Network Working Group
Request for Comments: 1191
Obsoletes: RFC 1063

J. Mogul
DECWRL
S. Deering
Stanford University

November 1990

Path MTU Discovery

Статус этого документа

Этот документ определяет стандарт IAB Standards Track для сообщества Internet, и требует обсуждения и внесения усовершенствований. Для выяснения состояния стандартизации и статуса данного протокола необходимо обратиться к "IAB Official Protocol Standards". Распространение данного документа не ограничено.

Оглавление

Аннотация

Данный документ описывает методику определения maximum transmission unit (MTU) на произвольном маршруте в Internet. Он определяет небольшие изменения для маршрутизаторов, генерирующих один тип ICMP сообщений. Для маршрута, пролегающего через маршрутизатор, который не был изменен, может методика может выдавать не верное значение MTU.

Благодарности

Это документ является продуктом IETF MTU Discovery Working Group. Механизм, предложенный здесь был первоначально предложен Джефом Купером (Geoff Cooper) [2], который в двух коротких абзацах изложил все основные идеи, потребовавшие от рабочей группы месяцев работы.

1 Введение.

Когда один IP хост имеет большой объем данных, которые требуется отправить другому хосту, то данные передаются как серии IP дейтаграмм. Желательно, чтобы дейтаграммы были максимального размера, не требующего фрагментации и на одном из участков маршрута от отправителя к получателю. Данный размер дейтаграммы далее упоминается как Path MTU (PMTU), и равен наименьшему из MTU, используемых на каждом хопе маршрута. Недостатком IP является отсутствие стандартного механизма для определения PMTU на произвольном маршруте.

Замечание: Path MTU, это тоже, что в [1] именуется как «Эффективный MTU для отправки» ("Effective MTU for sending" (EMTU_S).) PMTU ассоциируется с маршрутом, который является специфической комбинацией IP адресов отправителя и получателя и возможно типом обслуживания Type-of-service (TOS).

В существующей практике [1] используется меньшее из 576 и MTU для первого хопа в качестве PMTU для любого адресата, который не подключен к той же сети или подсети, что и источник. Во многих случаях результатом является использование размеров дейтаграмм, меньших, чем необходимо, потому что в большинстве случаев маршруты имеют PMTU большее, чем 576. Хост, посылающий дейтаграммы намного меньшие, чем PMTU напрасно тратит ресурсы Internet и возможно достигает производительности, близкой к оптимальной. Кроме того, существующая практика не

предотвращает полностью фрагментацию дейтаграмм, так как существуют тракты передачи с PMTU меньшим, чем 576.

Ожидается, что будущие протоколы маршрутизации будут способны предоставлять точную информацию о PMTU в пределах своей области маршрутизации, возможно за исключением многоуровневых маршрутизирующих иерархиях. Не понятно, на сколько скоро они станут доступны, так что в течение следующих нескольких лет Internet нуждается в простом механизме определения PMTU без перерасхода ресурсов и до того, как все хосты и маршрутизаторы будут модернизированы.

2 Краткий обзор протокола.

В этом документе мы описываем технику, использующую бит Don't Fragment (DF) в заголовке IP дейтаграммы для динамического определения PMTU маршрута. Основная идея состоит в том, что хост использует в качестве PMTU значение MTU первого хопа для данного маршрута. Все дейтаграммы отправляются с установленным флагом DF. Если какая-то дейтаграмма слишком велика для того, чтобы быть отправленной одним из маршрутизаторов дальше без фрагментации, то данный маршрутизатор отбрасывает эту дейтаграмму и посылает ICMP сообщение Destination Unreachable с кодом «fragmentation needed and DF set» [7]. Поле того, как хост получил такое сообщение (в дальнейшем называемое «дейтаграмма слишком большая»), хост уменьшает значение PMTU для этого маршрута.

Процесс определения PMTU заканчивается после того, как хост оценит PMTU достаточно маленьким, для того чтобы отправлять дейтаграммы без фрагментации.

Хост может закончить процесс определения PMTU, прекратив устанавливать флаг DF в заголовке дейтаграммы. Это может произойти, например, если хост считает допустимым отправлять фрагментированные дейтаграммы в данных обстоятельствах. В нормальных условиях хост должен отправлять все дейтаграммы с установленным флагом DF, так что если маршрут изменится и новый PMTU станет меньше, то это будет обнаружено.

К сожалению, сообщение «дейтаграмма слишком большая», в том виде, в каком она определена в настоящее время не сообщает MTU для того хопа, перед которым была отброшена дейтаграмма, являющаяся слишком большой. Таким образом хост отправитель дейтаграммы точно не может определить, на сколько следует уменьшит значения PMTU. Чтобы исправить это, мы предлагаем использовать неиспользуемое поле в заголовке сообщения «дейтаграмма слишком большая» для переноса значения MTU для следующего хопа. Это единственное требование к маршрутизаторам в поддержку определения PMTU.

PMTU может изменяться с течением времени, из-за изменений в топологии маршрутизации. Уменьшения PMTU детектируются с помощью сообщения «дейтаграмма слишком большая», за исключением тех случаев, когда хост прекратил устанавливать бит DF. Чтобы детектировать увеличение PMTU, хост периодически увеличивает значение MTU. В большинстве случаев результатом является отбрасывание дейтаграммы и отправка сообщения «дейтаграмма слишком большая», потому что в большинстве случаев значение PMTU не меняется.

Так как этот механизм по существу гарантирует, что хост не будет получать никаких фрагментов дейтаграмм от хоста осуществляющего определения PMTU, это может помогать во взаимодействии хостов, которые не могут производить сборку фрагментированных дейтаграмм.

3 Требования к хосту.

Когда хост получает сообщение «дейтаграмма слишком большая», он **ДОЛЖЕН** уменьшить значение PMTU в соответствии с полем Next-Hop MTU (см. п.4). Мы не определяем точное поведение хоста в этих обстоятельствах, так как различные приложения могут иметь различные требования и различные архитектурные реализации могут иметь различные полезные стратегии.

Мы требуем, чтобы после получения сообщения «дейтаграмма слишком большая» хост **ДОЛЖЕН** пытаться избегать появления большого количество таких сообщений в ближайшем будущем. Хост может или прекратить уменьшать размеры дейтаграмм или прекратить устанавливать флаг DF в заголовке дейтаграмм. Понятно, что формирователь стратегии может продолжать выявлять сообщения «дейтаграмма слишком большая», но каждое из этих сообщений (так же как и отброшенные дейтаграммы) потребляют ресурсы Internet, поэтому хост **ДОЛЖЕН** заставить процесс определения PMTU сходиться.

Хост использующей определение MTU **ДОЛЖЕН** реагировать на уменьшение MTU настолько быстро, насколько это возможно. Хост МОЖЕТ определять увеличение PMTU, но так как это требует посылки дейтаграмм большего размера, чем действующий PMTU, и потому что PMTU вероятнее всего не будет увеличиваться, то хост **ДОЛЖЕН** делать это не очень часто. Попытка детектировать увеличение PMTU (посылкой дейтаграммы большего размера, чем действующее значение PMTU) **ДОЛЖНА** быть не раньше, чем через 5 минут после получения сообщения «дейтаграмма слишком большая» для данного адресата, или не раньше чем через 1 минуту после того, как было предпринято последнее успешное увеличение PMTU. Мы рекомендуем установить значения этих таймеров в два раза больше, чем эти минимальные значения (10 минут и 2 минуты соответственно).

Хост должен быть способен работать с сообщениями «дейтаграмма слишком большая» не содержащими информации об MTU на следующем хопе, в связи тем, что не выполнимо выполнить модернизацию всех маршрутизаторов Internet за конечное время. Сообщения «дейтаграмма слишком большая» от не модернизированного маршрутизатора должно определяться по наличию нуля в (новом) поле Next-Hop MTU. (Этого требует спецификация протокола ICMP [7] – неиспользуемые поля должны быть заполнены нулями). В разделе 5 мы обсуждаем возможные стратегии для хоста, отрабатывающему сообщение «дейтаграмма слишком большая» в старом стиле (полученном от не модернизированного маршрутизатора).

Хост **НИКОГДА НЕ ДОЛЖЕН** принимать значение PMTU меньше чем 68 октетов.

Хост **НЕ ДОЛЖЕН** увеличивать значение PMTU в ответ на содержимое сообщения «дейтаграмма слишком большая». Сообщение предполагающее увеличение PMTU является либо старой дейтаграммой, плавающей по Internet, либо фальшивый пакет, являющийся частью атаки на отказ в обслуживании, либо результатом множественности маршрутов до адресата.

3.1 Опция TCP MSS

Хост, производящий определение PMTU должен придерживаться правила, что он не должен посылать дейтаграммы более 576 октетов, если он не имеет разрешения получателя на это. Для TCP соединения это означает, что хост не должен посылать дейтаграммы больше чем 40 октетов плюс максимальный размер сегмента (Maximum Segment Size-MSS) посланный его партнером.

Замечание: TCP MSS определен как допустимый размер дейтаграммы минус 40 [9]. Значение по умолчанию 576 октетов как максимальный размер дейтаграммы определяет размер по умолчанию для TCP MSS в 536 октетов.

Раздел 4.2.2.6 «Требования для хоста Internet-коммуникационный уровень» [1] говорит:

> Некоторые реализации TCP, посылают опцию MSS только если хост получатель подключен к другой посети. Конечно, в целом уровень TCP может не иметь соответствующей информации для принятия этого решения, так что предпочтительно оставить уровню IP задачу определения подходящего MTU для маршрута.

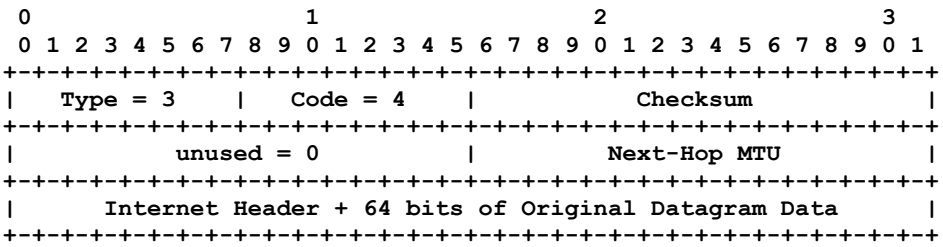
Фактически, многие реализации TCP всегда посылают опцию MSS, однако устанавливают это значение в 536, если адресат не является локальным. Это поведение было правильным, когда Интернет был полон хостов, не придерживавшихся данного правила, что дейтаграммы больше 576 октетов не должны отправляться на нелокальные компьютеры. Теперь, когда большинство хостов следуют этому правилу, ограничение опции TCP MSS не является необходимым.

Кроме того, выполнение этого правила делает неспособным механизм PMTU Discovery определять PMTU более 576, так что хосты больше **ДОЛЖНЫ** умешать значение, посылаемое в опции MSS. Значение MSS должно быть на 40 байт меньше, чем размер самой большой дейтаграммы, которую хост способен пере собрать (MSS_R, как определено в [1]). Во многих случаях будет иметь место архитектурное ограничение в 65495 (65535-40). Хост может посылать значение MSS полученное из MTU присоединенной сети; это не должно вызывать проблемы с PMTU Discovery.

Замечание: В настоящее время мы не видим никаких причин посылать MSS больше, чем максимальный MTU присоединенной сети, и мы не рекомендуем использовать 65495. Весьма возможно, что некоторые реализации IP имеют ошибки со знаковым битом, что делает опасным использование такого большого MSS.

4 Спецификация маршрутизатора.

Когда маршрутизатор не может переслать дейтаграмму, потому что она превышает размеры MTU на следующем хопе и у нее в заголовке установлен бит Don't Fragment, он должен будет выслать источнику дейтаграммы ICMP сообщение «Адресат недостижим» с кодом, обозначающим «требуется фрагментация и установлен флаг DF». Для поддержки техники Path MTU Discovery, описываемой в данном документе, маршрутизатор **ДОЛЖЕН** вставить значение MTU сети на следующем хопе в младшие 16 бит поля ICMP заголовка, которое помечено как неиспользуемое.



Значение, которое несет поле Next-Hop MTU это:

> Размер в октетах самой большой дейтаграммы, которая может быть отправлена по пути оригинальной дейтаграммы, без фрагментации на этом маршрутизаторе. Размер включает IP заголовок и IP данные и не включает низкоуровневые заголовки.

Это поле никогда не будет содержать значение меньше чем 68, потому что каждый маршрутизатор «должен быть способен отправлять дейтаграммы длинной по 68 октетов без фрагментации» [8].

5 Обработка сообщений в старом стиле на хосте

В этом разделе мы выделим несколько возможных стратегий для хоста, получающего сообщение «дейтаграмма слишком большая» от не модифицированного маршрутизатора (т.е. с нулевым полем Next-Hop MTU). Этот раздел не является частью описание протокола.

Самая простая вещь, которую может сделать хост в ответ на такое сообщение – это прекратить установку бита DF и принять, что это PMTU равно текущему PMTU или 576, в зависимости от того, что меньше. Таким образом, хост возвращается к тому же самому PMTU, как это делается в текущей практике (см. раздел 3.3.3. «Рекомендации для хостов Internet-коммуникационные уровни» [1]). Эта стратегия имеет преимущество, так как она отрабатывает быстро, что делает ее не хуже, чем там та, что используется в существующей практике. Она не работает, однако, для предотвращения фрагментации в некоторых случаях, и дает наиболее эффективное использование утилизации в других случаях.

Более сложные стратегии «ищут» точную оценку PMTU, посылая пакеты разного размера с установленным битом DF. Хорошей поисковой стратегией можно назвать ту стратегию, которая получает точную оценку PMTU при малом количестве потерянных пакетов в процессе поиска.

Некоторые стратегии применяют алгоритмические функции к предыдущей оценке PMTU для того чтобы получить следующую оценку PMTU. Например, старая оценка PMTU может быть умножена на константу (скажем, 0.75). Мы НЕ рекомендуем их; они сходятся слишком долго и могут недооценить правильное значение PMTU.

Более сложный подход делать бинарный поиск по размеру пакета. Он сходится несколько быстрее, хотя все еще требуется 4-5 шагов для того чтобы перейти от FDDI MTU к Ethernet MTU. Серьезный недостаток состоит в том, что требуется сложная реализация алгоритма индикации занижения текущей оценки PMTU. Мы так же не рекомендуем пользоваться этой стратегией.

Появилась одна стратегия, которая работает, кажется, весьма хорошо. Она стартует с обзора значений, которые практически в относительно небольшом количестве используются в Internet. Таким образом, вместо того, чтобы слепую проводить поиск среди произвольно выбранных значений, мы можем искать только среди тех, которые действительно могут использоваться. Более того, так как у разработчиков прослеживается тенденция выбирать MTU подобными путями, возможно накопление групп похожих значений MTU и использовать наименьшее значение в группе как наше поисковое «плато». (Понятно, что лучше недооценить значение MTU на несколько процентов, чем переоценить его на один октет).

В разделе 7, мы описываем, как мы пришли к таблице репрезентативных плато MTU (representative MTU plateaus) для использования при оценке MTU. Сходимость с этой таблицей такая же хорошая, как бинарный поиск в самом худшем случае и намного лучше в обычном случае (например, требуется только два шага для перехода от FDDI MTU к Ethernet MTU).

Любая поисковая стратегия должна иметь некоторую «память» о предыдущих оценках, чтобы делать следующие. Один подход должен использовать оценки PMTU, кэшированные в настоящее время, но фактически лучше информация, полученная из сообщения «дейтаграмма слишком большая». Все ICMP сообщения «адресат недоступен» содержат, включая упомянутое, IP заголовок оригинальной дейтаграммы, которая была слишком велика для передачи без фрагментации. Так как поле Total Length может быть меньше, чем текущее PMTU, но, тем не менее, больше, чем фактическое PMTU, то это может послужить входными данными методу, производящему следующую оценку PMTU.

Замечание: маршрутизаторы, основанные на 4.2BSD Unix, посылают неправильное значения поля Total Length оригинальной дейтаграммы. Значение, посылаемое этими маршрутизаторами, является суммой оригинального поля Total Length и Header Length (выраженное в октетах). Так как хост может получить такое сообщение «дейтаграмма слишком большая», зная, что дейтаграмма была получена от одного из таких маршрутизаторов, хост должен быть консервативен. Если возвращенное поле Total Length не меньше, чем текущая оценка PMTU, то оно должно быть уменьшено до четверти значения поля Header Length.

Стратегию, которую мы рекомендуем, состоит в том, чтобы использовать следующую оценку PMTU как наибольшее основное значение, которое меньше, чем то, что возвращается в поле Total Length (исправленное, если необходимо, в соответствии с замечанием выше).

6 Реализация хоста.

Этот раздел осуждает реализацию PMTU Discovery в программном обеспечении хоста. Это - не спецификация, а скорее набор предложений.

Раздел рассматривает следующие вопросы:

- Какой уровень или уровни включают в себя реализацию PMTU Discovery;
- Где информация PMTU будет кэширована?
- Как устаревшая PMTU информация будет удаляться?
- Что должны делать транспортный уровень и верхние уровни?

6.1 Иерархическое представление

В архитектуре протокола IP решение о том, какой должен быть размер отправляемого пакета производится протоколами, находящимися на уровне выше уровня IP. Мы будем называть такие протоколы протоколами пакетирования. Протоколы пакетирования обычно являются транспортными протоколами (например TCP), но они могут быть протоколами более высокого уровня (например, протоколы построенные поверх UDP).

Реализация PMTU Discovery на уровне протоколов пакетирования упрощает решение некоторых межуровневых проблем, но имеет несколько недостатков: реализация возможно должна будет переделана для каждого реализации протокола, это усложнит совместное использование информации PMTU между разными уровнями пакетирования и состояние ориентированное на соединение, поддерживаемое некоторыми уровнями пакетирования могут с трудом расширяться для сохранения информации PMTU на длительные периоды.

По этой причине мы полагаем, что уровень IP должен хранить информацию PMTU и что уровень ICMP должен обрабатывать сообщения «дейтаграмма слишком большая». Уровни пакетирования должны оставаться способными реагировать на изменения PMTU, изменением размера дейтаграмм, которые они посылают и должны быть способны определить те дейтаграммы, которые посланы с установленным флагом DF. Мы не хотим, чтобы уровень IP просто установил DF бит в каждом пакете, так как возможно, что уровень пакетирования, возможно приложение UDP вне ядра, неспособно изменить этот размер дейтаграммы. Протоколы, вовлеченные во внутреннюю фрагментацию, пускай не элегантные, бывают иногда успешными (NFS будет первым примером) и мы не хотим нарушать их работу.

Для поддержки иерархического представления, протоколы пакетирования требуют расширения сервиса IP описанного в [1]:

Способ изучения изменений в значении MMS_S ("maximum send transport-message size") это вычитание из PMTU минимального размера IP заголовка.

6.2 Сохранение информации PMTU

В общем, уровень IP должен ассоциировать каждое значение PMTU, которое было им получено, с определенным путем. Путь идентифицируется с адресом отправителя, адресом получателя и типом сервиса IP. (Некоторые реализации не записывают адрес источника, это применимо для хостов с одним интерфейсом, которые имеют только один возможный адрес отправителя).

Замечание: Некоторые пути могут быть далее отличены по различным классификациям защиты. Детали таких классификаций не рассматриваются в данном документе.

Очевидное место для хранения этой ассоциации - поле в таблице маршрутизации. Хост не будет иметь маршрута для любого возможного адресата, но он будет иметь возможность кэшировать маршрут на хост для каждого активного адресата. (Это требование уже налагается необходимость обрабатывать ICMP сообщения по перенаправлении).

Когда первый пакет послан на хост и отсутствует маршрут на хост (per-host route), то маршрут выбирается и маршрутов на сеть (per-network routes) или из набора маршрутов по умолчанию. Поля PMTU в этих маршрутах должны быть инициализированы значением MTU для первого хопа и не должны изменяться в процессе определения PMTU. (При определении PMTU создаются или изменяются маршруты на хост). До того, как получено сообщение «дейтаграмма слишком большая» значение, соответствующее изначально выбранному маршруту, является точным.

После того как получено сообщение «дейтаграмма слишком большая», уровень ICMP определяет новую оценку PMTU (или из ненулевого значения MTU следующего хопа или с помощью методов, описанных разделе 5). Если маршрут на хост для этого пути не существует, то он создается (почти как при обработке ICMP Redirect; новый маршрут использует тот же маршрутизатор, как и текущий маршрут). Если оценка PMTU ассоциированная с маршрутом на хост выше, чем новая оценка, то ее значение изменяется.

Уровни пакетирования должны быть извещены об уменьшении PMTU. Любой уровень пакетирования (например, TCP соединение) который активно использует путь, должен быть уведомлен об уменьшении оценки PMTU.

Замечание: когда сообщение «дейтаграмма слишком большая» содержит заголовок исходной дейтаграммы, который ссылается на UDP пакет уровень TCP должен быть уведомлен, если любое из этих соединений использует этот путь.

Также, протокол, который послал дейтаграмму, вызвавшую появление сообщения «дейтаграмма слишком большая», должен быть уведомлен, что эта дейтаграмма была отброшена, даже если оценка PMTU не изменилась, так чтобы он мог вызвать повторную отправку отброшенной дейтаграммы.

Замечание: механизм уведомления должен быть аналогичен механизму, используемому для предоставления уведомлений ICMP Source Quench. В некоторых реализациях (таких как системы, построенные на основе BSD 4.2) существующий механизм уведомлений не способен идентифицировать вовлеченное TCP соединение, что требует наличия дополнительного механизма.

В качестве альтернативы, реализация может избегать использования асинхронного механизма уведомления об уменьшении PMTU, откладывая уведомление до следующей попытки послать дейтаграмму большее, чем оценка PMTU. При таком подходе, когда сделана попытка ПОСЛАТЬ дейтаграмму с установленным битом DF, и дейтаграмма больше чем оценка PMTU, функция SEND должна закончиться сбоем и вернуть соответствующий признак ошибки. Этот подход может быть более подходящим для уровня пакетирования без установления соединения (какие как UDP), который (в некоторых реализациях) трудно «уведомлять» от уровня ICMP. В этом случае, обычные механизмы, основанные на тайм-аутах, использовались бы, чтобы восстановить отброшенные дейтаграммы.

Важно понять, что уведомление уровней пакетирования, использующих путь для изменения PMTU, отличается от уведомления специфичных уровней, о том, что пакет был отброшен. Последнее должно быть сделано только практически (т.е. асинхронно с точки зрения уровня пакетирования), в то время как первое может быть отсрочено до того момента, когда уровень пакетирования будет создавать пакет. Повторная передача должна быть сделана только для тех пакетов, о которых известно, что они отброшены, на что указывает сообщение «дейтаграмма слишком большая».

6.3 Удаление устаревшей информации PMTU.

Сетевая топология является динамической, маршруты меняются через некоторое время. PMTU, обнаруженный для данного адресата может быть неправилен, если начинает использоваться новый маршрут. Таким образом, PMTU информация, кэшируемая хостом, устаревает.

Сетевая топология является динамической, маршруты меняются через некоторое время. PMTU, обнаруженный для данного адресата может быть неправилен, если начинает использоваться новый маршрут. Таким образом, PMTU информация, кэшируемая хостом, устаревает.

Замечание: реализация должна предоставлять средства для изменения длительности таймаута, включая установку на «бесконечность». Например, хост подключенный к сети FDDI, которая в свою очередь подключена к Internet через медленную последовательную линию никогда не начнет процесс определения нового нелокального MTU, так что ему не придется сталкиваться с отброшенными дейтаграммами каждые 10 минут.

Верхний уровень не должен передавать дейтаграммы в ответ на увеличение оценки PMTU, так как это повышение никогда не вызовет отбрасывание дейтаграмм.

Один из подходов к реализации старения PMTU это добавление поля временной метки в таблицу маршрутизации. Это поле инициализируется «зарезервированным» значением, показывающим, что это значение PMTU никогда не изменялось. Всякий раз, когда значение PMTU было уменьшено в ответ на сообщения «дейтаграмма слишком большая» временная метка устанавливается на текущее время.

Спустя минуту, управляемая таймером процедура пробежит по таблице маршрутизации и для каждого маршрута значение временной метки, которого не является «резервным» и больше чем значение таймаута:

- Установит значение PMTU разное значению PMTU на первом хопе.
- Известит уровень пакетирования, использующий данный маршрут об увеличении.

Оценка PMTU может исчезнуть из таблицы маршрутизации, если маршрут на хост был удален из таблицы; это может случиться в ответ на сообщение ICMP Redirect или потому, что некоторые демоны маршрутной таблицы удаляют старые маршруты после нескольких минут. Также, на хостах имеющих несколько интерфейсов изменения топологии может быть результатом использования разных интерфейсов. Когда это случается, если уровень пакетирования не оповещен, тогда может продолжиться использование кэшированных значений PMTU, которые теперь слишком малы. Одно решение состоит в том, чтобы уведомлять уровень пакетирования о возможном изменении значения PMTU, всякий раз, когда сообщение о перенаправлении вызывает изменение маршрута и всякий раз, когда маршрут просто удален из таблицы маршрутизации.

Замечание: более сложные методы для определения увеличения значения PMTU описаны в разделе 7.1.

6.4 Действия уровня TCP.

Уровень TCP должен отслеживать значение PMTU для адресата, он не должен посылать дейтаграммы размером больше PMTU. Простая реализация может узнавать это значение у уровня IP (используя интерфейс GET_MAXSIZES описанный в [1]) каждый раз, когда создает сегмент, но это было бы не эффективно. Кроме того, реализации TCP, которые используют алгоритм медленного старта [4] обычно вычисляют и кэшируют несколько значений, полученных из PMTU.Может быть проще получить асинхронное уведомление, полученное при изменении PMTU, так, чтобы эти переменные могли быть модифицированы.

Реализации TCP должны так же сохранять значение MSS (которое по умолчанию равно 536), и не посылать сегменты больше этого MSS, не зависимо от PMTU.

В реализациях основанных на BSD 4.x это требует добавления дополнительного поля в запись состояния TCP.

Наконец, когда получено сообщение «дейтаграмма слишком большая», подразумевается, что дейтаграмма была отброшена маршрутизатором, который послал это сообщение. Достаточно обработать это как любой другой отброшенный сегмент и ждать, пока таймер повторной передачи не истечет, чтобы вызвать повторную передачу сегмента. Если процесс определения PMTU требует несколько шагов для правильной оценки PMTU, это может задержать соединение во много раз.

С другой стороны, повторная передача могла быть сделана в непосредственном ответе на уведомление о том, что значение PMTU было изменено, но только для определенного подключения, указанного в сообщении «дейтаграмма слишком большая». Размер дейтаграммы используемый при повторной передаче должен быть, конечно, не больше чем новый PMTU.

Замечание: **НЕЛЬЗЯ** повторно передать в ответ на каждое сообщения «дейтаграмма слишком большая», так как выброс сегментов с завышенными размерами приведет к нескольким повторным передачам тех же самых данных. Если новая оценка PMTU все еще не верна, процесс повторится, что приведет к экспоненциальному росту числа отправленных лишних сегментов.

Это означает, что уровень TCP должен быть способен распознать когда сообщение «дейтаграмма слишком большая» действительно уменьшает PMTU, который уже был использован для отправки дейтаграмм на данном соединении и что нужно игнорировать другие уведомления.

Современные реализации TCP используют алгоритмы «медленного старта» и «предотвращения скоплений» («congestion avoidance») для увеличения эффективности работы. [4]. В отличие от повторной передачи, вызванной истечением таймером повторной передачи, повторная передача, вызванная сообщением «дейтаграмма слишком большая» не должна менять окно накопления (congestion window). Это должно, однако, вызывать срабатывание механизма медленного старта (т.е. только один сегмент должен быть повторно передан, пока не подтверждения не начнут поступать снова).

Эффективность работы TCP может быть уменьшена, если максимальное окно отправителя не кратно используемому размеру сегмента (это не размер окна накопления, который всегда является кратным размеру сегмента). Во многих системах, (например, основанных на BSD 4.2) размер сегмента обычно кратен 1024 октетам, так что надлежащие отношения уставятся по умолчанию. Если используется определение PMTU, то конечно размер сегмента может не быть кратен посылаемому пространству (may not be a submultiple of the send space) и он может изменяться в период существования соединения; это означает, что уровень TCP может быть нуждается в изменении размера передающего окна, когда PMTU меняет свое значение. Максимальный размер окна должен быть установлен равным наибольшему кратному размеру сегмента (PMTU - 40), что должно быть меньше или равно размеру буфера пространства отправителя (sender's buffer space size).

Процесс определения PMTU не должен воздействовать на значение, отправляемое у опции MSS, потому что это значение используется на другом конце соединения, которое может использовать несвязанное значение PMTU.

6.5 Проблемы для других транспортных протоколов.

Некоторые транспортные протоколы (такие как ISO TP4 [3]) не позволяют перепакетирование (repacketize) во время повторной отправки. То есть, должна быть сделана попытка передачи дейтаграммы определенного размера и ее содержимое не может быть разбито на маленькие дейтаграммы для повторное передачи. В таком случае, оригинальная дейтаграмма должна быть повторно передана без установленного бита DF, позволяющего фрагментировать дейтаграмму при необходимости для того, чтобы достичь адресата. Последующие дейтаграммы, когда они передаются в первый раз, должны быть не больше, чем PMTU и должны иметь установленный бит DF.

Sun Network File System (NFS) использует протокол вызова удаленных процедур (Remote Procedure Call - (RPC) protocol) [11], во многих случаях посылающий дейтаграммы, которые должны быть фрагментированы даже на первом хопе. Это может улучшить эффективность работы с некоторых случаях, но как известно вызывает проблемы с надежностью и производительностью, особенно когда клиент и сервер разделены маршрутизаторами.

Мы рекомендуем использовать механизм определения PMTU в реализациях NFS всякий раз, когда в процесс обмена вовлечены маршрутизаторы. Большинство реализаций NSF позволяют изменять размер RPC дейтаграммы во время монтирования (mount-time) (косвенно изменяя эффективный размер блока файловой системы) но могут требовать некоторых модификаций для поддержки изменений в дальнейшем.

Также, поскольку одиночные операции NFS на могут быть разбиты на несколько UDP дейтаграмм, некоторое операции (прежде всего те, которые оперируют с именами файлов и директорий) требуют минимального размера дейтаграммы, который может быть больше, чем PMTU. Реализации NFS не должны уменьшать размер дейтаграмм ниже этого порога, даже если механизм определения PMTU предлагает меньшее значение. (Конечно, в этом случае дейтаграммы не должны быть посланы с установленным флагом DF).

6.6 Интерфейс управления.

Мы предлагаем, чтобы реализация предоставляла для системных утилит путь:

- определять на каком маршруте будет выполняться определение PMTU;
- изменять значение PMTU, ассоциированное с данным маршрутом.

Формирователь может быть выполнен путем связывания флага с маршрутом, когда пакет передается через этот маршрут с установленным флагом, уровень IP оставляет бит DF скинутым не зависимо от того, что требует верхний уровень.

Эти особенности могут использоваться для работы через сеть с аномальной ситуацией или реализацией протокола маршрутизации, который способен получать значение PMTU.

Реализации должны также предоставлять путь для изменения таймаута старения информации PMTU.

7 Вероятные значения для PMTU.

Алгоритмы, рекомендованные в разделе 5 для «поиска» пространства PMTU основаны на таблице значений, которая строго ограничивает поисковое пространство. Мы описываем здесь таблицу значений PMTU, которые представляют все главные технологии, используемые в Internet.

В таблице 7-1, данные представлены в порядке уменьшения PMTU, и сгруппированы так, что каждый набор одинаковых MTU, ассоциирован с «плато», равному наименьшему MTU в группе. Когда плато представляет больше одного MTU, таблица показывает максимальное отклонение в процентах, связанное с этим плато.

Мы не ожидаем, что значения в таблице, особенно для верхних уровней MTU, будут правильными всегда. Значения, находящиеся в таблице являются предложением для разработчиков, а НЕ спецификацией или требованием. Разработчики должны использовать актуальные ссылки для сбора наборов плато, это важно, поскольку таблица не содержит слишком много значений и поэтому процесс поиска PMTU может тратить впустую ресурсы Internet. Для клиентов, не имеющих исходных текстов операционных систем, разработчики должны предусмотреть удобный метод обновления таблиц в из системах (например, в ядрах Unix, основанных на BSD, таблица может быть изменена с помощью новой команды «ioctl»).

Замечание: возможно, является хорошей идеей добавить несколько значений в таблицу равных малым степеням числа 2 плюс 40 (для IP и TCP заголовка), если таких значений там не существует, так как кажется это резонный непроизвольный путь выбрать произвольные значения.

Эта таблица может также содержать записи для значений немного меньших, чем большие степени 2, в этом случае MTU определено около этих значений (в это случае для записей таблицы лучше быть маленькими, чем быть большими, иначе следующее самое маленькое плато будет выбрано вместо этого).

7.1 Наилучший способ определить увеличение PMTU.

Раздел 6.3 предлагает детектировать увеличение PMTU путем периодического повышения оценки PMTU до MTU на первом хопе. Вероятнее всего, этот процесс будет просто «переопределять» текущую оценку PMTU, ценой нескольких отброшенных дейтаграмм, поэтому это не должно выполняться часто.

Лучший подход это периодически повышать оценку PMTU до следующего самого высокого значения PMTU в таблице плато (или до MTU на первом хопе, если оно меньше). Если увеличенного значение не верно, то перед тем как будет найдено верное значение, будет затрачено время порядка времени прохождения пакета туда-обратно. Если увеличенное значение все еще слишком мало, то большее значение будет попробовано несколько позже.

Поскольку может требоваться несколько таких периодов для обнаружения существенного увеличения PMTU, мы рекомендуем этот короткий таймаут использовался после увеличения оценки, и более длинный тайм-аут использовался после ее уменьшения, произошедшего в результате получение сообщения «дейтаграмма слишком большая».

Табл. 7-1: Распространенные MTU

Плато	MTU	Комментарий
65535	65535 65535	Официальный максимум MTU Hyperchannel
32000		На всякий случай
17914	17914	16Mb IBM Token Ring
8166	8166	IEEE 802.4
4352 (1%)	4464	IEEE 802.5 (4Mb max)
	4352	FDDI (Revised)
2002 (2%)	2048	Wideband Network
	2002	IEEE 802.5 (4Mb recommended)
1492 (3%)	1536	Exp. Ethernet Nets
	1500	Ethernet Networks
	1500	Point-to-Point (default)
	1492	IEEE 802.3
1006	1006	SLIP
	1006	ARPANET
508 (13%)	576	X.25 Networks
	544	DEC IP Portal
	512	NETBIOS
	508	IEEE 802/Source-Rt Bridge
	508	ARCNET
296	296	Point-to-Point (low delay)
68		Официальный минимум MTU

Например, после того как оценка PMTU была уменьшена, таймаут должен быть установлен в 10 мин. После того как этот интервал истек и было попробовано большее значение, таймаут может быть установлен гораздо более короткий (скажем 2 минуты). Ни в коем случае таймаут не должен быть короче, чем время на пересылку пакета туда-обратно, если оно известно.

8 Вопросы безопасности

Механизм PMTU делает возможным два вида атак на отказ в обслуживании, оба основанные на посылке ложных сообщений «дейтаграмма слишком большая» злонамеренной стороной.

В первой атаке, фальшивые сообщения «дейтаграмма слишком большая» показывают, что PMTU намного меньше реальной. Это не должно полностью остановить весь поток данных, поскольку атакуемый хост никогда не будет использовать PMTU меньше абсолютного минимума, но 8 октетов данных на одну дейтаграмму замедлят обмен.

Во второй атаке фальшивые сообщения указывают PMTU больший, чем в реальности. Это может вызвать временную блокаду, так как дейтаграммы посылаемые атакуемым хостом будут отброшены одним из маршрутизаторов. В течение одного периода, равного по длительности, времени необходимому для пересылки пакета туда-обратно, хост будет исправлять ошибку (получив сообщение «дейтаграмма слишком большая» от этого маршрутизатора). Частое повторение этой атаки приведет к увеличению дейтаграмм, которые будут отброшены. Хост, конечно, не должен повышать оценку PMTU, основываясь на сообщении «дейтаграмма слишком большая», чтобы не быть подверженным данной атаке.

Злонамеренная сторона могла бы создавать проблемы, если бы могла остановить обработку истинных сообщений «дейтаграмма слишком большая», но имеются более простые атаки на отказ в обслуживании.

Ссылки

[1] R. Braden, ed. Requirements for Internet Hosts -- Communication Layers. RFC 1122, SRI Network Information Center, October, 1989.

[2] Geof Cooper. IP Datagram Sizes. Electronic distribution of the TCP-IP Discussion Group, Message-ID <8705240517.AA01407@apolling.imagen.uucp>.

[3] ISO. ISO Transport Protocol Specification: ISO DP 8073. RFC 905, SRI Network Information Center, April, 1984.

[4] Van Jacobson. Congestion Avoidance and Control. In Proc. SIGCOMM '88 Symposium on Communications Architectures and Protocols, pages 314-329. Stanford, CA, August, 1988.

[5] C. Kent and J. Mogul. Fragmentation Considered Harmful. In Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology. August, 1987.

[6] Drew Daniel Perkins. Private Communication.

[7] J. Postel. Internet Control Message Protocol. RFC 792, SRI Network Information Center, September, 1981.

[8] J. Postel. Internet Protocol. RFC 791, SRI Network Information Center, September, 1981.

[9] J. Postel. The TCP Maximum Segment Size and Related Topics. RFC 879, SRI Network Information Center, November, 1983.

[10] Michael Reilly. Private Communication.

[11] Sun Microsystems, Inc. RPC: Remote Procedure Call Protocol. RFC 1057, SRI Network Information Center, June, 1988.

Адреса авторов

Jeffrey Mogul

Digital Equipment Corporation Western Research Laboratory 100 Hamilton Avenue

Palo Alto, CA 94301

Phone: (415) 853-6643

EMail: mogul@decwrl.dec.com

Steve Deering

Xerox Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

Phone: (415) 494-4839

EMail: deering@xerox.com

Перевод на русский язык

Игорь Шеваров

issh@onego.ru. Ноябрь 2003 г.