АДМИН ДЛЯ НАЧИНАЮЩИХ

Маленькие секреты сетевых утилит. Интерпретируем вывод ping, traceroute и whois для отладки

Содержание статьи

Команды ping, traceroute и whois — в числе первых вещей, о которых узнают начинающие админы. Многие, кто не специализируется на сетях, ими и ограничиваются, и совершенно зря. С помощью стандартных инструментов можно извлечь гораздо больше информации о проблеме, чем может показаться.

---

ping

Команда `ping example.com` известна каждому, даже далекому от сетей человеку. Она отправляет удаленному хосту пакеты ICMP echo, на которые, по идее, он должен ответить таким же пакетом.

Однако этот протокол не просто так называется Internet Message Control Protocol. Его функции далеко не только диагностические, а диагностические функции куда шире, чем «ответил — не ответил».

Что может сказать ping?

Зачастую, если хост назначения недостижим, от ping действительно можно получить только `request timeout` и ничего больше. Если успешный ответ всегда исходит от самого хоста назначения, то сообщения об ошибках доставки — от промежуточных маршрутизаторов. По стандарту промежуточные маршрутизаторы могут, но не обязаны уведомлять отправителя. Часто и не уведомляют — по соображениям производительности, и обвинить их не в чем.

Но уж если тебе пришел ответ от промежуточного маршрутизатора, он обычно информативен. К примеру, ответ `destination host unreachable` должен отправляться только тогда, когда хост находится в одной локальной сети с маршрутизатором и не отвечает. Самый простой способ увидеть эту ошибку — пингануть заведомо несуществующий адрес в своей же сети: к примеру, если твоя сеть 192.168.0.0/24 и хоста 192.168.0.200 в ней нет, выполнить `ping 192.168.0.200`.

Такой ответ может прийти только от последнего маршрутизатора на пути к хосту.

А вот `network unreachable` говорит об отсутствии маршрута к указанной сети у одного из хостов на пути. Эта ошибка может возникнуть в любом месте пути, поэтому нужно обратить внимание на отправителя.

Чаще всего эта проблема у тебя самого: слетели настройки маршрутов или хост не

получил маршрут от сервера DHCP. Но такой ответ может прийти и от промежуточного маршрутизатора:

```
From 192.0.2.100 icmp_seq=1 Destination Net Unreachable
```

Если ты видишь такую картину, что-то серьезно пошло не так. Если хост достижим из других сетей, вполне возможно, что у провайдера проблема с настройками BGP. Я как минимум один раз сталкивался с тем, что крупный провайдер ошибочно фильтровал маршруты из сети, которую он считал зарезервированной для использования в будущем, хотя на тот момент IANA уже полгода как передала ее RIPE NCC и многие люди получили адреса из нее.

WWW

Если не хочешь быть как тот провайдер, можно воспользоваться автоматически обновляемыми списками несуществующих адресов вроде Cymru Bogon Reference

Ошибки семейства `destination host/net prohibited` означают, что пакет был отброшен правилом межсетевого экрана. Впрочем, никто не обязывает отвечать отправителю именно так или вообще отвечать. К примеру, в Linux правила вида `iptables -j REJECT` по умолчанию выдают `destination port unreachable`, если явно не указать `--reject-with`, причем указать можно любой тип, даже `icmp-net-unreachable`.

Но это все о простом `ping` без опций. Некоторые проблемы лучше всего выявляются дополнительными опциями.

Определяем проблемы с MTU

Пользователи VPN нередко могут столкнуться с недоступностью определенных ресурсов именно через туннель, даже если без туннеля из той же сети они прекрасно работают.

Распространенная причина таких проблем — некорректно настроена сеть назначения, из-за чего пакеты перестают проходить через туннель. Поскольку MTU (Maximum Transmission Unit — максимальный размер пакета) для туннелей меньше, чем

стандартный для интернета размер 1500, правильная работа соединений требует работающего path MTU discovery. Увы, работает он не всегда, и самый простой способ его сломать — запретить протокол ICMP, «чтобы не пинговал кто попало».

Такие патологические случаи среди сетевых админов встречаются редко, но именно сломанный `PMTU discovery`, увы, распространен.

Выявить эту проблему можно, указав размер пакета опцией `-s`: `ping -s1300 www.example.com`. Если стандартный пинг с размером пакета в 64 байта проходит, но с какого-то размера пакета (например, `-s 1450`) пинг внезапно перестает работать, то поздравляю, ты нашел проблему. Пиши админу, чтобы включил MSS clamping, или включай сам, если ты и есть админ.

Ищем глубокую инспекцию пакетов

Многие решения для DPI не смотрят так уж глубоко, а просто ищут фиксированные строки в пакетах. В некоторых случаях определить наличие такого DPI на пути можно с помощью одного ping. В Linux для этого есть опция `--pattern`. Один недостаток — сомнительную строку придется вручную кодировать в hex, но, если установить генератор пакетов нет никакой возможности, может пригодиться.

TTL exceeded

Еще одна ошибка, которую на практике можно увидеть только с дополнительной опцией, — `Time to live exceeded`. Поле Time To Live у пакетов IPv4 (Hop Limit в IPv6) ограничено значением 255, но интернет — «тесная сеть», и средний путь не достигает и одной десятой максимальной длины. Изначальная цель этого механизма — предотвратить бесконечную пересылку пакетов по кругу при возникновении петли маршрутизации, но современные протоколы исключают петли. Тем не менее никто не мешает указать TTL заведомо короче ожидаемой длины пути:

```
$ ping -t1 9.9.9.9
PING 9.9.9.9 (9.9.9.9) 56(84) bytes of data.
From 10.91.19.1 icmp_seq=1 Time to live exceeded
```

traceroute

Именно по принципу, описанному выше, и работает команда `traceroute`: отправляет пакеты сначала с TTL = 1, затем TTL = 2 и так далее и ждет ответов TTL exceeded от

каждого промежуточного хоста.

На первый взгляд traceroute представляется таким же простым инструментом, как ping. На самом деле из его вывода тоже можно извлечь больше данных, чем кажется. В то же время там можно увидеть проблему, которой не существует в реальности, а реальные проблемы могут никак не отображаться.

Задержки

Как обычная команда traceroute, так и инструменты вроде MTR весьма популярны для поиска «узких мест» в сети. MTR показывает статистику задержек для каждого хоста на пути.

Однако интерпретация этих данных не так очевидна. Предположим, ты видишь на первом хосте задержку в 20 миллисекунд, а на втором — 950. Не спеши радоваться, что нашел узкое место, и ругать админов той сети. Задержки выдачи ICMP TTL exceeded могут не иметь ничего общего с задержками передачи пакетов.

В нашем сценарии 950 миллисекунд — это именно время, которое прошло от отправки пакета до получения ответа ICMP. Сколько ушло времени на передачу тестового пакета — неизвестно. Сколько ушло на доставку ответа — тоже. Сколько прошло времени между получением пакета и отправкой ответа?

Будет большой ошибкой считать, что это время близко к нулю. Во-первых, пересылкой пакетов часто занимается аппаратная фабрика коммутации, а для управляющей операционной системы используется весьма скромный процессор. Поскольку сообщения ICMP всегда генерируются программно, этот процесс гораздо медленнее. Даже для чисто программных решений задачи вроде ответов на пинги и отправки TTL exceeded куда менее приоритетны, чем маршрутизация.

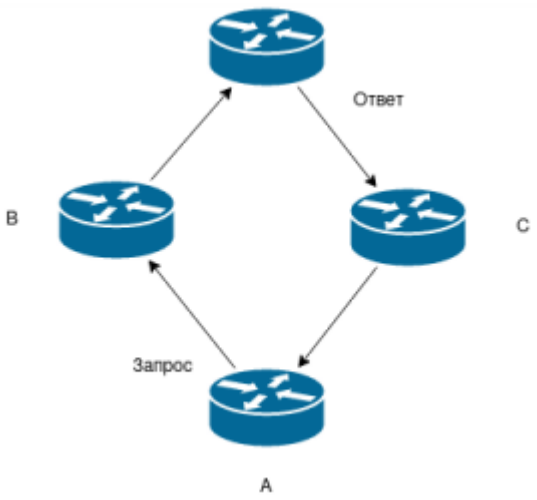
Поэтому сам по себе всплеск задержек в выводе traceroute или MTR ничего не означает. Вот если все задержки на последующих хостах больше 950 миллисекунд, тогда уже есть повод ругаться с админами.

Асимметричные пути, балансировка, MPLS

Не меньшей ошибкой будет считать, что трафик всегда возвращается к тебе тем же путем, который ты видишь в traceroute. Для транзитных маршрутизаторов, в отличие от клиентских, асимметричная маршрутизация — явление обычное, даже неизбежное.

Если сеть провайдера A подключена к сетям B и C каналами с одинаковой пропускной

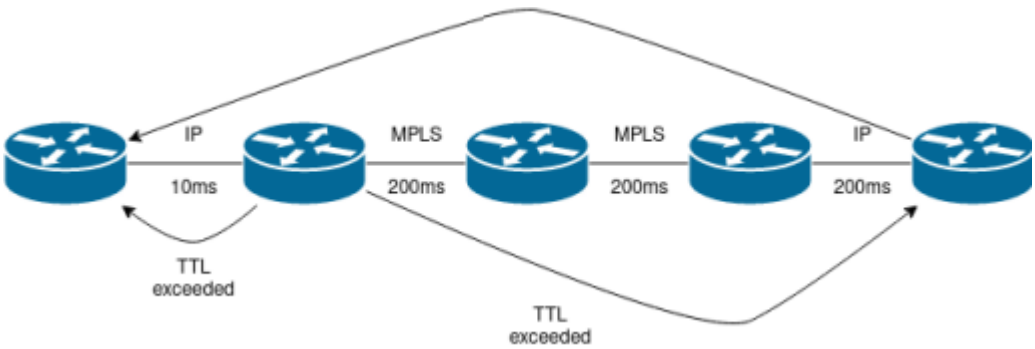
способностью и качеством, будет вполне естественно распределить исходящий трафик по этим каналам. Однако даже если админы провайдера A предпочитают отправлять большую часть трафика через сеть B, над входящим трафиком у них нет почти никакого контроля. Можно искусственно «ухудшить» анонсы маршрутов в сети C, но и в этом случае нет гарантии, что клиенты сети B проводят ту же политику. Вполне возможно, что они как раз предпочитают сеть C.



Предположим, что сеть C на самом деле плохая. Увидим ли мы это в traceroute? Очевидно, нет, поскольку истечение TTL всегда случается на прямом пути, а не на обратном. Никакого способа увидеть полный граф сети не существует.

Еще более интересной ситуацию делает MPLS. Поскольку кадры MPLS инкапсулируют пакеты IP целиком и для сетей на концах LSP он выглядит как прямое физическое соединение, огромная часть внутренней структуры транзитной сети оказывается невидимой.

Эта ситуация делает отладку сложнее не только для пользователей, но и для самих провайдеров, поэтому иногда используют MPLS ICMP tunneling, который позволяет генерировать корректные ответы. Однако, поскольку отправителем пакетов IP выступает последний маршрутизатор в логическом соединении MPLS (до него никакой обработки IP не делается), это будет выглядеть как множество хостов с нулевой задержкой между ними.



whois

Предположим, ты нашел проблемную сеть с помощью ping или traceroute или увидел адреса злоумышленников в логах. Как теперь найти, с кем ругаться? Здесь тебе на помощь придет whois.

Чаще всего команду whois используют для получения информации о доменах (`whois example.com`). На самом деле в базах данных RIR (Regional Internet Registries — RIPE NCC, ARIN...) существует куда больше типов объектов.

Каждый выделенный ресурс, будь то номер автономной системы или адрес сети, имеет в базе данных свою запись, из которой можно узнать его принадлежность.

К примеру, ты хочешь пожаловаться, что авторы «Хакера» учат молодежь плохому. Можно отрезолвить xakep.ru и выполнить `whois 178.248.232.27`. Но ты подозреваешь, что информация о самом адресе не совпадает с информацией о сети хостера. Не страшно, whois понимает и адреса сетей:

```
$ whois 178.248.232.0/24
[Querying whois.ripe.net]
[whois.ripe.net]
...
% Information related to '178.248.232.0 - 178.248.239.255'
% Abuse contact for '178.248.232.0 - 178.248.239.255' is 'abuse@qrator.net'
inetnum: 178.248.232.0 - 178.248.239.255
netname: RU-QRATOR-20100512
country: RU
org: ORG-LA267-RIPE
admin-c: QL-RIPE
tech-c: QL-RIPE
status: ALLOCATED PA
remarks: QRATOR filtering network
```

Здесь нам сразу показывают abuse contact, но, если его вдруг нет, всегда можно копать дальше и смотреть whois про любое поле. Например, `whois ORG-LA267-RIPE`. По правилам, у каждой организации есть abuse contact.

Оттуда же можно извлечь информацию об автономных системах. Просто допиши AS перед номером. Например, кому принадлежит автономная система 6939?

```
$ whois AS6939
[Querying whois.radb.net]
[whois.radb.net]
aut-num: AS6939
as-name: HURRICANE
descr: Hurricane Electric
```

Как поступать с людьми, которые шлют тебе спам и брутфорсят твои сервисы, — решать тебе, но во многих случаях написать письмо хостеру на abuse contact не будет лишним. Большинству хостеров и провайдеров нарушители закона и порядка ни к чему, так что, если у тебя есть доказательства, вероятность, что сервер нарушителя заблокируют, ненулевая.

Looking glass

Многие провайдеры предоставляют сервисы looking glass, с помощью которых можно просмотреть отладочную информацию с **их** маршрутизаторов.

Они доступны через веб-интерфейс, но иногда и через Telnet/SSH.

Увы, общего способа его найти не существует, и на сайтах провайдеров эти адреса чаще всего не пишут. Тем не менее сам термин настолько стандартный, что запрос $provider looking glass в любой поисковой системе приведет тебя к цели. Например, `https://duckduckgo.com/?q=rostelecom+looking+glass` приводит нас к `http://lg.ip.rt.ru`.

Большинство из них позволяют выполнить ping и traceroute, а самые полные предоставляют еще и информацию о маршрутах BGP и возможность выбора маршрутизаторов из разных регионов.

Заключение

Люди постоянно создают новые и полезные инструменты, но начинать отладку всегда можно со старых, испытанных и присутствующих в любой системе — их же вполне может оказаться достаточно.

Даниил Батурин

Координатор проекта VyOS (https://vyos.io), «языковед», функциональщик, иногда сетевой администратор