Internet Engineering Task Force (IETF)
Request for Comments: 8899
Updates: 4821, 4960, 6951, 8085, 8261
Category: Standards Track
ISSN: 2070-1721

G. Fairhurst
T. Jones
University of Aberdeen
M. Tüxen
I. Rüngeler
T. Völker
Münster University of Applied Sciences

September 2020

Packetization Layer Path MTU Discovery for Datagram Transports

Определение MTU для пути на уровне пакетизации для транспорта на основе дейтаграмм

Аннотация

Этот документ задаёт механизм определения MTU на пути для уровня пакетизации дейтаграмм (Datagram Packetization Layer Path MTU Discovery или DPLPMTUD). Это отказоустойчивый метод определения MTU на пути (Path MTU Discovery или PMTUD) для уровня пакетизации дейтаграмм (Packetization Layer или PL). Он позволяет уровню PL или использующему его приложению на основе дейтаграмм определять возможность поддержки на пути текущего размера дейтаграмм. Это можно использовать для определения и сокращения размера сообщений при столкновении отправителя с «чёрной дырой» для пакетов. Метод также позволяет проверить путь через сеть для определения максимального размера пакетов. Это обеспечивает для транспортировки дейтаграмм функциональность, эквивалентную PLPMTUD для протокола TCP, заданную в RFC 4821 и обновлённую этим документом. Документ также обновляет рекомендации по использованию UDP, указывая этот метод, а также обновляет протокол SCTP.

Документ включает заметки по реализации для встраивания Datagram PMTUD с транспорт дейтаграмм IETF и приложения, использующие транспорт дейтаграмм.

Эта спецификация обновляет RFC 4960, RFC 4821, RFC 6951, RFC 8085, RFC 8261.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc8899.

Оглавление

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

В IETF определён транспорт дейтаграмм с использованием протоколов UDP, SCTP (Stream Control Transmission Protocol) и DCCP (Datagram Congestion Control Protocol), а также протоколов, работающих на основе этого транспорта (например, SCTP/UDP, DCCP/UDP, QUIC/UDP), и прямой доставки дейтаграмм на основе сетевого уровня IP. В этом документе описан отказоустойчивый метод определения MTU на пути (PMTUD), который можно применять с этими транспортными протоколами (или приложениями, использующими такой транспорт) для определения подходящего размера пакетов при передаче через Internet.

1.1. Классическое определение MTU на пути

Классический механизм определения максимального блока для передачи по пути (Path Maximum Transmission Unit Discovery или PMTUD) можно применять с любым транспортом, способным обрабатывать сообщения ICMP о слишком больших пакетах (Packet Too Big или PTB) (например, [RFC1191] и [RFC8201]). В этом документе термин «сообщение PTB» относится к сообщениям IPv4 ICMP Unreachable (ти 3) с кодом необходимости фрагментации (Fragmentation Needed, тип 3, код 4) [RFC0792] и ICMPv6 Packet Too Big (тип 2) [RFC4443]. При получении отправителем сообщения PTB он снижает эффективное значение MTU до размера, указанного как MTU на канале в сообщении PTB. Классический механизм PMTUD задаёт метод периодического увеличения размера пакетов в попытке определить рост поддерживаемого PMTU. Пакеты, размер которых превышает эффективное значение PMTU, называют пакетами зондирования или просто зондами.

Пакеты, не предназначенные для зондирования, фрагментируются до текущего эффективного PMTU или будут отброшены с возвратом кода ошибки. Приложениям может предоставляться примитив, позволяющий им считывать максимальный размер пакета (Maximum Packet Size или MPS), который выводится из текущего эффективного PMTU.

Классический механизм PMTUD подвержен отказам протокола. Один из отказов возникает при использовании пакетов больше действующего значения PMTU которые блокируются (все дейтаграммы больше фактического PMTU отбрасываются). Это может быть обусловлено тем, что сообщения PTB по каким-то причинам не передаются отправителю (см., например, [RFC2923]). Примеры случаев, когда сообщения PTB не доставляются, приведены ниже.

- Ограничена скорость генерации сообщений ICMP, в результате чего сообщения PTB для отправителя могут не создаваться (см. параграф 2.4 в [RFC4443]).
- Сообщения ICMP могут фильтроваться промежуточными устройствами, включая межсетевые экраны (МСЭ) [RFC4890]. МСЭ может быть настроен на блокирование входящих сообщений ICMP и это будет препятствовать доставке сообщений PTB передающей конечной точкой, находящейся за МСЭ.
- Когда маршрутизатор, отправляющий сообщение ICMP, отбрасывает туннельный пакет, сообщение ICMP направляется на вход туннеля. Эта конечная точка туннеля отвечает за пересылку сообщения ICMP,

обработку включённого в него пакета из поля данных с целью исключения туннельных заголовков и возврат корректно форматированного сообщения ICMP отправителю [TUNNELS]. Отказ в этих операциях препятствует доставке сообщения PTB исходному отправителю.

- Асимметрия в пересылке может приводит к тому, что не будет обратного пути к исходному отправителю, что может воспрепятствовать доставке сообщения ICMP. Эта проблема может возникнуть при использовании маршрутизации на основе правил или ECMP[1], а также при балансировке трафика приложений на промежуточных устройствах. Примером может служить маршрутизатор ECMP, выбирающий путь к серверу на основе байтов данных IP. В этом случае при возникновении проблем для переданного серверу пакета после маршрутизатора ECMP тому потребуется направить сообщение ICMP исходному отправителю.

- Возникают также ситуации, когда следующий интервал пересылки (next-hop) не способен принять пакет по причине его размера, например, в результате некорректной настройки уровня 2 между узлами, скажем, MTU в коммутаторе L2 или неверная настройка MRU[2]. Если пакет отбрасывается каналом, это не вызовет передачу сообщения PTB исходному отправителю.

Другой отказ может возникать, когда узел, не находящийся на пути, передаст сообщение PTB в попытке вынудить отправителя к смене эффективного PMTU [RFC8201]. Отправитель может защититься от таких сообщений, используя вложенную в сообщение PTB часть пакета для проверки генерации сообщения PTB в ответ на действительно отправленный пакет. Однако такая проверка возможна не всегда и примеры таких ситуаций приведены ниже.

- От создающего сообщение ICMP маршрутизатора RFC 792 [RFC0792] требует включать в него лишь 64 бита данных из исходного пакета IP. Этого может быть недостаточно для проверки сообщения отправителем.

  Примечание. RFC 1812 [RFC1812] рекомендует маршрутизаторам IPv4 включать в сообщение максимально возможную часть часть исходного пакета, пока дейтаграмма ICMP не превышает 576 байтов. Маршрутизаторы IPv6 включают максимально возможное число байтов исходного пакета, пока пакет ICMPv6 не превышает 1280 байтов [RFC4443].

- Использование туннелей и/или шифрования может снизить размер возвращаемой части пакета, повышая риск недостаточности включённых данных для проверки отправителем.

- Даже при включении в сообщение PTB достаточной части исходного пакета, на сетевом уровне может не быть контекста, достаточного для проверки сообщения, поскольку она зависит от сведений об активных транспортных потоках на конечном узле (например, использованная пара сокет-адрес и другие данные из протокольного заголовка).

- При инкапсуляции (туннелировании) пакета через шифрованный транспорт вход туннеля может не иметь контекста или вычислительных ресурсов для восстановления транспортного заголовка, нужного для проверки.

- При генерации сообщения ICMP маршрутизатором в сегменте сети, вставившем заголовок в пакет, включённая в сообщение часть исходного пакета может содержать дополнительные протокольные заголовки, которых не было в исходном пакете и которые отправитель PL не обрабатывает или не знает, как обработать. Это может помешать отправителю при проверке сообщения PTB.

- Транслятор адресов (Network Address Translation или NAT), преобразующий заголовок пакета, должен также транслировать сообщения ICMP и обновлять содержащуюся в них часть исходного пакета [RFC5508]. При некорректной трансляции отправитель не сможет связать сообщение с PL, передавшим пакет и проверить сообщение ICMP не удастся.

1.2. Определение MTU для пути на уровне пакетизации

Термин «уровень пакетизации» (Packetization Layer или PL) был введён для описания уровня, отвечающего за размещение блоков данных в поле данных (payload) пакетов IP и выбор подходящего MPS. Эта функция зачастую реализуется транспортным протоколом (например, DCCP, RTP, SCTP, QUIC), но может выполняться и другими методами инкапсуляции, работающими над транспортным уровнем.

В отличие от PMTUD, механизм определения MTU на пути уровня пакетизации (PLPMTUD) [RFC4821] применяет метод, не основанный на получении и проверке сообщений PTB. Поэтому данный механизм более устойчив к отказам по сравнению с классическим PMTUD и стал рекомендуемым подходом к определению PMTU [BCP145].

Этот документ обновляет [RFC4821], задавая метод PLPMTUD для дейтаграммных PL, а также обновляет [BCP145], указывая заданный здесь метод для использования с дейтаграммами UDP вместо метода из [RFC4821].

Метод применяет общую стратегию, где PL передаёт пакеты зондирования в поиске наибольшего размера нефрагментированных дейтаграмм, которые можно передать по пути через сеть. Пакеты-зонды передаются для исследования с использованием наибольшего размера. Если зонд успешно доставлен (как определено PL), PLPMTU увеличивается до размера этого зонда. При обнаружении «чёрной дыры» (например, когда пакеты размером PLPMTU постоянно не доставляются) значение PLPMTU снижается. PLPMTUD для дейтаграмм обеспечивает гибкость реализации. С одной стороны, его можно настроить лишь на обнаружение чёрных дыр и восстановление с повышенной отказоустойчивостью по сравнению с Classical PMTUD. С другой стороны, обработку PTB можно полностью отключить и PLPMTUD заменит Classical PMTUD. PLPMTUD может также включать проверку согласованности без повышения риска потери данных при зондировании для определения Path MTU. Например, сведения, доступные на уровне PL или вышележащем уровне позволяют проверить сообщения PTB перед их использованием.

1.3. Определение Path MTU для служб на основе дейтаграмм

В разделе 5 этого документа представлен набор алгоритмов для протоколов на основе дейтаграмм, позволяющих определить наибольший размер нефрагментированной дейтаграммы, которая может быть передана по пути через сеть. Метод основан на свойствах PL, описанных в разделе 3 и применяемых к транспортным протоколам,

[1]Equal-Cost Multipath - множество равноценных путей.
[2]Maximum Receive Unit - максимальный принимаемый блок.

использующим IPv4 и IPv6. Метод не требует взаимодействия с нижележащими уровнями, но может использовать сообщения PTB, когда они доступны уровню PL.

В рекомендациях по определению размера сообщений параграфа 3.2 в [BCP145] сказано: «приложению **следует** использовать данные о MTU на пути от уровня IP или самостоятельно реализовать определение MTU на пути (Path MTU Discovery или PMTUD)», но не представлен механизм определения наибольшего размера нефрагментированных дейтаграмм на пути через сеть. Этот документ обновляет RFC 8085, задавая этот метод вместо PLPMTUD [RFC4821] и предоставляет механизм совместного использования определённого наибольшего размера как MPS (параграф 4.4).

Параграф 10.2 в [RFC4821] рекомендует метод зондирования PLPMTUD для протокола SCTP (Stream Control Transport Protocol). SCTP использует пакеты зондирования содержащие блоки HEARTBEAT минимального размера с блоком PAD, как указано в [RFC4820]. Однако в RFC 4821 не задана полная спецификация. Этот документ обеспечивает её.

Протокол DCCP (Datagram Congestion Control Protocol) [RFC4340] требует от реализаций поддержки Classical PMTUD и говорит, что отправитель DCCP: «**должен** поддерживать значение максимального размера пакета (MPS), разрешённого для каждой активной сессии DCCP». Протокол также определяет текущее значение MPS для контроля перегрузки (CCMPS[1]), поддерживаемое на сетевом пути. Документ рекомендует использование PMTUD и предлагает применять пакеты управления (DCCP-Sync) в качестве зондов, поскольку это не вносит риска потери данных приложения. Определённый в этой спецификации метод может применяться с DCCP.

В разделах 4 и 5 определены протокольные механизмы и спецификация определения MTU для пути на уровне пакетизации дейтаграмм (Datagram Packetization Layer Path MTU Discovery или DPLPMTUD). В разделе 6 задан метод для транспорта дейтаграмм и представлены сведения для реализации PLPMTUD с другим транспортом на основе дейтаграмм и приложениями, использующими такой транспорт. В разделе 6 представлены также рекомендации для конечных точек SCTP, обновляющие [RFC4960], [RFC6951], [RFC8261] с целью использования описанного здесь метода вместо предложенного в [RFC4821].

2. Терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

Ниже приведены определения используемых в документе терминов. Соответствующие термины напрямую скопированы из [RFC4821], применяются также определения из [RFC1122].

Acknowledged PL - PL с подтверждениями
: Уровень PL, включающий механизм, который может подтверждать доставку дейтаграмм удалённой конечной точке PL (например, SCTP). Обычно получатель PL возвращает подтверждения, соответствующие принятым дейтаграммам, которые можно использовать для обнаружения блокировки пакетов («чёрных дыр»). См. также Unacknowledged PL.

Actual PMTU - фактическое значение PMTU
: Значение PMTU в сети между PL отправителя и получателя, которое алгоритм DPLPMTUD пытается определить.

Black Hole - "чёрная дыра"
: «Чёрная дыра» возникает, когда отправитель не знает, доставлены ли его пакеты получателю. С DPLPMTUD связаны два типа «чёрных дыр»:
  - пакеты попадают в «чёрную дыру», когда они не доставляются конечному получателю (например, отправитель передаёт пакеты определённого размера с неизвестным заранее PMTU и они отбрасываются в сети);
  - «чёрная дыра» ICMP возникает, когда отправитель не знает, что пакеты не доставлены адресату, поскольку сообщения PTB не приходят в PL исходного отправителя.

Classical Path MTU Discovery - классический механизм определения Path MTU
: Classical PMTUD - процесс, описанный в [RFC1191] и [RFC8201], где узлы полагаются на сообщения PTB для определения наибольшего размера нефрагментированных пакетов, которые можно передать по сетевому пути.

Datagram - дейтаграмма
: Блок данных транспортного протокола, передаваемый в поле данных (payload) пакета IP.

DPLPMTUD
: Определение MTU для пути на уровне пакетизации дейтаграмм - это PLPMTUD с использованием протокола доставки дейтаграмм.

Effective PMTU - эффективное значение PMTU
: Текущая оценка значения PMTU, используемая PMTUD. Это эквивалент PLPMTU, выведенного PLPMTUD с добавлением размера всех заголовков ниже уровня PL, включая заголовки IP.

EMTU_S
: Эффективное значение MTU для передачи (EMTU_S), определённое в [RFC1122] как «максимальный размер передаваемой дейтаграммы IP для конкретной комбинации отправитель – получатель».

EMTU_R
: Эффективное значение MTU для приёма (EMTU_S), определённое в [RFC1122] как «максимальный размер дейтаграммы, которая может быть собрана».

Link - канал
: Средство связи или среда, через которые узлы могут взаимодействовать на канальном уровне, т. е. ниже уровня IP. Примерами могут служить ЛВС Ethernet и туннели уровня Internet (или более высокого).

Link MTU - MTU для канала
: Размер (в байтах) наибольшего пакета IP с учётом заголовка IP и данных, который может быть передан через канал. Отметим, что это корректней называть IP MTU в соответствии с терминологией, применяемой другими органами стандартизации. Значение учитывает заголовок IP, но не учитывает заголовки канального уровня и кадрирование, которые не относятся к IP. Другие органы стандартизации обычно учитывают в MTU для канала и

[1]Congestion control MPS.

заголовки канального уровня. Данная спецификация следует [RFC4821], где сказано: «Для всех каналов **должно** обеспечиваться соответствие своим MTU – при недетерминированном получении каналом пакета с размером больше установленного для канала MTU такие пакеты **должны** отбрасываться».

MAX_PLPMTU
Наибольшее значение PLPMTU, которое механизм DPLPMTUD пытается использовать (см. 5.1.2. Константы).

MIN_PLPMTU
Наименьшее значение PLPMTU, которое механизм DPLPMTUD пытается использовать (см. 5.1.2. Константы).

MPS
Максимальный размер пакета - это размер наибольшего блока данных приложения, который может быть передан по пути через сеть уровнем PL в одной дейтаграмме (см. 4.4. Максимальный размер пакета).

MSL
Максимальный срок жизни сегмента - это наибольшая задержка пакета, ожидаемая на пути. В [BCP145] задано значение 2 минуты.

Packet - пакет
Заголовок IP с расширениями и опциями, а также данные IP (payload).

Packetization Layer (PL) - уровень пакетизации
Уровень сетевого стека, помещающий данные в пакеты и выполняющий функции транспортного протокола. Примерами PL являются TCP, SCTP, SCTP на основе UDP, SCTP на основе DTLS, QUIC.

Path - путь
Набор каналов и маршрутизаторов, через которые проходит поток между узлом-источником и узлом-получателем.

Path MTU (PMTU) - MTU для пути
Минимальное из значение MTU для канала среди всех каналов на пути от источника к получателю, используемое PMTUD.

PTB
В этом документе термин «сообщение PTB» относится к сообщениям IPv4 ICMP Unreachable (тип 3) с ошибкой Fragmentation Needed (тип 3, код 4) [RFC0792] и ICMPv6 Packet Too Big (тип 2) [RFC4443].

PTB_SIZE
Значение, переданное в проверенном сообщении PTB, которое указывает MTU канала к следующему маршрутизатору на пути.

PL_PTB_SIZE
Значение, переданное в проверенном сообщении PTB, из которого исключены все заголовки, добавленные уровнями ниже PL.

PLPMTU
Значение PMTU уровня пакетизации, которое является оценкой максимального размера дейтаграммы PL, которая может быть передана по пути, контролируемому PLPMTUD.

PLPMTUD
Определение Path MTU для уровня пакетизации - описанный в этом документе метод для дейтаграммных PL, который является расширением Classical PMTU Discovery.

Probe packet - пакет зондирования
Дейтаграмма преднамеренного размера (обычно текущее значение PLPMTU или больше), передаваемая для проверки возможности сквозной доставки пакетов этого размера по пути через сеть.

Unacknowledged PL - PL без подтверждения
Уровень PL без механизма подтверждения доставки удалённой точке PL (например, UDP), которому требуется DPLPMTUD в качестве механизма обнаружения блокировки пакетов («чёрных дыр»). См. также Acknowledged PL.

3. Свойства, требуемые для Datagram PLPMTUD

Принципы [RFC4821] применимы к использованию метода с любым уровнем PL. Был определён TCP PLPMTUD со стандартными механизмами протокола TCP. Дейтаграммные PL, в отличие от TCP, требуют дополнительных механизмов и соображений в части реализации PLPMTUD, как указано ниже.

1. Управление PLPMTU. Для дейтаграммных PL значением PLPMTU управляет DPLPMTUD. Уровню PL **недопустимо** передавать дейтаграммы (кроме пробных пакетов), размер которых на уровне PL больше текущего PLPMTU.

2. Пакеты зондирования. От сетевого интерфейса ниже уровня PL **требуется** обеспечивать способ передачи пробных пакетов размером больше PLPMTU. В IPv4 пробные пакеты **должны** передаваться с флагом запрета фрагментирования (DF) в заголовке IP и без фрагментации конечной точкой сетевого уровня. В IPv6 пробные пакеты всегда передаются без фрагментации источником (как указано в параграфе 5.4 [RFC8201]).

3. Обратная связь при получении. От конечного получателя PL **требуется** метод обратной связи, указывающий отправителю DPLPMTUD приём пробного пакета конечным получателем PL. В разделе 6 представлены примеры подтверждения уровнем PL приёма пробных пакетов.

4. Восстановление при потере зондов. Для зондирования **рекомендуется** использовать пакеты, не содержащие пользовательских данных, которые потребуется повторять при потере. Большинство средств транспорта дейтаграмм разрешает это. Если пробный пакет содержит пользовательские данные, которые требуется повторять при потере, от PL (или вышележащих уровней) **требуется** организовать повторную передачу и/или восстановление при любых потерях. От PL **требуется** отказоустойчивость в случаях потери пакетов по другим причинам (включая ошибки канального уровня и перегрузку).

5. Параметры PMTU. Отправителю DPLPMTUD **рекомендуется** использовать сведения о максимальном размере пакета, которым он может по локальному каналу (например, MTU локального канала). Отправитель PL **может** использовать подобную информацию о максимальном размере пакета сетевого уровня, который получатель может воспринять (отметим, что это может быть меньше EMTU_R). Это избавляет реализации от попыток передать пробные пакеты, которые локальный канал не может передать. Слишком высокое значение может снизить эффективность алгоритма поиска. В некоторых приложениях также устанавливается максимальный размер транспортного блока (protocol data unit или PDU) и тогда не будет пользы от проверки больших размеров (если транспорт не позволяет мультиплексировать несколько PDU приложения в одну дейтаграмму).

6. Обработка сообщений PTB. Отправитель DPLPMTUD **может** использовать сообщения PTB от сетевого уровня как помощь при определении невозможности передачи пробного пакета через сеть. Полученные сообщения PTB **должны** проверяться перед их использованием для обновления информации PLPMTU [RFC8201]. Эта проверка подтверждает отправку сообщения PTB в ответ на пакет от отправителя и её нужно выполнять до реагирования метода определения PLPMTU на сообщение PTB. **Недопустимо** использовать сообщение PTB для увеличения PLPMTU [RFC8201], но можно инициировать зонд для проверки большего PLPMTU. Действительное значение PTB_SIZE преобразуется в PL_PTB_SIZE перед использованием в конечном автомате DPLPMTUD. Значение PL_PTB_SIZE, превышающее текущее проверенное значение, **следует** игнорировать (это сообщение PTB нужно отбросить без обработки, но можно использовать в качестве триггера перехода в режим устойчивости).

7. Зондирование и контроль перегрузок. PL **может** использовать контроллер перегрузок для решения вопроса об отправке пробного пакета. Если передача зондов ограничивается контроллером перегрузок, это может привести к задержке или приостановке передачи пробных пакетов во время перегрузки. Когда контроллер перегрузки не управляет передачей зондов, интервал отправки **должен** быть не меньше RTT. Потерю пробного пакета **не следует** считать индикацией перегрузки, а механизму контроля перегрузки **не следует** реагировать на неё [RFC4821], поскольку это может вести к необоснованному снижению скорости передачи. При обновлении PLPMTU (или MPS) **недопустимо** увеличивать окно перегрузок, измеренное в байтах [RFC4821]. Следовательно, рост размера пакета не вызывает увеличения скорости данных в байтах за секунду. PL с поддержкой окна перегрузки на основе ограничения числа ожидающих пакетов фиксированного размера **следует** изменить этот предел для компенсации фактического увеличения пакетов. Передача пробных пакетов может взаимодействовать с работой PL по сглаживанию пиков трафика и PL может потребоваться передача пробных пакетов в соответствии с этими методами.

8. Зондирование и управление потоком данных. Управление потоком данных в PL связано со сквозными потоками данных, использующими услуги PL. Управление потоком данных **не следует** применять к DPLPMTU, если в зондах не передаются пользовательские данные для удалённого приложения.

9. Общее состояние PLPMTU. Значение PMTU, рассчитанное из PLPMTU, **может** сохраняться в соответствующей записи, связанной с адресатом в кэше уровня IP и используемой другими экземплярами PL. В спецификации PLPMTUD [RFC4821] сказано: «Если PLPMTUD обновляет MTU для определённого пути, все сессии уровня пакетизации, использующие этот путь (см. параграф 5.2), **следует** уведомить об использовании нового MTU». Такие методы **должны** быть устойчивы к широкому спектру режимов пересылки базовой сети. В параграфе 5.2 [RFC8201] приведены рекомендации по кэширования данных PMTU и связи с метками потоков IPv6.

Кроме того, для разработки метода DPLPMTUD сформулирован ряд приведённых ниже принципов.

- PL **может** сегментировать блоки данных размером больше MPS в несколько дейтаграмм. Однако не все дейтаграммные PL поддерживают такое сегментирование. Методам **рекомендуется** избегать принуждения приложений к использованию произвольно малых значений MPS для передачи, пока выполняется поиск текущего поддерживаемого PLPMTU. Снижение MPS может снижать производительность приложения.

- Чтобы помочь приложениям выбрать подходящий размер блока данных, уровню PL **рекомендуется** обеспечивать примитив, возвращающий значение MPS, выведенное из PLPMTU, вышележащему уровню, который использует PL. Значение MPS может меняться при смене пути или потере пробных пакетов.

- Проверка пути. Методам **рекомендуется** обеспечивать устойчивость к смене пути, которая могла произойти послед подтверждения характеристик пути, и возможности получения несогласованных сведений о пути.

- Нарушение порядка дейтаграмм. От метода **требуется** устойчивость к возможному нарушению порядка или разделению трафика (включая пробные пакеты) между несколькими путями через сеть.

- Задержка и дублирование дейтаграмм. **Требуется** механизм обратной связи, устойчивый к возможности значительной задержки пакетов или их дублирования в сети.

- Момент зондирования. Методам **рекомендуется** определять, менялся ли путь после его измерения. Это позволить определить момент нового зондирования.

4. Механизмы DPLPMTUD

В этом разделе описаны механизмы, используемые в спецификации.

4.1. Пакеты зондирования PLPMTU

Метод DPLPMTUD основан на возможности отправителя PL генерировать пробные пакеты заданного размера. TCP может создавать такие пакеты выбирая соответствующий сегмент передаваемых данных [RFC4821]. В PL на основе дейтаграмм это потребовало бы запросить у приложения передачу блока данных, размер которого превышает генерируемых приложением, или использовать функции заполнения для увеличения размера дейтаграмм. Протоколы, разрешающие обмен управляющими сообщениями (без блока данных приложения), могут генерировать пробные пакеты, просто дополняя управляющее сообщение данными заполнения. Общий размер пробного пакета включает все заголовки и заполнение, а также передаваемые данные (например, поля опций протокола, связанные с защитой поля, такие как тег AEAD[1], и заполнения уровня TLS record).

От получателя **требуется** способность отличать блок данных от любого заполнения, чтобы то не передавалось приложению. В результате у отправителя есть три возможных способа создания пробного пакета.

Зондирование с использованием данных заполнения

Пробный пакет содержит данные управления и заполнение, требуемое для увеличения размера. Поскольку такие пакеты не передают представленного приложением блока данных, для них обычно не требуется повторная передача, но они все равно расходуют пропускную способность и требуют обработки на конечной точке.

[1]Authenticated Encryption with Associated Data - аутентифицированное шифрование со связанными данными.

Зондирование с использованием данных приложения и заполнения

Пробный пакет содержит блок данных приложения и заполнение, требуемое для увеличения размера.

Зондирование с использованием данных приложения

Пробный пакет содержит блок данных приложения, соответствующий размеру пробного пакета. Этот метод запрашивает у приложения блок данных нужного для зонда размера.

Уровень PL, использующий пробный пакет с данными приложения, для которого требуется защита от потери этого пакета, может выполнить повтор (восстановление) этого блока данных на транспортном уровне (например, повторяя передачу после обнаружения потери или дублируя блок данных в дейтаграмме без заполнения). Повторная передача блока данных может потребовать меньшего значения PLPMTU, что может вынудить PL использовать пакет меньшего размера для сквозного прохождения пути (для этого может использоваться фрагментация на сетевом уровне конечной точки или PL может заново сегментировать блок данных на несколько дейтаграмм).

Для упрощения реализации DPLPMTUD **может** ограничиться использованием лишь одного из указанных методов.

Пробные сообщения, передаваемые PL, **должны** содержать достаточно информации для однозначной идентификации зонда в интервале MSL (например, это может быть уникальный идентификатор от PL или реализации DPLPMTUD) с сохранением устойчивости к нарушению порядка и повторному использованию откликов на зонды и сообщений PTB.

4.2. Подтверждение размера пробных пакетов

Уровню PL нужен метод определения (подтверждения) приёма пробных пакетов другой стороной через сеть. Транспортные протоколы могут включать сквозные методы обнаружения и информирования о приёме конкретных переданных дейтаграмм (например, DCCP, SCTP, QUIC имеют функции keep-alive/heartbeat). При поддержке таких механизмов они **могут** применяться в DPLPMTUD для подтверждения приёма пробных пакетов.

PL без подтверждения приёма данных (например, UDP и UDP-Lite) сам не может определить факт отбрасывания пакета с размером больше фактического PMTU. Таким PL приходится полагаться при обнаружении потерь на протокол приложения. В разделе 6 задана эта функция для набора протоколов IETF.

4.3. Обнаружение блокировки и снижение PLPMTU

Приведённое ниже описание использует константы из параграфа 5.1.2 и переменные из параграфа 5.1.3. Обнаружение блокировки «чёрной дыры) сигнализируется неспособностью пути через сеть поддерживать текущий размер PLPMTU. Возможны три индикатора блокировки, указанных ниже.

- Проверенное сообщение PTB, указывающее, что PL_PTB_SIZE меньше текущего PLPMTU. Методу DPLPMTUD **недопустимо** полагаться лишь на эту индикацию.
- PL может использовать механизм зондирования DPLPMTUD для периодической генерации пробных пакетов, размер которых равен текущему PLPMTU (например, с таймером CONFIRMATION_TIMER, параграф 5.1.1). Таймер отслеживает приём подтверждений. Последовательная потеря зондов служит индикатором невозможности поддержки путём текущего значения PLPMTU (например, когда число переданных и не подтверждённых пробных пакетов PROBE_COUNT становится больше MAX_PROBES).
- PL может использовать событие, указывающее, что сетевой путь больше не поддерживает принятое отправителем значение PLPMTU. Это может быть реализованный в PL механизм обнаружения чрезмерных потерь с принятием на этом основании решения о том, что потери обусловлены недопустимым PLPMTU (как в PLPMTUD для TCP [RFC4821]).

Эти три метода могут давать разные картины передачи пробных пакетов и для них предполагается разная реакция на изменение фактического PMTU.

PL **может** запретить отправку зондов при отсутствии данных приложения с момента отправки предыдущего пробного пакета. Уровень PL, возобновляющий передачу пользовательских данных, **может** продолжить определение PLPMTU для каждого пути, что позволяет использовать актуальное значение PLPMTU. Однако это может вызвать передачу дополнительных пакетов.

Когда метод определяет, что текущее значение PLPMTU не поддерживается, DPLPMTUD устанавливает меньшие значения PLPMTU и MPS. После этого PL подтверждает пригодность нового PLPMTU на пути через сеть. Для пробного пакета может потребоваться размер меньше блока данных, создаваемого приложением.

4.4. Максимальный размер пакета

Результат зондирования определяет пригодное значение PLPMTU, используемое для установки MPS, применяемого приложением. MPS меньше PLPMTU, поскольку его значение уменьшается на размер заголовков PL (включая связанные с защитой поля, такие как тег AEAD и заполнение уровня TLS record). Связь между MPS и PLPMTUD показана на рисунке 1.

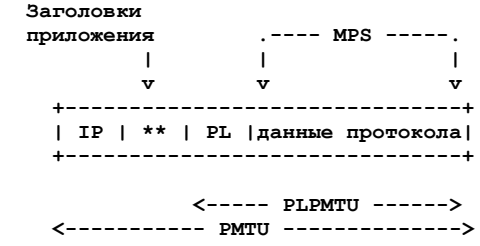

Рисунок 1. Связь между MPS и PLPMTU.

PL не может отправить пакет (кроме зонда) размером больше текущего PLPMTU на сетевом уровне. Для предотвращения этого PL **можно** спроектировать с сегментированием блоков данных больше MPS на несколько дейтаграмм.

DPLPMTUD стремится избежать фрагментации IP. Попытка передать блок данных размером больше MPS завершится отказом, если PL не может сегментировать данные. Для определения наибольшего блока данных, который можно передать, PL **следует** предоставлять приложениям примитив, возвращающий значение MPS, выведенное из текущего PLPMTU.

Если DPLPMTUD приводит к изменению MPS, приложению потребуется приспособиться к новому MPS. Может возникать случай, когда пакеты были переданы с размером меньше MPS, а затем было уменьшено значение PLPMTU. Если эти пакеты будут потеряны, PL **может** сегментировать данные с использованием нового MPS. Если PL не может повторно сегментировать переданную ранее дейтаграмму (например, [RFC4960]), отправитель отбрасывает дейтаграмму или может повторить передачу с использованием фрагментации на сетевом уровне для создания нескольких пакетов IP размером не более PLPMTU. Для IPv4 использование фрагментации передающей конечной точкой предпочтительней сброса бита DF в заголовке IPv4. Опыт эксплуатации показывает, что фрагментация IP может снижать гарантии доставки в Internet [RFC8900], что может снизить вероятность успеха при повторной передаче.

4.5. Отключение влияния PMTUD

Уровень PL, реализующий эту спецификацию, **должен** приостановить обработку на сетевом уровне, вынуждающую использовать PMTU [RFC1191][RFC8201], для каждого потока, применяющего DPLPMTUD, и вместо этого определять размер передаваемых в поток пакетов через DPLPMTUD. Это избавляет сетевой уровень от необходимости отбрасывать или фрагментировать переданные пакеты с размером больше PMTU.

4.6. Отклики на сообщения PTB

Этот метод требует от отправителя DPLPMTUD проверять полученные сообщения PTB до использования сведений из них. Отклик на сообщение PTB зависит от значения PL_PTB_SIZE, рассчитанного из PTB_SIZE в сообщении PTB, состояния конечного автомата PLPMTUD и применяемого протокола IP.

В параграфе 4.6.1 описана проверка для сообщений IPv4 ICMP Unreachable (тип 3) и ICMPv6 Packet Too Big, которые совместно называются в этом документе сообщениями PTB.

4.6.1. Проверка сообщения PTB

В этом параграфе описано использование и проверка сообщений PTB:

- простая реализация **может** игнорировать полученные сообщения PTB и PLPMTU не будет обновляться в результате их получения;
- уровень PL с поддержкой сообщения PTB **должен** проверять эти сообщения до дальнейшей обработки.

Уровень PL, получивший сообщение PTB от маршрутизатора или промежуточного устройства, выполняет проверку ICMP (раздел 4 в [RFC8201] и параграф 5.2 в [BCP145]). Поскольку DPLPMTUD работает на уровне PL, этот уровень должен проверить, что каждое принятое сообщение PTB создано в ответ на пакет, переданный конечной точкой PL, выполняющей DPLPMTUD.

Уровень PL **должен** проверить протокольные данные в полученной в сообщении ICMP PTB части исходного пакета, чтобы убедиться в отправке пакета им. Эта проверка включает сопоставление адресов IP, протокола, портов отправителя и получателя, чтобы направить сообщение PTB соответствующему PL.

При проверке **следует** использовать сведения, которые атакующему вне пути передачи сложно определить [BCP145]. Например, можно проверить значение поля в заголовке протокола известное лишь двум конечным точкам PL. Приложения на основе дейтаграмм, использующие общеизвестные порты у отправителя и получателя, должны также полагаться при проверке на другие сведения.

Проверка предназначена для защиты от пакетов, переданных узлами, не находящимися на пути через сеть. Не прошедшие проверку сообщения PTB **недопустимо** использовать в DPLPMTUD, как указано в разделе 8. Вопросы безопасности. Обработка сообщений PTB описана в параграфе 4.6.2.

4.6.2. Использование сообщений PTB

Проверенные сообщения PTB **можно** применять в алгоритме DPLPMTUD, но **недопустимо** напрямую использовать для установки PLPMTU. Перед использованием размера, указанного в сообщении PTB его нужно преобразовать в PL_PTB_SIZE. Значение PL_PTB_SIZE меньше PTB_SIZE, поскольку оно не включает заголовки уровней ниже PL, опции IP и расширения, добавленные к пакету PL.

Метод, использующий сообщения PTB, может повысить скорость определения алгоритмом подходящего PLPMTU, инициируя незамедлительное зондирование для размера PL_PTB_SIZE (создающего пакет сетевого уровня размером PTB_SIZE) вместо выполнения зондирования исключительно по таймеру.

Набор проверок предназначен для защиты от маршрутизаторов, сообщающих неожиданное значение PTB_SIZE. Уровню PL нужно также убедиться, что значение PL_PTB_SIZE меньше используемого размера пробных пакетов и не меньше минимально допустимого размера. Ниже приведена сводка использования сообщений PTB с константами из параграфа 5.1.2. Обработка зависит от PL_PTB_SIZE и текущих значение переменных, как показано ниже.

`PL_PTB_SIZE < MIN_PLPMTU`

- Непригодный размер PL_PTB_SIZE, см. параграф 4.6.1.
- Сообщение PTB нужно отбросить без обработки (PLPMTU не меняется).
- Информацию можно использовать как триггер включения режима устойчивости (параграф 5.3.3).

`MIN_PLPMTU < PL_PTB_SIZE < BASE_PLPMTU`

- Отказоустойчивый уровень PL **может** ввести состояние Error (5.2. Конечный автомат) для пути IPv4, когда значение PL_PTB_SIZE из сообщения PTB не меньше 68 байтов [RFC0791] но меньше BASE_PLPMTU.

- Отказоустойчивый уровень PL **может** ввести состояние Error (5.2. Конечный автомат) для пути IPv6, когда значение PL_PTB_SIZE из сообщения PTB не меньше 1280 байтов [RFC8200], но меньше BASE_PLPMTU.

BASE_PLPMTU <= PL_PTB_SIZE < PLPMTU

- Это может быть индикацией чёрной дыры. Для PLPMTU **следует** установить значение BASE_PLPMTU (для предотвращения ненужных потерь пакетов при возникновении блокировки).
- PL нужно запустить поиск для быстрого определения нового PLPMTU. Для инициализации алгоритма поиска можно использовать PL_PTB_SIZE из сообщения PTB.

PLPMTU < PL_PTB_SIZE < PROBED_SIZE

- PLPMTU остаётся действительным, но размер использованных для поиска пакетов (PROBED_SIZE) был больше фактического PMTU.
- PLPMTU не обновляется.
- PL может применять PL_PTB_SIZE из сообщения PTB в качестве следующей точки поиска при возобновлении алгоритма.

PL_PTB_SIZE >= PROBED_SIZE

- Несогласованный сигнал из сети.
- Сообщение PTB нужно отбросить без обработки (PLPMTU не меняется).
- Информацию можно использовать как триггер включения режима устойчивости.

5. PMTUD уровня пакетизации дейтаграмм

В этом разделе описывается PLPMTUD для дейтаграмм (DPLPMTUD). Метод может применяться в различных точках (* на рисунке 2) стека протоколов IP для определения PLPMTU, позволяющего приложения использовать подходящее значение MPS для текущего пути через сеть.

DPLPMTUD **следует** применять лишь на одном уровне между парой конечных точек. Поэтому вышележащему уровню PL или приложения следует избегать использования DPLPMTUD если этот метод уже включён на нижележащем уровне. Уровень PL **должен** установить значение MPS, указанное DPLPMTUD, с учётом всех дополнительных «издержек», вносимых PL.

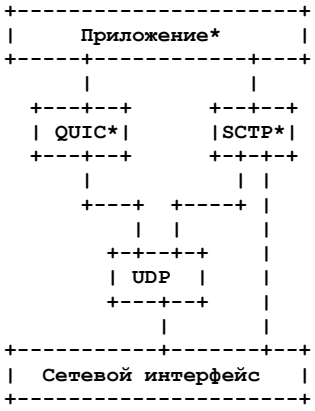

Рисунок 2. Примеры реализации DPLPMTUD.

Основная идея DPLPMTUD состоит в зондировании со стороны отправителя. Пробные пакеты передаются для определения максимального размера пользовательских сообщений, которые могут быть полностью переданы по пути через сеть между отправителем и получателем.

В последующих параграфах рассматриваются компоненты, требуемые для реализации, представлен обзор фаз работы, задан конечный автомат и алгоритм поиска.

5.1. Компоненты DPLPMTUD

В этом параграфе описаны таймеры, константы и переменные DPLPMTUD.

5.1.1. Таймеры

Метод использует 3 таймера.

PROBE_TIMER

Этот таймер настраивается на время максимального ожидания подтверждения пробного пакета. Для таймера **недопустимо** значение меньше 1 секунды и **следует** устанавливать значение больше 15 секунд. Рекомендации по выбору значения таймера приведены в параграфе 3.1.1 [BCP145].

PMTU_RAISE_TIMER

Время, в течение которого отправитель продолжает использовать текущее значение PLPMTU, перед входом в фазу поиска (Search Phase). Для этого таймера устанавливается значение 600 секунд в соответствии с рекомендацией PLPMTUD [RFC4821].

DPLPMTUD **может** запретить отправку пробных пакетов при отсутствии данных приложения с момента отправки предыдущей пробы. Уровень PL, предпочитающий использовать актуальное значение PMTU, после возобновления отправки пользовательских данных может продолжить определение PMTU для каждого пути. Однако это приведёт к передаче дополнительных пакетов.

CONFIRMATION_TIMER

При использовании PL с подтверждениями, применять этот таймер **недопустимо**. Для других PL таймер задаёт время, в течение которого отправитель PL ждёт перед подтверждением поддержки текущего PLPMTU. Это время

меньше PMTU_RAISE_TIMER и служит для уменьшения PLPMTU (например, при обнаружении чёрной дыры). Подтверждения должны быть достаточно частыми при передаче потока, чтобы передающий уровень PL не отправил в чёрную дыру слишком большой объем трафика. Рекомендации по установке этого таймера приведены в параграфе 3.1.1 [BCP145].

DPLPMTUD **может** запретить отправку пробных пакетов при отсутствии данных приложения с момента отправки предыдущей пробы. Уровень PL, предпочитающий использовать актуальное значение PMTU, после возобновления отправки пользовательских данных может продолжить определение PMTU для каждого пути. Однако это приведёт к передаче дополнительных пакетов.

DPLPMTUD задаёт разные таймеры, однако реализация может реализовать их функции с одним таймером.

5.1.2. Константы

MAX_PROBES

Максимальное значение счётчика зондов PROBE_COUNT (5.1.3. Переменные). MAX_PROBES ограничивает число последовательных зондов любого размера. Алгоритмы поиска выигрывают от значения MAX_PROBES больше 1, поскольку это повышает устойчивость к изолированной потере пакета. По умолчанию MAX_PROBES = 3.

MIN_PLPMTU

Наименьший размер PLPMTU, который DPLPMTUD пытается использовать. Конечной точке может потребоваться настройка MIN_PLPMTU для обеспечения пространства под заголовки расширения и иную инкапсуляцию ниже уровня PL. Значение может зависеть от интерфейса и пути. Для IPv6 этот размер не меньше размера на уровне PL, приводящего к 1280-байтовым пакетам IPv6, как указано в [RFC8200]. Для IPv4 этот размер не меньше размера PL, приводящего к 68-байтовым пакетам IPv4.

Примечание. От маршрутизатора IPv4 требуется способность пересылать дейтаграммы размером 68 байтов без дальнейшей фрагментации. Это включает заголовок IPv4 и фрагмент минимального размера 8 байтов. Кроме того, от получателей требуется способность собирать фрагментированные дейтаграммы размером по меньшей мере 576 байтов, как указано в параграфе 3.3.3 [RFC1122].

MAX_PLPMTU

Наибольшее значение PLPMTU. Оно должно быть не больше максимального размера пакета PL, который можно передать через выходной интерфейс (ограничивается MTU локального интерфейса). Значение должно также быть меньше максимального размера пакета PL, который может принять удалённая конечная точка (если этот размер известен), ограниченный EMTU_R. Значение параметра может быть ограничено реализацией или настройкой используемого уровня PL. Приложение или PL **может** выбрать меньшее значение MAX_PLPMTU, если не требуется передавать пакеты больше определённого размера.

BASE_PLPMTU

Настраиваемый размер, который считается подходящим для работы на большинстве путей. Это значение не меньше MIN_PLPMTU и меньше MAX_PLPMTU. Для большинства PL подходящим будет BASE_PLPMTU больше 1200 байтов. При использовании IPv4 эквивалентный размер не задан и по умолчанию для BASE_PLPMTU **рекомендуется** значение 1200 байтов.

5.1.3. Переменные

PROBED_SIZE

Текущий размер пробного пакета, определённый PL. Это предварительное значение PLPMTU, ожидающее подтверждения.

PROBE_COUNT

Число последовательно отправленных неудачных пробных пакетов. При каждом подтверждении пробного пакета для переменной устанавливается значение 0 (при поиске предполагается потеря некоторых зондов, поэтому однократная потеря пробного пакета не является индикацией проблемы PMTU.)

На рисунке 3 показаны связи между константами и переменными размера пакетов в момент, когда алгоритм DPLPMTUD выполняет зондирование пути для увеличения PLPMTU, передавая пробный пакет размера PROBED_SIZE. После подтверждения значение PLPMTU увеличивается до PROBED_SIZE, позволяя алгоритму DPLPMTUD увеличить PROBED_SIZE для отправки зонда с размером, приближающимся к фактическому PMTU.

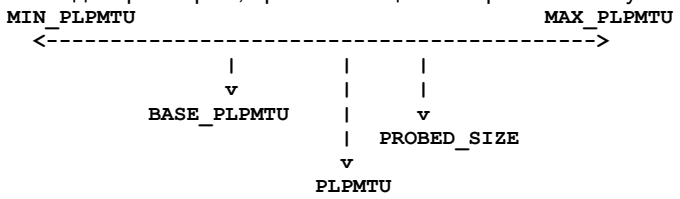

Рисунок 3. Связи между константами и переменными.

5.1.4. Фазы DPLPMTUD

В этом параграфе представлен высокоуровневый информативный обзор метода DPLPMTUD путём описания нескольких фаз работы метода. Более подробное описание конечного автомата дано в параграфе 5.2.

Base

Фаза Base подтверждает соединение с удаленным узлом, используя пакеты BASE_PLPMTU. Подтверждение связности неявно для ориентированного на соединения PL (может выполняться согласованием соединения PL), а в PL без соединений передаются пробные пакеты и используются подтверждения этих зондов, говорящие о достижимости партнёра. Отправитель также подтверждает применимость размера BASE_PLPMTU на пути через сеть. Это может быть достигнуто с помощью механизма PL (например, передачей пакета согласования размером BASE_PLPMTU) или отправкой зондов размера BASE_PLPMTU и подтверждением их получения.

Пробный пакет размера BASE_PLPMTU может быть отправлен сразу при входе в фазу Base (после проверки связности). PL без поддержки путей с PLPMTU меньше BASE_PLPMTU может упростить фазу до одного шага, выполняя проверку связности пробным пакетом размера BASE_PLPMTU.

После подтверждения DPLPMTUD переходит в фазу поиска (Search). Если фаза Base не смогла подтвердить BASE_PLPMTU, механизм DPLPMTUD переходит в фазу Error.

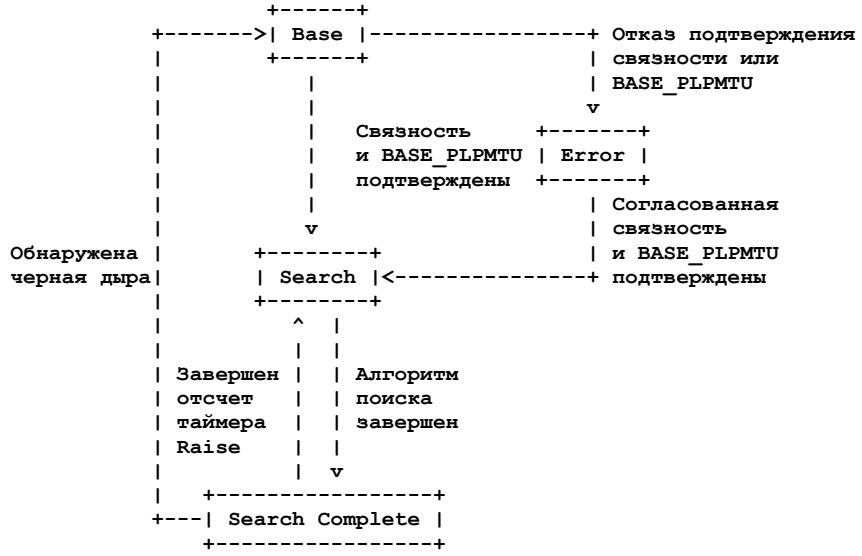

Рисунок 4. Фазы DPLPMTUD.

Search

Фаза Search использует алгоритм поиска для передачи пробных пакетов с целью увеличения PLPMTU. Алгоритм заключает, что найдено подходящее значение PLPMTU, переходом в фазу Search Complete.

PL может отвечать на сообщения PTB, используя PTB для продолжения или прерывания поиска (параграф 4.6).

Search Complete

Фаза завершённого поиска (Search Complete) начинается, когда PLPMTU поддерживается на сетевом пути. PL может использовать CONFIRMATION_TIMER для периодического повторения пробного пакета с текущим размером PLPMTU. Если отправитель не способен подтвердить достижимость (например, истекло время CONFIRMATION_TIMER) или PL сообщает о недостижимости, это считается обнаружением чёрной дыры и DPLPMTUD переходит в фазу Base.

Используется PMTU_RAISE_TIMER для периодического возобновления фазы поиска с целью определения возможности поднять PLPMTU. Обнаружение чёрной дыры вызывает переход в фазу Base.

Error

Фаза Error вводится при наличии конфликтующих или недействительных сведений PLPMTU для пути (например, отказ поддержки BASE_PLPMTU), которые делают невозможной работу DPLPMTUD и PLPMTU снижается.

DPLPMTUD остаётся в фазе Error, пока не будет найдено согласованное представление пути, а также подтверждена поддержка BASE_PLPMTU на пути (или DPLPMTUD приостанавливается).

Метод, который лишь уменьшает PLPMTU до подходящего размера, будет достаточным для обеспечения надёжной работы, но будет неэффективен при изменении фактического PMTU или при выборе методом (или иным путём) неоптимального PLPMTU.

Полная реализация DPLPMTUD предоставляет алгоритм, позволяющий отправителю DPLPMTUD увеличивать PLPMTU после изменения характеристик пути, таких как перенастройка канала с большим MTU или при смене набора проходимых каналов для сквозного потока (например, при смене маршрута или восстановлении после отказа).

5.2. Конечный автомат

Конечный автомат для DPLPMTUD показан на рисунке 5. При поддержке множества путей или многодомности автомат нужен для каждого пути. На рисунке для упрощения некоторые переходы не показаны.

Ниже перечислены состояния конечного автомата.

DISABLED

Начальное состояние до зондирования. В это состояние возможен переход из другого, если PL указывает потерю связности. Состояние сохраняется после того, как PL покажет связность с удаленным PL. При переходе в состояние BASE незамедлительно можно передать пробный пакет размером BASE_PLPMTU.

BASE

Состояние BASE служит для подтверждения поддержки размера BASE_PLPMTU на пути через сеть и предназначено для того, чтобы приложение могло продолжить работу при временном сокращении фактического PMTU. Это также позволяет избежать долгих периодов, когда отправитель, ищущий большее значение PLPMTU, не знает о том, что пакеты не были доставлены по причине блокировки их или ICMP (чёрная дыра).

При входе в состояние устанавливается PROBED_SIZE = BASE_PLPMTU и PROBE_COUNT = 0.

При каждой отправке пробного пакета запускается таймер PROBE_TIMER. Состояние прерывается, когда пробный пакет подтверждён, и отправитель PL переходит в состояние SEARCHING.

Состояние также прерывается, когда PROBE_COUNT достигает MAX_PROBES или проверяется полученное сообщение PTB. Это заставляет отправителя PL перейти в состояние ERROR.

SEARCHING

Состояние SEARCHING является основной частью зондирования и вводится по завершении зондирования для размера BASE_PLPMTU.

При каждом подтверждении пробного пакета устанавливается PROBE_COUNT = 0, PLPMTU = PROBED_SIZE, а значение PROBED_SIZE увеличивается с использованием алгоритма поиска (как описано в параграфе 5.3).

Когда переданный пробный пакет не подтверждён в интервале PROBE_TIMER, значение PROBE_COUNT инкрементируется и передаётся новый пробный пакет.

Состояние завершается переходом в SEARCH_COMPLETE, когда PROBE_COUNT достигает MAX_PROBES, принимается проверенное сообщение PTB, соответствующее последнему успешно проверенному размеру (PL_PTB_SIZE = PLPMTU), или подтверждается зонд размера MAX_PLPMTU (PLPMTU = MAX_PLPMTU).

При обнаружении чёрной дыры в состоянии SEARCHING отправитель PL переходит в состояние BASE.

SEARCH_COMPLETE

Состояние SEARCH_COMPLETE указывает завершения поиска. Это нормальное состояние поддержки, когда PL не выполняет зондирования для обновления PLPMTU. DPLPMTUD остаётся в этом состоянии до завершения отсчёта PMTU_RAISE_TIMER или обнаружения чёрной дыры.

Когда DPLPMTUD использует PL без подтверждений и находится в состоянии SEARCH_COMPLETE, таймер CONFIRMATION_TIMER периодически сбрасывает PROBE_COUNT и планирует пробный пакет размера PLPMTU. Если MAX_PROBES последовательных зондов размера PLPMTUD не подтверждены, метод переходит в состояние BASE. При использовании с подтверждаемым PL (например, SCTP) методу DPLPMTUD не следует продолжать генерацию зондов PLPMTU в этом состоянии.

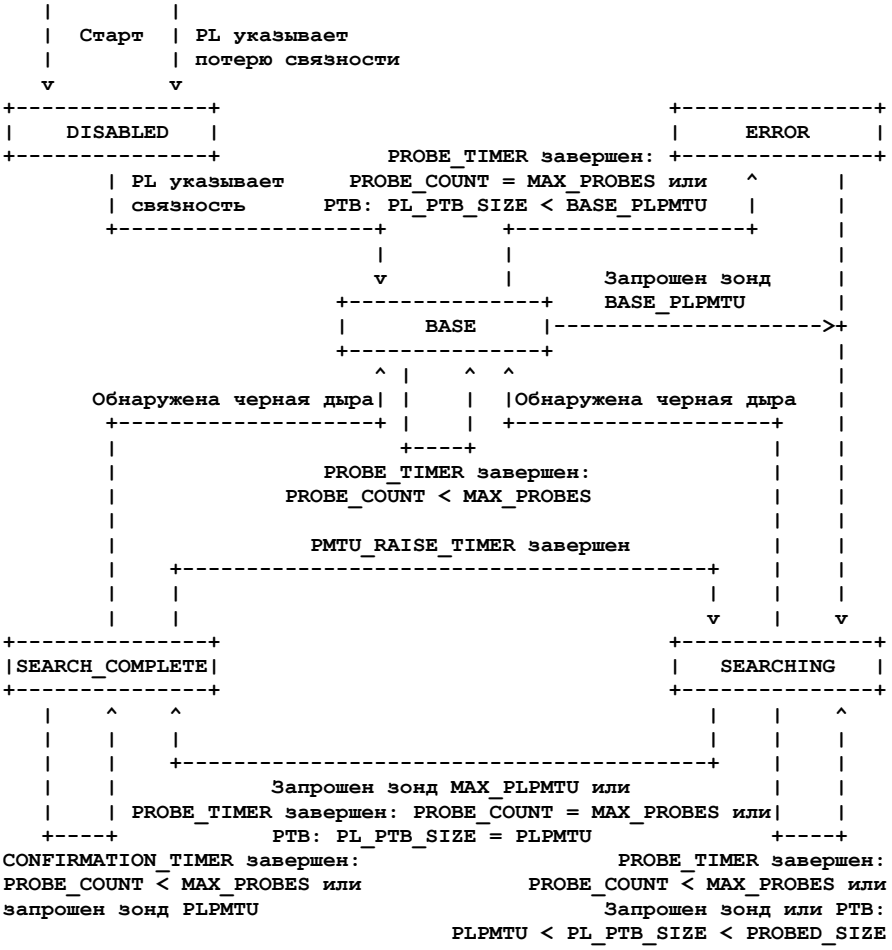

Рисунок 5. Конечный автомат для Datagram PLPMTUD.

ERROR

Состояние ERROR представляет случай, когда для сетевого пути неизвестно о поддержке PLPMTU размером хотя бы BASE_PLPMTU или имеются противоречивые сведения о пути через сеть, которые в ином случае привели бы к чрезмерным вариациям MPS, передаваемого вышележащему уровню. Состояние смягчает осцилляции конечного автомата (смены состояния по событиям). Оно передаёт умеренное значение MPS вышележащему уровню через PL. Состояние завершается, когда пробные пакеты больше не обнаруживают ошибок. Отправитель PL затем переходит в состояние SEARCHING.

Реализациям разрешено включать фрагментацию на конечных точках, если DPLPMTUD не может проверить MIN_PLPMTU за PROBE_COUNT проб. Если DPLPMTUD не может проверить MIN_PLPMTU, реализация будет переходить в состояние DISABLED.

Примечание. MIN_PLPMTU может совпадать с BASE_PLPMTU для упрощения действий в этом состоянии.

5.3. Поиск для увеличения PLPMTU

В этом параграфе описаны алгоритмы, используемые DPLPMTUD для поиска большего PLPMTU.

5.3.1. Зондирование для повышения PLPMTU

Реализации применяют алгоритм поиска по всему диапазону для определения возможности поддержки большего значения PLPMTU на пути через сеть. Метод определяет диапазон поиска, подтверждая минимальное значение PLPMTU, а затем используя зондирование для выбора PROBED_SIZE, не превышающего MAX_PLPMTU. Значение MAX_PLPMTU является меньшим из значений локального MTU и EMTU_R (когда оно получено от удалённой конечной точки). Приложение **может** снизить MAX_PLPMTU, установив максимальный размер передаваемых дейтаграмм.

Для счётчика PROBE_COUNT устанавливается 0 при передаче первого пробного пакета размером не меньше PLPMTU. Каждый пробный пакет, успешно доставленный удалённому партнёру, подтверждается на уровне PL (параграф 4.1).

При передаче каждого пробного пакета получателю запускается таймер PROBE_TIMER, который сбрасывается при получении уровнем PL подтверждения доставки пакета через сеть (параграф 4.1). Это подтверждает поддержку на пути размера PROBED_SIZE и значение PROBED_SIZE присваивается PLPMTU. Алгоритм поиска может продолжить передачу пробных пакетов, увеличивая размер.

Если отсчёт таймера завершается до получения подтверждения, пробный пакет не подтверждает PROBED_SIZE. При каждом завершении отсчёта PROBE_TIMER увеличивается значение PROBE_COUNT, таймер запускается снова и может быть передан пробный пакет того же или иного (заданного алгоритмом поиска). Максимальное число передаваемых зондов задано константой MAX_PROBES. Если PROBE_COUNT достигает значения MAX_PROBES, зондирование останавливается и отправитель PL переходит в состояние SEARCH_COMPLETE.

5.3.2. Выбор размера зондов

Алгоритм поиска определяет минимальное полезное увеличение PLPMTU. Для отправителя PL не имеет смысла проверять все размеры, поскольку это создаст ненужную нагрузку на путь. Реализации **следует** выбрать набор размеров пробных пакетов для максимального роста PLPMTU на каждом шаге поиска.

Реализация может оптимизировать процедуру поиска, выбрав размеры шагов из таблицы базовых размеров PMTU. При установке размера для следующего поиска разработчик должен также учитывать базовые размеры MPS, которые приложения стремятся использовать, а также базовые значения MTU, используемые в сети.

5.3.3. Устойчивость к несогласованности сведений о пути

Решение о повышении PLPMTU должно быть устойчивым к возможности получения несогласованной информации о пути через сеть. Такая несогласованность может быть, например, следствием потери пробных пакетов по другим причинам (т. е. не в связи с их размером) или частой смены пути, когда некоторые пакеты идут по одному пути, а иные - по другому с отличающимися характеристиками.

Отправитель PL может обнаружить несогласованность из последовательности подтверждённых зондов PLPMTU или последовательности принимаемых сообщений PTB. При обнаружении несогласованности сведений о пути отправитель PL может использовать другой режим поиска, который на некоторое время ограничивает предлагаемое значение MPS. Это позволяет избежать ненужной потери пакетов.

5.4. Устойчивость к несоответствию пути

Некоторые пути могут не поддерживать пакеты размером BASE_PLPMTU. Для обеспечения устойчивости к таким путям может быть реализовано состояние Error. Это позволяет использовать PLPMTU меньше желаемого значения для предотвращения отказов в связности. Для реализации этого можно использовать такие методы, как фрагментация IP в конечной точке, позволяющая отправителю PL взаимодействовать с партнёрами, используя пакеты размером меньше BASE_PLPMTU.

6. Спецификация зависящих от протокола методов

DPLPMTUD требует указания зависящих от протокола деталей для каждого применяемого уровня PL.

В следующем параграфе приведено руководство по реализации метода DPLPMTUD как части приложения, использующего UDP или UDP-Lite. Рекомендации применимы и к другим службам на основе дейтаграмм, не включающим конкретный транспортный протокол (например, к туннельной инкапсуляции). В последующих параграфах описаны возможные способы реализации DPLPMTUD как части транспортного сервиса, позволяющего приложениям использовать преимущества определения PLPMTU без необходимости самим реализовать этот метод при работе по протоколам SCTP и QUIC.

6.1. Поддержка DPLPMTUD в приложениях UDP и UDP-Lite

Действующие спецификации UDP [RFC0768] и UDP-Lite [RFC3828] не определяют в RFC метод поддержки PLPMTUD. В частности, транспорт UDP не поддерживает функций, требуемых для реализации PLPMTUD.

Метод DPLPMTUD можно реализовать как часть приложения, основанного на UDP или UDP-Lite, но полагающегося на функции вышележащего уровня для реализации метода [BCP145]. Некоторые примитивы, используемые DPLPMTUD, могут быть недоступны через Datagram API (например, возможность доступа к PLPMTU из кэша уровня IP или интерпретация сообщений PTB).

Кроме того, рекомендуется не применять определение PMTU на нескольких уровнях. Приложению **следует** избегать использования DPLPMTUD, если эти функции поддерживает базовая транспортная система. Единая поддержка PLPMTU обеспечивает преимущества как в возможности совместного использования разными процессами, так и в возможности согласовать зондирование для разных экземпляров PL.

6.1.1. Запрос к приложению

Приложению нужен механизм прикладного уровня (такой, как подтверждение сообщений), который запрашивает отклик принимающей конечной точки. Этому методу **следует** позволять отправителю проверять возвращённое в отклике значение для дополнительной защиты от вставки данных извне пути [BCP145]. Подходящими методами являются параметры, известные лишь паре конечных точек, такие как идентификатор сессии или порядковый номер.

6.1.2. Отклик приложения

Приложению нужен механизм прикладного уровня для передачи откликов от приёмной конечной точки. Этот отклик может указывать успешное получение зонда через сеть, а также указывать, что некоторые (или все) пакеты не попали к адресату.

6.1.3. Отправка пробных пакетов

Пробный пакет может содержать блок данных приложения, но это связано с риском потери пакета. Некоторые приложения могут предпочесть пробные пакеты, не содержащие данных приложения, чтобы не рисковать ими.

6.1.4. Начальная связность

Приложения, не имеющему других сведений от верхних уровней, подтверждающих связность с удаленным партнёром, **следует** реализовать механизм подключения с использованием подтверждённых зондов до перехода в фазу BASE.

6.1.5. Проверка пути

Приложения, не имеющему других сведений от верхних уровней, подтверждающих доставку дейтаграмм, **следует** реализовать таймер CONFIRMATION_TIMER для периодической передачи зондов в состоянии SEARCH_COMPLETE.

6.1.6. Обработка сообщений PTB

Приложение, способное и желающее принимать сообщения PTB, **должно** выполнять проверку ICMP в соответствии с параграфом 5.2 [BCP145]. Это требует от приложения проверки каждого принятого сообщения PTB, чтобы убедиться в его создании в ответ на передачу трафика и в том, что указанное в нем значение PL_PTB_SIZE меньше текущего размера зондов (см. параграф 4.6.2). Проверенное сообщение PTB **можно** использовать в качестве входных данных алгоритма DPLPMTUD, но **недопустимо** применять его напрямую для установки PLPMTU.

6.2. DPLPMTUD для SCTP

В параграфе 10.2 [RFC4821] определён рекомендуемый метод зондирования PLPMTUD для SCTP, а в параграфе 7.3 [RFC4960] конечным точкам рекомендуется применять методы RFC 4821 для каждого адреса получателя. Спецификация DPLPMTUD сохраняет практику применения PL для определения PMTU, но обновляет RFC4960 рекомендацией использовать заданный этим документом метод. **Рекомендуемый** метод генерации зондов заключается в добавлении к сообщению SCTP блока (chunk), содержащего лишь заполнение. Для создания пробного пакета **следует** присоединять блок PAD, определённый в [RFC4820], к блоку HEARTBEAT (HB). Это позволяет зондировать путь без влияния на передачу пользовательских сообщений и внесения ограничений в механизмы контроля перегрузок и управления потоком данных. Это предпочтительней использования блоков DATA (с заполнения, когда нужно) в качестве пробных пакетов.

В параграфе 6.9 [RFC4960] описано деление пользовательских сообщений на блоки DATA, передаваемые уровнем PL при использовании SCTP. При этом отмечено, что после отправки сообщения SCTP его нельзя сегментировать снова. В [RFC4960] описан метод повторной передачи блоков DATA при снижении MPS, а в параграфе 6.9 [RFC4960] описано применения для этого фрагментации IP. Данный документ не меняет этого поведения.

6.2.1. SCTP/IPv4 и SCTP/IPv6

6.2.1.1. Начальная связность

Базовый протокол задан [RFC4960] и является PL с подтверждениями. Поэтому отправитель может перейти в состояние BASE сразу после подтверждения связности.

6.2.1.2. Передача пробных пакетов SCTP

Пробный пакет содержит базовый заголовок SCTP, за которым следуют блоки HEARTBEAT и PAD. Блок PAD нужен для задания размера пробного пакета, а HEARTBEAT служит триггером возврата блока HEARTBEAT ACK. Получение блока HEARTBEAT ACK подтверждает успешную доставку зонда и на основании этого обновляются счётчики ассоциации и пути. Неудачные пробы при этом не учитываются и считаются последствием выбора слишком большого PLPMTU.

Отправитель SCTP должен быть способен определить суммарный размер пробного пакета. Блок HEARTBEAT может содержать параметр Heartbeat Information, включающий кроме информации, предложенной в [RFC4960], размер зонда, помогающий реализации связать HEARTBEAT ACK с размером переданного зонда. Отправитель также может использовать другие методы, такие как отправка nonce с проверкой этого значения в полученном отклике. Размер блока PAD определяется вычитанием из размера зонда размера базового заголовка SCTP и блока HEARTBEAT. Содержимое блока PAD может быть произвольным. При передаче на уровне IP размер PMTU также включает заголовок IPv4 или IPv6.

Зондирование может начаться сразу после согласования PL, до начала передачи данных. В таком случае (PMTU на превышает MTU для интерфейса) процесс займёт несколько периодов кругового обхода, в зависимости от числа передаваемых зондов DPLPMTUD. Для реализации PROBE_TIMER можно использовать таймер Heartbeat.

6.2.1.3. Проверка пути в SCTP

Поскольку SCTP является PL с подтверждениями, отправителю **недопустимо** реализовать таймер CONFIRMATION_TIMER в состоянии SEARCH_COMPLETE.

6.2.1.4. Обработка сообщений PTB в SCTP

Должна выполняться обычная проверка ICMP в соответствии с приложением C [RFC4960]. Это требует включения первых 8 байтов базового заголовка SCTP в сообщение PTB, возможного для ICMPv4 и обычного в ICMPv6.

Когда сообщение PTB проверено, значение PL_PTB_SIZE, рассчитанное из PTB_SIZE в сообщении PTB, **следует** применять в алгоритме DPLPMTUD при условии, что PL_PTB_SIZE меньше текущего размера зонда (параграф 4.6).

6.2.2. DPLPMTUD для SCTP/UDP

Инкапсуляция в UDP для протокола SCTP описана в [RFC6951]. Эта спецификация заменяет ссылку на RFC 4821 в параграфе 5.6 RFC 6951 ссылкой на этот документ (RFC 8899). RFC 6951 обновляется добавлением в конце параграфа 5.6 строки: «**Рекомендуемый** метод определения MTU на пути задан в RFC 8899».

6.2.2.1. Начальная связность

Отправитель может перейти в состояние BASE сразу после подтверждения связности SCTP.

6.2.2.2. Передача пробных пакетов SCTP/UDP

Зондирование может выполняться в соответствии с параграфом 6.2.1.2. Размер пробных пакетов включает 8 байтов заголовка UDP и его нужно учитывать при определении размера блока PAD.

6.2.2.3. Проверка пути с SCTP/UDP

SCTP является PL с подтверждениями, поэтому отправителю **недопустимо** реализовать таймер CONFIRMATION_TIMER в состоянии SEARCH_COMPLETE.

6.2.2.4. Обработка сообщений PTB в SCTP/UDP

Должна выполняться обычная проверка ICMP в соответствии с приложением C [RFC4960]. Это требует включения первых 8 байтов базового заголовка SCTP в сообщение PTB, возможного для ICMPv4 (отметим, что заголовок UDP также занимает часть включаемых в сообщение данных) и обычного в ICMPv6. Когда сообщение PTB проверено, значение PL_PTB_SIZE, рассчитанное из PTB_SIZE в сообщении PTB, **следует** применять в алгоритме DPLPMTUD при условии, что PL_PTB_SIZE меньше текущего размера зонда (параграф 4.6).

6.2.3. DPLPMTUD для SCTP/DTLS

Инкапсуляция в DTLS для SCTP определена в [RFC8261] и применяется для каналов данных в реализациях WebRTC. Эта спецификация заменяет ссылку на RFC 4821 в разделе 5 RFC 8261 ссылкой на этот документ (RFC 8899).

6.2.3.1. Начальная связность

Отправитель может перейти в состояние BASE сразу после подтверждения связности SCTP.

6.2.3.2. Sending SCTP/DTLS Probe Packets

Зондирование может выполняться в соответствии с параграфом 6.2.1.2. Размер пробных пакетов включает заголовок DTLS и его нужно учитывать при определении размера блока PAD.

6.2.3.3. Проверка пути с SCTP/DTLS

Поскольку SCTP является PL с подтверждениями, отправителю **недопустимо** реализовать таймер CONFIRMATION_TIMER в состоянии SEARCH_COMPLETE.

6.2.3.4. Обработка сообщений PTB в SCTP/DTLS

[RFC4960] не задаёт способа проверки содержимого сообщений SCTP/DTLS ICMP, равно как этот документ. Это может препятствовать обработке сообщений PTB на уровне PL.

6.3. DPLPMTUD для QUIC

QUIC [QUIC] представляет собой основанный на UDP уровень PL с обратной связью при получении. Данные UDP включают заголовок пакета QUIC, защищённое содержимое и поля проверки подлинности. Поддерживается заполнение и объединение пакетов для создания зондов нужного размера. С точки зрения DPLPMTUD, протокол QUIC можно считать PL с подтверждениями. В [QUIC] описан метод использования DPLPMTUD с пакетами QUIC.

7. Взаимодействие с IANA

Этот документ не требует действий IANA.

8. Вопросы безопасности

Вопросы безопасности при использовании UDP и SCTP рассмотрены в соответствующих RFC.

Для предотвращения чрезмерной нагрузки интервал между передаваемыми зондами **должен** быть не меньше RTT, а интервал между циклами зондирования определяется таймером PMTU_RAISE_TIMER.

Отправитель PL должен обеспечить защиту метода подтверждения приёма зондов от атакующих извне пути, способных внедрять пакеты в путь. Такая защита обеспечивается в протоколах IETF (например, TCP, SCTP) с помощью случайных значений порядковых номеров. Описание одного из способов защиты при использовании UDP приведено в параграфе 5.1 [BCP145]).

Возможны случаи, когда сообщения ICMP PTB (Packet Too Big) не доставляются в соответствии с политикой, настройкой или возможностями оборудования (см. 1.1. Классическое определение MTU на пути). Поэтому метод не полагается на получение сообщений PTB, но может использовать их. Сообщения PTB могут служить для принуждения узла к неоправданному снижению PLPMTU. Поэтому узел, использующий DPLPMTUD, **должен** проверять содержимое сообщений PTB и их соответствие переданным пакетам (т. е. сведения об ошибке должны соответствовать реально переданным дейтаграммам, см. параграф 4.6.1).

Расположенный на пути атакующий, способный создавать сообщения PTB, может генерировать фиктивные PTB, включая с них часть действительного пакета IP. Такие атаки могут служить для принудительного снижения PLPMTU. Устройство в пути может также принудительно снизить PLPMTU, реализуя правило отбрасывания пакетов больше определённого размера. Существует два способа ослабления таких атак. Во-первых, отправитель PL может не устанавливать PLPMTU ниже базового размера на основании лишь сообщений PTB, что достигается переходом в состояние BASE при получении таких сообщений. Во-вторых, сообщения PTB можно не обрабатывать и отправитель PL может отключить их обработку (например, в режиме устойчивости после обнаружения того, что последовательные зонды действительно показывают поддержку PTB_SIZE на пути).

Анализ вложенной в сообщение PTB части пакета может требовать от отправителя PL дополнительной обработки. **Следует** ограничивать эту обработку для предотвращения отказа в обслуживании при включении в сообщения произвольных заголовков. Ограничение скорости обработки может приводить к тому, что не все сообщения PTB будут получены PL, но метод DPLPMTUD устойчив к таким потерям.

Успешная обработка сообщения ICMP может вызвать зондирование, когда сообщенный в PTB размер действителен, но это не ведёт к прямому обновлению PLPMTU для пути. Такое поведение предотвращает чёрные дыры для данных, создаваемые путём задания избыточного для пути размера пакетов.

Информация о пути может быть нестабильной, например, в результате пересылки по нескольким путям с разными PMTU или изменения PMTU на одном из путей. В реализации PLPMTUD **следует** предусматривать смягчение эффекта таких изменений. Одним из возможных путей является обеспечение устойчивости метода (5.4. Устойчивость к несоответствию пути), предотвращающей флуктуации MPS.

Методы DPLPMTUD могут вносить заполнение для увеличения общего размера дейтаграммы в пробных пакетах. Размер пробного пакета включает все заголовки и заполнение добавленное к данным пакета (например, включение связанных с защитой полей, таких как тег AEAD и заполнение уровня TLS record). Значение заполнения не влияет на алгоритм поиска DPLPMTUD и поэтому должно выбираться в соответствии с политикой PL.

Если PL может использовать криптографические механизмы защиты конфиденциальности и целостности данных, при разработке следует избегать добавления чего-либо (например, заполнения) в пробные пакеты DPLPMTUD без защиты добавленных частей криптографическими механизмами.

9. Литература

9.1. Нормативные документы

[BCP145] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, March 2017, <https://www.rfc-editor.org/info/bcp145>.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., Ed., and G. Fairhurst, Ed., "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, DOI 10.17487/RFC3828, July 2004, <https://www.rfc-editor.org/info/rfc3828>.

[RFC4820] Tuexen, M., Stewart, R., and P. Lei, "Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)", RFC 4820, DOI 10.17487/RFC4820, March 2007, <https://www.rfc-editor.org/info/rfc4820>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication", RFC 6951, DOI 10.17487/RFC6951, May 2013, <https://www.rfc-editor.org/info/rfc6951>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

[RFC8261] Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "Datagram Transport Layer Security (DTLS) Encapsulation of SCTP Packets", RFC 8261, DOI 10.17487/RFC8261, November 2017, <https://www.rfc-editor.org/info/rfc8261>.

9.2. Дополнительная литература

[QUIC] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress[1], Internet-Draft, draft-ietf-quic-transport-29, 10 June 2020, <https://tools.ietf.org/html/draft-ietf-quic-transport-29>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <https://www.rfc-editor.org/info/rfc1812>.

[RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, DOI 10.17487/RFC2923, September 2000, <https://www.rfc-editor.org/info/rfc2923>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

[1]Документ опубликован в RFC 9000. *Прим. перев.*

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/info/rfc4821>.

[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <https://www.rfc-editor.org/info/rfc4890>.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, DOI 10.17487/RFC5508, April 2009, <https://www.rfc-editor.org/info/rfc5508>.

[RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", RFC 8900, BCP 230, September 2020, <https://www.rfc-editor.org/info/rfc8900>.

[TUNNELS] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019,<https://tools.ietf.org/html/draft-ietf-intarea-tunnels-10>.

Благодарности

Эта работа частично финансировалась в рамках программы European Union Horizon 2020 Research and Innovation по гранту No. 644334, A New, Evolutive API and Transport-Layer Architecture for the Internet (NEAT). Выраженные здесь взгляды принадлежат исключительно авторам.

Спасибо всем, кто участвовал в работе или комментировал документ, рабочим группам TSVWG и QUIC, а также Mathew Calder и Julius Flohr за предоставление предварительных реализаций.

Адреса авторов

Godred Fairhurst

University of Aberdeen

School of Engineering

Fraser Noble Building

Aberdeen

AB24 3UE

United Kingdom

Email: gorry@erg.abdn.ac.uk

Tom Jones

University of Aberdeen

School of Engineering

Fraser Noble Building

Aberdeen

AB24 3UE

United Kingdom

Email: tom@erg.abdn.ac.uk

Michael Tüxen

Münster University of Applied Sciences

Stegerwaldstrasse 39

48565 Steinfurt

Germany

Email: tuexen@fh-muenster.de

Irene Rüngeler

Münster University of Applied Sciences

Stegerwaldstrasse 39

48565 Steinfurt

Germany

Email: i.ruengeler@fh-muenster.de

Timo Völker

Münster University of Applied Sciences

Stegerwaldstrasse 39

48565 Steinfurt

Germany

Email: timo.voelker@fh-muenster.de

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru