1.1. Стек TCP/IP

Под понятием "TCP/IP" обычно понимают все, что связано с протоколами TCP и IP. Сюда входит семейство сетевых протоколов, прикладные программы, представляющие автономную систему, в рамках которой, происходит работа и ее функционирование [1]. В основе стека лежат протоколы IP и TCP, которые и дали ему название. Примерами протоколов, которые входят в состав стека являются FTP, IMAP4, POP3, HTTP, SCTP, UDP, ICMP, IGMP, OSPF, BGP, RIP. В таблице 1.1 представлены уровни моделей OSI и TCP/IP [2].

Таблица 1.1 Стек протоколов

Модель OSI	**Модель TCP/IP**	**Стек TCP/IP**
Прикладной	Прикладной	FTP, POP3, SMTP, IMAP4, ICQ, HTTP
Представительный		
Сеансовый		
Транспортный	Транспортный	TCP, UDP, SCTP
Сетевой	Сетевой	IP, ICMP, IGMP
Канальный	Канальный	802.3 (Ethernet), 802.5 (Token Ring), Fast Ethernet
Физический		Витая пара, оптоволокно, радиоволны

1.2. IP v.4 адресация. Маски

IP адрес версии 4 [3] адрес состоит из двух логических частей – адреса сети и адреса (интерфейса) хоста в сети и занимает пространство в 32 бита и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками (см. рис. 1.1), например, 135.15.78.110. Для гибкого использования пространства адресов вводится понятие «маска». *Маска* – это число, которое используется совместно с IP-адресом и содержит единицы (в двоичном представлении числа), которые соответствую битам (IPv4 адреса), относящихся к адресу сети, а в битах для номера хоста – нули.

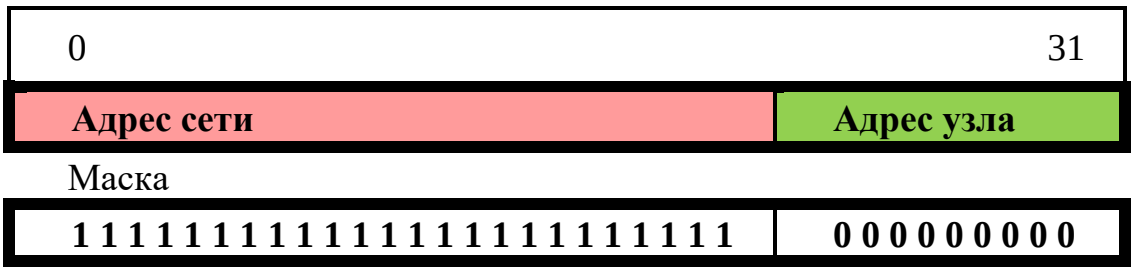

Рис. 1.1. Представление IPv4 адреса сети и маска

Например, адрес 135.15.78.110 с маской 255.255.0.0, говорит о том, что 135.15.0.0 это адрес сети, а число 0.0.78.110 это адрес узла в этой сети. Или существует краткая запись 135.15.78.110/16, означающаяся, что первые 16 бит отводятся под адрес сети.

Маска позволяет более рационально использовать вам выделенное пространство адресов.

Пример: Администратор получил в свое распоряжение адрес сети 169.201.0.0.

Требуется, чтобы сеть была разделена на три отдельных подсети с равным числом узлов в каждой. Необходимо найти минимальное число бит в которое «помещается» требуемое число подсетей. Для 3-х это будет два бита. Запишем в первые два бита в 3-м байте по единице, а в остальные 6 запишем единицы. Тогда получим в двоичном представлении число 11000000, или в десятичном представлении оно равно 192. Следовательно, взяв маску 255.255.192.0 мы выделим 2 бита в 3-ем байте под адреса подсетей.

Если же для предыдущего примера поставим обратную задачу – на построение максимального числа подсетей с определенным одинаковым числом узлов, например, 500. Тогда выделяется минимальное пространство под узлы, в которое помещается число 500, и оно заполняется нулями. В данном случае, требуется 9 бит. Для третьего и четвертого байта это будет двоичное число 11111110.00000000. Итак, получаем маску 255.255.254.0.

1.3. IP протокол v.4

Заголовок пакета

IP (Internet Protocol – протокол межсетевого взаимодействия), обеспечивает наилучшую попытку по передачи пакетов от отправителя к получателю через произвольное число промежуточных узлов [1, 4-6].

Протокол IP относится к протоколам без установления соединений. Каждый IP-пакет обрабатывается как независимая единица. Пакеты могут дублироваться и приходить не в порядке следования. Протокол не решает задачи надежной доставки сообщений от отправителя к получателю. За надежную доставка пакетов отвечают некоторые протоколы более высокого уровня стека TCP/IP.

Структура IP-пакета

IP-пакет состоит из заголовка и поля данных (см. IP версии 4 на рис. 1.2).

<table>
<tr><td>offset</td><td colspan="16">0 34 78 1516 31</td></tr>
<tr><td rowspan="2">0</td><td colspan="3">Версия (Version)</td><td colspan="3">Длина заголовка (IHL)</td><td colspan="5">Тип сервиса (TOS)</td><td colspan="8" rowspan="2">Общая длина (Total Length)</td></tr>
<tr><td>PR</td><td>D</td><td>T</td><td>R</td><td></td></tr>
<tr><td>4</td><td colspan="8">Идентификатор пакета (Identification)</td><td colspan="3">Флаги</td><td colspan="5">Смещение фрагмента (Fragment Offset)</td></tr>
<tr><td></td><td>DF</td><td>MF</td></tr>
<tr><td>8</td><td colspan="4">Время жизни (TTL)</td><td colspan="4">Протокол (Protocol)</td><td colspan="8">Контрольная сумма заголовка (Header Checksum)</td></tr>
<tr><td>12</td><td colspan="16">Адрес источника (Source Address)</td></tr>
<tr><td>16</td><td colspan="16">Адрес назначения (Destination Address)</td></tr>
<tr><td>20</td><td colspan="12">Опции (Options)</td><td colspan="4">Padding</td></tr>
</table>

Рис. 1.2 Формат заголовка IP-пакета (v4)

Номер версии (Version) – занимает 4 бита и указывает версию протокола IP. В настоящий момент используется версия 4 (IPv4) версию 6 (IPv6).

Длина заголовка (IHL – Internet Header Length) – занимает 4 бита и указывает значение длины заголовка, измеряемое в 32-битовых словах (4 байта). Минимальная длина заголовка составляет 20 байт и значение данного поля не должно быть меньше 5. Однако заголовок возможно увеличить до 60 байт за счет использования байт поля *Опции.*

Тип сервиса (TOS – Type Of Services) – занимает 1 байт и определяет тип сервиса обработки пакета. Как правило, это поле используется маршрутизаторами. Первые 3 бита отводятся под поле *приоритета (Precedence)* пакета и имеет значение от низкого – 0 (нормальный пакет) до самого высокого – 7 (пакет управляющей информации). Пакеты с большим значением имеют более высокий приоритет на обработку. Следующие 3 бита

определяют критерий выбора маршрута: установленный бит *D* (*Delay*) – выбор маршрута с минимальной задержкой доставки данного пакета; бит *T* (*Throughput*) – маршрут с максимальной пропускной способности; бит *R* (*Reliability*) – маршрут с максимальной надежностью доставки. Остальные биты зарезервированы и имеют нулевое значение.

Общая длина (Total Length) – занимает 2 байта и определяет общую длину пакета с учетом заголовка и поля данных.

Следующая строка отвечает за фрагментацию пакетов.

Идентификатор пакета (Identification) – занимает 2 байта и используется для идентификации фрагментированных пакетов. Все фрагменты исходного пакета должны иметь одинаковое значение этого поля. Фрагменты с одинаковым адресом источника, адресом назначения, типом протокола и идентификатором группируются.

Флаги (Flags) – занимают 3 бита и содержат флаги контроля, связанные с фрагментацией: установленный бит *DF* (*Do not Fragment*) запрещает фрагментацию данного пакета; установленный бит *MF* (*More Fragments*) показывает, что данный пакет является промежуточным (не последним) фрагментом; оставшийся бит зарезервирован.

Смещение фрагмента (Fragment Offset) – показывает месторасположение данных фрагмента в исходном пакете: занимает 13 бит и задает смещение в 8-байтовых словах поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Рассмотренные биты второй строки заголовка используются при сборке/разборке фрагментов пакетов.

Время жизни (TTL – Time To Live) – занимает 1 байт и определяет предельный срок, в течение которого пакет может перемещаться по сети. *Время жизни* данного пакета измеряется в секундах и задается отправителем. При прохождении пакета между маршрутизаторами и другими узлами сети (данный переход называется – hop), из текущего времени вычитается единица. Как только параметр времени жизни станет нулевым, до достижения получателя, то пакет удаляется.

Протокол (Protocol) – занимает 1 байт и указывает, данные какого протокола верхнего уровня содержит IP пакет (например, UDP или ICMP).

Контрольная сумма заголовка (Header Checksum) – занимает 2 байта, рассчитывается только по заголовку. Т.к. некоторые поля заголовка меняют свое значение (например, поле TTL) на пути прохождения пакета, контрольная сумма проверяется и повторно вычисляется при каждой обработке IP-заголовка. При вычислении контрольной суммы значение самого поля устанавливается в нуль. При обнаружении ошибки контрольной суммы пакет удаляется. Метод пересчёта контрольной суммы определён в RFC 1071 [7].

Адрес источника (Source Address) – содержит IP-адрес отправителя пакета и занимает 32 бита.

Адрес назначения (Destination Address) – содержит IP-адрес получателя пакета и занимает 32 бита.

Опции (Options) – является не обязательным и используется редко, например, при отладке сети. В частности, можно указать точный маршрут прохождения пакетов через маршрутизаторы, помещать временные метки прохождения шлюзов маршрутизации. Номера опций размещаются на сайте IANA [6].

Выравнивание (Padding) – имеет переменную длину и используется для выравнивания заголовка пакета по 32-битной границе. Выравнивание выполняется путем заполнения данного поля нулями.

Фрагментация

При передаче сообщения от одного IP-узла к другому может случиться так, что датаграммы будут передаваться по сети, для которой допустимый размер пакета данных (MTU, см. п 1.9) меньше размера датаграммы. В этом случае включается механизм *фрагментации* датаграмм (см.рис. 1.3).

Фрагментация IP-датаграмм необходима, когда ее размер превышает размер максимально допустимого пакета данных при передаче по сегменту сети.

Процедура фрагментации может разбить датаграмму на пакеты произвольной длины, а затем восстановить ее в первоначальном виде. Это обеспечивается полем Identification (которое однозначно определяет принадлежность фрагмента к исходному фрагментируемому пакету), полем смещения фрагмента (Fragment Offset) и флагом «следующего фрагмента» MF (More Fragments). Данной информации достаточно, чтобы узел, получивший информацию фрагментами, смог собрать исходную датаграмму.

В случае, когда датаграмма помечена как «не фрагментируемая» (установлен флаг DF), то она не может подвергаться разбиению на пакеты меньшей длины. Если же нет альтернатив по доставке такой датаграммы в точку назначения без фрагментации, она будет уничтожена.

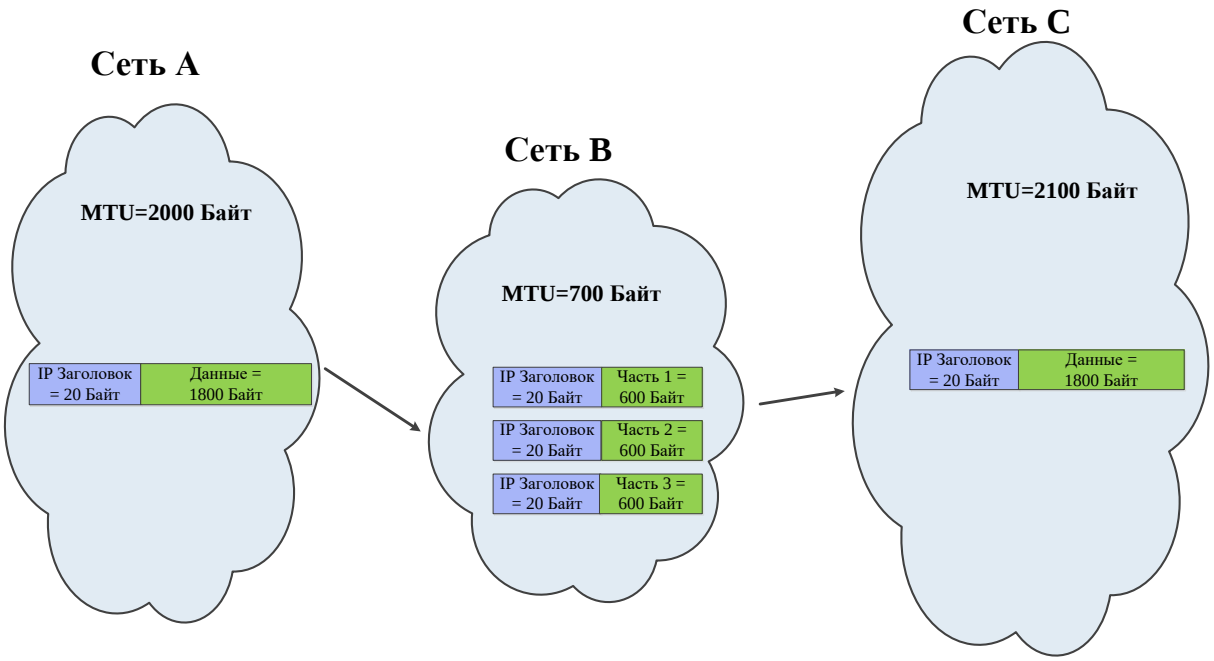

Рис. 1.3. Пример фрагментации пакетов

ICMP протокол

Протокол IP имеет ясную и элегантную структуру. В нормальных ситуациях IP очень эффективно использует для пересылки память и ресурсы. Однако, что произойдет в нестандартной ситуации? Что может прервать бесцельное блуждание датаграммы до завершения ее времени жизни после краха маршрутизатора и неисправности в сети? Кто предупредит приложение о прекращении отправки датаграмм в недостижимую точку назначения?

Средства для лечения таких неисправностей предоставляет *протокол управляющих сообщений Интернета* (Internet Control Message Protocol — ICMP) [1]. Он выполняет роль

сетевого помощника, способствуя маршрутизации в хостах и обеспечивая сетевого администратора средствами определения состояния сетевых узлов. Функции ICMP являются важной частью IP. Все хосты и маршрутизаторы должны быть способны генерировать и обрабатывать сообщения ICMP. При правильном использовании эти сообщения могут улучшить выполнение сетевых операций.

Сообщения ICMP пересылаются в датаграммах IP с обычным заголовком IP (см. рис. 2.3), имея в поле *протокола* значение 1.

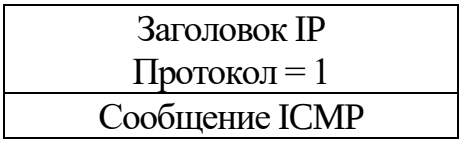

Рис.2.3 Пакетирование сообщения ICMP

ICMPопределен в RFC 792.RFC 1122 (требования к хостам) и RFC 1812 (требования к маршрутизаторам) содержат несколько очень полезных разъяснений. Исследованию маршрутов посвященRFC 1256. Исследование MTU по пути рассмотрено в RFC 1191, а дополнительные рекомендации представлены в RFC 1435.

2.3.1. Сообщения об ошибках ICMP

Бывают ситуации, приводящие к отбрасыванию (удалению из сети) датаграммы IP. Например, точка назначения может стать недоступной из-за обрыва связи. Или может завершиться время жизни датаграммы. Маршрутизатор не сможет переслать длинную датаграмму при запрещении фрагментации.

При отбрасывании датаграммы по адресу ее источника направляется сообщение ICMP, указывающее на возникшую проблему [1].

ICMPбыстро сообщит системе о выявленной проблеме. Это очень надежный протокол, поскольку указание на ошибки не зависит от наличия сетевого центра управления.

Однако в использовании сообщений ICMP имеются некоторые недостатки. Например, если недостижима точка назначения, то сообщение будет распространяться до источника по всей сети, а не на станцию сетевого управления.

Реально ICMP не имеет средств предоставить отчет об ошибках выделенному операционному центру. Для этого служит протокол SNMP.

Типы сообщений об ошибках

В таблице 2.2перечислены формальные имена сообщений об ошибках ICMP.

Таблица 2.2 Сообщения об ошибках

Сообщение	Описание
Destination Unreachable (недостижимая точка назначения)	Датаграмма не может достичь хоста назначения, утилиты или приложения
Time Exceeded (время закончилось)	Маршрутизатор определил завершение времени жизни, или закончилось время на сборку фрагментов в хосте назначения
Parameter Problem(проблема с параметром)	В заголовке IP неверный параметр

Source Quench (подавление источника)	Перегружен маршрутизатор или система назначения (системам рекомендуется *не отправлять* это сообщение)
Redirect (перенаправление)	Хост направил датаграмму на неверный локальный маршрутизатор

Обязанность по отправке сообщения ICMP

Протокол ICMP определяет, что сообщения *могут* или *должны* быть посланы в каждом случае, но он не требует выдавать сообщения ICMP о *каждой* ошибке.

В этом есть здравый смысл. Основным назначением маршрутизатора в сети является пересылка датаграмм. Перегруженный хост назначения должен уделять больше времени доставке датаграмм в приложения, а не указанию на ошибки удаленному хосту. Именно поэтому не формируются сообщения о случайном отбрасывании датаграммы.

Когда <u>не нужно</u> посылать сообщение ICMP

Напомним, что ICMP-сообщение об ошибке посылается, когда в сети не все благополучно. Важно, обеспечить, чтобы трафик ICMP не перегружал сети, делая ситуацию еще хуже. Для этого протокола, требуется ввести несколько очевидных ограничений. ICMP не должен формировать сообщения о:

- маршрутизации и доставке ICMP-сообщений messages;
- широковещательных и многоадресных датаграммах;
- фрагментах датаграмм, кроме первых;
- сообщениях, чей адрес источника не идентифицирует уникальный хост (например, IP-адреса источников 127.0.0.1 или 0.0.0.0).

Формат сообщения ICMP

Сообщение ICMP переносится в части данных датаграммы IP. Каждое сообщение ICMP начинается тремя одинаковыми полями: полем *типа* (Type), полем *кода* (Code), обеспечивающим более подробное описание ошибки, и полем *контрольной суммы* (Checksum). Формат оставшейся части сообщения определяется типом сообщения.

Сообщение об ошибке ICMP обрамляется заголовком IP. Добавляются первые 8 октетов датаграммы, которая привела к ошибке. Эти сведения позволяют проанализировать причину ошибки, поскольку содержат информацию о предполагаемом назначении датаграммы и целевом протоколе четвертого уровня. Дополнительные 8 байт позволяют определить коммуникационный элемент приложения(подробнее в [1]).

В сообщение включается и контрольная сумма ICMP, начиная от поля *Type.*

Сообщение Destination Unreachable

Существует много причин прекращения доставки датаграммы. Разорванная связь физически не позволит маршрутизатору достичь подсети назначения или выполнить пересылку в точку следующего попадания. Хост назначения может стать недоступным при отключении его для проведения профилактики.

Современные маршрутизаторы имеют хорошие средства обеспечения безопасности. Они могут быть сконфигурированы для просмотра входящего в сеть трафика. При запрещении сетевым администратором доступа к точке назначения датаграмма также не может быть доставлена.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Type =3	Code	Контрольная сумма
Не используется		
Заголовок Интернета 8 октетов данных исходной датаграммы		

Рис.2.3 Формат ICMP-сообщенияDestination Unreachable

Формат сообщения *Destination Unreachable*показан на рисунке2.3. Поле Type (в нашем случае 3) идентифицирует именно этот *mun* сообщения. Поле *Code* отражает причину отправки сообщения. Полный список кодов этого поля представлен в таблице 2.3.

Таблица 2.3 Коды ошибок сообщения Destination Unreachable

Код	**Смысл**
0	Сеть недостижима
1	Хост недостижим
2	Запрашиваемый протокол не поддерживается в точке назначения
3	Порт недостижим (недоступно удаленное приложение)
4	Необходима фрагментация, но установлен флаг "Не фрагментировать"
5	Неверен маршрут от источника
6	Неизвестна сеть назначения
7	Неизвестен хост назначения
8	Хост источника изолирован
9	Административно запрещены коммуникации с сетью назначения
10	Административно запрещены коммуникации с хостом назначения
11	Сеть недостижима для заданного типа обслуживания
12	Хост недостижим для заданного типа обслуживания

Сообщение Time Exceeded

Пересылаемая датаграмма может быть отброшена по тайм-ауту при уменьшении до нуля ее времени жизни (TTL). Еще один тайм-аут может возникнуть в хосте назначения, когда завершится время, выделенное на сборку, а прибыли еще не все фрагменты датаграммы. В обоих случаях формируется сообщение *Time Exceeded* для источника датаграммы. Формат этого сообщения показан на рисунке2.4.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Type =11	Code	Контрольная сумма
Не используется		
Заголовок Интернета 8 октетов данных исходной датаграммы		

Рис. 2.4 Формат **с**ообщение Time Exceeded

Значения кодов (см. таблицу 2.4) отражают причину тайм-аута.

Таблица 2.4 Коды сообщения Time Exceeded

Код	Смысл
0	Завершилось время жизни датаграммы
1	Завершилось время на сборку фрагментов датаграммы

Сообщение Parameter Problem

ICMP-сообщение *ParameterProblem*используется для отчета об ошибках, не специфицированных в кодах других сообщений. Например, в полях вариантов может появиться неверная информация, не позволяющая правильно обработать датаграмму, в результате чего датаграмма будет отброшена. Более часто проблемы с параметрами возникают из-за ошибок в реализации, когда система пытается записать параметры в заголовок IP.

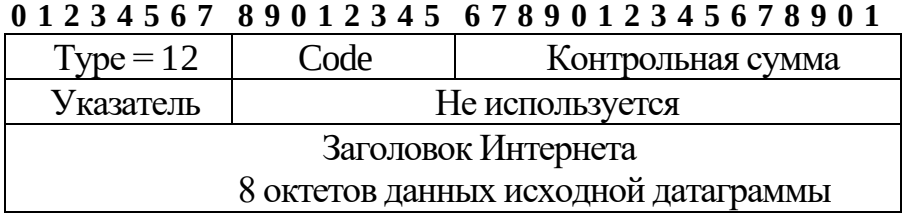

Рис. 2.5 ФорматICMP-сообщения ParameterProblem

Поле*Pointer*(указатель) сообщения *Parameter Problem*идентифицирует октет, в котором выявлена ошибка. На рисунке2.5показан формат сообщения *ParameterProblem,* а в таблице 2.5— значения кодов ошибок.

Таблица 2.5 Коды сообщения Parameter Problem

Код	Смысл
0	Значение в поле указателя специфицирует ошибочный октет
1	Отсутствует требуемый вариант (используется военными для указания на отсутствие параметров безопасности)
2	Неверная длина

Проблемы перегрузок

ПротоколIP очень прост: хост или маршрутизатор обрабатывают датаграмму и посылают ее как можно быстрее. Однако доставка не всегда проходит гладко. Могут возникнуть различные проблемы.

Когда один или несколько хостов отправляют трафик UDP на медленный сервер, то на последнем может возникнуть перегрузка, что приведет к отбрасыванию сервером некоторой части этого трафика.

Маршрутизатор может переполнить свои буферы и далее будет вынужден отбрасывать некоторые поступающие датаграммы. Медленное соединение через региональную сеть (например, на скорости 56 Кбит/с) между двумя скоростными локальными сетями (например, в 100 Мбит/с) может создать затор на пути следования датаграмм. Из-за этого в сети возникнут перегрузки, которые также приведут к отбрасыванию датаграммы и, следовательно, к созданию еще большего трафика.

Сообщение Source Quench

Сообщение *Source Quench* (подавление источника) показано на рисунке2.6. Оно позволяет попытаться решить проблему перегрузок, хотя и не всегда успешно. Механизмы для подавления источника перегрузки сети должны создавать разработчики конкретных продуктов, но остается открытым конкретный вопрос:

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Type =4	Code	Контрольная сумма
Не используется		
Заголовок Интернета 8 октетов данных исходной датаграммы		

Рис. 2.6 Формат ICMP-сообщения Source Quench

Когда и кому маршрутизатор или хост должен отправлять сообщение SourceQuench?

Обычно ICMP-сообщение указывает хосту источника на причину отбрасывания посланной им датаграммы. Однако при перегрузке такое сообщение может не дойти до этого хоста, генерирующего очень напряженный сетевой трафик. Кроме того, очень расплывчаты требования к обработке поступающих сообщений *Source Quench.*

Текущий документ по *требованиям к хостам* (RFC 1812) оговаривает в качестве особого пункта, что сообщения *Source Quench* вовсе *не нужно* посылать. Работа должна выполняться более совершенным механизмом управления нагрузкой в сети.

Сообщения Redirect

К локальной сети может быть подключено более одного маршрутизатора. Когда локальный хост посылает датаграмму не на тот маршрутизатор, последний пересылает ее и отправляет хосту источника ICMP-сообщение *Redirect*(перенаправление). Хост должен переключить последующий трафик на более короткий путь.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Type =5	Code	Контрольная сумма
Не используется		
Заголовок Интернета 8 октетов данных исходной датаграммы		

Рис. 2.7 Формат ICMP-сообщения Redirect

Сообщение *Redirect*используется и для выключения маршрутизатора системным администратором. Хост может быть сконфигурирован с единственным маршрутизатором по умолчанию; при этом он будет динамически определять возможности пересылки через другие маршрутизаторы.

Формат сообщения о перенаправлении показан на рисунке2.7. Коды этого сообщения перечислены в таблице 2.6. Некоторые протоколы маршрутизации способны выбирать путь доставки на основе содержимого поля *типа обслуживания* (TOS) датаграммы. Коды 2 и 3 предоставляют некоторые сведения для такого выбора.

Таблица 2.6 Коды перенаправления

Код	Смысл
0	Перенаправление датаграммы в сеть
1	Перенаправление датаграммы в хост
2	Перенаправление датаграммы в сеть на основе значения из поля типа обслуживания
3	Перенаправление датаграммы в хост на основе значения из поля типа обслуживания

Управление поступающими сообщениями ICMP

Что должен делать хост, получивший сообщение ICMP? Реализации различных разработчиков по-разному отвечают на этот вопрос. В некоторых из них хосты игнорируют все или многие такие сообщения. Стандарты TCP/IP оставляют большую свободу выбора в решении этого вопроса. Для различных типов сообщений ICMP предлагаются следующие рекомендации:

Destination Unreachable	Доставить ICMP-сообщение на транспортный уровень. Выполняемые действия должны зависеть от того, является ли причина вывода сообщения временной или постоянной (например, административный запрет на пересылку).
Redirect	Хост *обязан* обновить таблицу маршрутизации.
SourceQuench	Доставить ICMP-сообщение на транспортный уровень или в модуль обработки ICMP.
Time Exceeded	Доставить на транспортный уровень.
Parameter Problem	Доставить ICMP-сообщение на транспортный уровень с необязательным уведомлением пользователя.

Иногда ошибки должны обрабатываться совместно операционной системой, коммуникационным программным обеспечением и сетевым приложением.

2.3.2. Исследование MTU по пути

При пересылке большого объема данных (например, при копировании файлов по сети) с одного хоста на другой размер датаграмм существенно влияет на производительность. Заголовки IP и TCP требуют не менее 40 дополнительных байт [1].

- Если данные пересылаются в 80-байтовых датаграммах, дополнительная нагрузка составит 50%.
- Если данные пересылаются в 400-байтовых датаграммах, дополнительная нагрузка составит 10%.
- Если данные пересылаются в 4000-байтовых датаграммах, дополнительная нагрузка составит 1%.

Для минимизации дополнительной нагрузки лучше отсылать датаграммы наибольшего размера. Однако этот размер ограничивается значением максимального элемента пересылки (*Maximum Transmission Unit* — MTU) для каждого из носителей. Если датаграмма будет слишком большой, то она будет фрагментирована, а этот процесс снижает производительность. С точки зрения пользователя, качество сети определяется двумя параметрами: интервалом пересылки (от начала пересылки до ее завершения) и временем ожидания (задержкой доступа к сети, занятой другими пользователями). Увеличение размера датаграммы приводит к снижению интервала пересылки, но

увеличению ожидания для других пользователей. Грубо говоря, нагрузка на сеть будет выглядеть как пиковые импульсы с очень небольшой нагрузкой между ними, что считается самым неудачным вариантом загрузки сети. Гораздо лучше, когда сеть нагружается равномерно (*Прим. пер.).*

Многие годы хосты избегали фрагментации, устанавливая эффективное значение MTU для пересылки в 576 октетов для всех нелокальных хостов. Это часто приводило к ненужному снижению производительности.

Гораздо полезнее заранее знать наибольший допустимый размер датаграммы, которую можно переслать по заданному пути. Существует очень простой механизм *исследования MTU по пути* (*Path MTU discovery*), позволяющий узнать это значение. Для такого исследования:

- Флаг *"Не фрагментировать"* заголовка IP устанавливают в 1.
- Размер MTU по пути первоначально устанавливают в значение MTU для локального интерфейса.
- Если датаграмма будет слишком велика для одного из маршрутизаторов, то он пошлет обратно ICMP-сообщение *Destination Unreachable* с кодом 4.
- Хост источника уменьшит размер датаграммы и повторит попытку.

Какое же значение нужно выбрать для следующей попытки? Спецификация IP предполагает сохранение значения MTU и его доступность для протоколов транспортного уровня. Если маршрутизатор имеет современное программное обеспечение, то он будет включать в пересылаемое дальше по сети сообщение *DestinationUnreachable*размер MTU (см. рис. 2.8). Иногда средства защиты конфигурируются на полное исключение *всех* входящих сообщений ICMP, что не позволяет использовать механизм определения MTU по пути следования датаграммы.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Type = 3	Code	Контрольная сумма
Не используется		MTU для следующего попадания
Заголовок Интернета + 8 октетов данных исходной датаграммы		

Рис. 2.8 Сообщение Destination Unreachable приносит результат исследования размера MTU.

Поскольку пути пересылки могут меняться динамически, флажок *"Не фрагментировать"* нужно устанавливать во всех коммуникационных датаграммах. При необходимости маршрутизатор будут посылать сведения об обновлениях.

Если маршрутизатор использует устаревшее программное обеспечение, он не сможет предоставить значение MTU для следующего попадания. В этом случае значение для следующей попытки будет выбираться из списка стандартных размеров MTU с постепенным уменьшением для каждой новой попытки до достижения значения, нужного для коммуникации с удаленным хостом.

Разумеется, изменение пути следования может создать предпосылки для использования большего размера MTU. В этом случае система, согласовавшая небольшой размер MTU, будет пытаться его увеличить, если такое улучшение будет возможно.

2.3.3. Сообщения запросов ICMP

Не все сообщения ICMP сигнализируют об ошибках. Некоторые из них извлекают

из сети полезные сведения. Работает ли хост X? Не выключен ли хост Y? Как долго движется датаграмма до хоста Z и обратно? Какова маска подсети хоста источника?

Ответы на эти вопросы дают следующие сообщения ICMP[1]:

- *Эхо-запросы* и *эхо-ответы* обеспечивают обмен информацией между хостами и маршрутизаторами.
- Запросы и ответы о *маске адреса* позволяют системе исследовать присвоенную интерфейсу маску адреса.
- Запросы и ответы *временной метки* служат для извлечения сведений об установке времени на целевой системе. Ответы на такие запросы дают информацию, необходимую для оценки времени обработки датаграмм на хосте.

Эхо-запросы и эхо-ответы

Эхо-запросы (Echo Request) и *эхо-ответы* (Echo Reply) применяются для проверки активности системы. Код типа 8 применяется в запросах, а код 0 — в ответах. Количество октетов в поле данных переменно и может выбираться отправителем.

Отвечающая сторона должна послать обратно те же самые данные, которые были получены. Поле *идентификатора* служит для сравнения ответа с исходным запросом. Последовательный номер эхо-сообщения может применяться для тестирования, на каком участке произошел обрыв сети, и для вычисления приблизительного времени на путь туда и обратно. При этом идентификатор не меняется, а последовательный номер (начиная от 0) увеличивается на единицу для каждого сообщения. Формат эхо-сообщения показан на рисунке 2.9.

0 1 2 3 4 5 6 7	**8 9 0 1 2 3 4 5**	**6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1**
Type = 8 или 0	Code	Контрольная сумма
Идентификатор		Последовательный номер
Данные		

Рис. 2.9 Формат ICMP-сообщений Echo Request и EchoReply

Широко известная команда *ping*доступна почти во всех системах TCP/IP, а ее работа основана на ICMP-сообщениях для эхо-запросов и эхо-ответов.

Маска адреса

Напомним, что организация может разделить поле своего локального адреса на часть подсети и часть хоста. Когда включается система, она может быть сконфигурирована так, что не будет заранее знать, сколько бит было присвоено полю адреса подсети. Чтобы выяснить этот вопрос, система посылает широковещательный *запрос на определение маски адреса* (Address Mask Request).

Ответ должен быть получен от сервера, авторизированного для управления маской адреса сервера. Обычно в качестве такого сервера применяется маршрутизатор, но может использоваться и хост. В ответе в полях сети и подсети установлены единицы, определяя 32-разрядное поле маски адреса!

Сервер маски адреса может быть сконфигурирован так, что, даже при отключении от сети на какое-то время, он будет далее передавать широковещательные сообщения *Address Mask Reply*, как только станет активным. Это предоставляет шанс на получение нужной информации системам, которые были запущены в то время, когда сервер был неактивен.

На рисунке 2.10 показан формат *запроса маски адреса* и *ответа* на него. Тип 17

применяется для запроса, а тип 18 —для ответа. В общем случае можно игнорировать идентификатор и последовательный номер.

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Type = 17 или 18	Code	Контрольная сумма
Идентификатор		Последовательный номер
Маска адреса		

Рис. 2.10 Формат ICMP-сообщений Address Mask

На практике более предпочтительный метод определения маски адреса предоставляют протоколы загрузки, например *Dynamic Host Configuration Protocol* или *BOOTP*. Эти протоколы более эффективны, поскольку обеспечивают полный набор конфигурационных параметров. Кроме того, операции выполняются более точно, в том числе и некорректные.

Временная метка и ответ на Timestamp

Сообщение с ответом на *Timestamp*предоставляет сведения о времени в системе. Оно предназначено для оценки буферизации и обработки датаграммы на удаленной системе. Отметим следующие поля:

Originate time stamp (исходная временная метка)	Время последнего обращения к сообщению в системе-отправителе.
Receive time stamp временная метка получения)	Время первого обращения к сообщению отвечающей системы.
Transmit time stamp (временная метка пересылки)	Время последнего обращения к сообщению отвечающей системы.

По возможности, возвращаемое время должно измеряться в миллисекундах-относительно полуночи по универсальному времени (UniversalTime), которое ранее называлось временем по Гринвичу (GreenwichMeanTime). Большинство реализаций реально возвращает одно и то же время в полях *Receive time stamp* и *Transmit time stamp.*

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Type = 13 или 14	Code	Контрольная сумма
Идентификатор		Последовательный номер
Originate timestamp		
Receive timestamp		
Transmittimestamp		

Рис. 2.11 Формат ICMP-сообщений запросов и ответов о временной метке

Протокол ICMP обеспечивает очень простой способ синхронизации систем по времени. Однако это несколько грубая синхронизация, поскольку на нее влияют задержки в сети. Существует более совершенный *протокол сетевого времени* (Network Time Protocol), который был разработан для синхронизации по времени в Интернете.

Тип 13 используется для *запросов,* а 14 — для *ответов.* Формат сообщения представлен на рисунке 2.11.