Internet Engineering Task Force (IETF)
Request for Comments: 9347
Category: Standards Track
ISSN: 2070-1721

C. Hopps
LabN Consulting, L.L.C.

January 2023

Aggregation and Fragmentation Mode for Encapsulating Security Payload (ESP) and Its Use for IP Traffic Flow Security (IP-TFS)

Режим агрегирования и фрагментации для ESP и его применение в IP-TFS

Аннотация

Документ описывает механизм агрегирования и фрагментирования пакетов IP, инкапсулируемых в защищённые данные (Encapsulating Security Payload или ESP). Этот новый тип содержимого пакетов (payload) может применяться для разных целей, таких как сокращение издержек инкапсуляции для мелких пакетов IP, однако этот документ сосредоточен на улучшении защиты потоков трафика IP (IP Traffic Flow Security или IP-TFS) путём добавления конфиденциальности потока трафика (Traffic Flow Confidentiality или TFC) в трафик с инкапсуляцией IP. TFC обеспечивается путём сокрытия размера и частоты пакетов IP за счёт использования туннеля IPsec с постоянным размером и частотой передачи пакетов. Это позволяет контролировать перегрузку, а также использовать непостоянную скорость передачи трафика.

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.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке https://www.rfc-editor.org/info/rfc9347.

Оглавление

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

1. Введение

Анализом трафика [RFC4301] [AppCrypt] называют действия по извлечению сведений о данных, передаваемых через сеть. При непосредственном сокрытии данных путём шифрования [RFC4303] картина трафика в сообщении может раскрывать некоторые сведения из-за различий в форме и времени отправки [RFC8546] [AppCrypt]. Сокрытие размера и частоты передачи трафика называют конфиденциальностью потока трафика (TFC), в соответствии с [RFC4303].

[RFC4303] обеспечивает TFC за счёт заполнения (padding) шифрованных пакетов IP и возможности передавать пакеты, содержащие лишь заполнение (all-pad), что указывается номером протокола 59. Этому методу присуще важное ограничение, связанное с существенными издержками для пропускной способности.

Этот документ определяет для ESP режим агрегирования и фрагментирования (aggregation and fragmentation или AGGFRAG), а также задаёт применение ESP для IP-TFS. Это решение обеспечивает полную функциональность TFC без отмеченных издержек для пропускной способности. Это достигается за счёт применения туннелей IPsec [RFC4303] с фиксированным размером инкапсулированных пакетов, которые могут содержать целые пакеты IP, их фрагменты или несколько пакетов для максимального использования пропускной способности туннеля. Переменная скорость передачи разрешена, но конфиденциальность при её использовании не рассматривается в этом документе.

Сравнение издержек IP-TFS и решения TFC, предписанного в [RFC4303], дано в Приложении C.

Кроме того, IP-TFS обеспечивает беспристрастность в перегруженных сетях [RFC2914]. Это важно в случаях, когда пользователь IP-TFS не имеет полного контроля над доменами, через которые проходит туннель IP-TFS.

Заданные в этом документе механизмы, такие как AGGFRAG, согласуются с намерением разрешать работу без TFS, но этот вопрос выходит за рамки документа.

1.1. Термины и концепции

Ключевые слова **должно** (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.

Документ предполагает знакомство читателя с концепциями защиты IP, включая TFC, как описано в [RFC4301].

2. Туннель AGGFRAG

Как отмечено в разделе 1, режим AGGFRAG использует туннель IPsec [RFC4303] в качестве транспорта. Для целей IP-TFS в туннель AGGFRAG передаются пакеты фиксированного размера с постоянной скоростью.

Основными входными данными для алгоритма туннелирования является пропускная способность, используемая туннелем. Требуется задать два значения для обеспечения этой пропускной способности - размер инкапсулированных пакетов и скорость их передачи.

Фиксированный размер пакетов **может** быть задан вручную или определён с помощью таких методов, как PLMTUD[1] [RFC4821] [RFC8899] или PMTUD[2] [RFC1191] [RFC8201]. С методом PMTUD связаны известные проблемы, поэтому

[1]Packetization Layer MTU Discovery - определение MTU для уровня пакетизации.
[2]Path MTU Discovery - определение MTU для пути.

PLMTUD считается более надёжным вариантом. Для PLMTUD содержимое (payload) контроля перегрузок может применяться в качестве внутриканальных (in-band) зондов (см. параграф 6.1.2 и [RFC8899]).

С учётом размера инкапсулированных пакетов и запрошенной пропускной способности можно рассчитать скорость передачи пакетов путём деления запрошенной пропускной способности на размер инкапсулированного пакета.

Выходная (приёмная) сторона туннеля AGGFRAG **должна** разрешать и ожидать, что входная (передающая) сторона будет варьировать размер и скорость передачи инкапсулированных пакетов, если это не запрещено иными правилами.

2.1. Содержимое туннеля

Как было отмечено, одной из проблем, связанных с заполнением TFC [RFC4303] является избыточный расход пропускной способности, поскольку в инкапсулированном пакете может содержаться лишь 1 пакет IP. Для максимального пропускания в IP-TFS эта зависимость разрывается путём введения режима AGGFRAG для ESP.

Режим AGGFRAG агрегирует и фрагментирует вложенный трафик IP при инкапсуляции в пакеты туннеля IPsec. В IP-TFS инкапсулированные в туннель IPsec пакеты имеют фиксированный размер. Заполнение добавляется в туннельные пакеты лишь при отсутствии данных для передачи на момент отправки следующего пакета или запрете фрагментации получателем. Это достигается путём использования в поле ESP [RFC4303] Next Header значения AGGFRAG_PAYLOAD (6.1. Содержимое AGGFRAG_PAYLOAD).

Были предложены иные (не IP-TFS) варианты использования режима AGGFRAG, такие как повышение производительности за счёт агрегирования пакетов, а также решение проблемы MTU за счёт фрагментации. Эти вопросы не рассматриваются в документе, но применение таких методов документ не ограничивает.

2.2. Содержимое Payload

Содержимое данных (payload) AGGFRAG_PAYLOAD, определённое этим документом, включает 4- или 24-октетный заголовок, за которым следует полный или неполный блок данных или несколько таких блоков. На рисунке 1 показаны эти поля в пакете ESP. Полное описание AGGFRAG_PAYLOAD дано в параграфе 6.1.

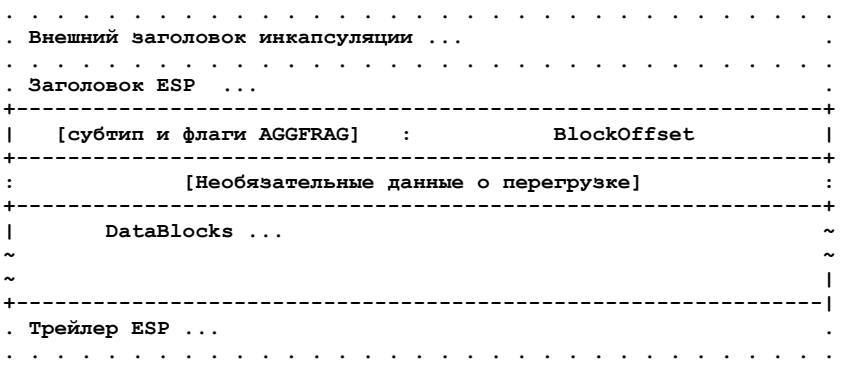

Рисунок 1. Схема пакета IPsec в режиме AGGFRAG.

BlockOffset имеет значение 0 или указывает некое смещение в DataBlocks или после блока данных. BlockOffset = 0 указывает, что DataBlocks начинается с нового блока данных. Отличное от 0 значение BlockOffset указывает начало нового блока данных, а исходные данные DataBlocks относятся к блоку, который ещё находится в процессе сборки. Если BlockOffset указывает значение после конца DataBlocks, это говорит, что следующий блок данных размещается в последующем инкапсулированном пакете. Наличие BlockOffset всегда указывает следующий доступный блок данных, что позволяет восстановить следующий вложенный пакет в случае потери внешнего инкапсулирующего пакета.

Пример потока пакетов для режима AGGFRAG представлен в Приложении A.

2.2.1. Блоки данных

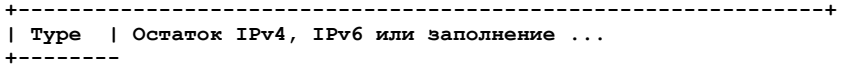

Рисунок 2. Схема блока данных.

Блок данных определяется 4-битовым кодом типа, за которым следует сам блок. Значения типов были аккуратно подобраны в соответствии со значениями полей IPv4 или IPv6, чтобы не возникало связанных с типом блока издержек при инкапсуляции пакетов IP. Размер блока данных извлекается из инкапсулированного поля IPv4 Total Length или IPv6 Payload Length.

2.2.2. Заполнение в конце

Поскольку блок данных указывается первыми 4 битами, заполнение требуется лишь при отсутствии данных для инкапсуляции. Для этого применяется специальный блок данных - Pad Data Block.

2.2.3. Фрагментация, порядковые номера и данные лишь с заполнением

Чтобы получатель мог собрать фрагментированные вложенные пакеты, отправитель должен передавать фрагменты вложенных пакетов вплотную друг к другу в логическом потоке внешних пакетов (т. е. с использованием последовательных порядковых номеров ESP). Однако отправителю разрешено вставлять содержимое (payload), состоящее только из заполнения (all-pad, т. е. данные с BlockOffset = 0 и одним блоком заполнения) между пакетами, передающими фрагменты вложенного пакета. Такое чередование с пакетами all-pad позволяет отправителю передавать туннельные пакеты независимо от вычислительных потребностей при инкапсуляции. Когда получатель собирает вложенный пакет, имея принятые данные all-pad, он инкрементирует порядковый номер пакета, в котором ожидается прибытие следующего фрагмента вложенного пакета.

С учётом изложенного выше, получателю требуется обрабатывать полученные с нарушением порядка внешние пакеты ESP перед сборкой. В ESP уже имеется возможность обнаружения атак с повторным использованием пакетов (replay), которая обычно реализуется на основе окна. Аналогичное скользящее окно на основе порядкового номера может применяться для восстановления порядка в потоке внешних пакетов. При получении большего (более нового) порядкового номера в пакете окно сдвигается вперёд, а при получении более старых пакетов ESP с номерами вне окна, такие пакеты отбрасываются. Выбор размера окна зависит от степени нарушения порядка, с которой сталкивается пользователь. Предлагается по умолчанию использовать окно размеров в 3 пакета, если нет причин его менять. Поскольку степень разупорядочения пакетов трудно предсказать, размер окна **следует** делать настраиваемым пользователем. Реализации **могут** динамически подстраивать размер окна на основе фактического нарушения порядка доставки пакетов.

Следует отметить, что при отправке IP-TFS непрерывного потока пакетов не требуется явный таймер для потерянных пакетов, однако **рекомендуется** использовать такой таймер. Если реализация не использует таймер потери пакетов и считает внешний пакет потерянным лишь при его выходе за пределы окна упорядочения, вложенный трафик может задерживаться на величину, не превышающую произведение размера окна упорядочения и скорости передачи пакетов. Эта задержка может быть существенной при малой скорости передачи или большом размере окна. Поскольку таймер потери пакетов влияет на задержку доставки вложенных пакетов, реализация или пользователь могут установить для него значение, пропорциональное скорости в туннеле.

Хотя ESP гарантирует рост порядковых номеров в передаваемых последовательно пакетах, последовательной генерации порядковых номеров не требуется (например, можно использовать лишь чётные номера при условии их постоянного увеличения). Пропуски порядковых номеров не подходят для этого документа, поэтому номер в потоке **должен** возрастать на 1 для каждого следующего пакета.

При использовании AGGFRAG_PAYLOAD вместе с обнаружением повторов размер окна для того и другого **можно** сократить до меньшего из двух значений. Это обусловлено тем, что пакеты за пределами меньшего окна, попадающие в большее, будут отбрасываться механизмом, с которым связано меньшее окно. Поддерживать одинаковые окна не требуется. Действительно, в некоторых случаях, таких как очень медленный туннель, где подойдёт очень малый или нулевой размер окна упорядочения, пользователю может потребоваться большое окно обнаружения повторов для регистрации повторно использованных пакетов. Кроме того, большие окна обнаружения повторов можно реализовать с очень малыми издержками по сравнению с большими окнами упорядочения.

Поскольку порядковые номера сбрасываются при переключении защищённых связей SA (например, при смене ключей Child SA), отправителям **недопустимо** передавать фрагменты вложенного пакета в разных SA.

> Примечание по значениям BlockOffset. Отправитель **должен** кодировать BlockOffset в соответствии с непосредственно предшествовавшим пакетом, содержащим не только заполнение. В частности, если непосредственно предшествующий пакет, содержащий не только заполнение, завершается Pad Data Block, значение BlockOffset **должно** быть 0, поскольку блоки Pad Data не фрагментируются. Значение BlockOffset должно соответствовать оставшемуся размеру, подразумеваемому полем размера во фрагментированном вложенном пакете.

2.2.3.1. Дополнительное заполнение

Когда пропускная способность туннеля не используется полностью, отправитель **может** заполнять текущий пакет инкапсуляции для доставки вложенного пакета без фрагментирования в следующем внешнем пакете. Преимущество состоит в исключении фрагментации вложенного пакета при наличии пиков предлагаемой нагрузки (трафик без пиков не будет фрагментироваться естественным путём). Отправитель **может** также задать минимальный размер фрагментов (например, процентной долей AGGFRAG_PAYLOAD) чтобы избежать фрагментации за счёт пропускной способности туннеля. Это ведёт к росту сложности и дополнительным задержкам для вложенного трафика. Основное преимущество предотвращения фрагментации состоит в минимизации потерь вложенных пакетов при наличии потерь внешних пакетов. Целесообразность (например, величина и тип потерь с учётом различных форм вложенного трафика и нагрузки) и значения допустимой (добавленной) задержки требуют дополнительного исследования, выходящего за рамки этого документа.

Хотя использование заполнения для предотвращения фрагментации не препятствует совместимости, неверное использование заполнения может снизить эффективную пропускную способность туннеля. Отправитель, реализующий какой-либо из означенных подходов, должен позаботиться о сохранении эффективной пропускной способности и общей полезности туннеля при использовании заполнения.

2.2.4. Пустое содержимое

Для поддержки передачи сведений контроля перегрузок (см. ниже) с использованием SA без поддержки AGGFRAG_PAYLOAD разрешено передавать содержимое (payload) AGGFRAG_PAYLOAD без блоков данных (т. е. размер данных ESP совпадает с размером заголовка AGGFRAG_PAYLOAD). Такое содержимое называется пустым (empty payload). В настоящее время это применимо лишь при работе без использования IKEv2.

2.2.5. Отображение полей заголовков IP

В [RFC4301] даны некоторые рекомендации по части отображения значений из внутреннего заголовка IP во внешний, а именно, бита запрета фрагментирования (Don't Fragment или DF) [RFC0791], поля дифференцированного обслуживания (Differentiated Services или DS) [RFC2474] и поля явного уведомления о перегрузке (Explicit Congestion Notification или ECN) [RFC3168]. В отличие от [RFC4301], режим AGGFRAG может и часто будет инкапсулировать более одного пакета IP в пакет ESP. Это вносит дополнительные ограничения для отображения полей.

2.2.5.1. Бит DF

В режиме AGGFRAG бит DF не отображается во внешний заголовок, поскольку он не связан с функциональностью туннеля AGGFRAG. В режиме AGGFRAG не требуется IP-фрагментация вложенных пакетов и на них не влияет фрагментация внешних пакетов инкапсуляции.

2.2.5.2. Значение ECN

Значение ECN не требуется отображать, поскольку любая перегрузка, связанная с туннелем IP-TFS с постоянной скоростью передачи, по своей природе не связана с потоком вложенного трафика. Отправитель **может** по-прежнему устанавливать ECN во вложенных пакетах в соответствии со спецификацией ECN [RFC3168] [RFC4301] [RFC6040].

2.2.5.3. Поле DS

По умолчанию поле DS **не следует** копировать, хотя отправитель **может** разрешить переопределение поведения в конфигурации. Отправителю **следует** также разрешать установку значения DS в конфигурации.

2.2.6. IPv4 TTL, IPv6 Hop Limit и сообщения ICMP

Изменение полей IPv4 TTL [RFC0791] и IPv6 Hop Limit [RFC8200] во внутренних заголовках описано в [RFC4301].

В [RFC4301] указано, как применять политику для аутентифицированных и разрешённых пакетов с ошибками ICMP (например, Destination Unreachable), прибывающих или пересланных через конечную точку, в частности обрабатывать, игнорировать или пересылать такие пакеты. За одним исключением этот документ не меняет обработку таких пакетов, заданную в [RFC4301].

Единственным отличием туннеля AGGFRAG при обработке ошибок ICMP является PMTU. При разрешённой в туннеле AGGFRAG фрагментации не требуется генерировать ошибку ICMP Too Big при получении входного трафика, поскольку внутренние пакеты естественным путём фрагментируются при инкапсуляции AGGFRAG.

Если фрагментация в туннеле AGGFRAG отключена, обработка входящего трафика в точности соответствует обработке в туннеле ESP без AGGFRAG. В явном виде пакеты IPv4 с флагом DF и пакеты IPv6, которые не помещаются в данные (payload) внешнего пакета, будут вызывать соответствующую ошибку ICMP Too Big, как описано в [RFC4301], а пакеты IPv4 без флага DF будут фрагментироваться IP, как описано в [RFC4301].

Выходящие из туннеля пакеты обрабатываются в соответствии с [RFC4301].

Остальные аспекты PMTU и обработка сообщений ICMP Too Big (т. е. связанные с размером внешних пакетов туннеля AGGFRAG/ESP) также соответствуют [RFC4301].

2.2.7. Эффективное значение MTU для туннеля

В отличие от [RFC4301] в туннелях AGGFRAG обычно не применяется эффективное значение MTU (EMTU), поскольку пакеты IP всех размеров передаются без фрагментации IP перед входом в туннель. При этом отправитель может разрешать явную настройку MTU для туннеля.

Если фрагментация в туннеле AGGFRAG отключена, значение EMTU и поведение туннеля не отличается от обычного для туннелей [RFC4301].

2.3. Исключительное использование SA

Этот документ не задаёт смешанного использования SA с разрешённым AGGFRAG_PAYLOAD. Отправитель **должен** передавать в SA, настроенных на режим AGGFRAG, только данные (payload) AGGFRAG_PAYLOAD.

2.4. Режимы работы

Как и обычные IPsec/ESP SA, защищённые связи AGGFRAG SA являются односторонними. Двухсторонняя работа IP-TFS обеспечивается организацией двух AGGFRAG SA, по одной для каждого направления.

Туннель AGGFRAG, применяемый для IP-TFS, может работать в 2 режимах - с контролем и без контроля перегрузок.

2.4.1. Режим без контроля перегрузок

В режиме без контроля перегрузок IP-TFS передаёт пакеты фиксированного размера через туннель AGGFRAG с постоянной скоростью. Размер пакетов постоянный и не подстраивается автоматически независимо от перегрузок (например, потери пакетов).

По тем же причинам, которые указаны в [RFC7510], режим без контроля перегрузок **должен** применяться лишь в тех случаях, когда у пользователя есть полный административный контроль над всеми участками туннеля. В иных случаях применение этого режима **недопустимо**. Это нужно для того, чтобы пользователь мог гарантировать пропускную способность а также быть уверенным в отсутствии негативного влияния на перегрузки в сети [RFC2914]. В этом случае администратор уведомляется о потере пакетов (например, через syslog, уведомления YANG, ловушки SNMP и т. п.), чтобы любой отказ, связанный с нехваткой пропускной способности, можно было устранить. **Рекомендуется** также использовать выключатели, описанные в параграфе 2.4.2.1.

Пользователям, выбравшим режим без контроля перегрузок, нужно понимать, что в этом режиме пакеты передаются с постоянной скоростью, занимая фиксированную полосу, и при возникновении перегрузки изменения не вносятся. Поэтому при отсутствии гарантированной пропускной способности для туннеля на его работу и остальную сеть может оказываться негативное влияние.

Одним из ожидаемых применений режима без контроля перегрузок является гарантия доступности полной пропускной способности туннеля и предпочтения перед другим (нетуннельным) трафиком. Фактически типовым применением является организация межсайтового (site-to-site) трафика, полностью передаваемого по туннелю IP-TFS.

Режим без контроля перегрузок подходит также при работе ESP по протоколу TCP [RFC9329]. Однако использование TCP считается лишь резервным решением для IPsec и крайне нежелательно. Это также является одной из причин, по которым протокол TCP не был выбран для инкапсуляции IP-TFS вместо AGGFRAG.

2.4.2. Режим с контролем перегрузок

В режиме с контролем перегрузок IP-TFS адаптируется к насыщению сети, снижая скорость передачи пакетов при перегрузке и увеличивая её при снижении нагрузки. Поскольку издержки возникают на уровне пакета, задание

максимального фиксированного размера пакетов и изменение скорости позволяют минимизировать транспортные издержки.

Результатом алгоритма контроля перегрузок является подстройка скорости отправки пакетов на входе туннеля. Хотя этот документ на требует конкретного алгоритма контроля перегрузок, на основе опыта рекомендуется применять алгоритм, соответствующий [RFC5348]. Принципы контроля перегрузок описаны в [RFC2914]. В [RFC4342] имеется пример алгоритма из [RFC5348], соответствующего требованиям IP-TFS (т. е. предназначенного для пакетов фиксированного размера и скорости отправки, зависящей от насыщения).

Входными данными для дружественного к TCP алгоритма управления скоростью, описанного в [RFC5348], являются частота потерь у получателя и оценка отправителем времени кругового обхода (round-trip time или RTT). Эти значения предоставляются IP-TFS с использованием данных о насыщении в полях заголовка, описанных в разделе 3. В частности, этих значений достаточно для реализации алгоритма, описанного в [RFC5348].

Сведения о перегрузке **должны** передаваться по меньшей мере 1 раз за интервал RTT. До определения RTT сведения **следует** постоянно передавать от отправителя и получателя, чтобы можно было оценить значение RTT. Отсутствие этих сведений в течение нескольких интервалов RTT подряд следует считать индикатором перегрузки, заставляющим отправителя снижать скорость передачи. Например, в [RFC4342] это называется «временем ожидания без откликов» (no feedback timeout) и равно 4 интервалам RTT. При возникновении такого тайм-аута скорость отправки снижается вдвое в соответствии с [RFC4342].

Реализация **может** всегда включать сведения о перегрузке в свой заголовок данных AGGFRAG, если они передаются через SA с поддержкой IP-TFS. Поскольку IP-TFS обычно работает с большим размером пакетов, сведения о перегрузке будут занимать незначительную часть доступной для туннеля пропускной способности. Реализация, выбравшая постоянную передачу сведений о насыщении, **может** выбрать обновление лишь значения полей заголовка LossEventRate и RTT, передаваемые в каждом интервале RTT.

При выборе алгоритмов контроля перегрузок следует учитывать, что IP-TFS не обеспечивает гарантированной доставки трафика IP, поэтому подтверждения (acknowledgement или ACK) для каждого пакета не требуются и не предоставляются.

Важно отметить, что переменная скорость отправки пакетов по туннелю AGGFRAG с контролем перегрузок не является приватной и определяется насыщением сети, но пока форма и время отправки потока инкапсулированного (внутреннего) трафика не влияют напрямую на (внешнее) насыщение сети, изменение скорости туннелирования не снижают уровень конфиденциальности внутреннего потока трафика.

2.4.2.1. Выключатели

В дополнение к контролю перегрузок реализациям, поддерживающим режим без контроля перегрузок, **следует** реализовать выключатели (circuit breaker) [RFC8084] как метод восстановления в крайних случаях. Когда такие выключатели разрешены, реализации **следует** также разрешать отчёты контроля перегрузок, чтобы выключатели имели сведения, требуемые для их работы.

Соображения о перегрузке псевдопроводов [RFC7893] применимы к заданным в этом документе механизмам, особенно в части текста о неэластичном трафике.

Один из вариантов выключателя с медленным отключением, который реализация может обеспечить, использует 2 значения - число постоянных потерь, определяющее переключение, и интервал времени, в течение которого эти потери должны наблюдаться для отключения. Эти 2 значения должны задаваться в пользовательской конфигурации. Когда выключатель срабатывает, туннельный трафик отключается и вносится запись в системный журнал (log) или подаётся иной сигнал, указывающий необходимость вмешательства в работу.

2.5. Сводка для обработки у получателя

В SA со включённым AGGFRAG получатель должен выполнять несколько задач.

Получатель **может** обрабатывать входящее содержимое AGGFRAG_PAYLOAD сразу по прибытии, насколько это возможно, т. е. если входящий пакет AGGFRAG_PAYLOAD содержит полные внутренние заголовки, получателю следует сразу же извлечь и передать их. Для неполных пакетов получатель должен хранить части пакетов в памяти, пока они не выходят за пределы окна переупорядочения или не будут получены недостающие части пакетов. В последнем случае выполняется сборка частей. Если в AGGFRAG_PAYLOAD содержится несколько пакетов, их **следует** передавать в порядке размещения в AGGFRAG_PAYLOAD (т. е. сохранить порядок их получения на другой стороне). Издержки этого метода заключаются в росте нарушения порядка доставки вложенных пакетов из-за их агрегирования.

Вместо описанного выше метода получатель **может** восстановить порядок содержимого AGGFRAG_PAYLOAD (2.2.3. Фрагментация, порядковые номера и данные лишь с заполнением) и лишь после этого передавать вложенные пакеты. Издержками этого метода является задержка при потере пакетов вплоть до значения таймера потерь (или полного окна переупорядочения, если таймер потерь не применяется). Кроме того, это может вызывать дополнительные всплески выходного потока. Эти всплески могут возникать, когда потерянный пакет отбрасывается из окна переупорядочения, а оставшиеся в окне внешние пакеты сразу же обрабатываются и передаются друг за другом.

Кроме того, при включённом контроле перегрузок получатель передаёт данные такого контроля (6.1.2. Формат содержимого AGGFRAG_PAYLOAD при контроле перегрузок) отправителю, как указано в параграфе 2.4.2 и разделе 3.

Отметим получение некорректных значений BlockOffset - для учёта неверного поведения отправителей получателю **следует** аккуратно обрабатывать случаи, когда BlockOffset в последовательных пакетах и/или общий для них внутренний пакет не согласуются. Получатель **может** отбросить вложенный пакет или один или оба внешних пакета.

3. Индикация перегрузки

Для поддержки режима с контролем перегрузок получателю нужно знать частоту потери пакетов и примерное значение RTT [RFC5348]. Чтобы получить эти значения, получатель передаёт данные контроля перегрузок отправителю через

свою SA. Таким образом, для поддержки контроля перегрузок получатель **должен** иметь парную SA к отправителю (это всегда выполняется при создании туннеля с использованием IKEv2). Если SA к отправителю не поддерживает AGGFRAG_PAYLOAD, для передачи сведений служит пустое содержимое AGGFRAG_PAYLOAD (только заголовки).

Для расчёта частоты потерь в соответствии с [RFC5348] получателю нужна оценка RTT, поэтому отправитель сообщает эту оценку в поле заголовка RTT. При старте это 0, пока не будет получена реальная оценка RTT. Для оценки отправителем своего значения RTT он помещает метку времени в поле заголовка TVal. При первом приёме метки TVal получатель записывает значение TVal вместе с временем прибытия метки. При последующем получении того же TVal **недопустимо** обновлять записанное время.

Когда получатель передаёт свой заголовок контроля перегрузки, он помещает последнее записанное значение TVal в поле заголовка TEcho вместе с двумя значениями задержки Echo Delay и Transmit Delay. Значение Echo Delay указывает разницу между записанным временем прибытия TVal и текущим временем в микросекундах. Transmit Delay указывает текущую задержку получателя при передаче через туннель (среднее время между отправкой пакетов на его стороне туннеля AGGFRAG).

Когда получатель принимает своё значение TVal в поле заголовка TEcho, он делает две оценки RTT. Первая указывает фактическую задержку, определяемую вычитанием TEcho из текущего времени и последующим вычитанием Echo Delay. Вторая оценка RTT определяется сложением полученного в заголовке значения Transmit Delay с собственной задержкой передачи у отправителя (среднее время между отправкой пакетов на его стороне туннеля AGGFRAG). Большее из полученных значений следует выбирать в качестве RTT. Две оценки RTT нужны для обработки разных комбинаций быстрого и медленного пути через туннель с большей или меньшей фиксированной скоростью в туннеле. Выбор большего из двух значений гарантирует, что RTT не будет меньше совокупной задержки передачи на основе скорости IP-TFS (вторая оценка) и фактического RTT на пути пакетов через туннель (первая оценка).

Получатель также рассчитывает и передаёт в поле заголовка LossEventRate частоту потерь для использования её отправителем. Это несколько отличается от [RFC4342], где отправителю периодически передаются все данные о потере пакетов для выполнения расчётов. В Приложении B представлен способ расчёта частоты потерь. Изначально это 0 (нет потерь), пока получатель не соберёт достаточно данных для оценки частоты потерь.

3.1. Поддержка ECN

В дополнение к обычным сведениям о потере пакетов режим AGGFRAG поддерживает использование битов ECN в заголовке инкапсуляции IP [RFC3168] для идентификации перегрузки. Если применение ECN разрешено и пакет приходит на выходную (приёмную) сторону с установленным битом перегрузки (Congestion Experienced или CE), получатель считает пакет отброшенным, хотя это не так. Получатель **должен** устанавливать бит E в любом заголовке содержимого AGGFRAG_PAYLOAD, со значением LossEventRate, выведенным из рассматриваемого CE.

В [RFC6040], обновляющем [RFC3168] и [RFC4301], задана маркировка внешнего поля ECN на основе поля ECN во вложенном заголовке. Поскольку в режиме AGGFRAG в одном внешнем пакете может быть несколько вложенных пакетов и нет очевидного способа сопоставить множество значений из вложенных пакетов с ECN в одном внешнем пакете, входной точке туннеля **следует** работать в режиме совместимости (compatibility), а не в режиме default из [RFC6040]. В частности, это означает, что входная (передающая) конечная точка туннеля всегда устанавливает в поле ECN значение Not-ECT [RFC6040].

4. Конфигурация туннелей AGGFRAG для IP-TFS

IP-TFS служит для развёртывания с минимальной настройкой. Всю конфигурацию IP-TFS следует задавать на входной (передача) стороне одностороннего туннеля. Предполагается поддержка работы без IKEv2 по крайней мере при локальной настройке. Документы YANG и MIB для IP-TFS заданы в [RFC9348] и [RFC9349].

4.1. Пропускная способность

Пропускная способность является опцией локальной конфигурации. Для режима без контроля перегрузки следует настраивать пропускную способность. В режиме с контролем перегрузки полоса может настраиваться или алгоритм контроля перегрузки будет определять и поддерживать максимальную пропускную способность. Стандартизированный метод настройки не требуется.

4.2. Фиксированный размер пакетов

Фиксированный размер пакетов для использования при туннельной инкапсуляции **может** настраиваться вручную или определяться автоматически с использованием таких методов, как PLMTUD [RFC4821] [RFC8899] или PMTUD [RFC1191] [RFC8201]. Поскольку у PMTUD имеются проблемы, PLMTUD считается более предпочтительным вариантом. Стандартизированный метод настройки не требуется.

4.3. Контроль перегрузок

Контроль перегрузок является опцией локальной конфигурации. Стандартизированный метод настройки не требуется.

5. IKEv2

5.1. Уведомления USE_AGGFRAG

Как отмечено выше, в туннелях AGGFRAG применяется содержимое ESP типа AGGFRAG_PAYLOAD. При использовании IKEv2 новое уведомление USE_AGGFRAG разрешает содержимое AGGFRAG_PAYLOAD для пары Child SA. Используемый метод похож на согласование USE_TRANSPORT_MODE, описанное [RFC7296]. Для запроса применения AGGFRAG_PAYLOAD с парой Child SA инициатор включает уведомление USE_AGGFRAG в содержимое SA, запрашивающее новую Child SA (при начальном IKE_AUTH или в процессе обмена CREATE_CHILD_SA). Если запрос принимается, отклик **должен** включать уведомление типа USE_AGGFRAG. При отклонении запроса Child SA будет создаваться без возможности использовать содержимое AGGFRAG_PAYLOAD. Если инициатора это не устраивает, он **должен** удалить Child SA.

Поскольку применение AGGFRAG_PAYLOAD в настоящее время определено лишь для туннелей, не использующих транспортный режим, уведомления USE_AGGFRAG **недопустимо** сочетать с уведомлениями USE_TRANSPORT. Уведомление USE_AGGFRAG содержит 1 октет флагов, задающих требования от отправителя уведомления. Если получатель не знает или не поддерживает какой-либо из флагов, ему **не следует** разрешать использование AGGFRAG_PAYLOAD (не отвечая уведомлением USE_AGGFRAG или, в случае инициатора, удаляя Child SA, если использование вновь созданной SA без AGGFRAG_PAYLOAD неприемлемо).

Тип уведомления и значения флагов определены в параграфе 6.1.4. Уведомления IKEv2 USE_AGGFRAG.

6. Форматы пакетов и данных

Определённые ниже форматы данных и флагов являются базовыми и могут применяться не только с IP-TFS, но такое использование выходит за рамки этого документа.

6.1. Содержимое AGGFRAG_PAYLOAD

Содержимое AGGFRAG указывается значением AGGFRAG_PAYLOAD (144) в ESP Next Header, зарезервированным в пространстве номеров протоколов IP. Первый октет содержимого указывает формат остальных данных.

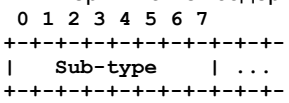

Рисунок 3. Формат содержимого AGGFRAG_PAYLOAD.

Sub-type

8-битовое значение, указывающее формат содержимого.

Этот документ определяет два субтипа содержимого, описанных ниже.

6.1.1. Формат содержимого AGGFRAG_PAYLOAD без контроля перегрузок

Содержимое AGGFRAG_PAYLOAD без контроля перегрузок включает 4 октета заголовка, за которым следует переменное число DataBlocks, как показано на рисунке 4.

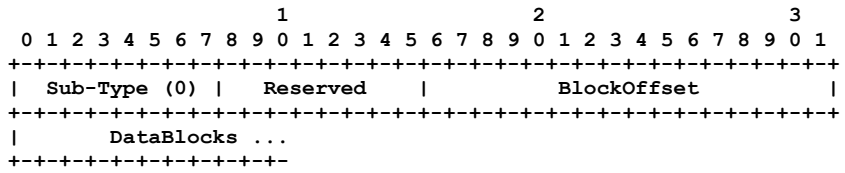

Рисунок 4. Формат содержимого без контроля перегрузок.

Sub-type

Октет, указывающий формат содержимого. При отсутствии контроля перегрузок имеет значение 0.

Reserved

Устанавливается значение 0 при создании и игнорируется при получении.

BlockOffset

16-битовое целое число без знака, указывающее количество октетов данных DataBlocks до начала нового блока данных. Если новый блок данных начинается в последующем содержимом (payload), BlockOffset указывает конец данных DataBlocks. В этом случае все данные DataBlocks относятся к текущему собираемому блоку. Если BlockOffset распространяется в последующее содержимое, значение учитывает лишь данные DataBlocks (т. е. не считаются последующие пакеты данных, не являющиеся DataBlocks, такие как октеты заголовка).

DataBlocks

Переменное число октетов с начала блока данных или продолжения предыдущего блока, за которыми могут следовать другие блоки данных.

6.1.2. Формат содержимого AGGFRAG_PAYLOAD при контроле перегрузок

Содержимое AGGFRAG_PAYLOAD с контролем перегрузок включает 24 октета заголовка и переменное число блоков данных DataBlocks, как показано на рисунке 5.

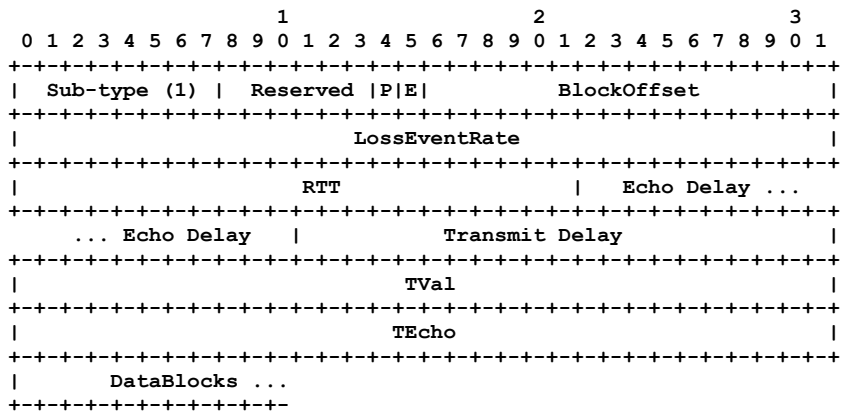

Рисунок 5. Формат данных контроля перегрузок.

Sub-type

Октет, указывающий формат содержимого. При использовании контроля перегрузок имеет значение 1.

Reserved

6-битовое поле, устанавливаемое в 0 при создании и игнорируемое при получении.

P

Флаг, указывающий, что происходит зондирование PLMTUD. Флаг может служит для предотвращения трактовки алгоритмом контроля перегрузок отсутствующих пакетов как потерь при работе алгоритма зондирования PLMTUD.

E

Флаг, указывающий, что биты ECN CE были получены и использованы для вывода сообщаемого LossEventRate.

BlockOffset

Такое же значение, как для формата без контроля перегрузок (см. параграф 6.1.1).

LossEventRate

32-битовое значение - 0 при отсутствии потерь, 1/LossEventRate в остальных случаях.

RTT

22-битовое значение, указывающее текущую оценку RTT отправителем (в микросекундах). Поле **может** иметь значение 0 до расчёта RTT отправителем. Для SA без поддержки AGGFRAG_PAYLOAD **следует** устанавливать 0. Если RTT не меньше 0x3FFFFF, **должно** устанавливаться значение 0x3FFFFF.

Echo Delay

21-битовое значение задержки между первым приёмом значения TVal получателем (в микросекундах) и отправкой его обратно в TEcho. Если задержка не меньше 0x1FFFFF, **должно** устанавливаться значение 0x1FFFFF.

Transmit Delay

21-битовое значение задержки передачи в микросекундах. Это фиксированная (или средняя) задержка между отправкой получателем пакетов в туннель IP-TFS. Если задержка не меньше 0x1FFFFF, **должно** устанавливаться значение 0x1FFFFF.

TVal

Неанализируемое (opaque) 32-битовое значение, возвращаемое получателем в поле TEcho последующих пакетов вместе со значением Echo Delay, указывающим время, занятое на возврат (echo).

TEcho

Неанализируемое 32-битовое значение из поля TVal в полученном пакете. Полученное значение TVal помещается в TEcho вместе со значением Echo Delay, указывающим время, прошедшее с момента получения TVal.

DataBlocks

Переменное число октетов с начала блока данных или продолжения предыдущего блока, за которыми могут следовать другие блоки данных. Для особого случая передачи данных контроля перегрузки в SA без поддержки IP-TFS это поле **должно** быть пустым (нулевой размер).

6.1.3. Блоки данных

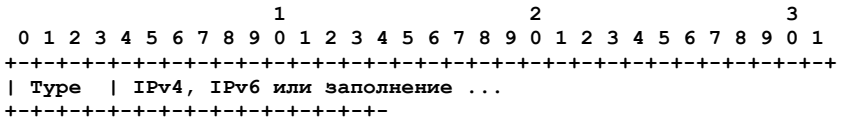

Рисунок 6. Формат блока данных.

Type

4-битовое поле, указывающее тип блока данных - 0x0 Pad Data Block, 0x4 блок данных IPv4, 0x6 блок данных IPv6.

6.1.3.1. Блок данных IPv4

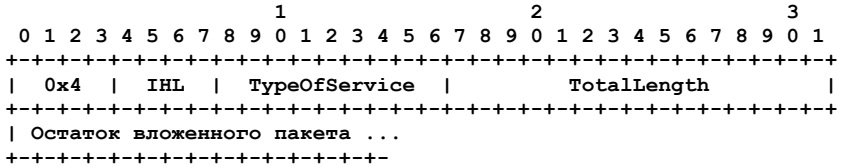

Рисунок 7. Формат блока данных IPv4.

Фактическое содержимое инкапсулированного заголовка IPv4, иными словами, начало этого блока является началом инкапсулированного пакета IP.

Type

4-битовое значение 0x4 указывает IPv4 (первый полубайт пакета IPv4).

TotalLength

16-битовое целое число без знака - поле Total Length вложенного пакета IPv4.

6.1.3.2. Блок данных IPv6

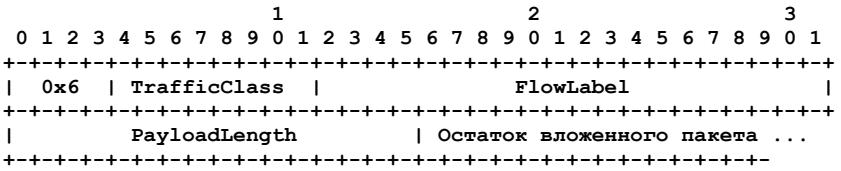

Рисунок 8. Формат блока данных IPv6.

Фактическое содержимое инкапсулированного заголовка IPv6, иными словами, начало этого блока является началом инкапсулированного пакета IP.

Type

4-битовое значение 0x6 указывает IPv6 (первый полубайт пакета IPv6).

PayloadLength

16-битовое целое число без знака - поле Payload Length вложенного пакета IPv6.

6.1.3.3. Блок данных заполнения

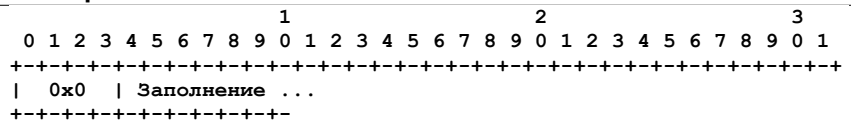

Рисунок 9. Формат блока данных заполнения.

Type
4-битовое значение 0x0 указывает блок данных заполнения.

Padding
Финальная часть инкапсулирующего пакета.

6.1.4. Уведомления IKEv2 USE_AGGFRAG

Как указано в параграфе 5.1, уведомление USE_AGGFRAG служит для согласования использования значения поля ESP Next Header AGGFRAG_PAYLOAD.

USE_AGGFRAG Notification Message State Type имеет значение 16442.

Данные (payload) уведомления содержат 1 октет флагов требований. В настоящее время заданы 2 флага, но будущие спецификации могут добавить определения флагов.

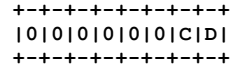

Рисунок 10. Флаги требований USE_AGGFRAG.

0
6 битов, которые **должны** устанавливаться в 0 при передаче, пока не будут заданы в новых спецификациях.

C
Флаг контроля перегрузки, устанавливаемый (1), когда отправитель указывает, что данные контроля перегрузок **должны** периодически возвращаться ему, как указано в разделе 3. Индикация перегрузки.

D
Флаг запрета фрагментирования, установка которого указывает, что отправитель уведомления не поддерживает приём фрагментов (т. е. вложенный пакет **должен** передаваться в одном блоке данных). Это значение применяется лишь к тому, что отправитель может получить. Отправитель **может** продолжать отправку фрагментов, пока получатель не указал аналогичное ограничение в своём уведомлении USE_AGGFRAG.

7. Взаимодействие с IANA

7.1. Значение ESP Next Header

Агентство IANA включило номер протокола IP 144 в реестр Protocol Numbers - Assigned Internet Protocol Numbers.

```
Decimal:  144
Keyword:  AGGFRAG
Protocol:  AGGFRAG encapsulation payload for ESP
Reference:  RFC 9347
```

7.2. Субтипы AGGFRAG_PAYLOAD

Агентство IANA создало реестр AGGFRAG_PAYLOAD Sub-Types в новой категории ESP AGGFRAG_PAYLOAD с политикой регистрации Expert Review [RFC8126] [RFC7120].

```
Name:  AGGFRAG_PAYLOAD Sub-Types
Description:  AGGFRAG_PAYLOAD Payload Formats
Reference:  RFC 9347
```

Исходное содержимое реестра приведено в таблице 1.

Таблица 1. Субтипы AGGFRAG_PAYLOAD.

Субтип	*Имя*	*Документ*
0	Формат без контроля перегрузок	RFC 9347
1	Формат с контролем перегрузок	RFC 9347
3-255	Резерв	

7.3. Тип уведомлений о статусе USE_AGGFRAG

Агентство IANA включило тип статуса USE_AGGFRAG в реестр IKEv2 Notify Message Types - Status Types.

```
Decimal:  16442
Name:  USE_AGGFRAG
Reference:  RFC 9347
```

8. Вопросы безопасности

Этот документ описывает механизмы агрегирования и фрагментирования с целью эффективной реализации TFC для трафика IP. Предполагается, что этот подход осложнит анализ трафика IPsec. В дополнение к обеспечиваемой этим механизмом защите IP-TFS применяет протоколы защиты [RFC4303] [RFC7296], поэтому соображения безопасности для них применимы и к IP-TFS.

Как отмечено в параграфе 3.1, биты ECN не защищены в IPsec и позволяют организовать скрытый канал. По этой причине применение ECN **не следует** включать по умолчанию.

Как отмечено в параграфе 2.4.2, для поддержки TFC инкапсулированный поток трафика не должен влиять на перегрузку предсказуемым способом, а если это происходит, следует применять режим без контроля перегрузок.

9. Литература

9.1. Нормативные документы

[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>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[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>.

9.2. Дополнительная литература

[AppCrypt] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C", 1996.

[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>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/RFC2914, September 2000, <https://www.rfc-editor.org/info/rfc2914>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4342] Floyd, S., Kohler, E., and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, DOI 10.17487/RFC4342, March 2006, <https://www.rfc-editor.org/info/rfc4342>.

[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>.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, DOI 10.17487/RFC5348, September 2008, <https://www.rfc-editor.org/info/rfc5348>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <https://www.rfc-editor.org/info/rfc7120>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <https://www.rfc-editor.org/info/rfc7510>.

[RFC7893] Stein, Y(J)., Black, D., and B. Briscoe, "Pseudowire Congestion Considerations", RFC 7893, DOI 10.17487/RFC7893, June 2016, <https://www.rfc-editor.org/info/rfc7893>.

[RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <https://www.rfc-editor.org/info/rfc8084>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8546] Trammell, B. and M. Kuehlewind, "The Wire Image of a Network Protocol", RFC 8546, DOI 10.17487/RFC8546, April 2019, <https://www.rfc-editor.org/info/rfc8546>.

[RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/info/rfc8899>.

[RFC9329] Pauly, T. and V. Smyslov, "TCP Encapsulation of Internet Key Exchange Protocol (IKE) and IPsec Packets", RFC 9329, DOI 10.17487/RFC9329, November 2022, <https://www.rfc-editor.org/info/rfc9329>.

[RFC9348] Fedyk, D. and C. Hopps, "A YANG Data Model for IP Traffic Flow Security", RFC 9348, DOI 10.17487/RFC9348, January 2023, <https://www.rfc-editor.org/info/rfc9348>.

[RFC9349] Fedyk, D. and E. Kinzie, "Definitions of Managed Objects for IP Traffic Flow Security", RFC 9349, DOI 10.17487/RFC9349, January 2023, <https://www.rfc-editor.org/info/rfc9349>.

Приложение A. Пример потока инкапсулированных пакетов IP

Ниже приведён пример потока вложенных пакетов IP внутри потока пакетов инкапсулирующего туннеля. Отметим, что инкапсулированные пакеты IP могут начинаться и заканчиваться в любом месте и в одном инкапсулирующем пакете может размещаться часть, целый или несколько вложенных пакетов.

```
 Offset: 0        Offset: 100    Offset: 2000    Offset: 600
[ ESP1  (1404) ][ ESP2  (1404) ][ ESP3  (1404) ][ ESP4  (1404) ]
[--750--][--750--][60][-240-][--3000----------------------][pad]
```

Рисунок 11. Потоки вложенных и внешних пакетов.

Каждое инкапсулирующее пространство ESP имеет фиксированный размер 1404 октета, из которых первые 4 занимает заголовок AGGFRAG. Поток инкапсулируемых пакетов IP (размер включает заголовок IP и данные) включает 2 пакета по 750 октетов, пакет в 60 октетов, пакет в 240 октетов и пакет в 3000 октетов.

BlockOffset в 4 заголовках содержимого AGGFRAG для этого потока пакетов имеют значения 0, 100, 2000 и 600, соответственно. Первый инкапсулирующий пакет (ESP1) имеет значение BlockOffset = 0, указывающее блок данных IP, следующий сразу после заголовка AGGFRAG. В следующем пакете (ESP2) BlockOffset указывает 100 внутрь к началу 60-октетного блока данных. Третий инкапсулирующий пакет (ESP3) содержит среднюю часть 3000-октетного блока данных, поэтому смещение указывает его конец в четвёртом инкапсулирующем пакете (ESP4), где смещение 600 указывает заполнение после завершения 3000-октетного пакета.

Приложение B. Передача и расчёт частоты потерь

Текущий опыт показывает, что контроль перегрузок следует выполнять дружественным к TCP способом. Такой алгоритм описан в [RFC5348]. Для этого варианта использования IP-TFS (как в [RFC4342]) (фиксированный) размер пакетов применяется как размер сегмента для алгоритма. Основная формула расчёта скорости передачи в алгоритме имеет вид

```
X = 1 / (R * (sqrt(2*p/3) + 12*sqrt(3*p/8)*p*(1+32*p^2)))
```

X - скорость передачи (пакет/сек), R - оценка RTT, p - частота потерь (обратное ей значение указывает получатель).

Кроме того, алгоритм [RFC5348] использует также значение X_recv (скорость приёма получателем). Для IP-TFS **можно** установить это значение в соответствии с текущей скоростью передачи туннельного отправителя (X).

Получатель IP-TFS, имеющий оценку RTT от отправителя, может применять описанный в [RFC5348] и [RFC4342] метод для сбора интервалов потерь и расчёта частоты потерь с использованием средневзвешенного значения, как указано. Получатель сообщает обратное этой частоте значение отправителю в поле LossEventRate заголовка содержимого AGGFRAG_PAYLOAD. У отправителя IP-TFS теперь имеются значения R и p, позволяющие корректно рассчитать скорость передачи. В соответствии с [RFC5348] отправителю также следует использовать механизм замедленного старта при организации IP-TFS SA.

Приложение C. Сравнения для IP-TFS

C.1. Сравнение издержек

Для сравнения издержек нужно рассчитать издержки ESP для обычных туннельных пакетов и пакетов AGGFRAG, поэтому нужно выбрать алгоритм шифрования и аутентификации. В этом примере используется алгоритм AES-GCM-256, что ведёт к издержкам IP+ESP в 54 октета.

```
54 = 20 (IP) + 8 (ESPH) + 2 (ESPF) + 8 (IV) + 16 (ICV)
```

Для IP-TFS были выбраны заголовки AGGFRAG_PAYLOAD без контроля перегрузок, добавляющие 4 октета, что даёт суммарные издержки в 58 октетов.

C.1.1. Издержки IP-TFS

Для сравнения принимаются издержки содержимого AGGFRAG в 58 октетов на внешний пакет. Поэтому издержки в расчёте на вложенный пакет получаются делением значения 58 на число требуемых внешних пакетов (дробные части разрешены). Издержки в процентах от размера вложенного пакета являются константой, зависящей от внешнего MTU.

```
OH = 58 / Размер внешних данных (Payload) / Размер вложенного пакета
OH % от размера вложенного пакета = 100 * OH / Размер вложенного пакета
OH % от размера вложенного пакета = 5800 / Размер внешнего пакета
```

Таблица 2. Издержки IP-TFS в процентах от размера вложенного пакета.

Тип	IP-TFS	IP-TFS	IP-TFS
MTU	576	1500	9000
PSize	518	1442	8942
40	11,20%	4,02%	0,65%
576	11,20%	4,02%	0,65%
1500	11,20%	4,02%	0,65%
9000	11,20%	4,02%	0,65%

C.1.2. Издержки ESP с заполнением

Издержки в расчёте на вложенный пакет для ESP с постоянной скоростью и заполнением (исходный IPsec TFC) составляют 36 октетов плюс заполнение, пока не требуется фрагментация.

Если нужна фрагментация вложенных пакетов для размещения во внешнем пакете IPsec, издержки составляют число внешних пакетов, требуемых для передачи фрагментированного вложенного пакета, умноженное на сумму внутренних издержек IP (20) и издержек внешнего пакета (54), за вычетом внутренних издержек IP, с добавлением заполнения в конце последнего пакета инкапсуляции. Размер заполнения - это число требуемых пакетов, умноженное на разность размера внешних данных (Outer Payload Size) и издержек IP, за вычетом размера вложенных данных (Inner Payload Size). Таким образом,

```
Размер вложенных данных = Размер пакета IP - Издержки IP
Размер внешних данных = MTU - Издержки IPsec

NF0 = Размер вложенных данных / (Размер внешних данных — Издержки IP)

NF = CEILING(NF0)

OH = NF * (Издержки IP + Издержки IPsec) - Издержки IP
     + NF * (Размер внешних данных - Издержки IP)
     - Размер вложенных данных

OH = NF * (Издержки IPsec + Размер внешних данных)
     - (Издержки IP + Размер вложенных данных)

OH = NF * (Издержки IPsec + Размер внешних данных)
     - Размер вложенного пакета
```

C.2. Сравнение издержек

Ниже приведены значения издержек для некоторых общепринятых значений L3 MTU с целью сравнения. Таблица 3 указывает число добавленных октетов для данного размера пакетов L3 MTU. В таблице 4 приведены издержки в процентах для тех же пакетов размера MTU.

Таблица 3. Сравнение издержек в октетах.

Тип	*ESP+Pad*	*ESP+Pad*	*ESP+Pad*	*IP-TFS*	*IP-TFS*	*IP-TFS*
L3 MTU	*576*	*1500*	*9000*	*576*	*1500*	*9000*
PSize	*522*	*1446*	*8946*	*518*	*1442*	*8942*
40	482	1406	8906	4,5	1,6	0,3
128	394	1318	8818	14,3	5,1	0,8
256	266	1190	8690	28,7	10,3	1,7
518	4	928	8428	58,0	20,8	3,4
576	576	870	8370	64,5	23,2	3,7
1442	286	4	7504	161,5	58,0	9,4
1500	228	1500	7446	168,0	60,3	9,7
8942	1426	1558	4	1001,2	359,7	58,0
9000	1368	1500	9000	1007,7	362,0	58,4

Таблица 4. Сравнение издержек в % от размера вложенных пакетов.

Тип	*ESP+Pad*	*ESP+Pad*	*ESP+Pad*	*IP-TFS*	*IP-TFS*	*IP-TFS*
MTU	*576*	*1500*	*9000*	*576*	*1500*	*9000*
PSize	*522*	*1446*	*8946*	*518*	*1442*	*8942*
40	1205,0%	3515,0%	22265,0%	11,20%	4,02%	0,65%
128	307,8%	1029,7%	6889,1%	11,20%	4,02%	0,65%
256	103,9%	464,8%	3394,5%	11,20%	4,02%	0,65%
518	0,8%	179,2%	1627,0%	11,20%	4,02%	0,65%
576	100,0%	151,0%	1453,1%	11,20%	4,02%	0,65%
1442	19,8%	0,3%	520,4%	11,20%	4,02%	0,65%
1500	15,2%	100,0%	496,4%	11,20%	4,02%	0,65%
8942	15,9%	17,4%	0,0%	11,20%	4,02%	0,65%
9000	15,2%	16,7%	100,0%	11,20%	4,02%	0,65%

C.3. Сравнение доступной пропускной способности

Ещё одним вариантом сравнения двух решения является пропускная способность, обеспечиваемое каждым из них. В последующих параграфах приведено сравнение процентной доли доступной пропускной способности. В качестве общеизвестной базы представлены значения для обычного (нешифрованного) трафика Ethernet и ESP.

C.3.1. Ethernet

Для расчёте доступной пропускной способности сначала вычисляются задержки в расчёте на пакет. Суммарные издержки Ethernet включают 14+4 октета заголовка и контрольной суммы (Cyclic Redundancy Check или CRC), а также 20 октетов кадрирования (преамбула, начало пакета и межпакетный интервал), что даёт 38 октетов. Минимальный размер содержимого (payload) составляет 46 октетов.

Таблица 5. Число октетов L2 на пакет.

Размер	*E + P*	*E + P*	*E + P*	*IPTFS*	*IPTFS*	*IPTFS*	*Enet*	*ESP*
MTU	*590*	*1514*	*9014*	*590*	*1514*	*9014*	*любое*	*любое*
OH	*92*	*92*	*92*	*96*	*96*	*96*	*38*	*74*
40	614	1538	9038	47	42	40	84	114
128	614	1538	9038	151	136	129	166	202
256	614	1538	9038	303	273	258	294	330
518	614	1538	9038	614	552	523	574	610
576	1228	1538	9038	682	614	582	614	650
1442	1842	1538	9038	1709	1538	1457	1498	1534
1500	1842	3076	9038	1777	1599	1516	1538	1574
8942	11052	10766	9038	10599	9537	9038	8998	9034
9000	11052	10766	18076	10667	9599	9096	9038	9074

Таблица 6. Число октетов для 10G Ethernet.

Размер MTU OH	E + P 590 92	E + P 1514 92	E + P 9014 92	IP-TFS 590 96	IP-TFS 1514 96	IP-TFS 9014 96	Enet любое 38	ESP любое 74
40	2,0M	0,8M	0,1M	26,4M	29,3M	30,9M	14,9M	11,0M
128	2,0M	0,8M	0,1M	8,2M	9,2M	9,7M	7,5M	6,2M
256	2,0M	0,8M	0,1M	4,1M	4,6M	4,8M	4,3M	3,8M
518	2,0M	0,8M	0,1M	2,0M	2,3M	2,4M	2,2M	2,1M
576	1,0M	0,8M	0,1M	1,8M	2,0M	2,1M	2,0M	1,9M
1442	678K	812K	138K	731K	812K	857K	844K	824K
1500	678K	406K	138K	703K	781K	824K	812K	794K
8942	113K	116K	138K	117K	131K	138K	139K	138K
9000	113K	116K	69K	117K	130K	137K	138K	137K

Таблица 7. Доля пропускной способности для 10G Ethernet.

Размер MTU OH	E + P 590 92	E + P 1514 92	E + P 9014 92	IP-TFS 590 96	IP-TFS 1514 96	IP-TFS 9014 96	Enet любое 38	ESP любое 74
40	6,51%	2,60%	0,44%	84,36%	93,76%	98,94%	47,62%	35,09%
128	20,85%	8,32%	1,42%	84,36%	93,76%	98,94%	77,11%	63,37%
256	41,69%	16,64%	2,83%	84,36%	93,76%	98,94%	87,07%	77,58%
518	84,36%	33,68%	5,73%	84,36%	93,76%	98,94%	93,17%	87,50%
576	46,91%	37,45%	6,37%	84,36%	93,76%	98,94%	93,81%	88,62%
1442	78,28%	93,76%	15,95%	84,36%	93,76%	98,94%	97,43%	95,12%
1500	81,43%	48,76%	16,60%	84,36%	93,76%	98,94%	97,53%	95,30%
8942	80,91%	83,06%	98,94%	84,36%	93,76%	98,94%	99,58%	99,18%
9000	81,43%	83,60%	49,79%	84,36%	93,76%	98,94%	99,58%	99,18%

Неожиданным результатом применения туннеля AGGFRAG (или иного туннеля с агрегированием пакетов) является то, что для мелких и средних пакетов доступная пропускная способность фактически больше, чем в обычном Ethernet. Это обусловлено сокращением издержек на кадрирование Ethernet. Платой за такую экономию полосу является рост задержек. Задержка связана определяется временем передачи несвязанных октетов во внешнем заголовке туннельного кадра. В таблице 8 показана задержка для некоторых размеров пакета на канале 10G Ethernet, а также задержка, вносимая заполнением при использовании ESP с заполнением.

Таблица 8. Добавленная задержка.

Размер MTU	ESP+Pad 1500	ESP+Pad 9000	IP-TFS 1500	IP-TFS 9000
40	1.12 мксек	7.12 мксек	1.17 мксек	7.17 мксек
128	1.05 мксек	7.05 мксек	1.10 мксек	7.10 мксек
256	0.95 мксек	6.95 мксек	1.00 мксек	7.00 мксек
518	0.74 мксек	6.74 мксек	0.79 мксек	6.79 мксек
576	0.70 мксек	6.70 мксек	0.74 мксек	6.74 мксек
1442	0.00 мксек	6.00 мксек	0.05 мксек	6.05 мксек
1500	1.20 мксек	5.96 мксек	0.00 мксек	6.00 мксек

Отметим, что значения задержки для разных решений очень близки, однако IP-TFS обеспечивает постоянную высокую пропускную способность, в некоторых случаях превосходящую пропускную способность Ethernet, а ESP с заполнением существенно сокращает доступную пропускную способность.

Благодарности

Спасибо Don Fedyk за помощь в рецензировании и редактировании этой работы. Спасибо Michael Richardson, Sean Turner, Valery Smyslov, Tero Kivinen за рецензии и многочисленные предложения по улучшению, а также Joseph Touch за рецензию (transport area) и предложения по улучшению.

Участник работы

Ниже указан человек, внёсший существенный вклад в этот документ.

Lou Berger
LabN Consulting, L.L.C.
Email: lberger@labn.net

Адрес автора

Christian Hopps
LabN Consulting, L.L.C.
Email: chopps@chopps.org

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru