Network Working Group
Request for Comments: 2460
Obsoletes: 1883
Category: Standards Track

S. Deering
Cisco
R. Hinden
Nokia

December 1998

Спецификация протокола IPv6

Internet Protocol, Version 6 (IPv6) Specification

Статус документа

В этом документе приведена спецификация проекта стандартного протокола Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущий статус стандартизации протокола можно узнать из документа «Internet Official Protocol Standards» (STD 1). Документ может распространяться свободно.

Авторские права

Тезисы

В этом документе приведена спецификация протокола Internet (IP[1]) версии 6 (Ipv6), который иногда называют также IP Next Generation[2] или IPng.

Оглавление

1. Введение

IPv6 представляет собой новую версию протокола IP, предлагаемую в качестве замены протокола IP версии 4 (IPv4) [RFC-791]. Основные отличия между IPv4 и IPv6 можно отнести к нескольким категориям:

Расширенные возможности адресации

IPv6 увеличивает размер адресов IP с 32 до 128 битов для поддержки большего числа уровней иерархии адресов, значительного увеличения числа адресуемых узлов и упрощения автоматической настройки адресов. Масштабируемость групповой маршрутизации (multicast routing) улучшается за счет добавления поля scope[3] в групповые адреса. Определен новый тип адресов - anycast, используемых для адресации пакетов, передаваемых одному (любому) узлу из группы.

Упрощение формата заголовка

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.

[1]Internet Protocol.
[2]Протокол IP следующего поколения.
[3]Область действия.

Поддержка меток потоков

Добавлена возможность помечать пакеты, относящиеся к определенному «потоку» трафика, для которого отправитель запросил специальную обработку (например, отличное от принятого по умолчанию качество обслуживания или обслуживание в реальном масштабе времени).

Аутентификация и приватность

В IPv6 добавлены опции для поддержки аутентификации, контроля целостности и (опционально) конфиденциальности данных.

В этом документе описан базовый заголовок Ipv6, а также изначально определенные для протокола IPv6 расширения и опции. Рассмотрены также вопросы, связанные с размером пакетов, семантика меток потоков и классов трафика, а также влияние IPv6 на протоколы вышележащих уровней. Формат и семантика адресов IPv6 определены в отдельном документе [ADDRARCH]. Протокол ICMP, поддержка которого требуется во всех реализациях Ipv6, описан в [ICMPv6].

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 такого уровня являются транспортные протоколы типа TCP и UDP, протоколы управления типа ICMP, протоколы маршрутизации типа OSPF, а также протоколы, «туннелируемые» через IPv6 (т. е., инкапсулированные в пакеты IPv6) типа IPX, AppleTalk или самого 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, туннели IPv4 или IPv6).

neighbors - соседи

Узлы, подключенные 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).

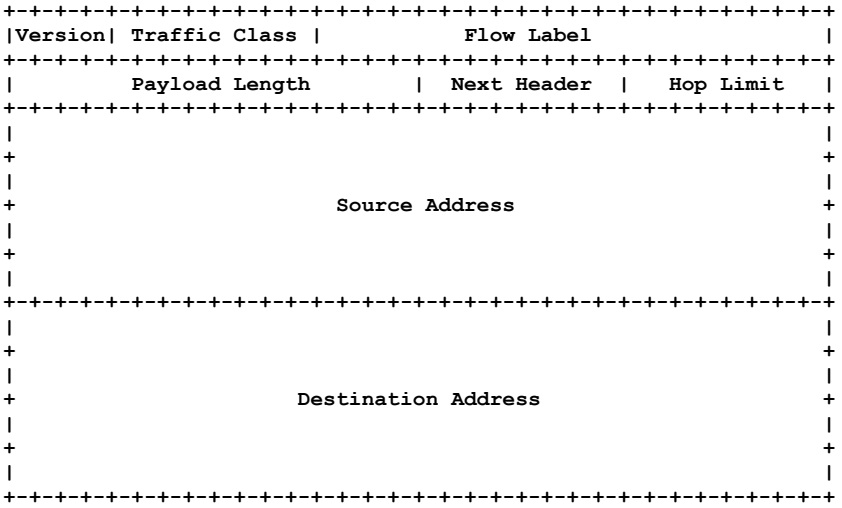
link MTU - максимальный передаваемый блок для канала

Максимальный передаваемый блок информации (максимальный размер пакета в октетах), который может быть передан через канал.

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 (см. раздел 7).

Flow Label - метка потока

20-битовая метка потока (см. раздел 6).

[1]Возможно (хотя и не вполне обычно), что устройство со множеством интерфейсов будет настроено на пересылку не адресованных ему пакетов, приходящих с некоторых (не всех) его интерфейсов, и отбрасывание подобных пакетов, приходящих с остальных его 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.

Payload Length - размер данных

16-битовое целое число без знака, показывающее размер поля данных IPv6 (часть пакета, следующая после заголовка) в октетах. Отметим, что все заголовки расширения (раздел 4) учитываются, как данные (т. е., размер таких заголовков включается в значение размера 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 [RFC-1700 и его преемники].

Hop Limit - предельное число пересылок

8-битовое целое число без знака. Значение поля уменьшается на 1 каждым узлом, пересылающим пакет. При достижении полем Hop Limit нулевого значения пакет отбрасывается.

Source Address - адрес отправителя

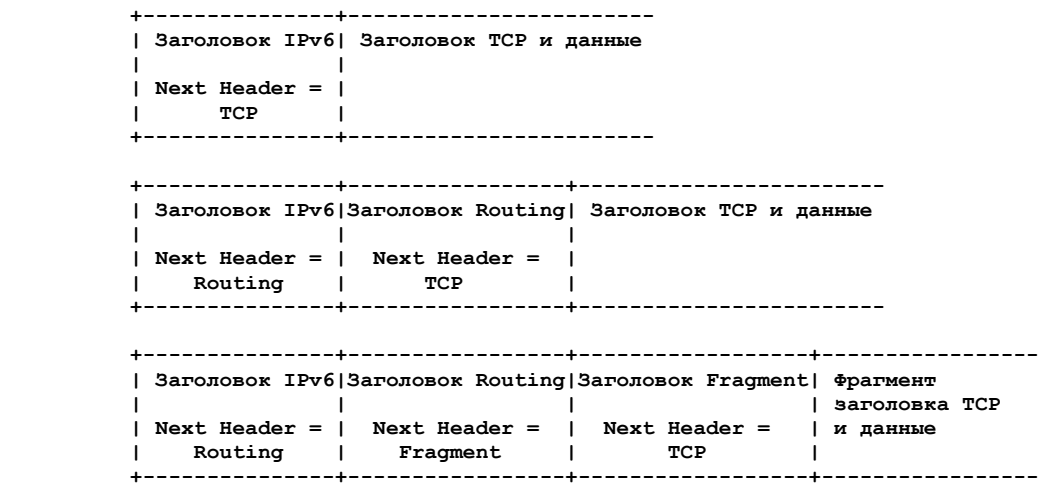
128-битовый адрес инициатора пакета (см. [ADDRARCH]).

Destination Address - адрес получателя

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

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



Заголовки расширения (за единственным исключением) не проверяются и не обрабатываются узлами на пути пересылки, пока пакет не достигнет узла (или множесства узлов при групповой адресации), указанного полем Destination Address в заголовке IPv6. На узле получателя при обычном демультиплексировании поля Next Header в заголовке IPv6 вызывается модуль для обработки первого заголовка расширения или заголовка вышележащего уровня, если заголовка расширения нет. Содержимое и семантика каждого заголовка расширения определяет потребность в обработке следующего заголовка. Следовательно, заголовки расширения должны обрабатываться строго в порядке их следования в пакете. Для получателя недопустимо сканирование пакета с целью поиска того или иного типа заголовка расширение для его обработки ранее, чем будут обработаны предшествующие заголовки расширения.

Упомянутое выше исключение относится к заголовку Hop-by-Hop Options, содержащему информацию, которая должна проверяться и обрабатываться каждым узлом на пути пересылки пакета, включая узлы источника и адресата. При наличии заголовка Hop-by-Hop Options он должен размещаться непосредственно после заголовка IPv6. Присутствие этого заголовка указывается нулевым значением поля Next Header в заголовке IPv6.

Если при обработке заголовка узел определил потребность в обработке следующего заголовка, на значение поля Next Header в текущем заголовке не распознано узлом, ему следует отбросить пакет и передать отправителю пакета сообщение ICMP Parameter Problem с ICMP Code = 1 (unrecognized Next Header type encountered[1]) и полем ICMP Pointer, указывающим смещение нераспознанного значения в исходном пакете. Такие же действия узлу следует выполнять при обнаружении Next Header = 0 в любом заголовке расширения.

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 (т. е., 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;
Routing (Type 0);
Fragment;
Destination Options;
Authentication;
Encapsulating Security Payload.

Первые 4 типа заголовков описаны в этом документе, а два оставшихся в [RFC-2402] и [RFC-2406], соответственно.

[1]Нераспознан тип следующего заголовка.

4.1 Порядок заголовков расширения

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

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

Каждый заголовок расширения следует включать не более одного раза, за исключением заголовка Destination Options, который может включаться дважды (один перед заголовком Routing, а другой 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 приведенные выше рекомендации по порядку следования.

При определении других заголовков расширения должен задаваться порядок их размещения относительно приведенного 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 изменен последующими спецификациями.

4.2 Опции

Два из определенных в настоящее время заголовков расширения (Hop-by-Hop Options и Destination Options) могут содержать переменное число опций, представленных в формате TLV[4], как показано ниже.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
|  Option Type  |  Opt Data Len |  Option Data
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
```

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 в заголовке должна обрабатываться строго в порядке из размещения в заголовке. Получателю недопустимо сканировать заголовки в поиске той или иной опции с целью ее 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 не известно:

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

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 расчете и проверке контрольной суммы пакета (authenticating value).

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 идентифицируется всеми 8 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 только одним из этих двух заголовков.

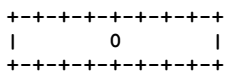
Для отдельных опций могут использоваться специфические требования по выравниванию, чтобы многооктетные поля в Option Data выравнивались по естественным границам. Требования по выравниванию для опций задаются с использованием нотации xn+y, показывающей, что поле Option Type должно представлять собой целое 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, а 8n+2 указывает 8-октетное значение со смещением 2 октета.

---

[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 порядку заголовков Authentication и Encapsulating Security Payload приведены в [RFC-2406].
[3] Для опций, обрабатываемых только адресатом пакета.
[4] Type-length-value - тип-размер-значение.

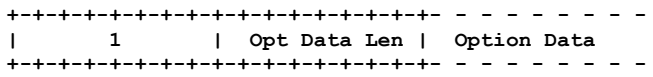
Для выравнивания заголовков по границе 8 октетов используется одна из двух опций заполнения, поддержка которых требуется от всех реализаций IPv6.

Pad1[1] (требований по выравниванию нет)



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

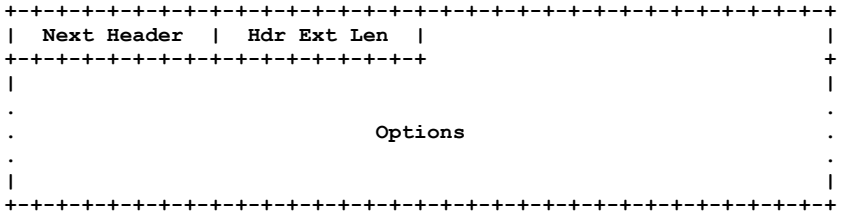
PadN (требований по выравниванию нет)



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 с нулевым значением.

Рекомендации по формату новых опций приведены в Приложении B.

4.3 Заголовок Hop-by-Hop Options



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

Next Header - следующий заголовок

8-битовый селектор, определяющий тип заголовка, следующего непосредственно за Hop-by-Hop Options. Используются те же значения, которые применяются в поле Protocol заголовков IPv4 [RFC-1700 и его преемники].

Hdr Ext Len - размер заголовка расширения

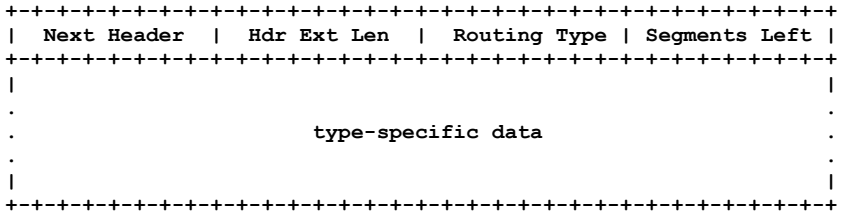
8-битовое целое число без знака, указывающее размер заголовка Hop-by-Hop Options в 8-октетных словах без учета первых 8 октетов.

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, которые пакет должен «посетить» на пути к адресату. Функционально этот заголовок очень похож на опции Loose Source и Record Route в заголовках IPv4. Заголовок Routing идентифицируется значением Next Header = 43 в предшествующем ему заголовке и использует 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 [RFC-1700 и его преемники].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка Routing в 8-октетных словах без учета 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 (т. е., число явно указанных промежуточных узлов), которые должны быть посещены 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.

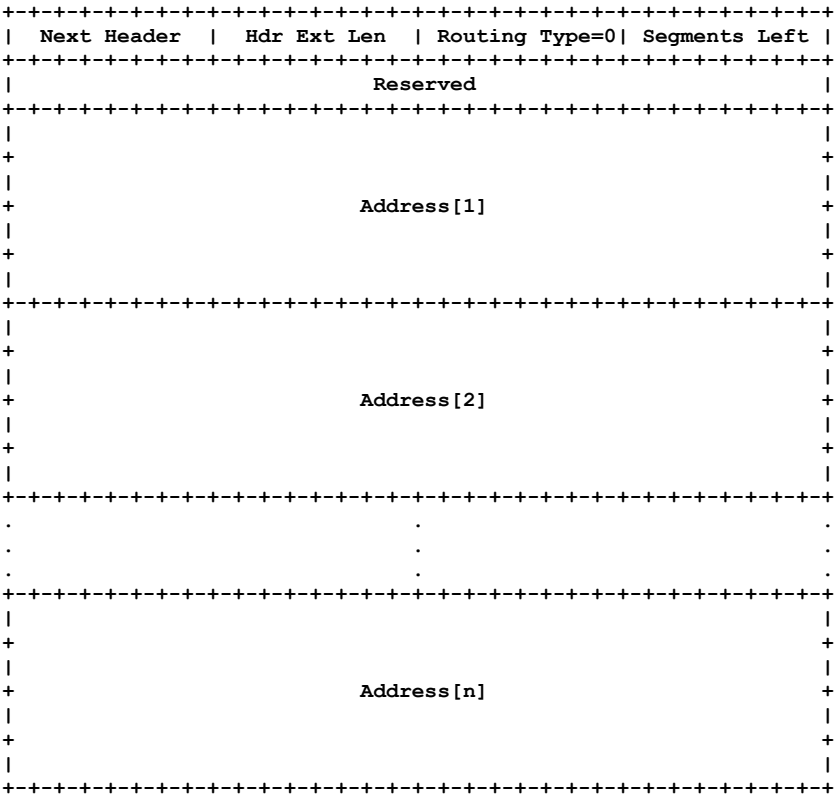
Если при обработке полученного пакета узел встречает заголовок Routing с неизвестным значением поля Routing Type, поведение узла определяется значением поля Segments Left, как описано ниже:

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

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 (указывает на нераспознанное значение Routing Type) по адресу Source Address.

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



Заголовок Routing типа 0[1] имеет формат, показанный на рисунке.

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 [RFC-1700 и последующие версии].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка Routing в 8-октетных словах без учета первых 8 октетов. Для заголовка Routing типа 0 значение поля Hdr Ext Len равно удвоенному числу адресов в этом заголовке.

Routing Type 0

Segments Left - число оставшихся сегментов

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

Reserved - резерв

32-битовое поле, зарезервированное на будущее. При передаче это поле заполняется нулями, а на приемной стороне игнорируется.

Address[1..n] - адреса

Вектор из 128-битовых адресов с номерами от 1 до n.

Групповые адреса не должны появляться в заголовках Routing типа Type 0 или в поле IPv6 Destination Address пакетов, содержащих заголовок Routing типа 0.

Заголовок Routing не проверяется и не обрабатывается, пока пакет не достигнет узла, указанного в поле Destination Address заголовка IPv6. На этом узле диспетчеризация поля Next Header предыдущего заголовка приводит к вызову модуля обработки заголовка Routing, который в случае Routing Type = 0, реализует приведенный ниже алгоритм.

```
if Segments Left = 0 {
   обработать следующий заголовок, тип которого указан полем Next Header в
   заголовке Routing
}
else if Hdr Ext Len имеет нечетное знаение {
      передать по адресу Source Address сообщение ICMP Parameter Problem с кодом 0,
      указывающее на поле Hdr Ext Len и отбросить пакет
}
else {
   рассчитать значение n (число адресов в заголовке Routing) путем деления
   значения поля Hdr Ext Len на 2;
   if Segments Left > n {
      передать по адресу Source Address сообщение ICMP Parameter Problem с кодом 0,
      указывающее на поле Segments Left и отбросить пакет
   }
```

[1]Использование заголовков Routing типа 0 запрещено RFC 5095. Причиной запрета является уязвимость, порождаемая этим типом заголовков. *Прим. перев.*

```
     else {
        уменьшить значение Segments Left на 1;
        рассчитать значение i (индекс следующего адреса из вектора адресов) путем
        вычитания значения Segments Left из n;
        if Address [i] или the IPv6 Destination Address является групповым {
           отбросить пакет
        }
        else {
           поменять местами IPv6 Destination Address и Address[i];
           if IPv6 Hop Limit ≤ 1 {
              передать сообщение ICMP Time Exceeded (Hop Limit Exceeded in  Transit)
              по адресу Source Address и отбросить пакет
           }
           else {
              уменьшить значение Hop Limit на 1;
              снова передать пакет модулю IPv6 для его отправки новому получателю
           }
        }
     }
  }
```

В качестве примера действия приведенного выше алгоритма рассмотрим случай, когда узел S передает пакеты получателю D, используя заголовок Routing для маршрутизации пакетов через промежуточные узлы I1, I2 и I3. Значения соответствующих полей заголовков IPv6 и Routing на каждом сегменте пути доставки показаны ниже.

Сегмент пути от S до I1:

```
Source Address = S                  Hdr Ext Len = 6
Destination Address = I1            Segments Left = 3
                                    Address[1] = I2
                                    Address[2] = I3
                                    Address[3] = D
```

Сегмент пути от I1 до I2:

```
Source Address = S                  Hdr Ext Len = 6
Destination Address = I2            Segments Left = 2
                                    Address[1] = I1
                                    Address[2] = I3
                                    Address[3] = D
```

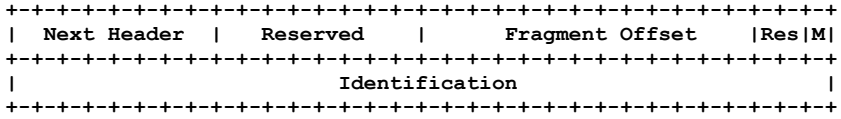
Сегмент пути от I2 до I3:

```
Source Address = S                  Hdr Ext Len = 6
Destination Address = I3            Segments Left = 1
                                    Address[1] = I1
                                    Address[2] = I2
                                    Address[3] = D
```

Сегмент пути от I3 до D:

```
Source Address = S                  Hdr Ext Len = 6
Destination Address = D             Segments Left = 0
                                    Address[1] = I1
                                    Address[2] = I2
                                    Address[3] = I3
```

4.5 Заголовок Fragment



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

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 [RFC-1700 и его преемники].

Reserved - резерв

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

Fragment Offset - смещение фрагмента

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

Res

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

Флаг M

1 - есть еще фрагменты; 0 - последний фрагмент.

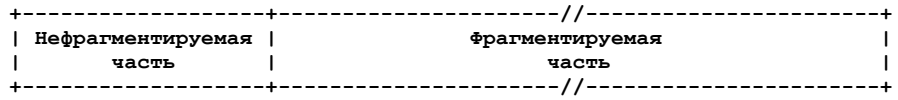
Identification - идентификация

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 фрагментов на приемной стороне.

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

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.



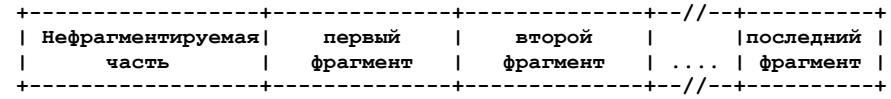
Изначальный (большой) нефрагментированный пакет будем называть исходным пакетом и рассматривать, как две части (см. рисунок).

Нефрагментируемая часть (Unfragmentable Part) состоит из заголовка IPv6 и всех заголовков расширения, которые должны обрабатываться узлами на пути к получателю (т. е., все заголовки расширения до Routing, включительно, если он имеется, или Hop-by-Hop Options, включительно, если он имеется).

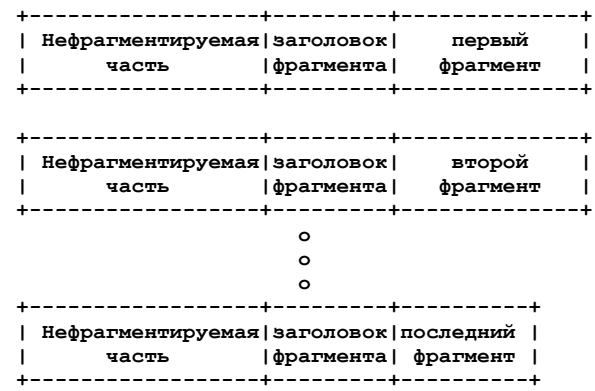
Фрагментируемая часть (Fragmentable Part) содержит остальную часть пакета, т. е., все заголовки расширения, которые требуется обрабатывать только на узле-получателе, заголовок и данные вышележащего уровня.

Фрагментируемая часть исходного пакета делится на фрагменты, размер каждого из которых (за исключением последнего) кратен 8 октетам. Фрагменты передаются в отдельных «фрагментированных пакетах», как показано на рисунке.

Исходный пакет



Фрагментированные пакеты



Каждый пакет с фрагментом состоит из нескольких компонент.

(1) Нефрагментируемая часть исходного пакета со значением поля Payload Length в исходном заголовке IPv6 в соответствии с реальным размером фрагментированного пакета (без учета самого заголовка IPv6) и значением поля Next Header в последнем заголовке нефрагментируемой части, равным 44.

(2) Заголовок фрагмента, включающий:

Значение Next Header, идентифицирующее первый заголовок фрагментируемой части исходного пакета.

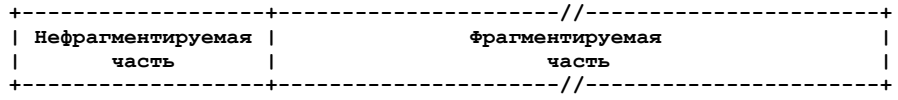
Поле Fragment Offset указывающее смещение фрагмента (в 8-октетных словах) от начала фрагментируемой части исходного пакета. Для первого (самого левого) фрагмента Fragment Offset = 0.

Флаг M имеет значение 0 для последнего фрагмента и 1 для всех прочих фрагментов.

Значение Identification, созданное для исходного пакета.

(3) Собственно фрагмент.

Размер фрагментов должен выбираться так, чтобы размер пакетов с фрагментами не превышал значения MTU на пути к получателю(ям).



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

Сборка фрагментов выполняется в соответствии с приведенными ниже правилами.

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

Нефрагментируемая часть собираемого из фрагментов пакета содержит все заголовки, вплоть до заголовка Fragment (но не включая его) первого пакета с фрагментом (т. е., Fragment Offset = 0) с двумя изменениями:

[1]«Недавно» означает «в пределах максимального вероятного времени жизни пакета, включая время передачи от отправителя к получателю и время ожидания сборки фрагментов». Однако узел-источник не обязан знать максимальное время жизни пакета. Вместо этого предполагается, что требование может быть удовлетворено за счет использования в поле Identification 32-битового кольцевого счетчика, значение которого инкрементируется для каждого фрагментируемого пакета. Разработчики сами могут выбрать поддержку одного счетчика для всего узла или организацию множества счетчиков (например, по одному счетчику для каждого из адресов узла, указываемых в поле отправителя, или отдельные счетчики для каждой активной пары отправитель-получатель).

Значение поля Next Header в последнем заголовке нефрагментируемой части берется из одноименного поля в заголовке Fragment первого фрагмента.

Значение поля Payload Length собираемого пакета вычисляется на основе размера нефрагментируемой части, а также размера и смещения последнего фрагмента. Например, значение этого поля для собранного исходного пакета можно вычислить по формуле:

```
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 за заголовком Fragment в первом
           пакете с фрагментом.
FO.last  = поле Fragment Offset в заголовке Fragment последнего фрагмента.
FL.last  = размер фрагмента, следующего за заголовком Fragment в последнем
           пакете с фрагментом.
```

Фрагментируемая часть исходного пакета восстанавливается из фрагментов, следующих после заголовков Fragment в каждом из пакетов с фрагментами. Размер каждого фрагмента определяется путем вычитания из значения поля Payload Length размера заголовков между заголовком IPv6 и самим фрагментом; положение фрагмента в Fragmentable Part определяется по значению поля Fragment Offset.

Заголовок Fragment не присутствует в собранном заново исходном пакете.

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

Если в течение 60 секунд с момента приема первого фрагмента получены не все фрагменты, требуемые для сборки, сборка должна быть прервана, а все полученные фрагменты - отброшены. Если первый фрагмент (т. е. пакет со значением Fragment Offset = 0) был получен, отправителю этого фрагмента следует передать сообщение ICMP Time Exceeded (Fragment Reassembly Time Exceeded[1]).

Если размер фрагмента, определенный из значения поля Payload Length в заголовке пакета с фрагментом, не кратен 8 октетам, а флаг M имеет значение 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.

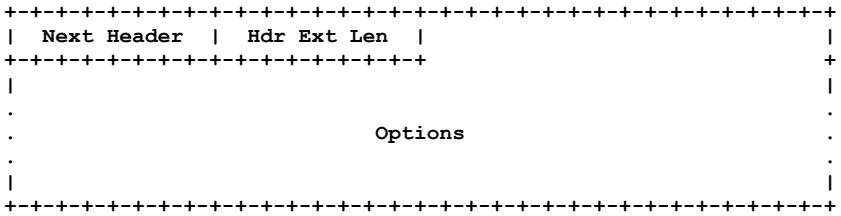
Если значения размера и смещения для фрагмента таковы, что значение поля Payload Length собранного из фрагментов пакета будет превышать 65 535 октетов, этот фрагмент должен быть отброшен, а его отправителю следует передать сообщение ICMP Parameter Problem с кодом 0, указывающее на поле Fragment Offset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

Перечисленные ниже ситуации являются нежелательными, но не трактуются, как ошибки.

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.

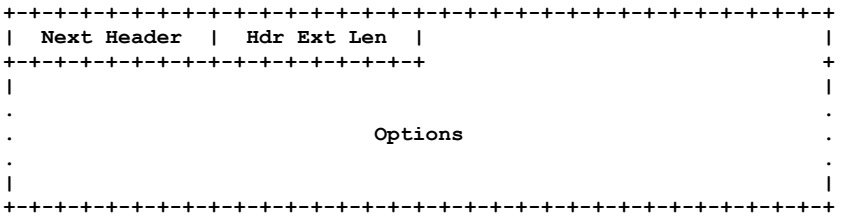
Значения Next Header в заголовках Fragment различных фрагментов одного исходного пакета могут различаться. Для сборки фрагментов используется только значение из пакета, содержащего фрагмент с нулевым смещением.

4.6 Заголовок Destination Options



Заголовок Destination Options используется для передачи дополнительной информации, которая будет просматриваться только на конечном узле(ах). Заголовок Destination Options идентифицируется 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 [RFC-1700 и его преемники].

Hdr Ext Len - размер заголовка расширения

8-битовое целое число без знака, указывающее размер заголовка Destination Options в 8-октетных словах без учета 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 возможных способа представления дополнительной информации для получателя в пакетах IPv6 - в виде опции заголовка Destination Options или в виде отдельного заголовка расширения. Примерами

[1] Истекло время сборки фрагментов.

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 (в случае, 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 (см. параграф 4.2).

4.7 Нет следующего заголовка

Значение 59 в поле Next Header заголовка IPv6 или любого заголовка расширения говорит об отсутствии последующих заголовков. 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.

5. Размер пакетов

IPv6 требует, чтобы каждый канал в сети имел значение MTU не менее 1280 октетов. 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 [RFC-1661]) 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 подключен непосредственно, узел должен быть способен принимать пакеты размером MTU для этого канала.

Для узлов IPv6 настоятельно рекомендуется поддержка механизма Path MTU Discovery [RFC-1981] 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 стороне получателя(ей). 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 .

В ответ на пакет IPv6, отправленный адресату IPv4 (т. е., пакет, преобразованный из IPv6 в IPv4), узел-источник IPv6 может получить сообщение ICMP Packet Too Big, указывающее, что Next-Hop MTU < 1280. В таких случаях от узла IPv6 не требуется снижать размер пакетов до значения меньше 1280. Узел в таких случаях должен использовать заголовок Fragment в пакетах так, чтобы маршрутизатор, выпроняющий преобразование IPv6 в IPv4 получил подходящее значение Identification для использования в последующих фрагментах IPv4. Отметим, что это может потребовать снижения размера данных в пакете до 1232 октетов (1280 - 40 октетов заголовка IPv6 и 8 октетов заголовка Fragment) или меньше, если используются и другие заголовки расширения[1].

6. Метки потоков

20-битовое поле Flow Label в заголовке IPv6 может быть использовано отправителем для маркировки последовательности пакетов, требующей специальной обработки в маршрутизаторах IPv6 (например, отличное от принятого по умолчанию качество сервиса или обслуживание пакетов в реальном масштабе времени). Во время подготовки этого документа этот аспект IPv6 оставался экспериментальным и был связан с изменением требований к поддержке потоков в сети Internet. Хосты и маршрутизаторы, не поддерживающие использование поля Flow Label, должны устанавливать для него нулевое значение в создаваемых пакетах, оставлять значение поля неизменным при пересылке пакетов и игнорировать поле в принимаемых пакетах.

Предполагаемая семантика и применение поля Flow Label описаны в Приложении A.

В отличие от RFC 1883 данная спецификация уменьшает размер метки потока до 20 битов. Упоминание 24-битовых меток на страницах 87 и 88 документа RFC 2205 следует обновить соответственно[2].

7. Классы трафика

8-битовое поле Traffic Class в заголовке IPv6 может использоваться отправителями и/или пересылающими пакеты маршрутизаторами для идентификации и разделения различных классов или приоритетов для пакетов IPv6. К моменту написания этого документа было выполнено множество экспериментов по использованию флагов Type of Service и/или Precedence протокола IPv4 для реализации различных форм дифференцированного обслуживания пакетов IP, отличающихся от явной организации потоков. Поле Traffic Class в заголовке IPv6 предназначено для поддержки аналогичных функций в IPv6.

[1] В http://www.rfc-editor.org/errata_search.php?eid=2843 было предложено удалить этот абзац (предложение отвергнуто). *Прим. перев.*
[2] Этот абзац отсутствует в исходном документе. См. http://www.rfc-editor.org/errata_search.php?eid=2541. *Прим. перев.*

Есть надежда, что эксперименты позволят достичь согласия по вопросу выбора методов классификации трафика, которые будут наиболее полезны для пакетов IP. Детальное описание синтаксиса и семантики всех или некоторых битов поля IPv6 Traffic Class (экспериментальных или планируемых для стандартизации) приведены в отдельных документах.

Ниже приведены общие требования, применимые к полю Traffic Class.

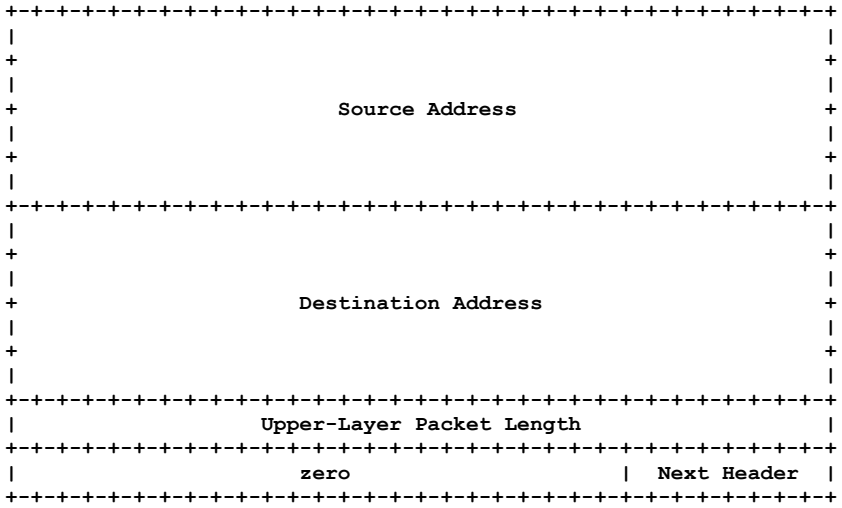
- Интерфейс к услугам IPv6 на узле должен обеспечивать протоколам вышележащего уровня механизм передачи битов Traffic Class в пакетах, порождаемых протоколом вышележащего уровня. По умолчанию все 8 битов поля должны иметь нулевые значения.

- Узлам, поддерживающим специфическое (экспериментальное или планируемое для стандартизации) использование некоторых или всех битов поля Traffic Class, дозволяется изменять значения этих битов в порождаемых, пересылаемых или принимаемых пакетах в соответствии с требованиями конкретного применения. Узлам следует игнорировать и оставлять неизменными все биты поля Traffic Class, для которых они не поддерживают специфического использования.

- Протоколам вышележащего уровня недопустимо предполагать, что значения битов Traffic Class в принятых пакетах совпадают со значениями этих битов на момент отправки пакетов источником.

8. Протоколы вышележащего уровня

8.1 Контрольные суммы вышележащего уровня



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

- Если пакет IPv6 включает заголовок Routing, поле Destination Address, используемое в псевдозаголовке, указывает конечного получателя. На узле-источнике этот адрес будет последним элементом в заголовке Routing, на приемной 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 собственных данных о размере и в этом случае 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 записывать этот факт в системный журнал.

Протокол ICMP для IPv6 [ICMPv6] включает описанный выше псевдозаголовок в расчет контрольной суммы (в отличие от протокола 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, идентифицирующее версию IPv6 протокола ICMP.

8.2 Максимальное время жизни пакета

В отличие от IPv4, узлы IPv6 не обязаны соблюдать максимальное время жизни пакетов. По этой причине поле «Time to Live» (TTL) 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 внесенное в

протокол изменение не имеет существенного практического значения. Любой протокол вышележащего уровня, который опирается на сетевой уровень (неважно, IPv4 или IPv6) для ограничения времени жизни пакетов, должен быть обновлен для обеспечения собственных механизмов детектирования и отбрасывания устаревших пакетов.

8.3 Максимальный размер данных вышележащего уровня

При расчете 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 (принятое по умолчанию или полученное от механизма Path MTU Discovery значение) 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 вычетом 60 октетов, поскольку размер минимального заголовка IPv6 (без заголовков расширения) на 20 октетов превышает минимальный размер заголовка IPv4.

8.4 Отклик на пакеты с заголовками Routing

Когда протокол вышележащего уровня шлет один или множество пакетов в ответ на принятый пакет с заголовком Routing, в пакеты откликов недопустимо включать заголовок Routing, который будет создаваться автоматически путем «обращения» полученного заголовка Routing, **пока** не будет проверена целостность и подлинность поля Source Address и заголовка Routing (например, путем использования 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 путем обращения заголовка Routing из принятого пакета **тогда и только тогда**, когда поле Source Address и заголовок Routing в полученном пакете были проверены отвечающей стороной.

Приложение A. Семантика и применение поля Flow Label

Поток представляет собой последовательность пакетов, передаваемых из конкретного источника конкретному адресату (индивидуальный или групповой адрес), для которых отправитель желает обеспечить специальную обработку на промежуточных маршрутизаторах. Требования к такой специальной обработке могут передаваться маршрутизаторам с помощью протокола управления (такого, как протоколо резервирования ресурсов) или через информацию, содержащуюся в пакетах самого потока (например, в опции hop-by-hop). Детальное рассмотрение таких протоколов и опций выходит за пределы данного документа.

Может существовать множество потоков от отправителя к получателю, наряду с трафиком, который не связан ни с одним потоком. Поток уникально идентифицируется комбинацией адреса отправителя и отличной от нуля меткой потока. Пакеты, не относящиеся к потокам, имеют нулевое значение метки потока.

Метка потока присваивается ему на узле — источнике потока. Новые метки потоков должны выбираться с использованием (псевдо)случайных значений, равномерно распределенных в диапазоне от 1 до FFFFF. Использование случайных значений выбрано для того чтобы любой набор битов поля метки Flow Label мог служить в качестве хэш-ключа для просмотра состояний, связанных с потоками, на маршрутизаторах.

Все пакеты, относящиеся к одному потоку, должны передаваться с одинаковыми значениями адресов отправителя и получателя, а также одинаковыми метками потока. Если любой из пакетов потока содержит заголовок Hop-by-Hop Options, все остальные пакеты этого потока должны генерироваться с такими же заголовками Hop-by-Hop Options (за исключением поля Next Header в заголовке Hop-by-Hop Options). Если любой из пакетов потока включает заголовок Routing, все остальные пакеты потока должны генерироваться с таким же содержимым заголовков расширения вплоть до заголовка Routing включительно (исключением является лишь поле Next Header в заголовке Routing). Маршрутизаторы и получатель могут, но не обязаны, проверять соблюдение приведенных выше условий. При обнаружении нарушения условий следует возвращать отправителю сообщение ICMP Parameter Problem с кодом 0, указывающее на старший октет поля Flow Label (т. е., смещение 1 в заголовке IPv6).

Максимальное время жизни любого состояния обслуживания потока вдоль его пути должно быть задано, как часть механизма организации состояния (например, с помощью протокола резервирования ресурсов или опции hop-by-hop). Отправителю недопустимо заново использовать метку для нового потока в течение максимального срока жизни любого состояния обслуживания потока, которое могло быть организовано до начало использования этой метки.

При перезапуске узла (например, в результате аварии) ему следует осторожно подходить к выбору меток, чтобы не использовать выделенную ранее метку, для которой могло еще не завершиться максимальное время жизни. Это может быть обеспечено путем записи использованных меток потоков на стабильное устройство хранения, информация в котором не теряется при авариях, или отказа от использования меток потоков до момента завершения времени жизни всех меток, которые могли быть выделены ранее. Если известно минимальное время перезагрузки узла, это время может быть вычтено из интервала ожидания перед началом выделения меток потока.

Не требуется, чтобы все или хотя бы большинство пакетов относились к потокам (т. е., имели отличные от нуля метки потока). Это замечание приведено здесь для того, чтобы разработчики не предполагали обратного. Например, было бы неразумно создать маршрутизатор, который будет обеспечивать адекватную производительность только в случае принадлежности большинства пакетов к потокам, или разработать схему компрессии заголовков, которая будет работать только для включенных в потоки пакетов.

[1] Максимальный размер сегмента TCP.

Приложение B. Рекомендации по формату опций

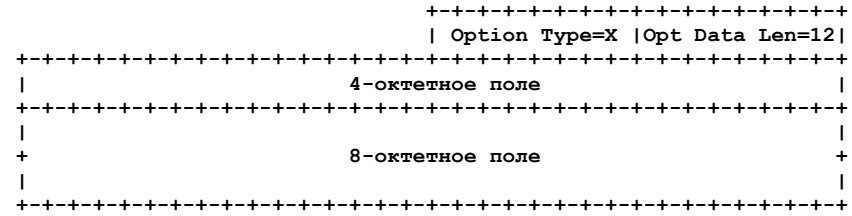
В этом приложении приведены некоторые рекомендации по расположению полей в новых опциях для использования с заголовками расширения Hop-by-Hop Options и Destination Options, как описано в параграфе 4.2. Рекомендации базируются на следующих допущениях:

- 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 учетом того, что размер заголовка должен быть кратным 8 октетам;
- можно предположить, что в заголовках с опциями число опций невелико (обычно одна опция).

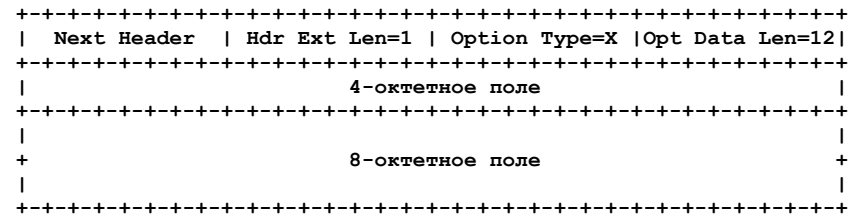
На основе этих допущений предлагается следующая схема размещения полей опции - поля упорядочиваются по возрастанию размера без внутреннего заполнения, а после этого выполняется выравнивание для опции в целом за счет выравнивания самого большого поля (максимальное заполнение для выравнивания может составить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

Пример 1

Если для опции X требуется два поля данных, размером 8 и 4 октета, их следует расположить так:

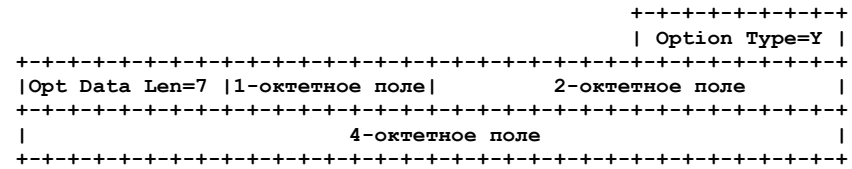


Выравнивание должно быть выполнено по формуле 8n+2, чтобы 8-октетное поле начиналось со смещением от начала содержащего опцию заголовка, кратным 8. Схема полного заголовка Hop-by-Hop Options или Destination Options, содержащего эту опцию, показана на рисунке.

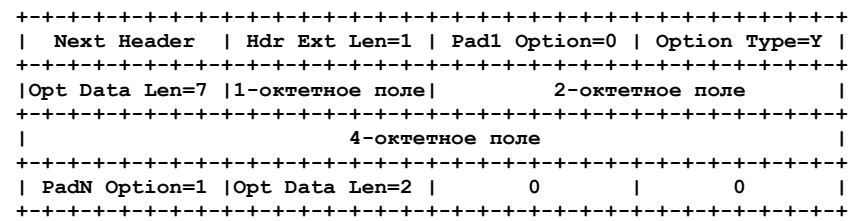


Пример 2

Если опция Y включает три поля размером 4, 2 и 1 октет, они будут располагаться следующим образом.

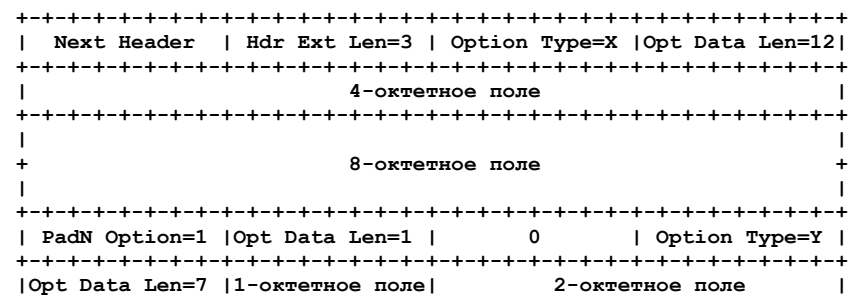


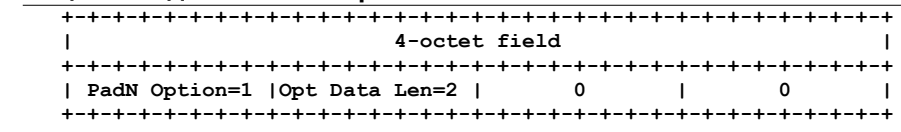
Выравнивание осуществляется по формуле 4n+3, чтобы 4-октетное поле имело смещение от начала содержащего опцию заголовка, кратное 4. Схема полного заголовка Hop-by-Hop Options или Destination Options, содержащего эту опцию, показана на рисунке.



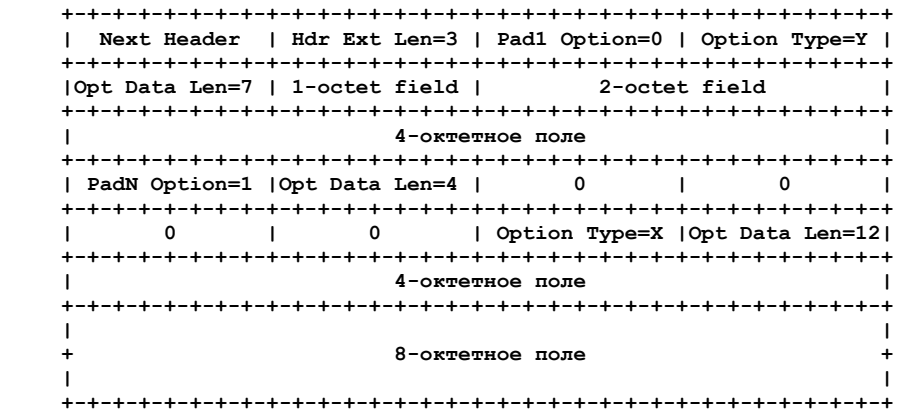
Пример 3

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





или



Вопросы безопасности

Вопросы безопасности IPv6 рассмотрены в документе «Архитектура безопасности для протокола Internet» [RFC-2401].

Благодарности

Адресы благодарят за множество полезных предложений членов рабочей группы IPng, исследовательской группы End-to-End Protocols, а также все сообщество Internet.

Адреса авторов

Stephen E. Deering

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134-1706

USA

Phone: +1 408 527 8213

Fax: +1 408 527 8254

EMail: deering@cisco.com

Robert M. Hinden

Nokia

232 Java Drive

Sunnyvale, CA 94089

USA

Phone: +1 408 990-2004

Fax: +1 408 743-5677

EMail: hinden@iprg.nokia.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Литература

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

[RFC-2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC-2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Protocol (ESP)", RFC 2406, November 1998.

[ICMPv6] Conta, A. and S. Deering, "ICMP for the Internet Protocol Version 6 (IPv6)", RFC 2463, December 1998.

[ADDRARCH] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

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

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

[RFC-1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. См. также: http://www.iana.org/numbers.html

[RFC-1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

Отличия от RFC 1883

Этот документ имеет ряд отличий от RFC 1883. Эти различия перечислены ниже с указанием версии документов Internet-Draft, в которых изменения были внесены.

02) Удалены все упоминания джамбограмм (jumbogram) и опции Jumbo Payload (перенесено в отдельный документ).

02) Большая часть описания меток потока (Flow Label) перенесена из раздела 6 в (новое) Приложение A.

02) В описании Flow Label (сейчас Приложение A) максимальное значение поля Flow Label FFFFFF заменено на FFFFF (т. е., меньше на одну F) для уменьшения размера Flow Label с 24 до 20 битов.

02) Приложение A переименовано в Приложение B.

02) Заменен текст раздела «Вопросы безопасности» (Security Considerations) во избежание путаницы между этой спецификацией и спецификациями IPsec.

02) Обновлен адрес электронной почты и место работы R. Hinden.

---

01) В разделе 3 изменено название поля Class на Traffic Class, а размер этого поля увеличен с 4 до 8 битов. Уменьшен размер поля Flow Label с 24 до 20 битов для компенсации увеличения размера поля Traffic Class.

01) В параграфе 4.1 восстановлен порядок заголовков Authentication и ESP, который был по ошибке перепутан в черновом варианте 00.

01) В параграфе 4.1 удалено поле Strict/Loose Bit Map и функциональность строгого задания маршрута из заголовка Routing типа 0, а также удалено ограничение числа адресов, которые могут передаваться в заголовке Routing типа 0 (раньше число адресов не должно было превышать 23 в связи с размером битового отображения strict/loose).

01) В разделе 5 минимальное значение IPv6 MTU изменено с 576 на 1280 октетов и добавлена рекомендация для каналов с настраиваемым значением MTU (например, PPP) поддерживать значение MTU не менее 1500 октетов.

01) В разделе 5 удалено требование к узлам не передавать фрагментированные пакеты, которые при сборке дают пакет размером более 1500 без наличия информации о размере буфера сборки на приемной стороне. Взамен рекомендуется не использовать такой фрагментации протоколам вышележащего уровня.

01) Ссылка на спецификацию механизма IPv4 Path MTU Discovery (RFC 1191) заменена ссылкой на спецификацию IPv6 Path MTU Discovery (RFC 1981) и удалены примечания в конце раздела 5, относящиеся к Path MTU Discovery, поскольку эти детали включены сейчас в RFC 1981.

01) В разделе 6 удалена спецификация «оппортунистической» организации потока, а также все ссылки на 6-секундное максимальное время жизни для таких «оппортуричстических» состояний.

01) В разделе 7 удалено описание внутренней структуры и сементики поля Traffic Class и указано, что эти описания представлены в отдельных документах.

---

00) В разделе 4 исправлено значение кода для индикации нераспознанного типа следующего заголовка (unrecognized Next Header type encountered) в сообщении ICMP Parameter Problem (код 2 заменен кодом 1).

00) В описании поля Payload Length в разделе 3 и поля Jumbo Payload Length в параграфе 4.3 даны разъяснения о том, что размер заголовков расширения учитывается в размере данных.

00) В параграфе 4.1 изменен порядок заголовков Authentication и ESP (это было ошибкой, которая была исправлена в версии 01).

00) В параграфе 4.2 даны разъяснения о том, что опции идентифицируются полным 8-битовым значением Option Type, а не только 5 битами поля Option Type. Также указано, что заголовки Hop-by-Hop Options и Destination Options используют общее пространство значений Option Type.

00) В параграфе 4.4 добавлено предложение, требующее от узлов при обработке заголовка Routing передавать сообщение ICMP Packet Too Big (например, вместо фрагментации) в ответ на получение пакета, размер которого слишком велик для канала на следующем интервале.

00) Имя поля IPv6 Priority заменено на Class и, соответственно, описание Priority в разделе 7 заменено описанием поля Class. Кроме того, это поле исключено из набора полей, которые должны совпадать во всех пакетах одного потока (раздел 6).

00) В псевдозаголовке (параграф 8.1) имя поля Payload Length заменено на Upper-Layer Packet Length. Разъяснено также, что для протоколов, поддерживающих свои данные о размере (например, UDP без jambo), в этом поле указывается размер от протокола вышележащего уровня (а не размер от IP).

00) Добавлен параграф 8.4, указывающий, что протоколам вышележащего уровня при отклике па полученные пакеты с заголовком Routing недопустимо включать обращённый заголовок Routing в пакеты отклика, пока полученный заголовок Routing не аутентифицирован.

00) Исправлены опечатки и грамматические ошибки.

00) Обновлены контактные данные авторов.

Полное заявление авторских прав