Internet Engineering Task Force (IETF)
Request for Comments: 8200
STD: 86
Obsoletes: 2460
Category: Standards Track
ISSN: 2070-1721

S. Deering
Retired
R. Hinden
Check Point Software

July 2017

Спецификация протокола IPv6

Internet Protocol, Version 6 (IPv6) Specification

Аннотация

Этот документ является спецификацией протокола IP версии 6 (IPv6) и отменяет RFC 2460.

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 http://www.rfc-editor.org/info/rfc8200.

Оглавление

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

1. Введение

Протокол IP версии 6 (IPv6) представляет собой новую версию протокола Internet (Internet Protocol или IP), разработанную для замены предшествующего протокола IP версии 4 (IPv4) [RFC791]. Основные отличия IPv6 от IPv4 можно разделить на несколько категорий.

- *Расширенные возможности адресации*

  IPv6 расширяет размер адресов IP с 32 до 128 битов для поддержки большего числа уровней иерархии адресов, многократного увеличения числа адресуемых устройств и упрощения процесса автоматической настройки адресов. Расширяемость групповой маршрутизации повышается за счёт добавления поля scope (область действия) в групповые адреса. Определён новый тип адресов - anycast, используемых для передачи пакета одному (любому) узлу из группы.

- *Упрощение формата заголовков*

  Некоторые поля заголовков IPv4 в новой версии протокола не используются или не обязательны. Это позволяет сократить издержки на обработку пакетов и расход полосы пропускания каналов на передачу заголовков IPv6.

- *Улучшенная поддержка расширений и опций*

  Изменение способов представления опций в заголовке IP позволяет обеспечить более эффективную пересылку, смягчить ограничения на размер опций и улучшить гибкость введения новых опций в будущем.

- *Поддержка меток потоков*

  Добавлена возможность помечать последовательности пакетов, для которых отправитель запрашивает трактовку в сети как для одного потока.

- *Поддержка аутентификации и приватности*

  В IPv6 добавлены расширения для поддержки проверки подлинности (аутентификации), контроля целостности и (необязательно) конфиденциальности данных.

В этом документе приведена спецификация базового заголовка IPv6, а также изначально определённых расширений и опций заголовков IPv6. Рассмотрены вопросы, связанные с размером пакетов, семантика меток потоков и классов трафика, а также влияние IPv6 на вышележащие протоколы. Формат и семантика адресов IPv6 рассмотрены в [RFC4291]. Версия ICMP для протокола IPv6, которую должны включать все реализации IPv6, описана в [RFC4443].

Порядок передачи данных для IPv6 совпадает с порядком передачи в IPv4, определенным в Приложении B к [RFC791].

Примечание. Поскольку этот документ заменяет собой [RFC2460], во всех упомянутых здесь документах ссылки на RFC 2460 следует считать ссылками на данный документ.

2. Терминология

node - узел
Устройство, реализующее IPv6.

router - маршрутизатор
Узел, пересылающий пакеты IPv6, не адресованные явно ему[1].

host - хост
Любой узел, не являющийся маршрутизатором[1].

upper layer - вышележащий уровень
Протокольный уровень, расположенный непосредственно над IPv6. Примерами являются транспортные протоколы TCP и UDP, протоколы управления типа ICMP, протоколы маршрутизации, такие как OSPF, а также протоколы, «туннелируемые» через IPv6 (т. е., инкапсулированные в пакеты IPv6), такие как IPX (Internetwork Packet Exchange - пакет межсетевого обмена), AppleTalk или IPv6.

link - канал
Коммуникационный объект или среда, посредством которых узлы могут взаимодействовать на канальном уровне (уровне, расположенном непосредственно под IPv6). Примерами могут служить сети Ethernet (с мостами или без них), каналы PPP, сети X.25, Frame Relay или ATM, а также туннели сетевого или вышележащих уровней (например, IPv4 или IPv6).

neighbors - соседи
Узлы, подключённые к одному каналу.

interface - интерфейс
Подключение узла к каналу.

address - адрес
Идентификатор уровня IPv6 для интерфейса или группы интерфейсов.

packet - пакет
Заголовок IPv6 и данные (payload).

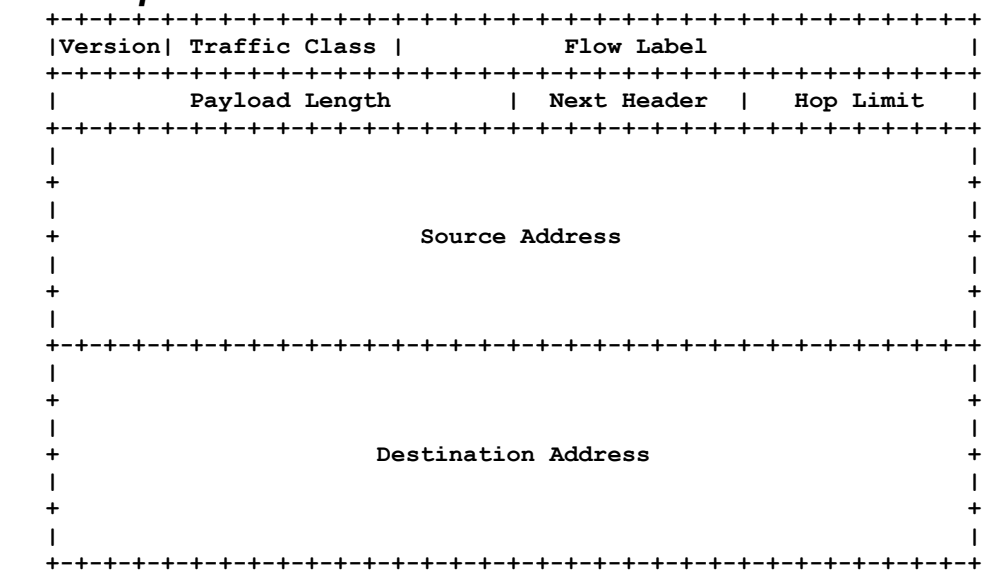
link MTU - максимальный передаваемый блок для канала
Максимальный блок информации (максимальный размер пакета в октетах), который можно передать через канал.

[1]Устройство со множеством интерфейсов может быть настроено на пересылку чужих пакетов, приходящих с некоторых (не всех) его интерфейсов, и отбрасывание подобных пакетов, приходящих с остальных интерфейсов. Такие устройства должны соответствовать требованиям к маршрутизаторам при получении пакетов от интерфейсов первой группы и взаимодействии с соседями через эти интерфейсы. Они должны также соответствовать требованиям к хостам при получении пакетов от интерфейсов второй группы и взаимодействии с соседями через эти интерфейсы.

path MTU - максимальный передаваемый блок для пути

Минимальное значение link MTU среди всех каналов на пути от отправителя к получателю.

3. Формат заголовка IPv6



Version - версия

4-битовое значение номера версии протокола IP (6).

Traffic Class - класс трафика

8-битовое поле классификатора трафика (7. Классы трафика).

Flow Label - метка потока

20-битовая метка потока (6. Метки потоков).

Payload Length - размер данных

16-битовое целое число без знака, показывающее размер поля данных IPv6 (часть пакета, следующая после заголовка) в октетах. Все заголовки расширения (4. Заголовки расширений IPv6) учитываются как данные (размер этих заголовков включается в размер данных пакета).

Next Header - следующий заголовок

8-битовый селектор, указывающий тип заголовка, следующего сразу после заголовка IPv6. Для этого поля используются те же значения, которые определены для поля Protocol в заголовке IPv4 [IANA-PN].

Hop Limit - предельное число пересылок

8-битовое целое число без знака. Значение поля уменьшается на 1 каждым узлом, пересылающим пакет. При получении пакета с Hop Limit = 0 или достижении полем Hop Limit нулевого значения после декрементирования, пакет отбрасывается. Узлу, который является конечным получателем пакета, следует не отбрасывать пакеты с Hop Limit = 0, а обрабатывать такие пакеты обычным способом.

Source Address - адрес отправителя

128-битовый адрес инициатора пакета (см. [RFC4291]).

Destination Address - адрес получателя

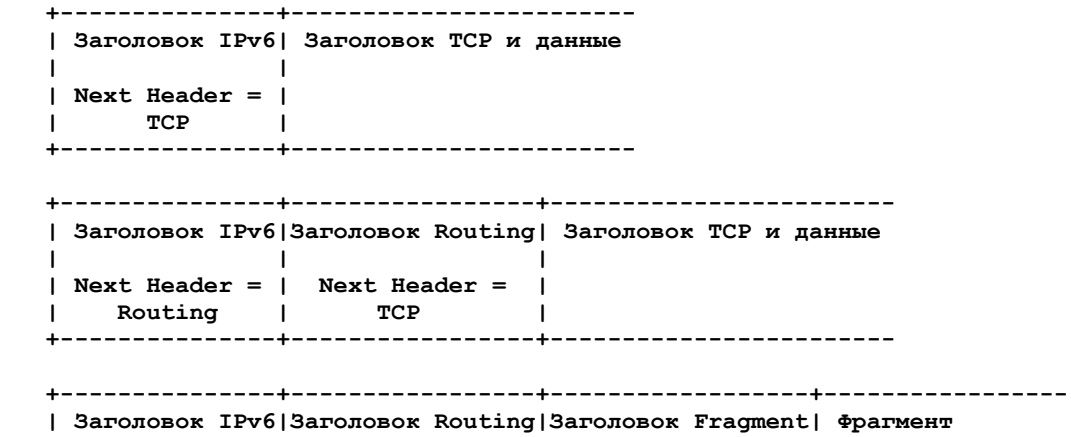
128-битовый адрес получателя пакета (возможно, не конечного, если присутствует заголовок Routing). См. [RFC4291] и параграф 4.4. Заголовок Routing.

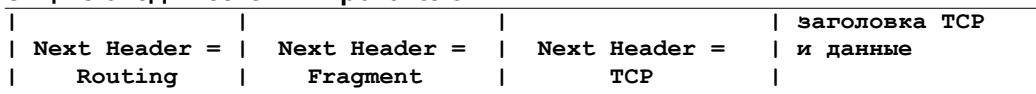
4. Заголовки расширений IPv6

В IPv6 необязательная информация сетевого уровня представляется в виде отдельных заголовков, которые могут размещаться в пакете между заголовком IPv6 и заголовком вышележащего уровня. Определено несколько таких заголовков, идентифицируемых значением поля Next Header.

Заголовки расширения нумеруются значениями из реестра IANA IP Protocol Numbers [IANA-PN] с использованием одинаковых значений для IPv4 и IPv6. При обработке последовательности значений Next Header в пакете первый элемент, не являющийся заголовком расширения [IANA-EH], показывает что следующий элемент в пакете относится к заголовку вышележащего уровня. Если такого заголовка нет, применяется специальное значение No Next Header.

Как показано в приведённых ниже примерах, пакет IPv6 может не включать расширенных заголовков или содержать один или несколько таких заголовков, каждый из которых указывается полем Next Header предшествующего заголовка.





Заголовки расширения (за исключением Hop-by-Hop Options) не обрабатываются, не добавляются и не удаляются узлами на пути доставки пакета, пока он не достигнет узла (или группы узлов в случае групповой адресации), указанного полем Destination Address в заголовке IPv6.

Заголовок Hop-by-Hop Options не добавляется и не удаляется промежуточными узлами, но может быть проверен и обработан любым узлом на пути до того, как достигнет узла (или группы узлов в случае групповой адресации), указанного полем Destination Address в заголовке IPv6. При наличии заголовка Hop-by-Hop Options он должен помещаться непосредственно после заголовка IPv6. Его присутствие указывается нулевым значением поля Next Header в заголовке IPv6.

Примечание. Хотя [RFC2460] требует от всех узлов проверки и обработки заголовка Hop-by-Hop Options, сейчас предполагается, что промежуточные узлы будут проверять и обрабатывать Hop-by-Hop Options только в тех случаях, когда это явно задано в их конфигурации.

На узле-получателе обычное демультиплексирование поля Next Header в заголовке IPv6 вызывает модуль для обработки первого заголовка расширения или заголовка вышележащего уровня (при отсутствии заголовков расширения). В зависимости от содержимого и семантики заголовка расширения выполняется (или не выполняется) обработка следующего заголовка. Поэтому заголовки расширения должны обрабатываться строго в порядке их размещения в пакете, получателю недопустимо сканировать пакет для поиска определённого заголовка расширения и обработки этого заголовка до обработки его предшественников.

Если по результату обработки заголовка получателю требуется обработать следующий заголовок, но значение Next Header в текущем заголовке не распознано, пакет следует отбросить и передать его отправителю сообщение ICMP Parameter Problem с ICMP Code = 1 (unrecognized Next Header type encountered) и полем ICMP Pointer, указывающим смещение непонятного значения в исходном пакете. Такие же действия следует выполнять в случае нулевого значения поля Next Header в любом заголовке за исключением базового заголовка IPv6.

Размер каждого заголовка расширения кратен 8 для выравнивания последующих заголовков по границе 8 октетов. Многооктетные поля в каждом заголовке расширения выравниваются по их естественным границам (поле размером n октетов размещается со смещением от начала заголовка, кратным n для значений n = 1, 2, 4 или 8).

Полная реализация IPv6 включает поддержку следующих заголовков расширения:

Hop-by-Hop Options;
Fragment;
Destination Options;
Routing;
Authentication;
Encapsulating Security Payload.

Первые 4 типа заголовков описаны в этом документе, а два оставшихся - в [RFC4302] и [RFC4303], соответственно. Действующий список заголовков расширения IPv6 можно найти в [IANA-EH].

4.1. Порядок заголовков расширения

При включении в пакет более 1 заголовка расширения рекомендуется размещать заголовки в следующем порядке:

IPv6;
Hop-by-Hop Options;
Destination Options[1];
Routing;
Fragment;
Authentication[2];
Encapsulating Security Payload[2];
Destination Options[3];
заголовок вышележащего уровня.

Каждый заголовок расширения следует включать не более 1 раза, за исключением заголовка Destination Options, который может включаться дважды (перед заголовком Routing и перед заголовком вышележащего уровня).

Если заголовком вышележащего уровня является другой заголовок IPv6 (туннелирование или инкапсуляция IPv6 в IPv6), за ним могут следовать свои заголовки расширения, к которым также относятся приведённые выше рекомендации по порядку следования.

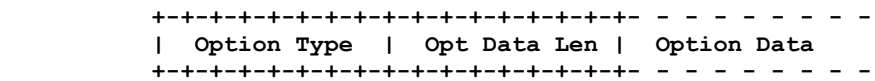
При определении других заголовков расширения должен задаваться порядок их размещения относительно приведённого выше списка.

Узлы IPv6 должны воспринимать и пытаться обрабатывать заголовки расширения при любом порядке и числе вхождений однотипных заголовков расширения в одном пакете. Исключением является заголовок Hop-by-Hop Options, который может присутствовать только непосредственно после заголовка IPv6. Тем не менее, источникам пакетов IPv6 настоятельно рекомендуется включать заголовки расширения в указанном выше порядке, если он не будет изменён последующими спецификациями.

[1]Для опций, которые будут обрабатываться первым получателем, указанным в поле IPv6 Destination Address, плюс получатели, указанные далее в заголовке Routing.
[2]Дополнительные требования к относительному порядку заголовков Authentication и Encapsulating Security Payload приведены в [RFC4303].
[3]Для опций, обрабатываемых только адресатом пакета.

4.2. Опции

Два из определённых в настоящее время заголовков расширения (Hop-by-Hop Options и Destination Options) могут содержать переменное число опций, представленных в формате TLV[1], как показано ниже.



Option Type - тип опции

8-битовый идентификатор типа опции.

Opt Data Len - размер данных опции

8-битовое целое число без знака, указывающее размер поля Option Data в октетах.

Option Data - данные опции

Поле переменного размера, определяемое типом опции.

Последовательность опций должна обрабатываться строго в порядке из размещения в заголовке. Получателю недопустимо сканировать заголовки в поиске той или иной опции с целью её обработки ранее предшествующих.

Идентификаторы Option Type представляются таким образом, что два старших бита задают действие, которое должен выполнить узел IPv6, если значение Option Type ему неизвестно:

00 пропустить опцию и продолжить обработку заголовка;

01 отбросить пакет;

10 отбросить пакет и, независимо от того, указывает ли поле Destination Address групповой адрес, передать отправителю (Source Address) сообщение ICMP Parameter Problem (Code=2, неизвестное значение Option Type);

11 отбросить пакет и, если поле Destination Address не содержит групповой адрес, передать сообщение ICMP Parameter Problem с кодом 2 (неизвестное значение Option Type) по адресу отправителя.

Третий по старшинству бит Option Type указывает, может ли поле Option Data изменяться на пути к адресату. При наличии в пакете заголовка Authentication для любой опции, данные которой могут меняться в пути, значение поля Option Data должно считаться нулевым при расчёте и проверке контрольной суммы пакета.

0 - Option Data не меняется в пути.

1 - Option Data может меняться в пути.

Три описанных выше старших бита считаются частью поля Option Type, но не являются независимыми от типа опции (т. е. конкретная опция указывается всеми битами поля Option Type, а не только 5 младшими битами этого поля).

Заголовки Hop-by-Hop Options и Destination Options используют общее пространство значений Option Type. Однако спецификация конкретной опции может ограничивать применение типа одним из этих двух заголовков.

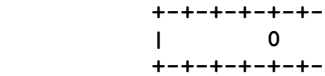
Для отдельных опций могут использоваться специальные требования, чтобы многооктетные поля в Option Data выравнивались по естественным границам. Требования по выравниванию для опций задаются с использованием нотации xn+y, показывающей, что поле Option Type должно представлять собой целое число, размер которого в октетах кратен x, со смещением y октетов от начала заголовка. Например,

2n указывает 2-октетное значение с произвольным смещением от начала заголовка;

8n+2 указывает 8-октетное значение со смещением 2 октета.

Имеется для варианта, которые можно применять при необходимости выравнивания опций путём заполнения до границы, кратной 8 октетам. Эти варианты заполнения должны поддерживаться всеми реализациями IPv6.

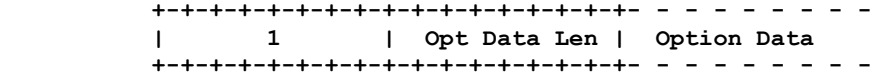
Pad1 (требований по выравниванию нет)



Опция Pad1[2] используется для вставки одного октета заполнения в область опций заголовка. Если требуется заполнить более одного октета, следует использовать описанную ниже опцию PadN, а не набор опций Pad1.

PadN (требований по выравниванию нет)

Опция PadN используется для вставки в область Options заголовка двух или более октетов заполнения. Для N октетов заполнения поле Opt Data Len содержит значение N-2, а поле Option Data - N-2 октетов 0.



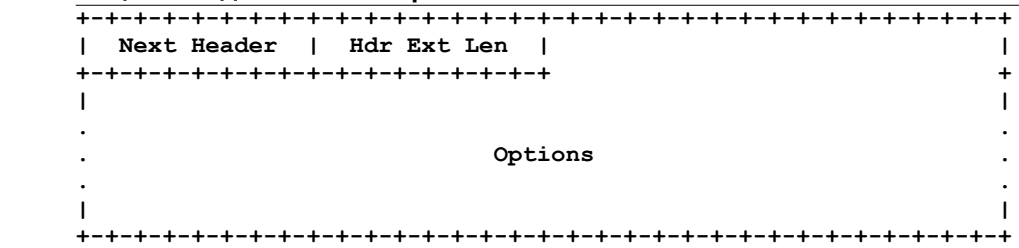
Рекомендации по формату новых опций приведены в Приложении A.

4.3. Заголовок Hop-by-Hop Options

Заголовок Hop-by-Hop Options используется для передачи дополнительной информации, которая может проверяться и обрабатываться на каждом узле по пути доставки пакета. Заголовок Hop-by-Hop Options указывается значением Next Header = 0 в заголовке IPv6 и использует формат, показанный на рисунке.

[1]Type-length-value - тип, размер, значение.

[2]Формат опции Pad1 отличается от обычного TLV, опция не включает полей размера и значения.



Next Header - следующий заголовок

8-битовый селектор типа заголовка, следующего непосредственно за Hop-by-Hop Options. Используются те же значения, которые применяются в поле Protocol заголовков IPv4 [IANA-PN].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка в 8-октетных словах без учёта первых 8 октетов.

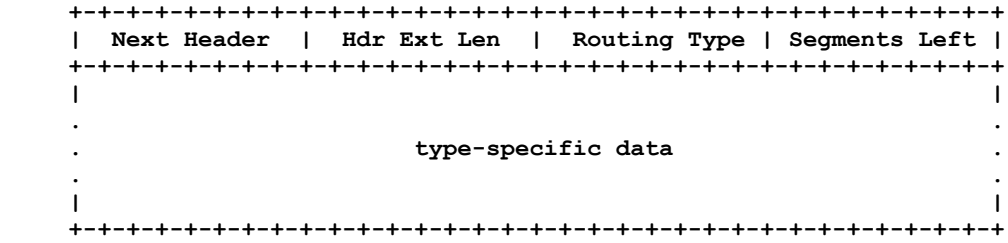
Options - опции

Поле переменного размера, содержащее одну или множество опций в формате TLV, как описано в параграфе 4.2. Общий размер заголовка Hop-by-Hop Options должен быть кратным 8 октетам.

В этом документе для данного заголовка расширения определены только опции Pad1 и PadN (параграф 4.2).

4.4. Заголовок Routing

Заголовок Routing используется источником пакетов IPv6 для указания одного или множества промежуточных узлов, через которые пакет должен пройти на пути к адресату. Функционально этот заголовок очень похож на опции Loose Source и Record Route в заголовках IPv4. Заголовок Routing указывается значением Next Header = 43 в предшествующем ему заголовке и имеет показанный на рисунке формат.



Next Header - следующий заголовок

8-битовый селектор, определяющий тип заголовка, следующего непосредственно за Routing. Используются те же значения, которые применяются в поле Protocol заголовков IPv4 [IANA-PN].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка в 8-октетных словах без учёта первых 8 октетов.

Routing Type - тип маршрутизации

8-битовый идентификатор конкретного варианта заголовка Routing.

Segments Left - число оставшихся сегментов

8-битовое целое число без знака, показывающее число оставшихся сегментов маршрута (число явно указанных промежуточных узлов), которые должны быть пройдены на оставшемся пути к адресату.

type-specific data - данные

Поле переменного размера, формат которого определяется значением поля Routing Type. Размер этого поля должен быть таким, чтобы полный размер заголовка Routing был кратен 8 октетам.

Если при обработке полученного пакета узел встречает заголовок Routing с неизвестным значением поля Routing Type, поведение узла определяется значением поля Segments Left, как описано ниже.

Если Segments Left = 0, узел должен игнорировать заголовок Routing и перейти к обработке следующего заголовка в пакете, тип которого указан полем Next Header в заголовке Routing.

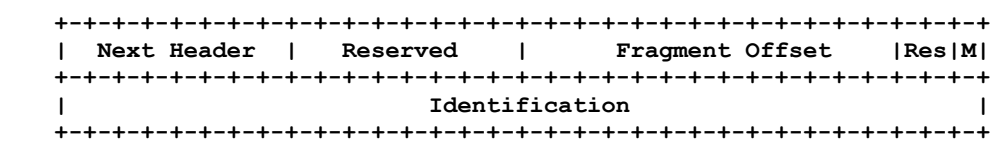
Если значение поля Segments Left отлично от нуля, узел должен отбросить пакет и передать сообщение ICMP Parameter Problem с кодом 0 (нераспознанное значение Routing Type) по адресу отправителя.

Если после обработки заголовка Routing получивший пакет узел определяет, что пакет будет пересылаться в канал, для которого значение MTU меньше размера пакета, узел должен отбросить пакет и передать сообщение ICMP Packet Too Big по адресу отправителя.

Определённые в настоящее время заголовки IPv6 Routing и их статус описаны в [IANA-RH]. Рекомендации по распределению значений для IPv6 Routing Header даны в [RFC5871].

4.5. Заголовок Fragment

Заголовок Fragment используется отправителем IPv6 для передачи пакетов, размер которых превышает значение path MTU для адресата[1]. Заголовок Fragment указывается значением Next Header = 44 в непосредственно предшествующем заголовке и имеет формат, показанный на рисунке.



[1]В отличие от IPv4 фрагментация в IPv6 выполняется только отправителями, а не маршрутизаторами на пути доставки (5. Размер пакетов).

Next Header - тип фрагментируемой части

8-битовый селектор, определяющий исходный тип заголовка фрагментируемой части (Fragmentable Part) исходного пакета (см. определение ниже). Используются те же значения, которые применяются в поле Protocol заголовков IPv4 [IANA-PN].

Reserved - резерв

8-битовое резервное поле. При передаче это поле заполняется нулями, а на приёмной стороне игнорируется.

Fragment Offset - смещение фрагмента

13-битовое целое число без знака, указывающее смещение (в 8-октетных блоках) данных, размещённых вслед за этим заголовком, относительно начала фрагментируемой части исходного пакета.

Res

2-битовое резервное поле. При передаче это поле заполняется нулями, а на приёмной стороне игнорируется.

Флаг M

1 - есть ещё фрагменты, 0 - последний фрагмент.

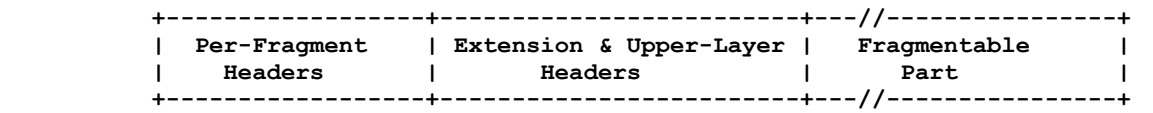
Identification - идентификация

32 бита (см. описание ниже).

Для передачи пакета, размер которого превышает значение MTU на пути к адресату, узел-источник может поделить такой пакет на фрагменты и передать каждый фрагмент в форме отдельного пакета для последующей сборки на приёмной стороне.

Для каждого пакета, который будет фрагментироваться, узел-источник генерирует значение Identification. Это значение для фрагментированного пакета должно отличаться от значений для фрагментированных пакетов, переданных «недавно»[1] с такими же значениями полей Source Address и Destination Address. При наличии заголовка Routing для фрагментированных пакетов Destination Address трактуется, как адрес конечного получателя.

Исходный нефрагментированный пакет будем рассматривать, как состоящий из трёх частей, показанных на рисунке.



Часть Per-Fragment Headers должна состоять из заголовка IPv6 и всех заголовков расширения, которые должны обрабатываться узлами на пути к получателю, т. е. всех заголовков вплоть до Routing (при наличии) или Hop-by-Hop Options (при наличии), включительно. Если ни одного из двух упомянутых заголовков нет, эта часть не включает заголовков расширения.

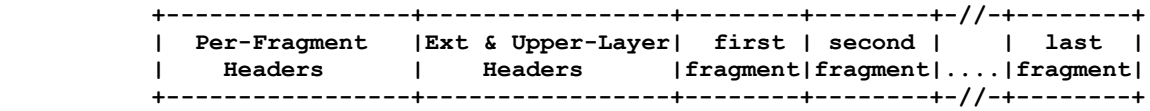
Следующая часть включает все оставшиеся заголовки расширения, которые не вошли в Per-Fragment Headers. При этом заголовки ESP[2] не считаются заголовками расширения. Заголовком вышележащего уровня является первый заголовок, не являющийся заголовком расширения IPv6. Примерами такого заголовка могут быть заголовки TCP, UDP, IPv4, IPv6, ICMPv6, ESP.

Fragmentable Part представляет оставшуюся часть пакета после заголовка вышележащего уровня или любого другого заголовка (например, основного заголовка IPv6 или заголовка расширения, в котором поле Next Header имеет значение No Next Header).

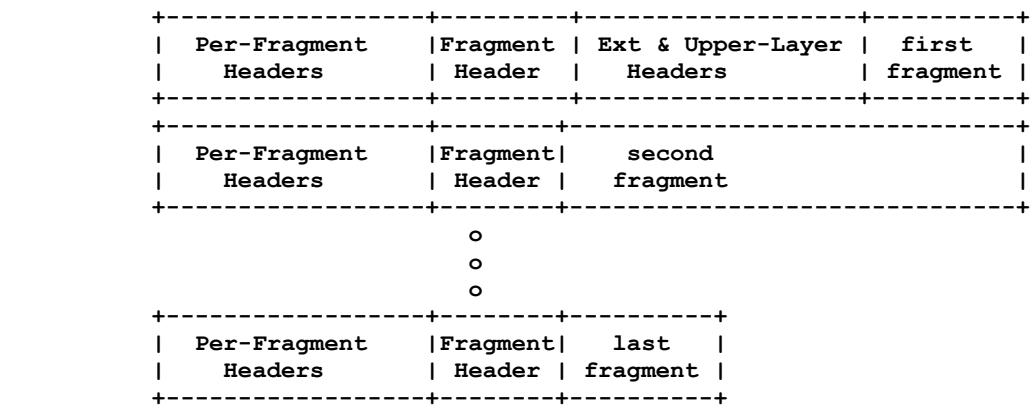
Fragmentable Part исходного пакета разбивается на фрагменты, размеры которых следует выбирать так, чтобы размер полученных в результате пакетов не превышал значения MTU на пути к адресатам. Каждый фрагмент, за исключением последнего (самого правого) имеет размер, кратный 8 октетам.

Фрагменты передаются в отдельных «фрагментированных пакетах», как показано ниже.

Исходный пакет



Фрагментированные пакеты



Пакет с первым фрагментом состоит из 4 частей, описанных ниже.

[1]Недавно означает «в пределах максимального вероятного времени жизни пакета, включая время передачи от отправителя к получателю и ожидания сборки фрагментов». Однако источник не обязан знать максимальное время жизни пакета и предполагается, что требование может быть выполнено с помощью алгоритма, обеспечивающего низкую частоту повторного использования значений. Примеры таких алгоритмов описаны в [RFC7739].

[2]Encapsulating Security Payload - инкапсуляция защищённых данных.

(1) Заголовки Per-Fragment исходного пакета с полем Payload Length в исходном заголовке IPv6, измененным в соответствии с размером данного фрагмента (без учёта самого заголовка IPv6), и полем Next Header в последнем заголовке Per-Fragment, содержащим значение 44.

(2) Заголовок Fragment, содержащий перечисленные ниже поля.

Next Header со значением, указывающим первый заголовок после заголовков Per-Fragment в исходном пакете.

Значение Fragment Offset, указывающее смещение фрагмента (в 8-октетных блоках) от начала Fragmentable Part исходного пакета. В первом (самом левом) фрагменте Fragment Offset = 0.

Флаг M со значением 1, поскольку это первый фрагмент.

Значение Identification, созданное для исходного пакета.

(3) Заголовки расширения (при наличии) и заголовок вышележащего уровня. Эти заголовки должны включаться в первый фрагмент[1].

(4) Первый фрагмент.

Пакеты следующих фрагментов состоят из трёх частей.

(1) Заголовки Per-Fragment исходного пакета с полем Payload Length в исходном заголовке IPv6, указывающим размер данного фрагмента (без учёта самого заголовка IPv6), и полем Next Header в последнем заголовке Per-Fragment со значением 44.

(2) Заголовок Fragment содержащий перечисленные ниже поля.

Next Header со значением, указывающим первый заголовок после заголовков Per-Fragment исходного пакета.

Значение Fragment Offset, указывающее смещение фрагмента (в 8-октетных блоках) от начала Fragmentable Part исходного пакета.

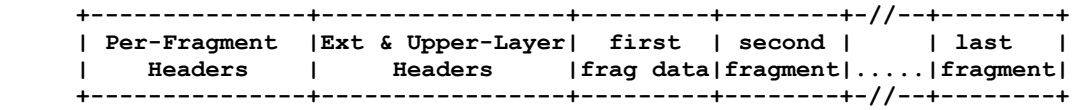
Флаг M со значением 0 для последнего (самого правого) фрагмента и 1 для остальных.

Значение Identification, созданное для исходного пакета.

(3) Сам фрагмент.

При фрагментировании пакета перекрытие фрагментов недопустимо.

На приёмной стороне фрагменты пакета собираются заново в исходный пакет, как показано на рисунке.



Сборка фрагментов выполняется в соответствии с приведёнными ниже правилами.

При сборке исходного пакета используются только фрагментированные пакеты с совпадающими значениями полей Source Address, Destination Address и Identification.

Заголовки Per-Fragment собранного пакета включают все заголовки вплоть (но не включая) до заголовка Fragment пакета с первым фрагментом (т. е. пакета с Fragment Offset = 0), с учётом указанных ниже изменений.

Поле Next Header последнего заголовка Per-Fragment берётся из поля Next Header в заголовке Fragment пакета с первым фрагментом.

Значение поля Payload Length в собранном пакете вычисляется на основе размера заголовков Per-Fragment, а также размера и смещения последнего фрагмента. Ниже приведена формула для расчёта.

```
PL.orig = PL.first - FL.first - 8 + (8 * FO.last) + FL.last
```

где

```
PL.orig  =  поле Payload Length собранного пакета.
PL.first =  поле Payload Length первого фрагмента.
FL.first =  размер фрагмента, следующего после заголовка Fragment в пакете первого фрагмента.
FO.last  =  поле Fragment Offset заголовка Fragment в пакете последнего фрагмента.
FL.last  =  размер фрагмента после заголовка Fragment в пакете последнего фрагмента.
```

Fragmentable Part собираемого пакета складывается из фрагментов, следующих за заголовком Fragment в каждом из фрагментированных пакетов. Размер каждого фрагмента вычисляется путём вычитания из значения поля Payload Length размера всех заголовков, содержащихся между заголовком IPv6 и самим фрагментом, относительное расположение Fragmentable Part определяется на основе значения Fragment Offset.

Заголовок Fragment в собранном пакете не присутствует.

Если фрагмент является полной дейтаграммой (Fragment Offset = 0 и флаг M сброшен), он не требует какой-либо сборки и его следует обрабатывать как полностью собранный пакет (скорректировать значения Next Header и Payload Length, удалить заголовок Fragment и т. д.). Любые другие фрагменты, соответствующие этому пакету (с такими же значениями IPv6 Source Address, IPv6 Destination Address и Fragment Identification), следует обрабатывать независимо.

При сборке фрагментов может возникать несколько ситуаций, приводящих к ошибкам, которые описаны ниже.

- Если в течение 60 секунд с момента приёма первого фрагмента получены не все фрагменты, требуемые для сборки, сборка должна быть прервана, а все полученные фрагменты отброшены. Если первый фрагмент (пакет со значением Fragment Offset = 0) был получен, отправителю этого фрагмента следует передать сообщение ICMP Time Exceeded (истекло время сборки фрагментов).

[1]Это ограничивает размер заголовков (с учётом заголовка Upper-Layer) величиной MTU на пути к адресатам.

- Если размер фрагмента, определённый из значения поля Payload Length в заголовке пакета с фрагментом, не кратен 8 октетам, а флаг M = 1, этот фрагмент должен отбрасываться, а отправителю фрагмента следует передать сообщение ICMP Parameter Problem с кодом 0, указывающее на поле Payload Length пакета с фрагментом.

- Если значения размера и смещения для фрагмента таковы, что значение поля Payload Length собранного из фрагментов пакета будет превышать 65 535 октетов, этот фрагмент должен быть отброшен, а его отправителю следует передать сообщение ICMP Parameter Problem с кодом 0, указывающее на поле Fragment Offset в пакете с фрагментом.

- Если первый фрагмент не включает всех заголовков до заголовка Upper-Layer, фрагмент следует отбросить, а его отправителю передать сообщение ICMP Parameter Problem с кодом 3 и полем Pointer = 0.

- Если любой из участвующих в сборке фрагментов перекрывается с другим фрагментом того же пакета, сборка пакета должна быть прервана, а все полученные фрагменты отброшены. Передавать сообщение ICMP об ошибке не следует. Отметим, что в сети могут возникать дубликаты фрагментов. Реализации могут не считать такие фрагменты перекрывающимися с другими, а просто отбрасывать полные дубликаты, сохраняя остальные фрагменты того же пакета.

Ниже перечислены ситуации, которые представляются достаточно редкими, но не считаются ошибками.

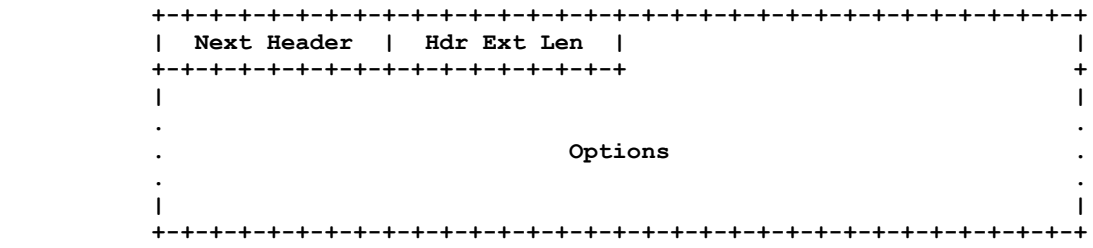
Число и содержимое заголовков, предшествующих заголовку Fragment в разных фрагментах одного исходного пакета могут различаться. Независимо от того, какие заголовки присутствуют в каждом фрагменте перед заголовком Fragment, они обрабатываются до того, как фрагменты будут помещены в очередь на сборку. В собранном пакете остаются лишь заголовки из фрагментированного пакета с Offset = 0.

Значения Next Header в заголовках Fragment различных фрагментов одного исходного пакета могут различаться. Для сборки фрагментов используется только значение из пакета, содержащего фрагмент с Offset = 0.

Другие поля в заголовке IPv6 также могут изменяться в процессе сборки фрагментов. Спецификации, использующие такие поля, могут включать дополнительные инструкции, если базового механизма с использованием значений лишь из фрагмента с Offset = 0 недостаточно. Например, в параграфе 5.3 [RFC3168] описано, как комбинируются биты ECN из разных фрагментов для получения ECN в собранном пакете.

4.6. Заголовок Destination Options

Заголовок Destination Options используется для передачи дополнительной информации, которая будет просматриваться только на конечных узлах. Заголовок Destination Options указывается значением Next Header = 60 в предшествующем непосредственно заголовке и имеет формат, показанный на рисунке.



Next Header - следующий заголовок

8-битовый селектор, определяющий тип заголовка, следующего непосредственно за Destination Options. Используются те же значения, которые применяются в поле Protocol заголовков IPv4 [IANA-PN].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка в 8-октетных словах без учёта первых 8 октетов.

Options - опции

Поле переменной длины, такой, что полный размер заголовка Destination Options кратен 8 октетам. Поле опций содержит одну или множество опций в формате TLV, как описано в параграфе 4.2.

В этом документе определены только две опции получателя (Pad1 и PadN), описанные в параграфе 4.2.

Отметим, что существует два способа представления дополнительной информации для получателя в пакетах IPv6 - в виде опции заголовка Destination Options или в виде отдельного заголовка расширения. Примерами второго варианта могут служить заголовки Fragment и Authentication. Выбор конкретного варианта зависит от того, какие действия желательны на узле-адресате в случае непонимания дополнительной информации.

- Если желательным действием является отбрасывание пакета получателем и передача (если адрес получателя не является групповым) отправителю сообщения ICMP Unrecognized Type, дополнительная информация может быть представлена в отдельном заголовке или в опции заголовка Destination Options со значением 11 в двух старших битах поля Option Type. Выбор конкретного варианта может определяться размером опции, более эффективным выравниванием или разбором.

- Если желательно иное действие, дополнительная информация должна представляться в виде опции заголовка Destination Options, для которого два старших бита поля Option Type имеют значение 00, 01 или 10, задающее требуемое действие (4.2. Опции).

4.7. Нет следующего заголовка

Значение 59 в поле Next Header заголовка IPv6 или любого заголовка расширения говорит об отсутствии последующих заголовков в пакете. Если поле Payload Length в заголовке IPv6 показывает наличие октетов после завершения заголовка, в котором Next Header = 59, эти октеты должны игнорироваться и пересылаться без изменений.

4.8. Определение новых расширений заголовка и опций

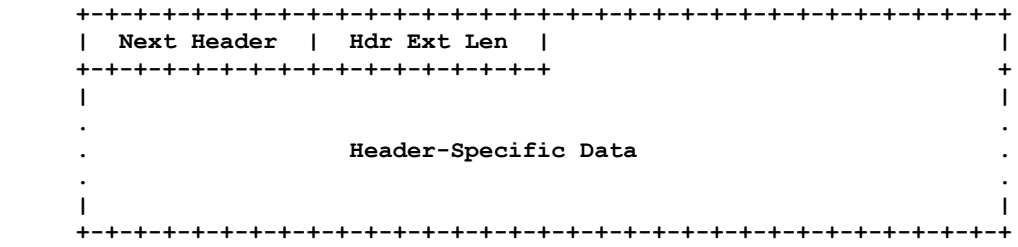
Определение новых заголовков расширения IPv6 не рекомендуется, если можно воспользоваться имеющимся заголовком путём задания для него новой опции. Предложение по добавлению заголовка расширения IPv6 должно включать подробное техническое разъяснение невозможности воспользоваться имеющимися заголовками для реализации желаемой новой функции. Дополнительная информация представлена в [RFC6564].

Примечание. Новые заголовки расширения, требующие поэтапной (hop-by-hop) обработки, добавлять недопустимо, поскольку, как отмечено в разделе 4 данного документа, заголовок Hop-by-Hop Options является единственным заголовком расширения с поэтапной обработкой.

Создавать новые опции с поэтапной (hop-by-hop) обработкой не рекомендуется, поскольку на узлах может быть задано игнорирование заголовка Hop-by-Hop Options, отбрасывание пакетов с таким заголовком или направление пакетов по пути медленной обработки. Разработчикам, предполагающим определить новые опции hop-by-hop, следует принимать это во внимание и чётко обосновать необходимость любой новой опции hop-by-hop до ее стандартизации.

Вместо определения новых заголовков расширения рекомендуется применять заголовок Destination Options для передачи дополнительной информации, которая будет проверяться только конечными получателями, поскольку в этом случае обеспечивается более эффективная обработка и совместимость с прежними версиями.

Если новые заголовки расширения всё-таки определяются, они должны использовать показанный ниже формат.



Next Header - следующий заголовок

8-битовый селектор, определяющий тип заголовка, следующего непосредственно после заголовка расширения. Используются те же значения, что и в поле Protocol заголовков IPv4 [IANA-PN].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка в 8-октетных словах без учёта первых 8 октетов.

Header Specific Data

Поле переменного размера в зависимости от расширения заголовка.

5. Размер пакетов

IPv6 требует, чтобы каждый канал в сети Internet имел значение MTU не менее 1280 октетов. Это называется минимальным MTU канала IPv6. Для всех каналов, которые не поддерживают передачу пакетов размером 1280 октетов, должна обеспечиваться фрагментация и сборка на уровне ниже IPv6.

Каналы с настраиваемым значением MTU (например, PPP [RFC1661]) должны настраиваться на использование значений MTU не менее 1280 октетов. Рекомендуется устанавливать значение MTU не менее 1500 октетов для обеспечения возможности инкапсуляции (например, туннелирования) без фрагментации на уровне IPv6.

Из каждого канала, к которому узел подключён непосредственно, он должен быть способен принимать пакеты размером MTU для этого канала.

Для узлов IPv6 настоятельно рекомендуется поддержка механизма Path MTU Discovery [RFC8201] для обнаружения и использования преимуществ MTU > 1280 октетов. Однако минимальные реализации IPv6 (например, в загрузочных ПЗУ) могут ограничиваться передачей пакетов, размер которых не превышает 1280 октетов, и не поддерживать Path MTU Discovery.

Для передачи пакетов, размер которых превышает path MTU, узел может использовать заголовок IPv6 Fragment для фрагментации пакета на стороне отправителя и сборки фрагментов на приёмной стороне. Однако такой фрагментации следует избегать во всех приложениях, которые могут подстраивать размер пакетов под измеренное значение path MTU (например, до 1280 октетов).

Узел должен быть способен воспринимать фрагментированные пакеты, размер которых после сборки достигает 1500 октетов, и может воспринимать пакеты, размер которых после сборки фрагментов превышает 1500 октетов. Протоколам или приложениям вышележащего уровня, зависящим от фрагментации IPv6, для передачи пакетов с размером больше MTU для пути не следует передавать пакетов размером более 1500 октетов, если нет уверенности в том, что получатель может собирать из фрагментов пакеты размером более 1500 октетов .

6. Метки потоков

20-битовое поле Flow Label в заголовке IPv6 используется отправителем для обозначения последовательности пакетов, которую в сети следует считать единым потоком. Определение IPv6 Flow Label приведено в [RFC6437].

7. Классы трафика

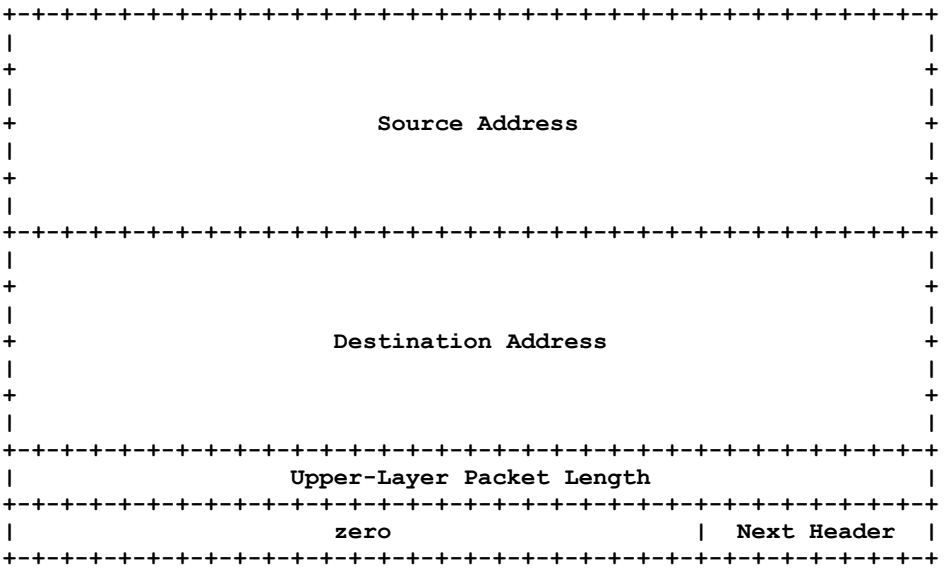
8-битовое поле Traffic Class в заголовке IPv6 используется в сети для управления трафиком. Значения битов Traffic Class в принятом пакете или фрагменте могут отличаться от установленных отправителем.

Использование поля Traffic Class для дифференцированных услуг (Differentiated Services) и явного уведомления о перегрузке (Explicit Congestion Notification или ECN) задано в [RFC2474] и [RFC3168].

8. Протоколы вышележащего уровня

8.1. Контрольные суммы

Любой транспортный протокол или иной протокол вышележащего уровня, который включает адреса из заголовка IP в расчёт контрольной суммы, должен быть изменён для использования с протоколом IPv6, в котором применяются 128-битовые адреса IPv6 взамен 32-битовых адресов IPv4. Приведённый рисунок показывает псевдозаголовок TCP и UDP для IPv6.



- Если пакет IPv6 включает заголовок Routing, поле Destination Address, используемое в псевдозаголовке, указывает конечного получателя. На источнике этот адрес будет последним элементом заголовка Routing, на приёмной стороне этот адрес будет указан в поле Destination Address заголовка IPv6.
- Значение Next Header в псевдозаголовке указывает протокол вышележащего уровня (например, 6 для протокола TCP или 17 для UDP). Оно будет отличаться от значения Next Header в заголовке IPv6, если между этим заголовком и заголовком вышележащего уровня имеются заголовки расширения.
- Поле Upper-Layer Packet Length в псевдозаголовке содержит размер заголовка и данных вышележащего уровня (например, заголовка и данных TCP). Некоторые протоколы вышележащего уровня поддерживают собственную информацию о размере (например, поле Length в заголовке UDP), для таких протоколов это будет размер, указанный в псевдозаголовке. Другие протоколы (такие, как TCP) не поддерживают своих данных о размере и размер, указанный в псевдозаголовке, будет значением поля Payload Length из заголовка IPv6 за вычетом размера всех заголовков расширения между заголовками IPv6 и вышележащего уровня.
- В отличие от IPv4, поле контрольной суммы для пакетов UDP, порождаемых узлом IPv6, не является опциональным, т. е. при генерации пакета UDP узел IPv6 должен рассчитать контрольную сумму UDP для пакета и псевдозаголовка и при получении нулевого значения контрольной суммы заменить его значением FFFF для включения в заголовок UDP. Получатель IPv6 должен отбрасывать пакеты UDP с нулевым значением контрольной суммы и этот факт следует записывать в системный журнал.
- Как исключение для принятого по умолчанию поведения, протоколы, использующие UDP для туннельной инкапсуляции, могут разрешать нулевые значения контрольной суммы (zero-checksum mode) для конкретного порта или набора портов при отправке и/или приёме пакетов. Каждый узел, поддерживающий режим zero-checksum, должен следовать рекомендациям документа «Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums» [RFC6936].

Протокол ICMP для IPv6 [RFC4443] включает описанный выше псевдозаголовок в расчёт контрольной суммы (в отличие от ICMP для IPv4, где псевдозаголовок не учитывается в контрольной сумме). Это сделано для защиты ICMP от ошибочной доставки или повреждения тех полей в заголовках IPv6, от которых зависит ICMP и которые, в отличие от IPv4, не учитывается в контрольной сумме сетевого уровня. Поле Next Header в псевдозаголовке для ICMP содержит значение 58, указывающее версию IPv6 для протокола ICMP.

8.2. Максимальный срок жизни пакета

В отличие от IPv4, узлы IPv6 не обязаны соблюдать максимальное время жизни пакетов. По этой причине поле TTL (Time to Live) протокола IPv4 переименовано в «максимальное число интервалов» (Hop Limit) в IPv6. На практике очень немногие реализации IPv4 соответствуют требованиям по ограничению времени жизни пакетов, поэтому внесённое в протокол изменение не имеет существенного практического значения. Любой протокол вышележащего уровня, который опирается на сетевой уровень (IPv4 или IPv6) для ограничения времени жизни пакетов, должен быть обновлён путём обеспечения собственных механизмов детектирования и отбрасывания устаревших пакетов.

8.3. Максимальный размер данных вышележащего уровня

При расчёте максимального размера данных, доступного протоколу вышележащего уровня, этот протокол должен принимать во внимание больший размер заголовков IPv6 по сравнению с IPv4. Например, в IPv4 опция TCP MSS[1] рассчитывается как максимальный размер пакета (принятый по умолчанию или полученный от Path MTU Discovery) за вычетом 40 октетов (20 октетов минимального заголовка IPv4 и 20 октетов минимального заголовка TCP). При использовании TCP с протоколом IPv6 значение MSS должно рассчитываться как максимальный размер пакета за

[1]Максимальный размер сегмента TCP.

вычетом 60 октетов, поскольку размер минимального заголовка IPv6 (без заголовков расширения) на 20 октетов превышает минимальный размер заголовка IPv4.

8.4. Отклики на пакеты с заголовками Routing

Когда протокол вышележащего уровня шлёт один или множество пакетов в ответ на принятый пакет с заголовком Routing, в пакеты откликов недопустимо включать заголовок Routing, который будет создаваться автоматически путём обращения полученного заголовка Routing, **пока** не будет проверена целостность и подлинность поля Source Address и заголовка Routing (например с помощью заголовка Authentication из полученного пакета). Иными словами, в ответ на полученные пакеты с заголовком Routing можно передавать только следующие типы пакетов:

- пакеты отклика без заголовков Routing;
- пакеты отклика с заголовками Routing, которые **не** были получены обращением заголовка Routing из принятого пакета (например, с заголовком Routing, определяемым локальной конфигурацией);
- пакеты отклика с заголовками Routing, полученными путём обращения заголовка Routing из принятого пакета, **тогда и только тогда**, когда поле Source Address и заголовок Routing в полученном пакете были проверены отвечающей стороной.

9. Взаимодействие с IANA

В RFC 2460 указано множество реестров IANA, включая:

- Internet Protocol Version 6 (IPv6) Parameters [IANA-6P];
- Assigned Internet Protocol Numbers [IANA-PN];
- ONC RPC Network Identifiers (netids) [IANA-NI];
- Network Layer Protocol Identifiers (NLPIDs) of Interest [IANA-NL];
- Protocol Registries [IANA-PR].

Агентство IANA обновило эти реестры ссылками на данный документ.

10. Вопросы безопасности

IPv6 с точки зрения базового формата и передачи пакетов имеет параметры безопасности похожие на IPv4. Основные проблемы безопасности перечислены ниже.

- *Перехват* - элементы сети на пути передачи могут видеть целиком (содержимое и метаданные) каждую дейтаграмму IPv6.
- *Повторное использование* (Replay) - атакующий может сохранить перехваченные пакеты и повторно передать их тому же адресату.
- *Вставка пакетов* - атакующий может подделать пакеты, придав им нужные свойства, и отправить их в сеть.
- *Удаление пакетов* - атакующий может удалить пакеты из канала передачи.
- *Изменение пакетов* - атакующий может извлечь пакет из канала передачи, изменить его и снова передать в сеть.
- *MITM-атаки*[1] - атакующий может разорвать путь передачи, представившись отправителю получателем, а получателю отправителем.
- *DoS-атаки*[2] - атакующий передаёт большие объёмы легитимного трафика адресату с целью перегрузки последнего.

Пакеты IPv6 можно защитить от просмотра, повторного использования, вставки, изменения и MITM-атак с помощью «Security Architecture for the Internet Protocol» [RFC4301]. В дополнение к этому можно применять протоколы типа TLS[3] или SSH[4] для защиты трафика приложений, передаваемого по протоколу IPv6.

Не задано механизма защиты от DoS-атак. Методы защиты от атак на службы выходят за рамки этого документа.

Адреса IPv6 существенно длиннее адресов IPv4 и это осложняет сканирование адресного пространства Internet или даже одной сети (например, ЛВС). Дополнительная информация приведена в [RFC7707].

Предполагается, что адреса узлов IPv6 станут «более видимыми» в сети Internet по сравнению с адресами IPv4, поскольку технологии трансляции адресов будут применяться реже. Это создаёт некоторые добавочные проблемы приватности за счёт упрощения идентификации конечных точек. Дополнительная информация приведена в [RFC7721].

Архитектура заголовков расширения IPv6 значительно расширяет возможности, но и вызывает новые проблемы безопасности. Как отмечено ниже, вопросы, связанные с заголовками расширения Fragment, были решены, однако очевидно, что любое новое расширение потребует тщательной проверки в плане безопасности и взаимодействия с имеющимися заголовками расширения. Дополнительная информация дана в [RFC7045].

Данная версия спецификации IPv6 снимает множество проблем безопасности, присущих прежней версии спецификации IPv6 [RFC2460]. Эти проблемы перечислены ниже.

[1]Man-in-the-middle - перехват и изменение пакетов с участием человека.
[2]Denial-of-service - атака, нацеленная на отказ в обслуживании.
[3]Transport Layer Security - защита на транспортном уровне.
[4]Secure Shell.

- Пересмотрен текст по обработке фрагментов, являющихся полными дейтаграммами (Fragment Offset = 0 и M = 0). При получении такого пакета его следует трактовать, как собранный из фрагментов. Все остальные соответствующие фрагменты следует обрабатывать независимо. Процесс фрагментации был изменён для предотвращения фрагментов, являющихся полными дейтаграммами (Fragment Offset = 0 и M = 0). Дополнительная информация представлена в [RFC6946] и [RFC8021].
- Из раздела 5 исключён параграф, требовавший включения заголовка Fragment в исходящие пакеты при получении сообщения ICMP Packet Too Big с информацией о том, что Next-Hop MTU < 1280. Дополнительная информация приведена в [RFC6946].
- Изменён текст, относящийся к перекрытию фрагментов IPv6, и узлам теперь недопустимо создавать такие фрагменты. Введено требование отбрасывания (без уведомления) дейтаграммы целиком, если в процессе сборки обнаружилось наложение фрагментов. Добавлено разъяснение, что передавать сообщение ICMP в таких случаях не следует. Дополнительная информация приведена в [RFC5722].
- Пересмотрен текст в части фрагментации заголовков и сейчас все заголовки вплоть до первого заголовка вышележащего уровня должны помещаться в первый фрагмент. Дополнительная информация приведена в [RFC7112].
- Учтены обновления из [RFC5095] и [RFC5871] в части удаления текста с описанием заголовков Routing типа 0 (RH0), с указанием того, что рекомендации по этим заголовка приведены в RFC 5871, а тип RH0 удалён из списка требуемых заголовков расширений.

Вопросы безопасности, относящиеся к отдельным частям IPv6, включая адресацию, ICMPv6, Path MTU Discovery и т. п., рассмотрены в соответствующих спецификациях.

11. Литература

11.1. Нормативные документы

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[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, <http://www.rfc-editor.org/info/rfc2474>.

[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, <http://www.rfc-editor.org/info/rfc3168>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[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, <http://www.rfc-editor.org/info/rfc4443>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <http://www.rfc-editor.org/info/rfc6437>.

11.2. Дополнительная литература

[Err2541] RFC Errata, Erratum ID 2541, RFC 2460.

[Err4279] RFC Errata, Erratum ID 4279, RFC 2460.

[Err4657] RFC Errata, Erratum ID 4657, RFC 2460.

[Err4662] RFC Errata, Erratum ID 4662, RFC 2460.

[IANA-6P] IANA, "Internet Protocol Version 6 (IPv6) Parameters", <https://www.iana.org/assignments/ipv6-parameters>.

[IANA-EH] IANA, "IPv6 Extension Header Types", <https://www.iana.org/assignments/ipv6-parameters>.

[IANA-NI] IANA, "ONC RPC Network Identifiers (netids)", <https://www.iana.org/assignments/rpc-netids>.

[IANA-NL] IANA, "Network Layer Protocol Identifiers (NLPIDs) of Interest", <https://www.iana.org/assignments/nlpids>.

[IANA-PN] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers>.

[IANA-PR] IANA, "Protocol Registries", <https://www.iana.org/protocols>.

[IANA-RH] IANA, "Routing Types", <https://www.iana.org/assignments/ipv6-parameters>.

[RFC1661] Simpson, W., Ed., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, DOI 10.17487/RFC1661, July 1994, <http://www.rfc-editor.org/info/rfc1661>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

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

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <http://www.rfc-editor.org/info/rfc4302>.

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

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <http://www.rfc-editor.org/info/rfc5095>.

[RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, DOI 10.17487/RFC5722, December 2009, <http://www.rfc-editor.org/info/rfc5722>.

[RFC5871] Arkko, J. and S. Bradner, "IANA Allocation Guidelines for the IPv6 Routing Header", RFC 5871, DOI 10.17487/RFC5871, May 2010, <http://www.rfc-editor.org/info/rfc5871>.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <http://www.rfc-editor.org/info/rfc6564>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <http://www.rfc-editor.org/info/rfc6936>.

[RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, DOI 10.17487/RFC6946, May 2013, <http://www.rfc-editor.org/info/rfc6946>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <http://www.rfc-editor.org/info/rfc7045>.

[RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <http://www.rfc-editor.org/info/rfc7112>.

[RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <http://www.rfc-editor.org/info/rfc7707>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <http://www.rfc-editor.org/info/rfc7721>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <http://www.rfc-editor.org/info/rfc7739>.

[RFC8021] Gont, F., Liu, W., and T. Anderson, "Generation of IPv6 Atomic Fragments Considered Harmful", RFC 8021, DOI 10.17487/RFC8021, January 2017, <http://www.rfc-editor.org/info/rfc8021>.

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

Приложение A. Рекомендации по формату опций

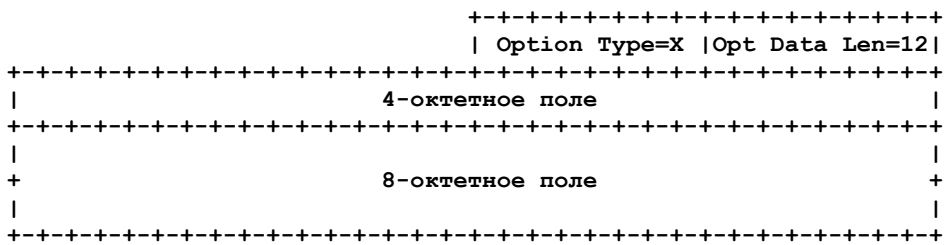
В этом приложении приведены некоторые рекомендации по расположению полей в новых опциях для использования с заголовками расширения Hop-by-Hop Options и Destination Options, как описано в параграфе 4.2. Рекомендации базируются на следующих допущениях:

- желательно выравнивать многооктетные поля в Option Data по их естественным границам - т. е. поля размером n октетов следует размещать с кратным n смещением от начала заголовка Hop-by-Hop Options или Destination Options (для n = 1, 2, 4, 8);
- желательно минимизировать размер заголовков Hop-by-Hop Options и Destination Options с учётом того, что размер заголовка должен быть кратным 8 октетам;
- можно предположить, что в заголовках с опциями число опций невелико (обычно одна опция).

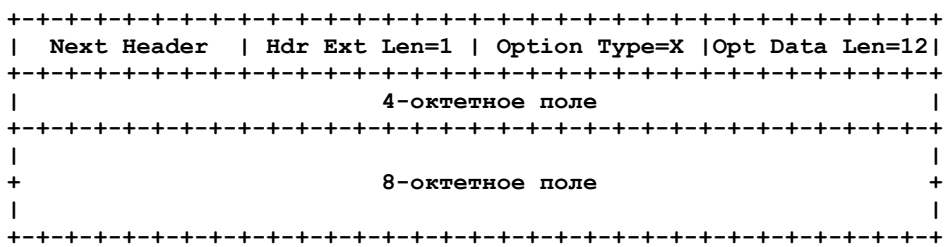
На основе этих допущений предлагается следующая схема размещения полей опции - поля упорядочиваются по возрастанию размера без внутреннего заполнения, а после этого выполняется выравнивание для опции в целом за счёт выравнивания самого большого поля (максимальное заполнение для выравнивания может составить 8 октетов). Этот подход проиллюстрирован примерами.

Пример 1

Если для опции X требуется два поля данных, размером 8 и 4 октета, их следует расположить, как показано на рисунке.

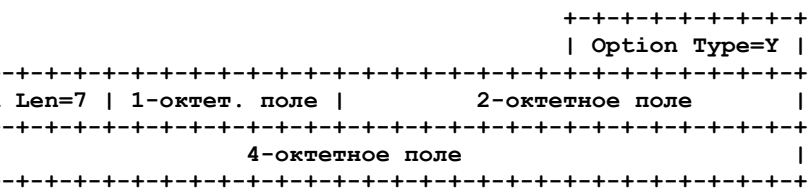


Выравнивание должно быть выполнено по формуле 8n+2, чтобы 8-октетное поле начиналось с кратным 8 смещением от начала содержащего опцию заголовка. Схема полного заголовка Hop-by-Hop Options или Destination Options, содержащего эту опцию, показана на рисунке.

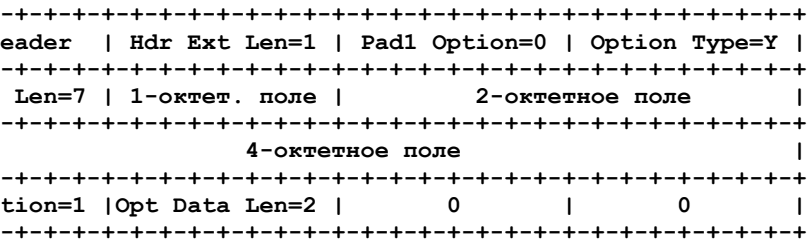


Пример 2

Опция Y включает три поля размером 4, 2 и 1 октет, они будут располагаться, как показано на рисунке.

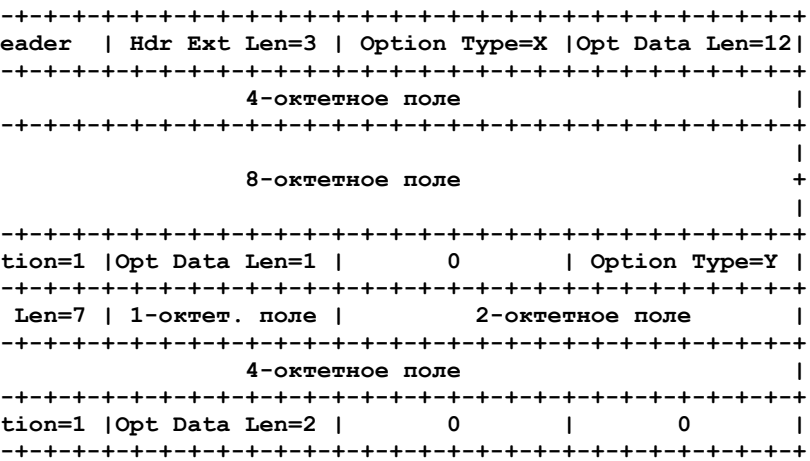


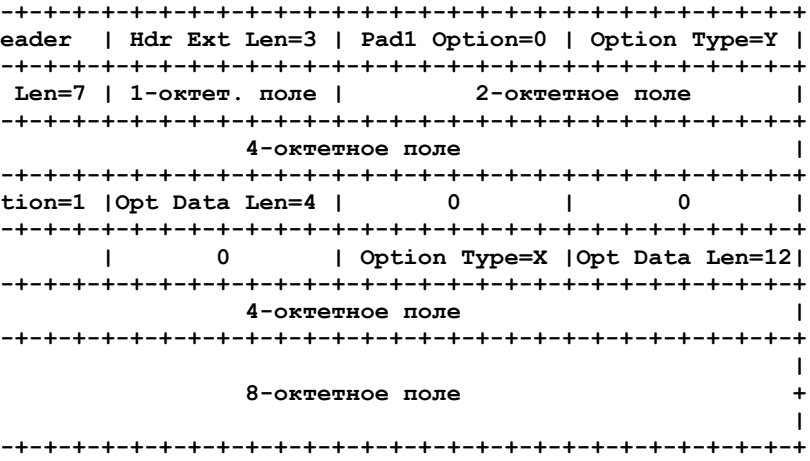
Выравнивание осуществляется по формуле 4n+3, чтобы 4-октетное поле имело кратное 4 смещение от начала содержащего опцию заголовка. Схема полного заголовка Hop-by-Hop Options или Destination Options, содержащего эту опцию, показана на рисунке.



Пример 3

Заголовок Hop-by-Hop Options или Destination Options с опциями X и Y из примеров 1 и 2 будет иметь один из приведённых ниже форматов в зависимости от порядка расположения опций





Приложение B. Отличия от RFC 2460

Ниже перечислены отличия данного документа от RFC 2460.

- Удалено упоминание IP Next Generation в аннотации (Abstract).
- В раздел 1 добавлен текст, указывающий, что порядок передачи данных совпадает с принятым для протокола IPv4 в RFC 791.
- Уточнён текст раздела 3 в части декрементирования Hop Limit.
- Добавлено разъяснение, что заголовки расширения (кроме Hop-by-Hop Options) не обрабатываются, не добавляются и не удаляются узлами на пути доставки пакетов.
- Требование по обработке заголовка Hop-by-Hop Options заменено на «может» и добавлено примечание, указывающее ожидания в части заголовка Hop-by-Hop Options.

- В раздел 4 добавлен параграф, разъясняющий, как нумеруются заголовки расширения и какой заголовок относится к вышележащему уровню.
- В конце раздела 4 добавлена ссылка на реестр IANA IPv6 Extension Header Types.
- Включены обновления из RFC 5095 и RFC 5871 с целью удалить описание RH0 и указать, что рекомендации по для Routing указаны в RFC 5871, а тип RH0 исключён из списка требуемых заголовков расширения.
- Пересмотрен параграф 4.5 в части фрагментации IPv6 с учётом обновлений из RFC 5722, RFC 6946, RFC 7112 и RFC 8021.
  - Пересмотрен текст по обработке фрагментов, являющихся полными дейтаграммами (Fragment Offset = 0 и M = 0). При получении такого пакета его следует трактовать как собранный из фрагментов. Все остальные соответствующие фрагменты следует обрабатывать независимо. Процесс фрагментации был изменён для предотвращения фрагментов, являющихся полными дейтаграммами (Fragment Offset = 0 и M = 0).
  - Изменён текст, относящийся к перекрытию фрагментов IPv6, и узлам теперь недопустимо создавать такие фрагменты. Введено требование отбрасывания (без уведомления) дейтаграммы целиком, если в процессе сборки обнаружилось наложение фрагментов. Добавлено разъяснение, что передавать сообщение ICMP в таких случаях не следует.
  - Пересмотрен текст в части фрагментации заголовков и сейчас все заголовки вплоть до первого заголовка вышележащего уровня должны помещаться в первый фрагмент. Изменённый текст описывает процессы фрагментации и сборки, а также включает дополнительный случай возможной ошибки.
  - Обновлён текст в части обработки заголовков Fragment для случая точного совпадения фрагментов.
  - Обновлён текст описания заголовка Fragment в части включения AH[1] и случая No Next Header.
  - Вместо термина Unfragmentable Headers используется термин Per-Fragment headers в тексте, посвящённом заголовку Fragment.
  - Из раздела 5 удалён абзац, требовавший включения заголовка Fragment в исходящие пакеты ICMP Packet Too Big с информацией о том, что Next-Hop MTU < 1280.
  - Изменён текст, проясняющий ограничение MTU и 8-байтовые ограничения, а также отмечающий ограничения для заголовков в первом фрагменте.
- В параграф 4.5 добавлен текст, поясняющий, что некоторые поля в заголовке IPv6 могут изменяться в процессе сборки фрагментов, а также возможность задания дополнительных инструкций по сборке в других спецификациях (см., например, параграф 5.3 в [RFC3168]).
- Включено обновление из RFC 6564 в форме нового параграфа 4.8 с рекомендациями по определению новых заголовков расширения и опций.
- В раздел 5 добавлен текст, определяющий минимальное значение IPv6 MTU для канала.
- Упрощён текст раздела 6, относящийся к меткам потоков (Flow Label) и удалено Приложение A (Семантика и применение поля Flow Label). Вместо прежнего текста добавлена ссылка на действующую спецификацию поля IPv6 Flow Label в [RFC6437] и поля Traffic Class в [RFC2474] и [RFC3168].
- В раздел 8 включено обновление из RFC 6935 (IPv6 and UDP Checksums for Tunneled Packets). Добавлено исключение из принятого по умолчанию поведения при обработке пакетов UDP с нулевой контрольной суммой для туннелей.
- В разделе 9. Взаимодействие с IANA ссылки на RFC 2460 заменены ссылками на данный документ.
- Пересмотрен и расширен раздел 10. Вопросы безопасности.
- В раздел «Благодарности» добавлен текст с благодарностями авторам обновлённых документов.
- Обновлены ссылки с указанием действующих документов и проведено разделение ссылок на нормативные и информационные.
- Внесены изменения, обусловленные информацией об ошибках RFC 2460.

  Erratum ID 2541 [Err2541] - отмечено, что RFC 2460 не обновляет RFC 2205, где размер метки потока был изменён на 24 бита с 20, заданных RFC 1883. Эта проблема была решена в RFC 6437, где определены метки потоков. Данная спецификация упоминает RFC 6437. Других изменений не требуется.

  Erratum ID 4279 [Err4279] - отмечено, что в спецификации не описано поведение пересылающего узла при получении пакетов с Hop Limit = 0. В разделе 3 данной спецификации проблема устранена.

  Erratum ID 4657 [Err4657] - предложено отказаться от добавления заголовков расширения на всех узлах, кроме источника пакета. Вопрос решён в разделе 4. Заголовки расширений IPv6.

  Erratum ID 4662 [Err4662] - предложено отказаться от проверки, обработки, изменения, добавления и удаления заголовков расширения (с единственным исключением) на промежуточных узлах в пути доставки. Вопрос решён в разделе 4. Заголовки расширений IPv6.

  Erratum ID 2843 - это сообщение отвергнуто (Rejected) и никаких изменений не внесено.

[1]Authentication Header - заголовок аутентификации.

Благодарности

Авторы выражают свою признательность за многочисленные предложения членам рабочей группы Ipng, исследовательской группы End-to-End Protocols и всему сообществу Internet.

Авторы также признательным авторам обновлённых RFC, которые были включены в этот документ для получения спецификацией IPv6 статуса Internet Standard. Это Joe Abley, Shane Amante, Jari Arkko, Manav Bhatia, Ronald P. Bonica, Scott Bradner, Brian Carpenter, P.F. Chimento, Marshall Eubanks, Fernando Gont, James Hoagland, Sheng Jiang, Erik Kline, Suresh Krishnan, Vishwas Manral, George Neville-Neil, Jarno Rajahalme, Pekka Savola, Magnus Westerlund и James Woodyatt.

Адреса авторов

Stephen E. Deering

Retired

Vancouver, British Columbia

Canada

Robert M. Hinden

Check Point Software

959 Skyway Road

San Carlos, CA 94070

United States of America

Email: bob.hinden@gmail.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru