Network Working Group
Request for Comments: 4821
Category: Standards Track

M. Mathis
J. Heffner
PSC

March 2007

Определение Path MTU на уровне пакетизации

Packetization Layer Path MTU Discovery

Статус документа

Этот документ является спецификацией стандарта Internet, предназначенного для сообщества Internet, и служит приглашением к дискуссии в целях развития протокола. Сведения о текущем состоянии стандартизации протокола можно найти в документе Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Тезисы

Этот документ описывает отказоустойчивый метод определения MTU для пути (PMTUD[1]), который основывается на TCP или каком-либо ином уровне пакетизации (Packetization Layer) для зондирования пути через Internet с помощью прогрессивно увеличивающихся в размере пакетов. Метод описывается, как расширение RFC 1191 и RFC 1981, в которых определен основанный на ICMP метод определения Path MTU для протокола IP версий 4 и 6, соответственно.

Оглавление

1. Введение

Этот документ описывает метод PLPMTUD[2], являющийся расширением методов определения Path MTU, описанных в [RFC1191] и [RFC1981]. В отсутствие сообщений ICMP корректное определение MTU начинается с передачи мелких пакетов и последующих попыток увеличения их размера. Основная часть алгоритма реализована над уровнем IP, в транспортном уровне (например, TCP) или другом «протоколе пакетизации», отвечающем за определение границ пакетов.

[1]Path MTU Discovery.
[2]Packetization Layer Path MTU Discovery - определение MTU на уровне пакетизации.

Этот документ не является обновлением RFC 1191 и RFC 1981, однако он позволяет корректно определять MTU без ICMP, что неявно ослабляет некоторые требования к алгоритмам, заданным этими документами.

Описанные в этом документе методы основаны на имеющихся протоколах. Они применимы для многих транспортных протоколов, работающих на основе IPv4 и IPv6. Эти методы не требуют взаимодействия с нижележащими уровнями (за исключением согласования приемлемого размера пакетов) или партнерами. Поскольку описанные методы применяют только отправители, разные варианты реализации не будут вызывать проблем при взаимодействии.

Для большей ясности при описании отдается предпочтение терминологии TCP и IPv6. В разделе, посвященном терминам, представлены аналогичные термины IPv4 и концепции для терминов IPv6. В некоторых ситуациях конкретные детали описаны раздельно для IPv4 и IPv6.

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с [RFC2119].

Документ является результатом работы группы Path MTU Discovery (PMTUD) под эгидой IETF и в значительной мере является наследником RFC 1191 и RFC 1981 в части терминологии, идей и отдельных фрагментов текста.

2. Обзор

PLPMTUD представляет собой метод динамического определения MTU для пути в TCP или других протоколах пакетизации путем зондирования с помощью постепенно увеличивающихся пакетов. Наибольшая эффективность метода обеспечивается при использования вместе с механизмом Path MTU Discovery на основе ICMP, описанным в RFC 1191 и RFC 1981, но этот метод решает множество проблем отказоустойчивости, поскольку он не зависит от доставки сообщений ICMP.

Этот метод примени для TCP и других протоколов транспортного или прикладного уровня, которые отвечают за выбор границ пакетов (например, размер сегмента) пакетов.

Общей стратегией для уровня пакетизации является определение подходящего значения Path MTU путем зондирования пути постепенно увеличивающимися в размере пакетами. Если пробный пакет доставляется получателю, эффективное значение Path MTU увеличивается до размера пробного пакета.

Потеря отдельного пробного пакета (с получением ICMP Packet Too Big или без него) трактуется, как предел MTU, а не индикатор перегрузки. Только в этом случае протоколу пакетизации разрешается повторно передать пропущенные данные без изменения размера окна насыщения.

При возникновении тайм-аута или потери дополнительных пакетов в процессе зондирования, проба считается неубедительной (например, потеря пробного пакета не обязательно говорит о превышении Path MTU). Кроме того, потери трактуются подобно другим индикаторам перегрузки - изменение окна или скорости является обязательным в соответствующих стандартах контроля насыщения [RFC2914]. Зондирование может быть возобновлено после задержки, определяемой природой обнаруженного сбоя.

PLPMTUD использует метод поиска для определения Path MTU. Каждый последующий пробный пакет сужает диапазон поиска MTU путем повышения нижнего порога при успешном зондировании или снижения верхнего порога при неудаче, постепенно приходя к верному значению Path MTU. Для большинства транспортных уровней поиск следует останавливать после того, как диапазон поиска будет достаточно сужен, чтобы преимущества от повышения эффективного значения Path MTU еще превышали издержки, связанные с дальнейшим поиском.

Наиболее вероятен (и наименее серьезен) отказ при зондировании в результате перегрузки на канале. В этом случае приемлем повтор передачи пробного пакета того же размера после полной адаптации уровня пакетизации к перегрузке и восстановления нормальной работы. В остальных случаях дополнительные потери и тайм-ауты указывают на проблемы с каналом или уровнем пакетизации. В таких случаях желательно использовать более продолжительную задержку в зависимости от серьезности ошибки.

Можно использовать дополнительную проверку для обнаружения ситуаций, когда повышение MTU увеличивает скорость потери пакетов. Например, если канал организован через множество физических соединений с несогласованными значениями MTU, возможны ситуации, когда пробные пакеты будут доставляться даже в в случае превышения максимального размера пакетов на отдельных физических линиях. В таких случаях увеличение Path MTU до размера пробного пакета будет повышать частоту потери пакетов и значительной потере производительности. После увеличения MTU новый размер можно проверить, наблюдая частоту потерь.

Механизм PLPMTUD обеспечивает дополнительную гибкость реализациям классического метода Path MTU Discovery. Его можно настроить только на восстановление «черных дыр» ICMP с целью повышения отказоустойчивости Path MTU Discovery или, в качестве другой крайности, полностью отменить обработку ICMP и применять PLPMTUD взамен Path MTU Discovery.

Классический механизм Path MTU Discovery подвержен протокольным отказам («зависание» соединений), если сообщения ICMP Packet Too Big (PTB) не доставляются или не обрабатываются по той или иной причине [RFC2923]. С помощью PLPMTUD классический механизм Path MTU Discovery можно изменить путем включения дополнительной проверки согласованности без повышения риска повисания соединений в результате паразитного отказа при дополнительных проверках. Такие изменения классического Path MTU Discovery выходят за рамки этого документа.

В предельном случае все сообщения ICMP PTB могут безусловно игнорироваться, а PLPMTUD может использоваться в качестве единственного метода определения Path MTU. В такой конфигурации PLPMTUD работает параллельно с контролем перегрузок. Сквозной транспортный протокол подстраивает свойства потока данных (размер окна или пакетов), а потеря пакетов используется для определения приемлемости корректировки. Этот метод представляется более согласованным со «сквозным принципом» Internet, нежели использование сообщений ICMP, содержащих переписанные заголовки протоколов разных уровней.

Основные сложности реализации PLPMTUD обусловлены необходимость размещения на одном узле нескольких различных частей этого механизма. В общем случае для каждого протокола пакетизации нужна своя реализация PLPMTUD. Кроме того, естественным механизмом совместного использования данных Path MTU одновременными или последовательными соединениями является кэш информации о путях на уровне IP. Разным протоколам пакетизации нужны средства доступа и обновления общего кэша на уровне IP. В этом документе PLPMTUD описывается в терминах основных подсистем без полного описания его сборки в готовую реализацию.

Большая часть описанных здесь деталей реализации представляет собой рекомендации, основанные на опыте использования более ранних версий Path MTU Discovery. Эти рекомендации обусловлены желанием максимально повысить отказоустойчивость PLPMTUD в неидеальных условиях сетей.

Документ не включает полного описания реализации. В нем лишь очерчены детали, не оказывающие влияния на взаимодействие с другими реализациями и испытывающие заметное внешнее влияние на критерии оптимальности (например, эвристика поиска и кэширования MTU). Остальные детали рассмотрены явно, поскольку с ними связаны вопросы взаимодействия возможности для некоторых (возможно, весьма утонченных) случаев.

В разделе 3 представлен глоссарий используемых терминов.

В разделе 4 описаны детали PLPMTUD, оказывающие влияние на взаимодействие с другими стандартами и протоколами Internet.

В разделе 5 рассмотрено деление PLPMTUD на уровни и управление кэшем информации о путях на уровне IP.

В разделе 6 описаны общие свойства уровня пакетизации (Packetization Layer) и требуемые для PLPMTUD свойства.

В разделе 7 описано применение пробных пакетов для определения Path MTU.

Раздел 8 рекомендует использовать фрагментацию IPv4 в конфигурации, имитирующей функциональность IPv6, для минимизации будущих проблем при переходе на IPv6.

В разделе 9 описан программный интерфейс для реализации PLPMTUD в приложениях, которые самостоятельно определяют границы пакетов, а также средствах диагностики проблем, которые могут конфликтовать с Path MTU Discovery.

В разделе 10 рассмотрены детали реализации для конкретных протоколов, включая TCP.

3. Терминология

Ниже приведены определения используемых в документе терминов.

IP

IPv4 [RFC0791] или IPv6 [RFC2460].

Node - узел

Устройство, реализующее IP.

Upper layer - вышележащий уровень

Протокольный уровень, расположенный непосредственно над IP. Примерами являются транспортные протоколы типа TCP и UDP, протоколы управления типа ICMP, протоколы маршрутизации типа OSPF, а также IP или протоколы нижележащего уровня, «туннелируемые» (инкапсулированные) в IP, типа IPX, AppleTalk или самого IP.

Link - канал

Коммуникационное устройство или среда, через которую узлы могут взаимодействовать на канальном уровне (т. е., уровне, расположенном непосредственно под IP). Примерами могут служить Ethernet (плоская сеть или сеть с мостами), каналы PPP, сети X.25, Frame Relay или ATM[1], а также IP (и вышележащие) и другие «туннелируемые» протоколы типа туннелей через IPv4 или IPv6. Иногда для обозначения всего перечисленного используется болкк общий термин «нижележащий уровень» (lower layer).

Interface - интерфейс

Точка присоединения узла к каналу.

Address - адрес

Идентификатор уровня IP для интерфейса или группы интерфейсов.

Packet - пакет

Заголовок IP в комбинации с данными (payload).

MTU - максимальный передаваемый блок

Размер (в байтах) наибольшего пакета IP (заголовок и данные), который может быть передан по каналу или пути. Отметим, что более корректно было бы говорить IP MTU, поскольку аббревиатуру MTU уже используют в других стандартах.

Link MTU - MTU для канала

Размер (в байтах) наибольшего пакета IP, который может быть целиком (без деления на части) передан через канал. Следует отметить, что это определение отличается от определений, используемых другим органами стандартизации.

MTU для канала в документах IETF определяется, как IP MTU для этого канала. Размер учитывает заголовок IP, но не включает заголовки канального уровня и другое кадрирование, которое не является частью IP или данными IP.

Другие организации обычно определяют link MTU с учетом заголовков канального уровня.

Path - путь

Множество каналов, через которые пакет проходит от отправителя до получателя.

Path MTU, PMTU - MTU для пути

Минимальное значение link MTU среди всех каналов на пути между отправителем и получателем.

Classical Path MTU Discovery - классический метод определения MTU для пути

Процесс, описанный в RFC 1191 и RFC 1981, для определения MTU на пути с помощью сообщений ICMP PTB[2].

Packetization Layer - уровень пакетизации

Уровень стека сетевых протоколов, на котором данные сегментируются в пакеты.

Effective PMTU - эффективное значение MTU для пути

Текущее значение PMTU, используемое уровнем пакетизации для сегментирования.

PLPMTUD

Определение MTU для пути на уровне пакетизации (Packetization Layer Path MTU Discovery) - метод, описанный в этом документе и являющийся расширением классического механизма PMTU Discovery.

PTB (Packet Too Big) message - сообщение PTB

Сообщение ICMP, указывающее, что пакет IP слишком велик для пересылки. Это термин IPv6, соответствующий сообщению IPv4 ICMP «Fragmentation Needed and DF Set».

Flow – поток

[1]Asynchronous Transfer Mode - асинхронный режим передачи.
[2]Packet Too Big - пакет слишком велик.

Контекст, в котором могут вызываться алгоритмы MTU Discovery. Это, естественно, экземпляр протокола пакетизации, например, одна сторона соединения TCP.

MSS

Максимальный размер сегмента TCP[1] [RFC0793] - максимальный размер блока данных, доступный для уровня TCP. Обычно это Path MTU за вычетом размера заголовков IP и TCP.

Probe packet - пробный пакет, зонд

Пакет, который будет применяться для тестирования пути на предмет большего MTU.

Probe size - размер зонда

Размер пакета, который будет использоваться для тестирования пути с целью определения MTU (включая заголовок IP).

Probe gap - пропуск пробных пакетов

Данные, которые были потеряны и должны быть переданы повторно, если пробный пакет не доставлен.

Leading window - ведущее окно

Любые неподтвержденные данные в потоке на момент передачи пробного пакета.

Trailing window - трейлерное (ведомое) окно

Любые данные в потоке после передачи пробного пакета, но до подтверждения его получения.

Search strategy - стратегия поиска

Эвристика, используемая для выбора последовательности размеров пробных пакетов, сходящейся на приемлемом значении Path MTU, как описано в параграфе 7.3.

Full-stop timeout - тайм-аут полной остановки

Тайм-аут, когда ни один из переданных после некого события пакетов (включая повторы), не подтвержден получателем. Это принимается в качестве индикации того или иного отказа в сети (например изменение маршрутизации с переходом на канал с меньшим MTU). Более подробное описание приведено в параграфе 7.7.

4. Требования

Для всех каналов **должно** обеспечиваться соответствие своим MTU - при недетерминированном получении каналом пакета с размером больше установленного для канала MTU такие пакеты **должны** отбрасываться.

В достаточно давнем прошлом было множество мелких устройств, которые не применяли MTU, но были способный без гарантий доставлять чрезмерно большие пакеты. Например, некоторые старые повторители побитовые Ethernet способны пересылать пакеты произвольных размеров, но не могут делать это стабильно по причине ограниченной стабильности аппаратных таймеров. Это единственная причина для размещения PLPMTUD на нижележащих уровнях. Важно явно указать это требование для предотвращения стандартизации и внедрения в будущем технологий, не совместимых с PLPMTUD.

Все хостам **следует** использовать фрагментацию IPv4 в режиме имитации функциональности IPv6. Всю фрагментацию **следует** выполнять на хосте и во всех пакетах IPv4, включая фрагменты, **следует** устанавливать флаг DF, чтобы предотвратить их дальнейшее фрагментирование в сети (см. раздел 8).

Приведенные ниже требования относятся только к реализациям, включающим PLPMTUD.

Для использования PLPMTUD уровень пакетизации **должен** иметь механизм уведомления о потерях, который обеспечивает отправителя своевременной и точной индикацией потери конкретного пакета в сети.

Во всех случаях, за исключением потери единственного пробного пакеты, алгоритмы контроля перегрузок **должны** работать, как обычно. В отмеченном случае обычный механизм снижения перегрузки (изменение окна или скорости передачи) **следует** подавить. При прочих потерях данных стандартных контроль перегрузок **должен** сохраняться.

Подавление контроля перегрузок **должно** быть ограничено по частоте так, чтобы оно происходило реже чем худшие случаи потери пакетов для контроля перегрузок TCP при сравнимой скорости передачи данных по тому же пути (т. е., меньше, чем частота потерь TCP-friendly [tcp-friendly]). Это **следует** исполнять посредством требования минимального продвижения между корректировкой подавленного контроля перегрузок (по причине отказа пробы) и следующей попыткой зондирования, которое равно одному периоду кругового обхода для каждого пакета, разрешенного окном контроля насыщения. Этот вопрос дополнительно рассмотрен в параграфе 7.6.2.

Всякий раз при увеличении MTU переменные состояния для насыщения **должны** масштабироваться заново, чтобы не увеличивать размер окна в байтах (или скорость передачи данных в байт/с).

При снижении MTU (например, после обработки сообщений ICMP PTB) переменные состояния для насыщения **следует** масштабировать заново, чтобы не увеличивать размер окна в пакетах.

Если PLPMTUD обновляет MTU для определенного пути, все сессии уровня пакетизации, использующие этот путь (см. параграф 5.2), **следует** уведомить об использовании нового MTU и выполнении требуемой корректировки контроля перегрузок.

Все реализации **должны** включать механизмы, позволяющие приложениям селективно передавать пакеты, размер которых больше эффективного Path MTU, но меньше MTU в канале первого интервала пересылки. Это требуется для организации PLPMTUD с использованием протокола без организации явных соединений, а также для реализации средств диагностики, которые не используют системной реализации Path MTU Discovery. Дополнительная информация приведена в разделе 9.

Реализации **могут** пользоваться той или иной эвристикой при выборе начального эффективного значения Path MTU для каждого протокола. Протоколам без организации соединений и протоколам, не поддерживающим PLPMTUD, **следует** поддерживать свое значение, используемое по умолчанию в качестве начального эффективного Path MTU, которое может быть более консервативным (меньшим) по сравнению с начальным значением, используемым TCP и другими протоколами, подходящими для PLPMTUD. **Следует** задавать пределы начального эффективного Path MTU (eff_pmtu) по протоколам и маршрутам, а также верхний предел поиска (search_high). Дополнительная информация представлена в параграфе 7.2.

5. Уровни

Механизм Path MTU Discovery уровня пакетизации проще всего реализовать, разделив его функции между уровнями. Уровень IP лучше всего подходит для хранения общего состояния, сбора сообщений ICMP, отслеживания размера заголовков IP и управления информацией MTU, предоставляемой интерфейсами канального уровня. Однако

[1]Maximum Segment Size - максимальный размер сегмента.

процедуры, используемые PLPMTUD для зондирования и проверки Path MTU, весьма тесно связаны с функциями уровня пакетизации типа механизмов восстановления данных и состояний контроля перегрузок.

Отметим, что такая многоуровневая модель является прямым расширением рекомендаций текущих спецификаций PMTUD в RFC 1191 и RFC 1981.

5.1. Учет размера заголовков

Работа PLPMTUD на нескольких уровнях требует механизма для учета размера заголовков на всех уровнях от IP до уровня пакетизации (включительно). При передаче пакетов, не являющихся зондами, достаточно, чтобы уровень пакетизации обеспечивал верхнюю границу размера пакетов IP, не превышающую текущее эффективное значение Path MTU. На всех уровнях пакетизации, участвующих в классическом Path MTU Discovery, такое требование уже реализовано. Пр передаче зондов уровень пакетизации **должен** определить окончательный размер пробного пакета с учетом заголовка IP. Это требование вносится механизмом PLPMTUD и для его исполнения могут потребоваться новые межуровневые коммуникации в существующих реализациях.

5.2. Хранение информации PMTU

В этом документе используется концепция потока для определения области действия алгоритма Path MTU Discovery. Для многих реализаций поток будет естественным способом соответствовать экземпляру каждого протокола (т. е., каждому соединению или сессии). В таких реализациях описанные в этом документе алгоритмы выполняются в рамках каждой сессии каждого протокола. Наблюдаемое значение PMTU (eff_pmtu в параграфе 7.1) **может** использоваться разными потоками, проходящими через одно представление пути.

Кроме того, PLPMTUD можно реализовать так, что полное состояние будет связано с представлением пути. Такие реализации могут использовать множество соединений или сессий для каждой последовательности проб. Очевидно, что в некоторых средах такая модель обеспечит гораздо более быстрое схождение, особенно в случаях использования множеством приложений большого числа мелких соединений, каждое из которых слишком мало для полной поддержки процесса Path MTU Discovery.

В рамках одной реализации разные протоколы могут применять любую из этих двух моделей. По причине различий в протоколах по части генерации проб (параграф 6.2) и алгоритмов поиска MTU (параграф 7.3) может оказаться нежелательным совместное использование разными уровнями пакетизации общего состояния PLPMTUD. Это позволяет предположить, что некоторые протоколы смогут совместно использовать состояние зондирования, а другие смогут использовать совместно лишь найденное значение PMTU. В таких случаях разные протоколы будут иметь разные свойства в части схождения PMTU.

Для хранения кэшированного значения PMTU и других состояний общего пользования типа значений MTU, полученных из сообщений ICMP PTB, **следует** использовать уровень IP. В идеальном варианте это общедоступное состояние связывается с конкретным путем, по которому пакеты проходят между отправителем и получателем. Однако в большинстве случаев узел не будет иметь информации, достаточно для полной и точной идентификации такого пути. Взамен узлам приходится связывать значение PMTU с неким локальным представлением пути. Выбор такого представления остается за реализацией.

Реализация **может** использовать адрес получателя в качестве локального представления пути. Значение PMTU, связанное с получателем, будет меньшим из числа PMTU определенных для всего набора путей к этому получателю. Предполагается, что множество активных путей к данному получателю будет достаточно мало и во многих случаях путь окажется единственным. Такой подход обеспечивает использование пакетов оптимального размера для каждого получателя и хорошо интегрируется с концептуальной моделью хоста, описанной в [RFC2461], - значение PMTU будет храниться вместе с соответствующей записью в кэше адресатов. Поскольку трансляторы NAT[1] и другие промежуточные устройства могут одновременно показывать разные значения PMTU для одного адреса IP, сохранять **следует** минимальное значение.

В качестве представления пути **недопустимо** использовать сети или подсети, поскольку не существует единого механизма определения сетевой маски удаленного хоста.

Для пакетов с заданным отправителем маршрутом (source-routed), которые включают заголовок маршрутизации IPv6, опции IPv4 Loose Source и Record Route (LSRR) или Strict Source и Record Route (SSRR), такой маршрут **может** использоваться для локального представления пути. Реализация **может** использовать заданный отправителем маршрут в качестве локального представления пути.

Если применяются потоки IPv6, реализация **может** использовать триплет из метки потока (Flow label) и адресов отправителя и получателя [RFC2460][RFC3697] в качестве локального представления пути. Такая модель теоретически ведет к оптимальному выбору размера пакетов для потока, обеспечивая более тонкую детализацию значений MTU, нежели это делается на уровне получателей.

5.3. Трактовка IPsec

В этом документе не учитывается «уровень» IP Security (IPsec) [RFC2401], который логически размещается между IP и уровнем пакетизации. Реализации PLPMTUD могут трактовать IPsec как часть IP или уровня пакетизации, пока это не вступает в противоречие с реализацией. Если IPsec принимается как часть уровня IP, для каждой защищенной связи с удаленной точкой может ее трактовка как отдельного пути. Если IPsec считается частью уровня пакетизации, заголовок IPsec **должен** включаться при расчете размера заголовка уровня пакетизации.

5.4. Групповая адресация

Для групповых адресов получателей копии пакета могут проходить по разным путям до распределенных территориально узлов. Локальное представление «пути» к групповому получателю должно фактически учитывать потенциально большое число возможных путей.

По минимуму реализация **может** поддерживать одно значение MTU для всех групповых пакетов, исходящих от узла. Это значение **следует** делать достаточно малым, чтобы оно не превысило минимальное значение Path MTU на всех ветвях multicast-дерева. Если значение Path MTU, определенное традиционными для индивидуальных пакетов методами, меньше установленного для групповых пакетов MTU, значение MTU для групповых пакетов **может** быть

[1]Network Address Translator - транслятор сетевых адресов.

уменьшено до этого значения. Ясно, что эта модель приведет к использованию на многих путях размера пакетов меньше оптимального.

Если использующее групповую адресацию приложение получает полные отчеты о доставке пакетов (ен очевидно, поскольку это требование весьма слабо масштабируется), PLPMTUD **можно** реализовать на уровне групповых протоколов так, чтобы наименьшее значение MTU для путей группы становилось эффективным значением MTU для данной группы.

6. Общие свойства пакетизации

В этом разделе описаны общие свойства уровня пакетизации и характеристики, требуемые для реализации PLPMTUD. Описаны также некоторые общие вопросы реализации, относящиеся ко всем уровням пакетизации.

6.1. Механизм обнаружения потерь

Для уровня пакетизации важно наличие механизма своевременного и отказоустойчивого информирования о потере пакетов. PLPMTUD настраивает значение MTU на основе детектирования потери пакетов. Любая задержка или неточность при уведомлении о потере будет приводить к выбору некорректного MTU или замедлению схождения. Важно, чтобы механизм надежно отличал изолированные потери пробных пакетов от других потерь в окне перед зондом или после него.

Лучше всего, если уровень пакетизации использует явный механизм детектирования потерь типа «табло» SACK[1] [RFC3517] или ACK Vector [RFC4340] для того, чтобы отличить реальные потери от нарушения порядка, хотя достаточно и неявных механизмов типа TCP Reno.

PLPMTUD можно также реализовать в протоколах, которые в качестве основного механизма восстановления при потерях используют лишь тайм-ауты, однако тайм-ауты **не следует** использовать в качестве основного механизма обнаружения потерь при наличии других вариантов.

6.2. Генерация проб

Есть несколько способов изменения уровня пакетизации с целью генерации пробных пакетов. Разные методы вносят свои накладные расходы, которые можно разделить на три группы - сложность генерации пробных пакетов (рост сложности реализации Packetization Layer и дополнительные операции с данными), расход пропускной способности сети на передачу пробных пакетов и восстановление после отказов при пробах (издержки сети и протокола).

Для некоторых протоколов возможны расширения, позволяющие использовать произвольное заполнение любыми данными. Это существенно упрощает реализацию, поскольку пробы могут выполняться без участия протоколов вышележащих уровней и при отказе пробы обеспечиваются отсутствующие данные (probe gap) для заполнения текущего MTU при повторе. Этот метод может оказаться самым подходящим для протоколов, которые поддерживают произвольный размер опций или поддерживают внутри себя мультиплексирование.

Многие протоколы уровня пакетизации могут передавать чисто управляющие сообщения (без данных вышележащих протоколов) с заполнением произвольного размера. Например, для такой цели хорошо подходит блок SCTP PAD (см. параграф 10.2). При таком подходе преимущество обеспечивается за счет того, что в случае потери пробы не нужно ничего передавать повторно.

Эти методы не подходят для TCP, поскольку в нем нет отдельного поля размера или иного механизма, позволяющего отличить заполнение от данных. Для TCP единственным вариантом остается передача дополнительных данных в сегменте избыточного размера. У этой модели есть по крайней мере два варианта, описанных в параграфе 10.1.

В некоторых случаях может не найтись подходящего механизма для генерации проб в самом протоколе уровня пакетизации. В таких случаях остается полагаться на дополнительный протокол типа ICMP ECHO (ping). Более подробно это описано в параграфе 10.3.

7. Метод зондирования

В этом разделе описаны детали метода зондирования MTU, включая передачу пробных пакетов и обработку индикации ошибок для определения Path MTU.

7.1. Диапазоны размеров пакетов

В этом документе описан метод зондирования с использованием трех переменных состояния.

search_low
: Наименьший полезный размер зонда минус 1. Предполагается, что сеть способна доставлять пакеты размера search_low.

search_high
: Наибольший полезный размер зонда. Предполагается, что пакеты размером search_high слишком велики для доставки через сеть.

eff_pmtu
: Эффективное значение PMTU для данного потока. Это размер наибольшего пакета (не зонда), разрешаемый PLPMTUD для данного пути.

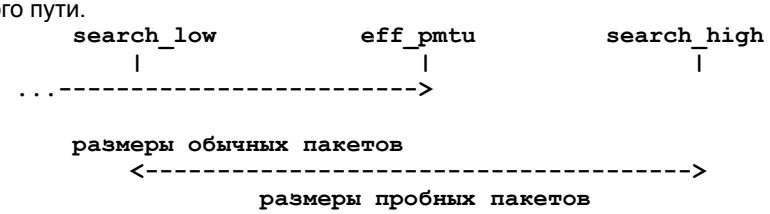

Рисунок 1.

При передаче обычных пакетов (не зондов) уровню пакетизации **следует** создавать пакеты размером не более eff_pmtu.

[1]Selective Acknowledgment - селективное подтверждение.

При передаче проб уровень пакетизации **должен** выбирать размер проб, который превышает search_low и не превышает search_high.

При зондировании с увеличением размера eff_pmtu всегда совпадает с search_low. В других состояниях (типа начальных условий) после обработки сообщения ICMP PTB или следующего PLPMTUD в другом потоке, проходящем по тому же пути, eff_pmtu may может отличаться от search_low. Обычно eff_pmtu будет не меньше search_low и меньше search_high. Обычно предполагается, но не требуется размер проб, превышающий eff_pmtu.

Для начальных условий, где еще нет информации о пути, значение eff_pmtu может превышать search_low. Начальное значение search_low **следует** выбирать достаточно малым, по производительность измерения может возрастать при выборе большего значения eff_pmtu (см. параграф 7.2).

Если eff_pmtu > search_low, это явно разрешает передавать не являющиеся пробами пакеты размером больше search_low. Когда такой пакет подтверждается, он эффективно служит «неявной пробей» и значение search_low **следует** увеличивать до размера подтвержденного пакета. Однако при потере «неявной пробы» **недопустимо** считать это потерей пробного пакета. Если значение eff_pmtu слишком велико, это условие может быть обнаружено только через сообщения ICMP PTB или детектирование «черной дыры» (см. параграф 7.7).

7.2. Выбор начальных значений

Начальное значение search_high **следует** делать равным размеру наибольшего пакета, который может поддерживаться потоком. Это значение может быть ограничено величиной MTU локального интерфейса, явным механизмом протокола типа опции TCP MSS или внутренним ограничением типа размера поля длины пакета. Кроме того, начальное значение search_high **может** быть ограничено конфигурационной опцией для предотвращения слишком большого размера пробных пакетов. Очевидно, что search_high будет совпадать с начальным значением Path MTU, рассчитанным с помощью классического алгоритма Path MTU Discovery.

Рекомендуется в качестве начального значения search_low устанавливать значение MTU, что очевидно будет работать в большинстве сред. Для современных технологий значение 1024 байта является достаточно безопасным. Начальное значение search_low **следует** делать настраиваемым.

Корректная работа механизма Path MTU Discovery очень важна для обеспечения отказоустойчивости и эффективной работы сети Internet. Любое серьезное изменение (например, описанное здесь) может оказывать негативное влияние, если оно вызывает неожиданные изменения в поведении протоколов. Выбор начального значения eff_pmtu определяет, в какой степени поведение реализации PLPMTUD соответствует классическому PMTUD в тех случаях, когда классического метода достаточно.

Консервативный подход заключается в установке значения eff_pmtu для search_high и использовании сообщений ICMP PTB для установки приемлемо низкого значения eff_pmtu. В такой конфигурации классический механизм PMTUD обеспечивает полную функциональность, а PLPMTUD вызывается лишь для восстановления при возникновении «черных дыр» ICMP с помощью процедуры, описанной в параграфе 7.7.

В некоторых случаях, когда известно, что классический механизм PMTUD столкнется с отказом (например, когда сообщения ICMP PTB административно запрещены из соображений безопасности), использование малых начальных значений eff_pmtu позволит предотвратить долгое ожидание, требуемое для детектирования «черной дыры». С другой стороны, использование слишком малого начального значения eff_pmtu может приводить к снижению производительности.

Отметим, что в качестве начального значения eff_pmtu может быть выбрано любое число из интервала search_low - search_high. Начальное значение eff_pmtu = 1400 байтов может быть хорошим выбором, поскольку оно безопасно почти для всех туннелей в сетях общего пользования и достаточно близко к оптимальному MTU на большинстве путей в современной сети Internet. Более эффективный выбор может обеспечить использование статистики недавних потоков - например, в качестве начального eff_pmtu для потока можно установить средний (медианный) размер пробных пакетов из всех успешных проб.

Поскольку стоимость PLPMTUD обусловлена, прежде всего, специфическими для протокола издержками на генерацию и обработку пробных пакетов, может оказаться целесообразным использование для каждого протокола своей эвристики выбора начального значения eff_pmtu. Для протоколов без организации соединений и других протоколов, которые могут не получать четкой индикации «черных дыр» ICMP, очень важно использовать более консервативные (меньшие) начальные значение eff_pmtu, как описано в параграфе 10.3.

Следует устанавливать конфигурационные опции на уровне протокола и маршрута для замены начальных значений eff_pmtu и других переменных состояния PLPMTUD.

7.3. Выбор размера пробы

Пробный пакет может иметь любой размер из описанного выше «диапазона размеров проб». Однако на выбор подходящего размера влияет множество факторов. В качестве простой стратегии выбора может использоваться двоичный поиск в уменьшением вдвое размера диапазона после каждой пробы. Однако для некоторых протоколов (типа TCP) отказы обходятся «дороже» успешных проб, поскольку данные в столкнувшейся с отказом пробе требуется передать заново. Для таких протоколов стратегия незначительного увеличения размера каждой следующей пробы может обеспечивать меньшие издержки. Для многих протоколов на уровне пакетизации и выше его преимущества в результате роста MTU могут расти ступенчато, поэтому из некоторых поддиапазонов размер проб выбирать просто не имеет смысла.

Для оптимизации может быть разумно выбрать пробы размером в некое общее или ожидаемое значение MTU, например 1500 байтов для стандартной технологии Ethernet или 1500 за вычетом размера заголовков протокола туннелирования.

Некоторые протоколы могут использовать другие механизмы определения размера проб. Например, протоколы, с неким естественным размером блока данных могут просто собирать сообщение из множества таких блоков, пока общий размер меньше search_high и возможно больше search_low.

Каждый протокол пакетизации **должен** определять схождение проб, т. е., размер проб уже достаточно мал и дальнейшее зондирование не уменьшит его. Когда зондирование сходится, **следует** устанавливать таймер. По завершении отсчета этого таймера search_high следует сбросить в начальное значение (см. выше), чтобы зондирование можно было восстановить. Таким образом, при изменении пути с ростом Path MTU это преимущество

может быть использовано. Для таймера **недопустимо** устанавливать значение меньше 5 минут и рекомендуется задавать 10 минут в соответствии с RFC 1981.

7.4. Условия зондирования

Прежде, чем передавать пробы, для потока **должны** быть выполнены по крайней мере перечисленные ниже условия:

- не должно быть незавершенных проб или потерь;
- если для последней пробы возник отказ или она не была завершена, отсчет тайм-аута для проб должен завершиться (см. параграф 7.6.2);
- размер доступного окна превышает размер пробы;
- для протоколов, использующих в пробах обычные данные, имеется достаточный для отправки пробы объем данных.

Кроме того, с алгоритмами своевременного обнаружения потерь в большинстве протоколов связаны предварительные условия, которые **следует** выполнить до отправки зонда. Например, TCP Fast Retransmit не обеспечивает надежных результатов пока вслед за зондом не будет передано достаточное число сегментов, поэтому отправителю **следует** сначала набрать достаточный объем данных в очереди и достаточно большое окно приема для отправки зонда и не менее Tcprexmtthresh [RFC2760] дополнительных сегментов. Это условие може препятствовать зондированию при некоторых состояниях протокола (например, незадолго до завершения работы соединения или при слишком малом окне).

Протоколы **могут** задерживать отправку не являющихся зондами пакетов для того, чтобы набрать объем данных, требуемый условиями зондирования. **Следует** применять алгоритм отложенной передачи, использующий тот или иной метод самомасштабирования для ограничения времени задержки отправки данных. Например, возврат ACK можно использовать для того. Чтобы предотвратить снижение размера окна сверх требуемого для отправки зонда значения.

7.5. Выполнение зондирования

После выбора размера зондов и выполнения перечисленных выше условий уровень пакетизации (Packetization Layer) **может** начать зондирование. Для этого он создает пробные пакеты так, чтобы их размер, с учетом всех заголовков IP, был равен размеру зонда. После отправки зонда происходит ожидание отклика, которое может закончиться одним из перечисленных ниже результатов:

Успех. Получение пробного пакета было подтверждено удаленным хостом.

Отказ. Протокольный механизм указал на потерю пробы, тогда как в предыдущем и последующем окне потери пакетов не наблюдалось.

Тайм-аут. Протокольный механизм указал на потерю пробы, в предшествующем окне потерь не было и нет возможности определить наличие потери пакетов в последующем окне. Например, потеря была обнаружена по тайм-ауту и использована повторная передача go-back-n.

Незавершенность. Проба была потеряна наряду с другими пакетами в предыдущем или следующем окне.

7.6. Реакция на результат зондирования

По завершении проб результат **следует** обрабатывать в соответствии с приведенным ниже описанием в зависимости от категории этого результата.

7.6.1. Успешная проба

Доставка пробного пакета говорит о том, что Path MTU не меньше размера этого пакета. Размер пробы устанавливается в качестве search_low. Если размер пробы превышает eff_pmtu, увеличивается значение eff_pmtu до размера пробы. Размер пробного пакета может оказаться меньше eff_pmtu, если поток не использовал полного MTU для пути в силу тех или иных ограничения (например, объем доступных данных в интерактивной сессии).

Отметим, что при маршрутизации пакетов потока по разным путям или путям с недетерминированным значением MTU, доставка одного пробного пакета не говорит о доставке остальных пакетов того же размера. В таких случаях для получения надежного результата уровню пакетизации **следует** проверять значение MTU, как описано в параграфе 7.8.

7.6.2. Неудачная проба

Когда теряются только пробы, это считается индикацией того, что Path MTU меньше размера зонда. В таких случаях потерю **не следует** считать индикацией перегрузки.

При отсутствии других индикаторов установите для search_high значение размера зонда минус 1. Значение eff_pmtu может быть больше размера зонда, если поток не использовал полное значение MTU для пути, поскольку здесь применяются иные ограничения типа доступности данных в интерактивном сеансе. Если eff_pmtu превышает размер зонда, значение eff_pmtu **должно** быть уменьшено так, чтобы оно не превышало search_high и **следует** уменьшать его до search_low, поскольку была обнаружена некорректность eff_pmtu, как после тайм-аута full-stop (параграф 7.7).

Если полученное сообщение ICMP PTB соответствует пробному пакету, для search_high и eff_pmtu **можно** установить значение MTU, указанное в этом сообщении. Отметим, что сообщение ICMP может быть получено до или после индикации потери протоколом.

Отказ пробы является одной из ситуаций, когда уровню пакетизации **следует** не принимать потерю как индикацию перегрузки. Поскольку существует некоторый риск того, что демпфирование контроля насыщения может давать непредвиденные последствия (даже для отдельного факта потери), **требуется** чтобы отказы проб были более редкими событиями по сравнению с обычными потерями при стандартном контроле перегрузок. В частности, после отказа пробы PLPMTUD **недопустимо** повторять пробу до истечения интервала, превышающего обычный интервал между событиями контроля перегрузок (см. раздел 4). Простейшей оценкой интервала до следующего события контроля перегрузок является число интервалов кругового обхода, совпадающее с числом пакетов в текущем окне контроля перегрузок.

7.6.3. Тайм-аут для пробы

Если потеря была обнаружена по тайм-ауту и восстановлена с помощью повторной передачи go-back-n, необходимо уменьшить окно насыщения. Относительно высокая цена пробу в этом случае может «заслуживать» увеличения интервала перед отправкой следующего зонда. **Рекомендуется** интервал, превышающий в 5 раз время для случая отказа без тайм-аута параграф 7.6.2).

7.6.4. Незавершенная проба

Наличие других потерь вблизи потери зонда может указывать на потерю зонда по причине перегрузки, а не из-за ограничения MTU. В таких случаях переменные состояния eff_pmtu, search_low и search_high **не следует** обновлять а пробу того же размера **следует** повторить, как только будут выполнены условия зондирования (т. е. на уровне пакетизации не останется не восстановленных потерь). В этот момент уместно повторить пробу, поскольку окно насыщения для потока будет в нижней точке, что минимизирует вероятность потери в результате перегрузки.

7.7. Тайм-аут полной остановки

При любых условиях тайм-аут полной остановки (full-stop timeout, иногда persistent timeout) **следует** считать указанием некого значимого разрушительного события в сети, такого как отказ маршрутизатора или смена маршрутизации на путь с меньшим MTU. Для TCP это возникает при наступлении порога тайм-аута R1, описанного в [RFC1122].

При возникновении тайм-аута полной остановки и отсутствии сообщения ICMP (PTB, Net unreachable и т. п. или сообщение ICMP проигнорировано по иной причине) с указанием причины **рекомендуемым** первым действием по восстановлению является трактовка этого события как обнаружение «черной дыры» ICMP, описанной в [RFC2923].

Отклик на обнаружение черной дыры зависит от текущих значений search_low и eff_pmtu. Если eff_pmtu > search_low, устанавливается eff_pmtu = search_low. В противном случае для eff_pmtu и search_low устанавливается начальное значение search_low. При дополнительных последующих тайм-аутах значение search_low и eff_pmtu **следует** уменьшать вдвое с нижней границей 68 байтов для IPv4 и 1280 байтов для IPv6. **Можно** устанавливать и более низкие значения для поддержки ограниченной работы по каналам связи с MTU меньше разрешенных спецификацияим IP значений.

7.8. Проверка MTU

Поток может проходить одновременно по нескольким путям, но реализация будет способна сохранить представление лишь одного пути для потока. Если пути имеют разные MTU, сохранение минимального среди всех путей значения MTU обеспечит корректное поведение. Если сообщения ICMP PTB доставляются, классическим механизм PMTUD будет работать корректно.

При отказе доставки ICMP, прерывающем PMTUD, соединение будет полагаться только на PLPMTUD. В этом случае PLPMTUD также может давать отказ, поскольку предполагает прохождение потока по пути с одним MTU. Проба с размером между минимальным и максимальным Path MTU может быть успешной. Однако при увеличении эффективного PMTU потока частота потерь значительно возрастет. Поток может все еще сохраняться, но результирующий уровень потерь вероятно станет неприемлемым. Например, при использовании двухстороннего циклического чередования может отбрасываться 50% полноразмерных пакетов.

Такое чередование зачастую операционно нежелательно по другим причинам (например, нарушение порядка пакетов) и его обычно стараются избежать хэшированием каждого потока на одном пути. Однако для повышения отказоустойчивости реализации **следует** обеспечивать ту или иную форму проверки MTU, чтобы при росте потерь в результате увеличения eff_pmtu можно было вернуться к меньшему MTU.

Рекомендуемой стратегией является запись значения eff_pmtu перед его увеличением. Затем, если частота потерь за интервал времени превышает порог (например, больше 10% за несколько интервалов RTO[1]), новое значение MTU считается непригодным. **Следует** восстановить сохраненное значение eff_pmtu и установить сниженное как при отказе пробы значение search_high. Реализациям PLPMTUD **следует** поддерживать проверку MTU.

8. Фрагментация на хосте

Уровням пакетизации **следует** избегать передачи сообщений, которые будут требовать фрагментирования [Kent87] [frag-errors]. Однако полностью предотвратить фрагментацию не всегда удается. Некоторые уровни пакетизации, такие как приложения UDP вне ядра, могут быть не в состоянии менять размер передаваемых сообщений и в результате размер дейтаграмм превысит Path MTU.

IPv4 разрешает таким приложениям передавать пакеты без установленного бита DF. Пакеты избыточного размера без флага DF будут фрагментироваться сетью или передающим хостом, если MTU в канале меньше размера пакта. В некоторых случаях пакеты могут фрагментироваться несколько раз, если на пути имеются каналы с уменьшающимися значениями MTU. Такой подход **не рекомендуется**.

Реализациям IPv4 **рекомендуется** использовать стратегию, имитирующую функциональность IPv6. Когда приложение передает дейтаграммы размером больше эффективного Path MTU, их **следует** фрагментировать до Path MTU на IP-уровне хоста, даже если размер меньше MTU на первом канале, напрямую подключенном к хосту. Для фрагментов **следует** установить бит DF, чтобы они снова не были фрагментированы в сети. Это будет минимизировать вероятность того, что приложения будут полагаться на фрагментацию IPv4 таким способом, который не может быть реализован в IPv6. По меньшей мере одна из основных операционных систем уже применяет такую стратегию. В разделе 9 описаны некоторые исключения из этого правила, когда приложения передают большие пакеты для зондирования или диагностики.

Поскольку протоколы, не реализующие PLPMTUD, все еще подвержены проблемам в результате возникновения черных дыр ICMP, может оказаться желательным ограничение таких протоколов «безопасным» MTU, с которым вероятно можно будет работать на любом пути (например, 1280). реализующим PLPMTUD протоколам можно работать с полным диапазоном, поддерживаемым нижним уровнем.

Отметим, что фрагментация IP делит данные на пакеты, поэтому является минимальной реализацией уровня пакетирования. Однако в ней нет механизма детектирования потерь, поэтому не может поддерживать естественную реализацию PLPMTUD. Для PLPMTUD на базе фрагментации нужен добавочный протокол (параграф 10.3).

[1]Retransmission timeout - тайм-аут повторной передачи.

9. Зондирование из приложений

Все реализации **должны** включать механизм, позволяющий приложениям, которые используют протоколы без организации соединений, передавать свои пробы. Это необходимо для реализации PLPMTUD в прикладном протоколе, как описано в параграфе 10.4, или для реализации диагностических средств отладки PMTUD. **Должен** быть механизм, позволяющий приложению передавать дейтаграммы размером больше eff_pmtu (оценка операционной системой значения Path MTU) без фрагментации. В пакетах IPv4 **должен** быть установлен флаг DF.

В настоящее время большинство операционных систем поддерживает два режима отправки дейтаграмм, один из которых «молча» фрагментирует слишком большие пакеты, а другой отвергает их. Ни один из этих режимов не подходит для реализации PLPMTUD в приложениях или диагностики проблем Path MTU Discovery. **Требуется** третий режим, который будет передавать дейтаграммы даже при размерах более текущей оценки Path MTU.

Реализация PLPMTUD в приложении также требует механизма, с помощью которого приложение может информировать операционную систему о результатах проб, как описано в параграфе 7.6, или напрямую обновлять search_low, search_high и eff_pmtu, описанные в параграфе 7.1.

Диагностические приложения полезны при поиске проблем PMTUD, например, вызванных неисправным маршрутизатором, который возвращает сообщения ICMP PTB с неверной информацией о размере. Такие проблемы наиболее быстро можно обнаружить с помощью инструмента, который может передавать пробы любого заданного размера, собирая и отображая все возвращенные сообщения ICMP PTB.

10. Конкретные уровни пакетизации

Все протоколы уровня пакетизации должны учитывать все аспекты, рассмотренные в разделе 6. Для многих протоколов решение этих вопросов является простым. В этом разделе рассматриваются конкретные детали реализации PLPMTUD с некоторыми протоколами. Следует надеяться, что представленные здесь описания будут достаточной иллюстрацией для разработчиков применительно к другим протоколам.

10.1. Метод зондирования с использованием TCP

В TCP нет механизма, позволяющего различать данные и заполнение. Поэтому протокол TCP должен генерировать пробы путем соответствующего сегментирования данных. Существует два подхода к сегментированию - с перекрытием и без перекрытия.

В варианте без перекрытия данные сегментируются так, что проба не имеет данных, которые перекрываются с любыми последующими сегментами. При потере пробы пропуск (probe gap) будет составлять полный размер пробы без заголовков. Данные из probe gap потребуется передать повторно в нескольких более мелких сегментах.

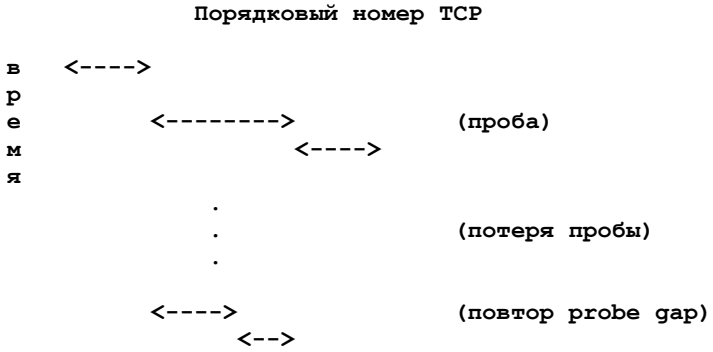

Рисунок 2.

Другим вариантом является такое перекрытие последующих данных с пробой, что пропуск пробы (probe gap) составит текущий размер MSS. При успешной пробе это увеличивает издержки, поскольку некоторые данные передаются дважды, но повторно будет передаваться лишь один сегмент после потери пробы. Когда проба успешна, вероятна будет генерация дубликатов подтверждений в результате передачи дубликатов данных. Важно, чтобы эти дубликаты не вызывали режим ускоренного повтора (Fast Retransmit). Поэтому применяющей такой подход реализации **следует** ограничивать размер проб трехкратным значением текущего MSS (что может вызвать не более 2 дубликатов подтверждений) или подобающим образом настроить порог дублирования подтверждений для данных сразу после успешной пробы.

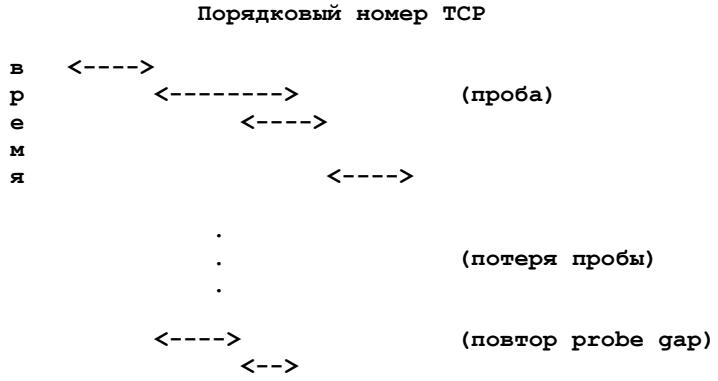

Рисунок 3.

Выбор применяемого метода сегментирования следует основывать на простоте и эффективности в данной реализации TCP.

10.2. Метод зондирования с использованием SCTP

В протоколе SCTP[1] [RFC2960] приложение пишет в SCTP сообщения, которые делят данные на небольшие блоки (chunk), подходящие для передачи через сеть. Каждому блоку назначается порядковый номер TSN[2]. После передачи TSN протокол SCTP не может изменить размер блока. Поддержка в SCTP множества путей обычно требует от протокола выбирать размер блока так, чтобы сообщения помещались в наименьший PMTU среди всех путей. Хотя это не требуется, реализация может собрать множество блоков в более крупные пакеты IP для отправки по путям с большим PMTU. Отметим, что в SCTP пробы PMTU должны выполняться на каждом пути к партнеру.

Рекомендуемым методом генерации проб является дабавление блока, содержащего лишь заполнение, к сообщению SCTP. Для создания пробы блок PAD, определенный в [RFC4820], **следует** присоединять к блоку HEARTBEAT (HB) минимального размера. Этот метод полностью совместим и современными реализациями SCTP.

SCTP **может** также использовать для проб метод, аналогичный описанному выше для TCP, применяя встроенные данные. Использование такого метода имеет преимущество в том, что при успешных пробах не возникает дополнительных издержек, однако при отказе пробы придется повторять передачу данных, что может влиять на производительность потока.

10.3. Метод зондирования для фрагментации IP

Имеется несколько протоколов и приложений, которые обычно передают большие дейтаграммы и полагаются для их доставки на фрагментацию IP. Давно известно, что это имеет некоторые нежелательные последствия [Kent87]. Недавно стало ясно, что фрагментация IPv4 недостаточно отказоустойчива для общего применения в современной сети Internet. 16-битовое поле идентификации IP недостаточно велико для предотвращения частого ошибочного связывания фрагментов IP, а контрольных сумм TCP и UDP недостаточно для предотвращения доставки полученных в результате поврежденных данных вышележащим протоколам [frag-errors].

Как упомянуто в разделе 8, протоколы дейтаграмм (такие как UDP) могут полагаться на фрагментацию IP как уровень пакетизации. Однако применение фрагментации IP для реализации PLPMTUD проблематично, поскольку на уровне IP нет механизма для определения доставки пакетов конечному получателю без прямого участия приложения.

Для поддержки фрагментации IP в качестве уровня пакетизации без необходимости изменять приложения реализации **следует** полагаться на совместное использование Path MTU, описанное в параграфе 5.2, а также дополнительный протокол для проб Path MTU. Имеется множество пригодных для этого протоколов, таких как ICMP ECHO и ECHO REPLY или дейтаграммы UDP в стиле traceroute, которые вызывают сообщения ICMP. Использование ICMP ECHO и ECHO REPLY будет проверять прямой и обратный путь, поэтому отправитель сможет лишь воспользоваться преимуществом меньшего из двух. Другие методы проверяют только прямой пути и поэтому более предпочтительны.

Все эти подходы связаны с множеством возможных проблем отказоустойчивости. Наиболее вероятные отказы обусловлены потерями, не относящимися к MTU (например, узлы, отбрасывающие пакеты некоторых протоколов).Такие потери, не связанные с MTU, могут помешать PLPMTUD увеличить MTU, заставляя использовать фрагментацию IP при работе с меньшим, чем нужно, MTU. Поскольку эти проблемы вряд ли будут влиять на возможности взаимодействия, они сравнительно безвредны.

Однако имеются и более серьезные отказы, например вызываемые промежуточными устройствами или маршрутизаторами, которые выбирают разные пути для различных протоколов или сессий. В таких средах вспомогательные протоколы могут корректно получать другие значения Path MTU, нежели основной протокол. Если вспомогательный протокол найдет большее значение MTU, чем будет у основного протокола, PLPMTUD может выбрать MTU, которое не подойдет для основного протокола. Хотя эта проблема потенциально серьезна, такая ситуация будет скорей всего восприниматься как некорректная многочисленными наблюдателями, которые постараются исправить ситуацию.

Поскольку протоколы без организации соединений могут не сохранять состояния, достаточного для эффективной диагностики черных дыр MTU, большую устойчивость к ошибкам обеспечило бы использование небольшого начального MTU (например, 1 Кбайт или меньше) до начала проверки пути с целью определения MTU. По этой причине реализациям, применяющим фрагментацию IP, **следует** использовать начальное значение eff_pmtu, выбранное в соответствии с параграфом 7.2, за исключением отдельного глобального управления для принятого по умолчанию начального eff_mtu в протоколах без организации соединений.

Протоколы без организации соединений создают также дополнительную проблему поддержки кэша с информацией о пути, поскольку здесь нет событий, соответствующих организации или разрыву соединения, которые могли бы служить для управления кэшем. Естественным решением будет сохранение в кэше неизменной записи для default path, которая имеет в качестве eff_pmtu фиксированное начальное значение для протокола без организации соединений. Вспомогательный протокол Path MTU Discovery будет вызываться один, как только число фрагментированных дейтаграмм для любого конкретного адресата достигнет настраиваемого порога (например, 5). Новые записи в кэше будут создаваться при обновлении вспомогательным протоколом значений eff_pmtu и удаляться по таймеру или алгоритму замены записей LRU[3].

10.4. Метод зондирования для приложений

Недостатки, связанные с фрагментацией IP и добавочным протоколом для выполнения Path MTU Discovery, можно устранить реализацией Path MTU Discovery в самом приложении, с использованием прикладного протокола. Приложение должно иметь тот или иной подходящий метод генерации проб и механизм своевременного обнаружения потери пробных пакетов.

В идеале прикладной протокол включает облегченную эхо-функцию, которая подтверждает доставку сообщения, и имеет механизм заполнения, позволяющий создавать пробы нужного размера так, чтобы заполнение не возвращалось в эхо. Такая комбинация (похожая на SCTP HB с заполнением - PAD) является **рекомендуемой**, поскольку приложение может раздельно измерять MTU каждого направления на путях с асимметричными MTU.

Для протоколов, которые не могут реализовать PLPMTUD с помощью «эхо и заполнения», имеются другие методы генерации проб. Например, протокол может иметь эхо-сигнал переменного размера для эффективного измерения

[1]Stream Control Transmission Protocol - протокол управления потоковой передачей.
[2]Transmission Sequence Number - порядковый номер передачи.
[3]Least Recently Used - самое недавнее использование.

минимального MTU на прямом и обратном пути или может иметь способ добавлять заполнение в обычные сообщения с реальными данными приложений. Могут быть и другие способы сегментирования данных приложения для создания проб или, в крайнем случае, может быть целесообразно расширить протокол новыми типами сообщений для поддержки определения MTU.

Отметим, что при необходимости добавить новые типы сообщений для поддержки PLPMTUD наиболее общим решением будет добавление сообщений ECHO и PAD, которые обеспечат максимальную широту взаимодействия реализации PLPMTUD в конкретном приложении с другими приложениями и протоколами на той же конечной системе.

Все методы зондирования требуют возможности передавать сообщения размером больше текущего значения eff_pmtu, описанного в разделе 9.

11. Вопросы безопасности

При любых условиях описанные в этом документе процедуры PLPMTUD защищены по меньшей мере в такой же степени, как современные стандартные процедуры Path MTU Discovery, описанные в RFC 1191 и RFC 1981.

Поскольку механизм PLPMTUD разрабатывался для отказоустойчивой работы без получения ICMP или иных сообщений из сети, его можно настроить на игнорирование сообщений ICMP глобально или на уровне приложения. В такой конфигурации он не может быть атакован, пока у злоумышленника нет возможности идентифицировать пакеты проб и вызывать их потерю. Атаки на PLPMTUD снижают производительность, но не столь сильно, как атаки на контроль перегрузок, вызывающие потерю любых пакетов. Такой злоумышленник может нанести гораздо больший, полностью нарушив работу определенных протоколов, например, DNS.

Поскольку протоколы пакетизации могут иметь общее состояние, враждебность одного протокола (в частности, приложения) может вредить работе других протоколов на том же хосте, снижая эффективное значение MTU. Если протокол пакетизации не является доверенным, нельзя разрешать ему запись в общее состояние.

12. Литература

12.1. Нормативные документы

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, March 2004.

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960[1], October 2000.

[RFC4820] Tuexen, M., Stewart, R., and P. Lei, "Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)", RFC 4820, March 2007.

12.2. Дополнительная литература

[RFC2760] Allman, M., Dawkins, S., Glover, D., Griner, J., Tran, D., Henderson, T., Heidemann, J., Touch, J., Kruse, H., Ostermann, S., Scott, K., and J. Semke, "Ongoing TCP Research Related to Satellites", RFC 2760, February 2000.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401[2], November 1998.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC3517] Blanton, E., Allman, M., Fall, K., and L. Wang, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", RFC 3517, April 2003.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[Kent87] Kent, C. and J. Mogul, "Fragmentation considered harmful", Proc. SIGCOMM '87 vol. 17, No. 5, October 1987.

[tcp-friendly] Mahdavi, J. and S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control", Technical note sent to the end2end-interest mailing list , January 1997, <http://www.psc.edu/networking/papers/tcp_friendly.html>.

[frag-errors] Heffner, J., "IPv4 Reassembly Errors at High Data Rates", Work in Progress[3], December 2007.

Приложение A. Благодарности

Многие идеи и даже часть текста этого документа заимствованы из RFC 1191 и RFC 1981.

[1]Этот документ заменен RFC 4960. *Прим. перев.*
[2]Этот документ заменен RFC 4301. *Прим. перев.*
[3]Работа опубликована в RFC 4963. *Прим. перев.*

В подготовку документа внесло свой вклад множество людей, включая Randall Stewart (текст для SCTP), Michael Richardson (материал из ранних идентификаторов туннелей, которые игнорируют DF, Stanislav Shalunov (идея о том, что чистый механизм PLPMTUD работает параллельно контролю перегрузок), Matt Zekauskas (за поддержку обсуждений при встречах). Спасибо первоначальным разработчикам Kevin Lahey, John Heffner и Rao Shoaib, которые предоставили конкретные отклики о недостатках ранних версий. Спасибо также всем, кто внес конструктивные замечания на встречах рабочей группы и в почтовой конференции. Авторы уверены, сто этот список достойных людей неполон.

Работа Matt Mathis и John Heffner была поддержана грантом компании Cisco Systems, Inc.

Адреса авторов

Matt Mathis

Pittsburgh Supercomputing Center

4400 Fifth Avenue

Pittsburgh, PA 15213

USA

Phone: 412-268-3319

EMail: mathis@psc.edu

John W. Heffner

Pittsburgh Supercomputing Center

4400 Fifth Avenue

Pittsburgh, PA 15213

US

Phone: 412-268-2329

EMail: jheffner@psc.edu

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

Полное заявление авторских прав

Интеллектуальная собственность

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.