12		проблемы с параметрами - parameter problem:		
	0	неверный IP заголовок - IP header bad		e
	1	отсутствует необходимая опция - required option missing		e
13	0	запрос временной марки - timestamp request	?	
14	0	отклик с временной маркой - timestamp reply	a	
15	0	информационный запрос - information request	?	
16	0	информационный отклик - information reply	a	
17	0	запрос маски адреса - address mask request	?	
18	0	отклик с маской адреса - address mask reply	a	

Net tools.

arpwatch
tcpdump
syslog
wget
nslookup
traceroute
trat
snort
nmap
whois
ipconfig
rancid
ntop
bro
net-snmp
dig
ping
iperf
wireshark
NDT
dummynet
mrtg

4. Утилита ping.

В качестве примера работы ICMP-сообщений вида (эхо-запросы и эхо-ответы) рассмотрим использование ICMP в популярной утилите ping.

ping — это служебная компьютерная программа, предназначенная для проверки соединений в сетях на основе TCP/IP.

Компьютер или маршрутизатор посылает по составной сети ICMP-сообщение эхо-запроса (Echo-Request), указывая в нем IP-адрес узла, достижимость которого нужно проверить. Узел, получивший эхо-запрос, формирует и отправляет эхо-ответ (Echo-Reply) отправителю запроса.

Так как эхо-запрос и эхо-ответ передаются по сети внутри IP-пакетов, то их успешная доставка означает нормальную работу всей транспортной системы составной сети.

Частота потери пакетов и время между отправкой запроса и получением ответа (RTT - Round Trip Time) позволяют косвенно определять загруженности каналов передачи данных и промежуточных устройств.

Полное отсутствие ICMP-ответов может означать, что удалённый узел (или промежуточный маршрутизатор) блокирует ICMP Echo-Reply или игнорирует ICMP Echo-Request.

Формат эхо-запроса и эхо-ответа показан на рис.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 0 or 8**								**Code = 0**								**Checksum**															
04	**Identifier**																**Sequence Number**															
08-n	**Data (эхо данные запроса-ответа)**																															

Рис. Формат ICMP-сообщений типа эхо-запрос и эхо-ответ

Поля Идентификатора запроса (Identifier), Номера последовательности (Sequence Number) и Данных (Data) в ICMP сообщении могут быть установлены по выбору отправителя, эти же значения будут возвращены сервером в отклике. Именно таким образом отправитель идентифицирует отклик на свой запрос.

Реализации ping, присутствующие в Unix, устанавливают в поле идентификатора ICMP сообщения идентификатор процесса, отправляющего запрос. Это позволяет программе ping идентифицировать вернувшийся ответ, если на одном и том же хосте в одно и то же время запущено несколько программ ping.

Номер последовательности начинается с 0 и увеличивается на единицу каждый раз когда посылается следующий эхо запрос. ping печатает номер последовательности каждого возвращенного пакета, позволяя нам увидеть, потерялся ли пакет, поменялась ли последовательность движения пакетов и был ли пакет продублирован. Так как IP является ненадежным сервисом доставки дейтаграмм, любое из трех вышеперечисленных условий может появиться при работе программы ping.

```
MBA-ys:~ ys$ ping www.ru
PING www.ru (217.112.35.75): 56 data bytes
64 bytes from 217.112.35.75: icmp_seq=0 ttl=55 time=30.994 ms
64 bytes from 217.112.35.75: icmp_seq=1 ttl=55 time=30.792 ms
64 bytes from 217.112.35.75: icmp_seq=2 ttl=55 time=30.676 ms
^C
--- www.ru ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 30.676/30.821/30.994/0.131 ms
```

TTL=55 – оставшееся время жизни пакета (при отправке узел www.ru установил 64).

Программа ping является одним из основных диагностических средств в сетях TCP/IP и входит в поставку всех современных сетевых операционных систем. Функциональность ping также реализована в некоторых встроенных ОС маршрутизаторов. Так как ping использует ICMP и создает raw-пакеты для её выполнения в unix-системах необходимы права суперпользователя. Чтобы обычные пользователи могли использовать ping на /bin/ping ставят SUID бит в правах доступа.

4.1. Опция записи IP маршрута.

В большинстве версий программы ping присутствует опция -R, которая включает опцию записи маршрута (RR – Record Route) для IP в исходящих дейтаграммах (которые содержат ICMP эхо запрос). При этом каждый маршрутизатор, который обрабатывает дейтаграмму, добавляет свой IP адрес в список, находящийся в дополнительном поле IP. Когда дейтаграмма достигает конечного пункта назначения, список IP адресов копируется в исходящий ICMP эхо отклик. Когда ping принимает эхо отклик, то она печатает полученный список IP адресов.

```
MBA-ys:~ ys$ ping -R www.ru
PING www.ru (217.112.35.75): 56 data bytes
64 bytes from 217.112.35.75: icmp_seq=0 ttl=54 time=47.817 ms
RR:     static-91.203.69.72.nano.lv (91.203.69.72)
        159.148.79.61
        latnet-serviss.10gigabitethernet1-2.core1.rix1.he.net (216.66.89.82)
        10ge3-4.core1.rix1.he.net (184.105.64.118)
        10gigabitethernet1-2.core1.sto1.he.net (194.68.123.187)
        mx01.stockholm.gldn.net (194.67.0.215)
        195.239.139.165
        217.112.35.1
        v76-u.valuehost.ru (217.112.35.75)
^C
--- www.ru ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 47.817/47.817/47.817/0.000 ms
```

где,
code - код
len - длина
ptr - указатель

Рис. Общий формат опции маршрута в IP заголовке.

5. Утилита traceroute.

В качестве примера работы ICMP-сообщений об ошибках рассмотрим использование ICMP в популярной UNIX утилите мониторинга сети traceroute, которая отличается от tracert для Windows.

Когда маршрутизатор не может передать или доставить IP-пакет, он отсылает узлу, отправившему этот пакет, сообщение о недостижимости узла назначения. Формат этого сообщения показан на рис. В поле типа помещается значение 3, а в поле кода — значение из диапазона 0-15, уточняющее причину, по которой пакет не был доставлен. Следующие за полем контрольной суммы четыре байта заголовка не используются и заполняются , нулями.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 3**								**Code = 0-15**								**Checksum**															
04	**Unused=0**																**Unused=0**															
08-15	**IP Header**																															
16-n	**+ опции IP**																															
n+1-n+8	**+ первые 8 байт данных исходного IP пакета вызвавшего ошибку**																															

Рис. Формат ICMP-сообщения об ошибке недостижимости узла назначения.

Помимо причины ошибки, указанной в заголовке (в полях типа и кода), дополнительная диагностическая информация передается в поле данных ICMP-сообщения. Именно туда помещается заголовок IP и первые 8 байт данных того IP-пакета, который вызвал ошибку.

В основе работы утилиты traceroute для Unix лежат ICMP-сообщения об ошибках «время истекло при передаче» и «порт недоступен», а tracert в Windows работает как ping (Echo-Reply).

traceroute — это служебная компьютерная программа, предназначенная для исследования маршрутов следования данных в сетях TCP/IP.

Программа 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 traceroute позволяет определить, на каком именно участке сети возникли неполадки.

В предыдущем разделе мы описали IP опцию записи маршрута. Зачем писать новое приложение, когда данная опция уже реализована в IP и ping? Существует две причины:

1. исторически не все маршрутизаторы поддерживают опцию записи маршрута, а traceroute не требует каких-либо специальных характеристик на промежуточных маршрутизаторах, кроме поддержки ICMP;
2. размер, предоставляемый для опций в IP заголовке, недостаточен для обработки большинства маршрутов, т.к. в поле опций IP заголовка входит всего 9 IP адресов.

5.1. Принцип работы traceroute.

Traceroute использует поле TTL в IP заголовке и несуществующий порт UDP при отправке, и ICMP при приёме. Поле TTL (время жизни) это 8-битное поле, которое отправитель устанавливает в какое-либо значение. Рекомендуемое исходное значение указано в Assigned Numbers RFC и в настоящее время равно 64. Более старые системы устанавливают это значение в 15 или 32. Мы видели в некоторых примерах работы программы Ping, что ICMP эхо отклики часто отправляются с TTL, установленным в максимальное значение - 255.

Для определения промежуточных маршрутизаторов traceroute отправляет серию IP пакетов целевому узлу, при этом каждый раз увеличивая на 1 значение поля TTL («время жизни»). Это поле указывает время, но обычно соответствует максимальному количеству маршрутизаторов, которое может быть пройдено пакетом.

Первый пакет отправляется с TTL, равным 1, и поэтому первый же маршрутизатор возвращает обратно сообщение ICMP тип=11 код=0 "время истекло при передаче" (**time exceeded in transit**), traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся traceroute на монитор компьютера).

Затем traceroute повторяет отправку пакета, но уже с TTL, равным 2, что позволяет первому маршрутизатору пропустить пакет дальше. Процесс повторяется до тех пор, пока не достигнет целевого узла.

Но, когда дейтаграмма с оставшимся TTL=1 прибывает на хост назначения, он не уничтожит ее и не сгенерирует ICMP сообщение об истечении времени.

Как traceroute определяет, что дейтаграмма достигла конечного пункта назначения?

Для этого traceroute посылает не просто IP пакеты, а инкапсулированные в них UDP или TCP дейтаграммы с несуществующим в целевой системе номером UDP или TCP порта (больше-равно чем 33434), что делает невозможным обработку этой дейтаграммы каким-либо приложением. При получении такой дейтаграммы, UDP модуль хоста назначения сгенерирует ICMP сообщение тип=3 код=3 "порт недоступен" (port unreachable).

Может получится так, что порт назначения будет открыт, но, то так как в процессе трассировки номер порта назначения будет инкрементироваться при каждой попытке (33434, 33435 и т д), то мы всё равно поймаем ответ от конечного узла о закрытом UDP порте, а для TCP порта сервер отправит на хост-инициатор например TCP ACK если для трассировки используются TCP SYN пакеты, что тоже будет являться триггером к окончанию трассировки.

Именно по типу принимаемых ICMP сообщений (либо об истечении времени, либо о недоступности порта, либо другие icmp) и по tcp-syn, traceroute узнаёт, доставлена ли дейтаграмма в пункт назначения.

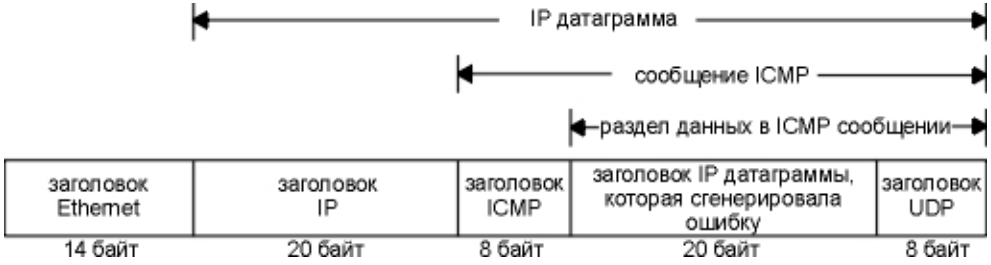


Рис. ICMP сообщение "порт UDP недоступен", вернувшееся в traceroute.

Далее приведен экран работы traceroute при трассировке хоста www.ru на MAC OS X.

```
MBA-ys:~ ys$ traceroute www.ru
traceroute to www.ru (217.112.35.75), 64 hops max, 52 byte packets
 1  192.168.111.1 (192.168.111.1)                                  1.820 ms  0.876 ms  0.857 ms
 2  85.254.142.1 (85.254.142.1)                                    1.355 ms  1.401 ms  1.300 ms
 3  static-91.203.69.73.nano.lv (91.203.69.73)                     1.525 ms  1.745 ms  1.762 ms
 4  159.148.79.62 (159.148.79.62)                                  1.501 ms  1.360 ms  1.489 ms
 5  10ge1-2.core1.rix1.he.net (216.66.89.81)                       1.668 ms  1.610 ms  1.401 ms
 6  10ge3-11.core1.sto1.he.net (184.105.64.117)                    15.174 ms  15.530 ms  21.709 ms
 7  netnod-ix-ge-a-sth-1500.goldentelecom.com (194.68.123.151)     30.567 ms  29.584 ms  31.759 ms
 8  * * *
 9  195.239.139.166 (195.239.139.166)                              32.059 ms  30.883 ms  46.140 ms
10  v76-u.valuehost.ru (217.112.35.75)                             31.073 ms !X  31.147 ms !X  30.127 ms !X
```

Первая строка, без номера содержит имя и IP адрес пункта назначения и указывает на то,

что величина TTL не может быть больше 64. Размер дейтаграммы установлен в 52 байт, из которых 20 байт отводится на IP заголовок, 8 байт на UDP заголовок и 24 байт на пользовательские данные. (В 24 байтах пользовательских данных могут содержаться номер последовательности, который увеличивается на единицу при отправке каждой следующей дейтаграммы, копия исходящего TTL и время, когда дейтаграмма была отправлена.)

Последовательность строк соответствует последовательности маршрутизаторов, образующих маршрут к заданному узлу (число хопов), далее отражается IP-адрес и доменное имя (если оно имеется) маршрутизатора, а затем время прохождения до соответствующего маршрутизатора. Утилита traceroute тестирует каждый маршрутизатор трижды, поэтому следующие три числа в строке — это значения RTT, вычисленные путем посылки трех пакетов, время жизни которых истекло на этом маршрутизаторе. Если ответ от какого-либо маршрутизатора не приходит за заданное время, то вместо времени на экране печатается звездочка (*).

Видно, что большинство интерфейсов маршрутизаторов зарегистрированы в службе DNS, а первые два, относящиеся к локальным маршрутизаторам, — нет.

В последней строке есть отметка о фильтрации пакета !X. Почему это произошло? Узел 217.112.35.75 получив пакет к www.ru дропает (уничтожает) его, так как он попадает под запрещающее правило на входящем интерфейсе и отправляет хосту-инициатору сообщение о том, что пакет был зафильтрован (ICMP Type 3 «Destination Unreachable» Code 13 — «Communication Administratively Prohibited»). Это тоже сигнал о недостижимости порта. Поэтому утилита traceroute получив такое сообщение, заканчивает свою работу так и не

добравшись до хоста назначения. В данном случае в выводе важно понять, что пакеты были именно зафильрованы, о чем нам подсказывает знак !X (в Unix) или знак !A (в Cisco).

Приведём ещё несколько примеров демонстрирующих использование для трассировки различных протоколов и портов для traceroute на Linux и прохождение пакетов по разным маршрутам (балансировка).

Пример traceroute с использованием UDP пакетов:

```
ys@ns:~$ traceroute -U -p 22 -w 1 -n www.lv
traceroute to www.lv (92.240.66.50), 30 hops max, 60 byte packets
 1  192.168.111.1  0.356 ms  0.254 ms  0.227 ms
 2  85.254.142.1  1.106 ms  0.984 ms  0.912 ms
 3  91.203.69.73  1.305 ms  1.333 ms  1.313 ms
 4  91.90.247.24  0.715 ms  0.677 ms  0.722 ms
 5  195.246.227.159  1.214 ms  1.818 ms  1.825 ms
 6  92.240.66.25  0.890 ms  0.945 ms  0.881 ms
```

Пример traceroute с использованием TCP SYN пакетов:

```
ys@ns:~$ sudo traceroute -T -p 80 -w 1 -n www.lv
traceroute to www.lv (92.240.66.50), 30 hops max, 60 byte packets
 1  192.168.111.1  0.364 ms  0.266 ms  0.249 ms
 2  85.254.142.1  0.867 ms  0.971 ms  0.811 ms
 3  91.203.69.73  1.164 ms  1.130 ms  1.094 ms
 4  91.90.247.24  0.627 ms  0.931 ms  0.906 ms
```

```
 5  195.246.227.159  0.978 ms  1.290 ms  1.153 ms
 6  92.240.66.50  1.070 ms  1.046 ms  0.993 ms
```

Пример traceroute для гуглового DNS с балансировкой нагрузки на 5-11 шагах.

```
MBA-ys:~ ys$ traceroute -n 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 52 byte packets
 1  192.168.111.1  1.831 ms  0.824 ms  1.033 ms
 2  85.254.142.1  1.581 ms  1.358 ms  1.893 ms
 3  91.203.69.73  1.824 ms  1.773 ms  1.704 ms
 4  91.90.247.24  29.791 ms  18.918 ms  1.338 ms
 5  91.90.239.20  1.404 ms  1.699 ms  1.490 ms
 6  62.115.57.125  2.025 ms  1.916 ms  1.733 ms
 7  62.115.139.154  11.678 ms
    80.91.246.190  11.341 ms
    62.115.139.154  11.593 ms
 8  80.91.249.219  11.525 ms
    213.155.133.19  13.889 ms
    62.115.114.165  11.892 ms
 9  62.115.61.30  15.393 ms  13.195 ms  15.672 ms
10  74.125.37.237  11.828 ms
    216.239.54.213  15.501 ms  15.621 ms
11  209.85.245.61  14.308 ms
    72.14.234.89  14.082 ms
    209.85.245.63  14.210 ms
12  8.8.8.8  31.388 ms  28.417 ms  30.272 ms
```

Пример traceroute с использованием ICMP Echo-Request.

Для этого ее следует запустить с ключом -I:

```
ys@ns:~$ sudo traceroute -I -w 1 -n academy.lv
traceroute to academy.lv (85.254.142.227), 30 hops max, 60 byte packets
 1  85.254.142.227  0.333 ms  0.293 ms  0.280 ms
```

5.2. Особенности traceroute.

- RTT **-** это не время прохождения пакетов между двумя соседними маршрутизаторами, а время, за которое пакет проделывает путь от источника до соответствующего маршрутизатора и обратно. Так как ситуация в Интернете с загрузкой маршрутизаторов постоянно меняется, то время достижимости маршрутизаторов не всегда нарастает монотонно, а может изменяться достаточно произвольным образом.
- Не существует гарантии, что маршрут, который используется сегодня, будет использоваться и завтра. Если маршрут изменится в процессе работы программы, Вы увидите это, потому что traceroute напечатает новые IP адреса для определенных TTL рядом с каждым из трёх времён RTT.
- Не существует гарантии того, что путь, по которому вернется ICMP сообщение, совпадет с путем, по которому traceroute отправила UDP дейтаграмму. Это означает, что время RTT, которое печатает программа, может не равняться двойному времени, потребовавшемуся на передачу исходящей дейтаграммы. (Возможен вариант, что UDP дейтаграмма дойдет от источника до маршрутизатора за 1 секунду, однако ICMP

сообщение проделает обратный путь за 3 секунды, при этом время RTT будет напечатано как 4 секунды.)

- IP адрес, который возвращается в сообщение ICMP, это чаще всего IP адрес интерфейса, на который маршрутизатор принял IP/UDP дейтаграмму. Тогда как при использовании опции записи маршрута записывается IP адрес исходящего интерфейса.

  Так как каждый маршрутизатор имеет 2 или более интерфейсов, запуск traceroute от хоста A к хосту B будет отличаться от того, который запущен с хоста B на хост A.

  На рисунке показаны две локальные сети, соединенные через маршрутизаторы. Если мы запустим traceroute с хоста сети 1 на хост в сети 2, IP адреса для маршрутизатора будут if1 и if3. При использовании другого пути, будут получены адреса if4 и if2.

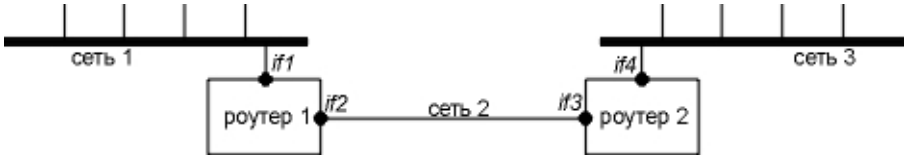


Рис. Идентификация интерфейсов программой traceroute.

- Так как traceroute создаёт raw-пакеты для её выполнения в unix-системах могут быть необходимы права суперпользователя. Чтобы обычные пользователи могли использовать traceroute на /usr/sbin/traceroute ставят SUID бит в правах доступа.

6. ICMP ошибки о перенаправлении маршрута.

ICMP ошибка перенаправления отправляется маршрутизатором на хост, пославший IP дейтаграмму, когда дейтаграмма должна быть послана на другой маршрутизатор. Подобная концепция довольно проста, три составные части этой концепции на рисунке. Возникает ICMP перенаправление только когда хост имеет выбор, на какой маршрутизатор послать пакет.

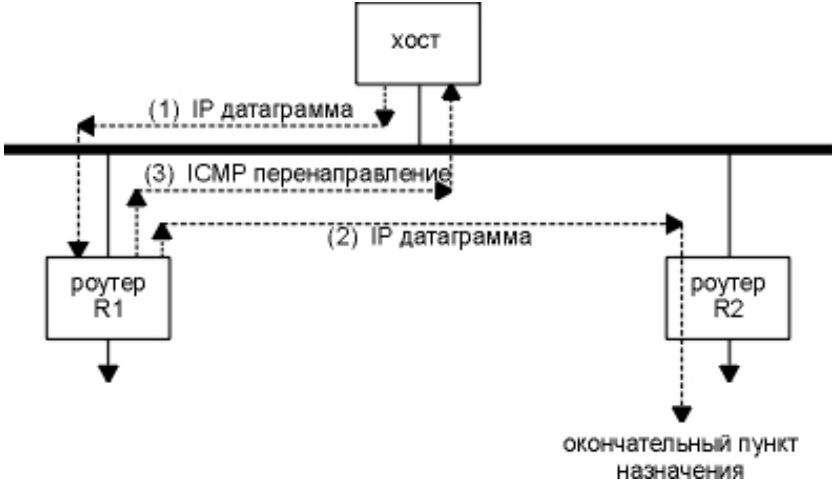


Рис. Пример ICMP перенаправления.

1. Предположим, что хост посылает IP дейтаграмму на R1. Подобное решение принято потому, что R1 - это маршрутизатор по умолчанию для этого хоста.
2. R1 принимает дейтаграмму, просматривает свою таблицу маршрутизации, и определяет, что маршрутизатором следующей пересылки является R2, и именно туда необходимо отправить дейтаграмму. Когда R1 отправляет дейтаграмму на R2, он определяет, что отправляет ее на тот же самый интерфейс, с которого дейтаграмма была получена (локальная сеть, к которой подключен хост и два маршрутизатора). В этом случае маршрутизатор отправляет ошибку перенаправления на хост, пославший дейтаграмму.
3. R1 посылает ICMP перенаправление на хост, сообщая тем самым, что следующие дейтаграммы необходимо посылать на R2 вместо R1.
4. Хост добавляет запись с маршрутом доступа к хосту (не к сети).

Перенаправления используются для того, чтобы позволить хосту с минимальным знанием о маршрутах поддерживать и обновлять свою таблицу маршрутизации. Как правило, формирование таблицы маршрутизации хоста начинается с создания маршрута по умолчанию (R1 из примера на рисунке), при этом с использованием перенаправления хост может обновить свою таблицу маршрутизации. ICMP перенаправление позволяет TCP/IP хостам полностью полагаться на интеллектуальность маршрутизаторов в вопросе выбора маршрутов.

Маршрутизаторы R1 и R2 в нашем примере должны точно представлять топологию подключенных сетей, тогда как хосты, подключенные к локальной сети, могут начинать свою маршрутизацию с маршрута по умолчанию (R1), узнавая затем более подробно о новых маршрутах из принятых перенаправлений.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 5**								**Code = 0-3**								**Checksum**															
04	**IP адрес маршрутизатора, который должен быть использован**																															
08-15	**IP Header**																															
16-n	**+ опции IP**																															
n+1-n+8	**+ первые 8 байт данных исходной IP дейтаграммы**																															

Рис. ICMP сообщение о перенаправлении.

Существуют четыре сообщения о перенаправлении, с различными значениями кода (code):

- 0 - перенаправление для сети
- 1 - перенаправление для хоста
- 2 - перенаправление для типа сервиса (TOS) и сети
- 3 - перенаправление для типа сервиса (TOS) и хоста

Правила работы ICMP перенаправления. Перенаправления генерируются только маршрутизаторами, ни в коем случае не хостами. Однако, перенаправления могут быть использованы только хостами, не маршрутизаторами. Считается, что маршрутизаторы используют протоколы маршрутизации и не нуждаются в перенаправлениях.

7. ICMP сообщения поиска и объявления маршрутизатора (ICMP Router Discovery and Router Advertisement Messages).

Одним из способов инициализации таблицы маршрутизации является способ, заключающийся в использовании ICMP объявлений маршрутизаторов.

Основной принцип заключается в том, что после загрузки хост рассылает широковещательные или групповые запросы с требованием сообщить ему о маршрутизаторе. Один или несколько маршрутизаторов отвечают с использованием сообщения об объявлении маршрутизатора. В дополнение, маршрутизаторы периодически рассылают широковещательные или групповые сообщения с объявлением маршрутизатора, позволяя каждому хосту, который примет эти сообщения, обновить свои таблицы маршрутизации.

RFC 1256 и RFC 4861 содержит формат этих ICMP сообщений. На рисунках далее показаны формат ICMP сообщения запроса маршрутизатора и формат ICMP сообщения объявления маршрутизатора, которое рассылается маршрутизаторами.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 10**								**Code = 0**								**Checksum**															
04	**Не используется, должен быть установлено в 0**																															

Рис. ICMP сообщение запроса маршрутизатора.

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 9**								**Code = 0**								**Checksum**															
04	**Количество адресов**								**Размер записи адреса =2**								**Время жизни**															
08	**IP адрес маршрутизатора 1**																															
16	**Уровень предпочтительности 1**																															
20	**IP адрес маршрутизатора 2**																															
24	**Уровень предпочтительности 2**																															
28-	**…**																															

Рис. ICMP сообщение объявления маршрутизатора.

- **Количества адресов** - в одном сообщении маршрутизатор может объявить несколько.
- **Размер записи адреса** - количество 32-битных слов для каждого адреса маршрутизатора и уровня предпочтения, оно всегда установлено в 2.
- **Время жизни** - количество секунд, в течение которого данное объявление адресов считается действительным.
- **IP адрес** – адрес одного или нескольких маршрутизаторов.
- **Уровень предпочтительности** - указывает на предпочтительность этого адреса в качестве адреса маршрутизатора по умолчанию, по сравнению с другими адресами маршрутизаторов в той же подсети. Большее значение указывает на большую предпочтительность.

7.1. Функционирование маршрутизатора.

Когда маршрутизатор стартует, он начинает **периодически рассылать объявления** на все интерфейсы, которые поддерживают групповой и широковещательный тип адресации.

В действительности эти объявления не периодические, они рассылаются случайным образом. Это сделано для того, чтобы объявления не перемешивались с другими маршрутизаторами в той же подсети. Обычный интервал между объявлениями составляет от 450 до 600 секунд (10 минут).

Время жизни по умолчанию для каждого объявления составляет 30 минут.

Поле времени жизни также используется, **когда интерфейс маршрутизатора выключается**. В этом случае маршрутизатор может передать последнее объявление с временем жизни, установленным в 0.

Помимо периодических объявлений, маршрутизатор отвечает на **запросы от хостов**. Он отвечает на запросы объявлением маршрутизатора.

Если в одной подсети существует несколько маршрутизаторов, задача системного администратора **сконфигурировать уровень предпочтительности** для каждого маршрутизатора. Например, основной маршрутизатор по умолчанию должен иметь более высокий уровень предпочтительности по отношению к запасному маршрутизатору.

7.2. Функционирование хоста.

При старте хост обычно посылает три запроса о поиске маршрутизатора с интервалом в 3 секунды. После того как принято объявление от маршрутизатора, запросы прекращаются.

Хост также слушает объявления от маршрутизатора. Эти объявления могут привести к смене маршрутизатора по умолчанию для данного хоста.

Если **подтверждающее объявление** не получено для текущего маршрута по умолчанию, он может быть удален по тайм-ауту.

Пока текущий маршрутизатор по умолчанию функционирует, он отправляет объявления каждые 10 минут с временем жизни в 30 минут. Это означает, что маршрут по умолчанию в таблице маршрутизации хоста не будет удален по тайм-ауту, даже если одно или два **объявления будут потеряны**.

8. ICMPv6.

ICMPv6 (Internet Control Message Protocol for the Internet Protocol Version 6 — межсетевой протокол управляющих сообщений для межсетевого протокола версии 6) — реализация ICMP для IPv6. ICMPv6 — неотъемлемая часть IPv6, отвечающая за сообщения об ошибках, диагностические функции (например, ping), поиск соседей, определение MTU и основа для расширения и реализации будущих аспектов управления межсетевым протоколом. ICMPv6 определен в RFC 4443.

ICMPv6-сообщения могут быть разделены на две категории: сообщения об ошибках (Type = 0-127) и информационные сообщения (Type = 128-255).

ICMPv6-сообщения инкапсулированы в пакеты IPv6, с полем Next Header = 58.

8.1. Формат ICMPv6 пакета.

ICMPv6 состоит из заголовка и полезных данных протокола. Заголовок содержит только три поля: тип (8 бит), код (8 бит) и контрольная сумма (16 бит).

Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	**Type = 0-255**								**Code = 0-127**								**Checksum**															
04-n	**ICMP Data**																															

8.2. Типы ICMPv6 сообщений.

Тип	Описание	RFC
1	Destination Unreachable	RFC 4443
2	Packet Too Big	RFC 4443
3	Time Exceeded	RFC 4443
4	Parameter Problem	RFC 4443
100	Private experimentation	
101	Private experimentation	
127	Reserved for expansion of ICMPv6 error messages	
128	Echo Request	RFC 4443
129	Echo Reply	RFC 4443
130	Multicast Listener Query	RFC 2710 и RFC 3810
131	Version 1 Multicast Listener Report	RFC 2710
132	Multicast Listener Done	RFC 2710
133	Router Solicitation	RFC 4861
134	Router Advertisement	RFC 4861
135	Neighbor Solicitation	RFC 4861
136	Neighbor Advertisement	RFC 4861
137	Redirect	RFC 4861

138	Router Renumbering	
139	ICMP Node Information Query	
140	ICMP Node Information Response	
141	Inverse Neighbor Discovery Solicitation Message	RFC 3122
142	Inverse Neighbor Discovery Advertisement Message	RFC 3122
143	Version 2 Multicast Listener Report	RFC 3810
144	Home Agent Address Discovery Request Message	RFC 3775
145	Home Agent Address Discovery Reply Message	RFC 3775
146	Mobile Prefix Solicitation	RFC 3775
147	Mobile Prefix Advertisement	RFC 3775
148	Certification Path Solicitation Message	RFC 3971
149	Certification Path Advertisement Message	RFC 3971
150	ICMP messages utilized by experimental mobility protocols such as Seamoby	RFC 4065
151	Multicast Router Advertisement	RFC 4286
152	Multicast Router Solicitation	RFC 4286
153	Multicast Router Termination	RFC 4286
200	Private experimentation	
201	Private experimentation	
255	Reserved for expansion of ICMPv6 informational messages	

9. Упражнения.

1. В конце раздела 1 мы привели 5 специальных условий, при которых сообщение об ошибке ICMP не посылается. Что произойдет, если эти 5 условий не соблюдены, а мы послали широковещательную UDP дейтаграмму на несоответствующий порт в локальном кабеле?
2. Если Ваша система имеет команду netstat, используйте ее, чтобы посмотреть, какой тип ICMP сообщений принимается и посылается системой.
3. Что произойдет, если IP модуль уменьшит входящий TTL на единицу, а затем проверит на равенство нулю?
4. Сравните имена узлов для команд ping –R www.ru и traceroute www.ru.
5. Сравните пути прохождения ping -R и traceroute с разных рабочих станций на один и тот же хост.