

Network Working Group
Request for Comments: 4213
Obsoletes: 2893
Category: Standards Track

E. Nordmark
Sun Microsystems, Inc.
R. Gilligan
Intransa, Inc.
October 2005

Базовые механизмы перехода для хостов и маршрутизаторов IPv6

Basic Transition Mechanisms for IPv6 Hosts and Routers

Статус документа

Этот документ содержит проект стандарта протокола Internet для сообщества Internet и служит приглашением к дискуссии в целях развития и совершенствования предложенного протокола. Информацию о текущем статусе протокола и состоянии стандартизации можно найти в документе Internet Official Protocol Standards (STD 1). Документ может распространяться свободно.

Авторские права

Copyright (C) The Internet Society (2005).

Тезисы

Этот документ задает механизмы совместимости с IPv4, которые могут быть развернуты на хостах и маршрутизаторах IPv6. Приведены спецификации двух механизмов - двойной стек и настраиваемое туннелирование. Механизм с двойным стеком протоколов предполагает полную реализацию обеих версий протокола IP (IPv4 и IPv6), а механизм настраиваемого туннелирования позволяет передавать пакеты IPv6 через инфраструктуру маршрутизации IPv4 без ее изменения.

Этот документ служит заменой RFC 2893.

Оглавление

1. Введение.....	1
1.1. Терминология.....	2
2. Работа Dual IP.....	2
2.1. Настройка адресов.....	2
2.2. DNS.....	3
3. Механизмы настраиваемого туннелирования.....	3
3.1. Инкапсуляция.....	3
3.2. MTU для туннеля и фрагментация.....	4
3.2.1. Статическое значение MTU для туннеля.....	4
3.2.2. Динамическое значение MTU для туннеля.....	4
3.3. Hop Limit.....	5
3.4. Обработка сообщений ICMPv4 об ошибках.....	5
3.5. Создание заголовка IPv4.....	6
3.6. Декапсуляция.....	6
3.7. Адреса Link-Local.....	7
3.8. Обнаружение соседей через туннели.....	8
4. Угрозы, связанные с подменой адресов отправителей.....	8
5. Вопросы безопасности.....	8
6. Благодарности.....	9
7. Литература.....	9
7.1. Нормативные документы.....	9
7.2. Дополнительная литература.....	9
8. Отличия от RFC 2893.....	10

1. Введение

Ключевым элементом успешного перехода на IPv6 является совместимость с большой базой установленных хостов и маршрутизаторов IPv4. Обеспечение совместимости с IPv4 при развертывании IPv6 упростит задачу перевода сети Internet на протокол IPv6. В этой спецификации определены два механизма, которые могут быть реализованы на хостах и маршрутизаторах IPv6 для обеспечения совместимости с хостами и маршрутизаторами IPv4.

Описанные в документе механизмы разработаны для развертывания на хостах и маршрутизаторах IPv6, которым требуется интероперабельность с хостами IPv4 и возможность использования инфраструктуры маршрутизации IPv4. Предполагается, что такая совместимость потребуется для большинства узлов Internet в течение длительного (возможно, неопределенного) срока.

Предлагаемые здесь механизмы включают:

- Dual IP (двойной стек IP) - метод обеспечения полной поддержки обеих версий протокола IP (IPv4 и IPv6) на хостах и маршрутизаторах.
- Настраиваемое туннелирование IPv6 через IPv4 - метод организации туннелей «точка-точка» за счет инкапсуляции пакетов IPv6 с заголовками IPv4 для передачи через маршрутную инфраструктуру IPv4.

Механизмы, определенные здесь, предназначены на роль основных инструментов перехода - расширяемого набора методов, которые разработчики и пользователи могут применять для упрощения перехода на новую версию протокола. Использование конкретных инструментов определяется реальными потребностями. Разработчики и сайты самостоятельно принимают решения о выборе конкретных методов решения своих задач.

В этом документе определяется базовый набор механизмов перехода, который не исчерпывает всего спектра имеющихся механизмов. В других документах описаны дополнительные механизмы перехода и обеспечения совместимости.

1.1. Терминология

Ниже определены основные термины, используемые в этом документе.

Типы узлов

IPv4-only node (только IPv4)

Хост или маршрутизатор, на котором развернут только протокол IPv4. Такие узлы не понимают протокол IPv6. Установленная база хостов и маршрутизаторов IPv4, существовавших до начала перехода относится к этому типу узлов.

IPv6/IPv4 node (IPv6 и IPv4)

Хосты и маршрутизаторы, поддерживающие протоколы IPv4 и IPv6.

IPv6-only node (только IPv6)

Хост или маршрутизатор, который поддерживает IPv6, но не поддерживает IPv4. Работа узлов этого типа в данном документе не рассматривается.

IPv6 node (узел IPv6)

Любой хост или маршрутизатор, поддерживающий IPv6. Узлы типа IPv6/IPv4 и IPv6-only относятся к узлам IPv6.

IPv4 node (узел IPv4)

Любой хост или маршрутизатор, поддерживающий IPv4. Узлы типа IPv6/IPv4 и IPv4-only относятся к узлам IPv4.

Методы, используемые при переходе

IPv6-over-IPv4 tunneling (туннелирование IPv6 через IPv4)

Метод инкапсуляции пакетов IPv6 в пакеты IPv4, которые могут передаваться через инфраструктуру маршрутизации IPv4.

Configured tunneling (настраиваемое туннелирование)

Туннелирование IPv6-over-IPv4 при котором адреса IPv4 конечных точек туннеля определяются конфигурационными параметрами этих точек. Все туннели предполагаются двухсторонними. Туннель обеспечивает (виртуальный) канал «точка-точка» для уровня IPv6 с использованием заданных в конфигурации адресов IPv4 в качестве адресов конечных точек нижележащего уровня.

Другие механизмы перехода, включая иные методы туннелирования, выходят за пределы этого документа.

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

2. Работа Dual IP

Наиболее простым способом обеспечения совместимости узлов IPv6 с узлами, поддерживающими только IPv4, является полная реализация протокола IPv4. Узлы, полностью поддерживающие оба протокола IPv4 и IPv6, называют узлами IPv6/IPv4. Такие узлы способны передавать и принимать как пакеты IPv4, так и пакеты IPv6. Они могут напрямую взаимодействовать с узлами IPv4, используя пакеты IPv4, и с узлами IPv6, используя пакеты IPv6.

Даже на узлах, способных поддерживать обе версии протокола, тот или иной из двух стеков может быть отключен. Включенный стек имеет выделенный ему адрес IP, но доступность для стеков тех или иных приложений явно не определена. Таким образом, узлы IPv6/IPv4 могут работать в одном из трех режимов:

- стек IPv4 включен, а IPv6 отключен;
- стек IPv6 включен, а IPv4 отключен;
- оба стека протоколов включены.

Узлы IPv6/IPv4 с отключенным стеком IPv6 будут работать, как узлы IPv4-only. Аналогично, узлы IPv6/IPv4 с отключенным стеком IPv4 будут работать, как IPv6-only. Узлы IPv6/IPv4 **могут** поддерживать конфигурационные параметры для включения и отключения их стеков IPv4 или IPv6.

Метод настраиваемого туннелирования, описанный в разделе 3, может использоваться в дополнение к методу Dual IP.

2.1. Настройка адресов

Поскольку узлы IPv6/IPv4 поддерживают обе версии протокола, для них в конфигурации могут задаваться адреса как IPv4, так и IPv6. Узлы IPv6/IPv4 используют механизмы IPv4 (например, DHCP) для получения адресов IPv4 и механизмы IPv6 (например, автоматическая настройка конфигурации [RFC2462] и/или DHCPv6) для получения адресов IPv6.

2.2. DNS

Система доменных имен (DNS¹) используется как в IPv4, так и в IPv6 для отображения имен хостов на адреса IP и обратно. Для адресов IPv6 в документе [RFC3596] определен новый тип записей о ресурсах - AAAA. Поскольку узлы IPv6/IPv4 должны обеспечивать интероперабельность с узлами IPv4 и IPv6, они должны поддерживать библиотеки преобразования, способные работать как с записями IPv4 типа A, так и с записями IPv6 типа AAAA. Отметим, что поиск записей A и AAAA не зависит от протокола, используемого для передачи пакетов DNS (IPv4 или IPv6), поэтому не делается допущения о том, что серверы DNS знают о возможностях поддержки IPv4/IPv6 на запрашивающих узлах.

Вопросы использования IPv6 с DNS и рабочие рекомендации более подробно рассмотрены в других документах (например, [DNSOPV6]).

Библиотеки преобразования DNS (resolver) на узлах IPv6/IPv4 **должны** поддерживать обработку записей обоих типов AAAA и A. Однако, если запросы возвращают записи AAAA с адресом IPv6 и записи A с адресом IPv4, библиотека преобразования **может** упорядочивать возвращаемые приложению результаты с учетом версии пакетов IP, используемых в коммуникациях с конкретным узлом - сначала IPv6 или сначала IPv4.

Приложениям **следует** поддерживать возможность задания желаемых типов записей (IPv4, IPv6 или оба типа [RFC3493]). Это определяет, какие семейства адресов преобразователь будет искать. Если приложение не указывает своего выбора или запрашивает оба типа, для библиотек преобразования **недопустима** фильтрация записей.

Поскольку большинство приложений пытаются использовать адреса в порядке, возвращаемом преобразователем, это может оказывать влияние на выбор версии IP «предпочитаемой» приложением.

Реальные механизмы упорядочения выходят за пределы этого документа. Более подробно выбор адресов рассмотрен в [RFC3484].

3. Механизмы настраиваемого туннелирования

В большинстве вариантов развертывания инфраструктура маршрутизации IPv6 будет строиться достаточно долго. Пока будет создаваться инфраструктура IPv6, существующая инфраструктура маршрутизации IPv4 будет продолжать функционирование и может использоваться для передачи трафика IPv6. Туннелирование обеспечивает возможность использования существующей инфраструктуры маршрутизации IPv4 для доставки трафика IPv6.

Хосты и маршрутизаторы IPv6/IPv4 могут туннелировать дейтаграммы IPv6 через области маршрутной топологии IPv4 за счет их инкапсуляции в пакеты IPv4. Туннелирование можно организовать множеством способов:

- **Router-to-Router.** Маршрутизаторы IPv6/IPv4, соединенные через инфраструктуру IPv4, могут туннелировать между собой пакеты IPv6. В этом случае туннель представляет собой один сегмент сквозного пути пакетов IPv6.
- **Host-to-Router.** Хосты IPv6/IPv4 могут туннелировать пакеты IPv6 на промежуточные маршрутизаторы IPv6/IPv4, доступные через инфраструктуру IPv4. Этот тип туннеля представляет собой первый сегмент сквозного пути доставки пакетов.
- **Host-to-Host.** Хосты IPv6/IPv4, связанные через инфраструктуру IPv4 могут туннелировать пакеты IPv6 между собой. В этом случае туннель будет представлять собой весь сквозной путь доставки пакетов.
- **Router-to-Host.** Маршрутизаторы IPv6/IPv4 могут туннелировать пакеты IPv6 конечным получателям IPv6/IPv4. В этом случае туннель представляет собой последний сегмент сквозного пути доставки пакетов.

Настраиваемое туннелирование может использоваться во всех перечисленных случаях, но больше всего оно подходит для туннелирования между маршрутизаторами по причине необходимости явной настройки конечных точек туннелей.

Основными элементами туннелирования являются:

- входной узел туннеля (инкапсулятор), создающий заголовок инкапсуляции IPv4 и передающий инкапсулированный пакет;
- выходной узел туннеля (декапсулятор), принимающий инкапсулированный пакет, выполняющий при необходимости сборку фрагментов, удаляющий заголовок IPv4 и обрабатывающий принятый пакет IPv6;
- инкапсулятору может потребоваться поддержка информации о состоянии для каждого туннеля, включающей такие данные, как MTU для туннеля, чтобы соответствующим образом обрабатывать пакеты IPv6, пересылаемые в туннель.

При настраиваемом туннелировании адреса конечных точек туннеля определяются на инкапсуляторе из конфигурационных параметров, хранящихся для каждого туннеля. Когда пакеты IPv6 передаются через туннель, адреса отправителя и получателя в инкапсулирующем заголовке IPv4 задаются в соответствии с параграфом 3.5.

Решение вопроса о туннелировании обычно принимается на основе маршрутной информации в инкапсуляторе. Обычно это происходит с использованием таблицы маршрутизации, которая направляет пакеты в соответствии с адресами их получателей, используя методы сравнения масок и префиксов.

Декапсулятор проверяет соответствие пакетов с номером протокола 41 настроенным на нем туннелем и принимает только пакеты, в которых адрес отправителя IPv4 соответствует настроенному на декапсуляторе туннелю. Следовательно, оператор должен гарантировать, что конфигурация адресов IPv4 для туннелей совпадает на стороне инкапсулятора и декапсулятора.

3.1. Инкапсуляция

Инкапсуляция дейтаграмм IPv6 в пакеты IPv4 показана на рисунке.

Кроме добавления заголовка IPv4 инкапсулятор решает некоторые более сложные вопросы:

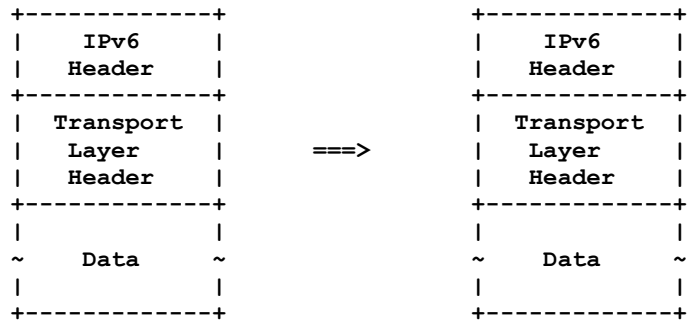
¹Domain Naming System.

- определяет необходимость фрагментирования и передачи сообщений ICMPv6 packet too big¹ отправителям пакетов;
- способ передачи сообщений ICMPv4 от маршрутизаторов вдоль туннеля отправителю исходных пакетов, как сообщений ICMPv6.

Эти вопросы рассмотрены в последующих параграфах.

3.2. MTU для туннеля и фрагментация

В примитивном варианте инкапсулятор может рассматривать процесс инкапсуляции, как IPv6, использующий протокол IPv4 в качестве канального уровня с очень большим значением MTU (до 65535-20 байтов; 20 байтов расходуется на инкапсулирующий заголовок IPv4). Инкапсулятору нужно лишь передавать сообщения ICMPv6 packet too big отправителям пакетов, размер которых превышает это значение MTU. Однако такая схема не будет эффективной и не обеспечит интероперабельности по перечисленным ниже трем причинам. Следовательно, **недопустимо** использовать такую примитивную схему.



Инкапсуляция IPv6 в IPv4.

- 1) Возникает избыточная фрагментация. Фрагментации на уровне IPv4 следует избегать, поскольку могут возникать проблемы с производительностью в результате потери блоков данных, размер которых меньше размера при повторе передачи [KM97].
- 2) Любая фрагментация IPv4 внутри туннеля (между инкапсулятором и декапсулятором) будет приводить к сборке фрагментов в конечной точке туннеля. Для туннелей, завершающихся на маршрутизаторах, потребуются дополнительный расход памяти и других ресурсов на сборку фрагментов IPv4 в полный пакет IPv6 до его пересылки.
- 3) У инкапсулятора может не быть возможности узнать о способности декапсулятора дефрагментировать такие пакеты IPv4 (см. параграф 3.6) и способности декапсулятора обрабатывать столь большие блоки IPv6 MRU².

Следовательно, инкапсулятору **недопустимо** трактовать туннель, как интерфейс с MTU = 64 кбайт, а взамен этого следует использовать статически заданное фиксированное значение MTU или **необязательный** механизм динамического определения MTU на основе MTU на пути IPv4 к конечной точке туннеля.

При реализации обоих механизмов решение о выборе конкретного варианта **следует** принимать на основе конфигурационных параметров, задаваемых для конечных точек каждого туннеля.

3.2.1. Статическое значение MTU для туннеля

Узел, использующий статическое значение MTU для туннеля, трактует туннельный интерфейс, как интерфейс с фиксированным MTU. По умолчанию значение MTU **должно** лежать в диапазоне от 1280 до 1480 байтов (включительно), но **следует** устанавливать значение 1280 байтов. Если используемое по умолчанию значение отличается от 1280, реализация **должна** иметь конфигурационный параметр, позволяющий изменить значение MTU.

Узел должен быть способен воспринимать фрагментированные пакеты IPv6, размер которых после сборки не превышает 1500 октетов [RFC2460]. Этот документ также включает требования (см. параграф 3.6) для количества сборок фрагментов IPv4 и IPv6 MRU, которые **должны** поддерживаться всеми декапсуляторами. Это гарантирует интероперабельность при всех фиксированных значениях MTU из диапазона 1280 - 1480 байтов.

Большие фиксированные значения MTU, поддерживаемые в соответствии с данной спецификацией, недопустимо использовать, пока на административном уровне не гарантируется возможность сборки пакетов такого размера на стороне декапсулятора.

Выбор подходящего значения MTU зависит от множества факторов, часть которых перечислена ниже:

- когда пакеты IPv4 с номером протокола 41 будут передаваться через среду, которая может иметь меньшее значение path MTU (например, IPv4 VPN³), выбор большого значения может привести к излишней фрагментации IPv4;
- при использовании туннеля для транспортировки туннелированных пакетов IPv6 (например, мобильный узел с настраиваемым туннелем IPv6-in-IPv4 и туннельный интерфейс Ipv6-in-IPv6), выбор слишком малого значения может приводить к излишней фрагментации IPv6.

Если предполагается наличие многоуровневой инкапсуляции, представляется разумным рассмотреть поддержку динамического определения MTU с целью минимизации фрагментирования и оптимизации размера пакетов.

При использовании статического MTU для туннеля, в инкапсулирующем заголовке IPv4 **недопустимо** устанавливать флаг DF⁴. В результате этого инкапсулятор не должен получать сообщений ICMPv4 о недопустимо большом размере для инкапсулированных им пакетов.

3.2.2. Динамическое значение MTU для туннеля

Динамическое определение MTU является **необязательным**. Однако при его реализации **следует** выполнять рекомендации, содержащиеся в данном документе.

¹Слишком большой пакет.

²Maximum Receive Unit - максимальный принимаемый блок.

³Virtual Private Network - виртуальная частная сеть.

⁴Don't Fragment - не фрагментировать.

Фрагментация внутри туннеля может быть сведена к минимуму за счет определения инкапсулятором значений IPv4 MTU в туннеле с помощью протокола IPv4 Path MTU Discovery Protocol [RFC1191] и записи результирующего значения MTU для всего пути. Уровень IPv6 в инкапсуляторе может в этом случае рассматривать туннель, как канальный уровень со значением MTU, равным IPv4 MTU для пути за вычетом размера инкапсулирующего заголовка IPv4.

Отметим, что это не предотвращает фрагментации IPv4 в тех случаях, когда значение MTU для пути IPv4 будет давать для IPv6 MTU значение менее 1280 байтов (любой канальный уровень, используемый IPv6, должен иметь значение MTU не менее 1280 байтов [RFC2460]). В этом случае уровень IPv6 «видит» канальный уровень с MTU = 1280, а инкапсулятор использует фрагментацию IPv4 для пересылки 1280-байтовых пакетов IPv6.

Инкапсулятору **следует** реализовать приведенный ниже алгоритм для решения вопроса об использовании фрагментации IPv4 при пересылке через туннель пакетов IPv6, размер которых превышает MTU для пути через туннель, и возврате отправителю сообщений ICMPv6 о недопустимом размере пакетов [RFC1981]:

```

if (IPv4 path MTU - 20) < 1280
    if размер пакета > 1280 байтов
        Передать сообщение ICMPv6 packet too big с MTU = 1280 и отбросить пакет
    else
        Инкапсулировать пакет без установки флага DF в заголовке IPv4. Полученный
        пакет IPv4 может быть фрагментирован уровнем IPv4 на инкапсуляторе или
        другом маршрутизаторе по пути IPv4.
    endif
else
    if размер пакета > (IPv4 path MTU - 20)
        Передать сообщение ICMPv6 «packet too big» с MTU = (IPv4 path MTU - 20) и
        отбросить пакет.
    else
        Инкапсулировать и установить флаг DF в заголовке IPv4.
    endif
endif
    
```

Инкапсуляторы с большим числом туннелей могут выбирать между статическим и динамическим определением MTU независимо для каждого туннеля. В случаях, когда один узел использует большое число туннелей, полезно кэшировать информацию о состоянии туннелей и отбрасывать неиспользуемые записи из кэша.

Отметим, что при использовании динамических значений MTU для туннелей могут возникать «черные дыры» IPv4 MTU, когда сообщения ICMPv4 о недопустимом размере пакетов будут отбрасываться межсетевыми экранами или не будут генерироваться маршрутизаторами [RFC1435, RFC2923].

3.3. Hop Limit

Туннели IPv6-over-IPv4 моделируются с точки зрения IPv6, как «один интервал». Туннель является «черным ящиком» для пользователей и не детектируется средствами диагностики сетей типа traceroute.

Модель на базе «одного интервала» реализуется за счет процессов инкапсуляции/декапсуляции, которые устанавливают значение поля IPv6 hop limit, как при пересылке через любой другой канал данных (т. е., они уменьшают значение поля hop limit на 1 при пересылке пакета IPv6). Исходный и конечный узлы не меняют значение этого поля.

Значение TTL в инкапсулирующем заголовке IPv4 выбирается по усмотрению реализации протокола. Предлагаемые в настоящее время значения опубликованы в документе Assigned Numbers [RFC3232][ASSIGNED]. Разработчики **могут** обеспечивать механизмы, позволяющие администратору задавать значение IPv4 TTL, как IP Tunnel MIB [RFC4087].

3.4. Обработка сообщений ICMPv4 об ошибках

В ответ на переданные в туннель пакеты инкапсулятор может получать сообщения ICMPv4 об ошибках от маршрутизаторов IPv4 в туннеле. Эти пакеты адресуются инкапсулятору, поскольку он является источником IPv4 для пакета.

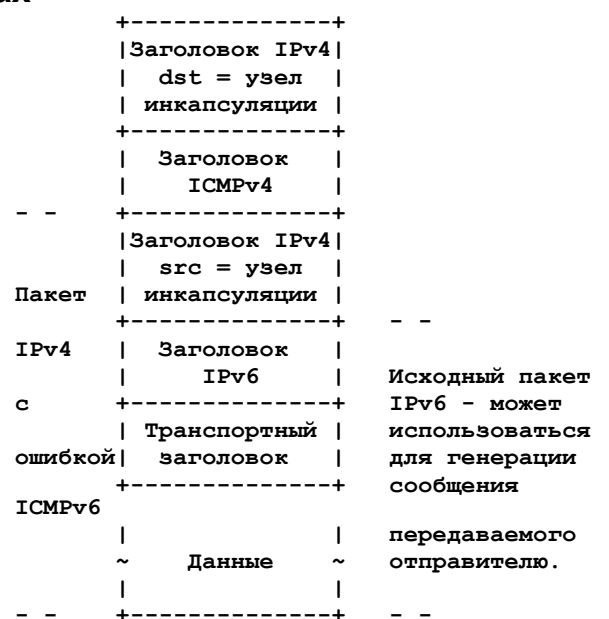
Обработка сообщений ICMPv4 применима только для динамического определения MTU, хотя эти функции могут использоваться и при статическом задании MTU для туннеля.

Сообщения ICMPv4 о недопустимо большом размере пакетов обрабатываются в соответствии с механизмом IPv4 Path MTU Discovery [RFC1191] и полученное в результате значение MTU для пути записывается уровнем IPv4. Это значение MTU для пути используется IPv6 для решения вопроса о генерации сообщений ICMPv6 packet too big, как описано в параграфе 3.2.2.

Обработка других типов сообщений ICMPv4 зависит от доступности информации из инкапсулированного пакета, вызвавшего ошибку.

Многие старые маршрутизаторы IPv4 возвращают лишь 8 байтов данных после заголовка IPv4 в вызвавшем ошибку пакете. Этой информации недостаточно для включения адресных полей заголовка IPv6. Более современные маршрутизаторы IPv4 могут возвращать количество данных после заголовка IPv4, достаточное для включения полного заголовка IPv6 и, возможно, некоторого объема данных (см. [RFC1812]).

Если объем данных из вызвавшего ошибку пакета достаточно велик, инкапсулятор **может** извлечь информацию из инкапсулированного пакета IPv6 и использовать ее для генерации сообщения ICMPv6 исходному отправителю IPv6, как показано на рисунке.



Сообщение ICMPv4, возвращаемое инкапсулятору

При получении сообщений ICMPv4, отличных от packet too big, полезно записывать эти события в системный журнал, как ошибки, связанные с туннелем. При достаточном объеме данных узел **может** передавать сообщения ICMPv6 типа unreachable¹ с кодом address unreachable² отправителю IPv6 (код address unreachable подходит, поскольку с точки зрения IPv6 туннель является каналом, а этот код служит для связанных с каналами ошибок [RFC2463]).

Отметим, что в случаях превышения MTU для пути IPv4, если данных, связанных с сообщением ICMPv4 об ошибке, недостаточно или сообщение ICMPv4 не вызывает генерации сообщения ICMPv6 при достаточном объеме данных, будет отбрасываться не менее двух пакетов (взамен отбрасывания не менее одного для случая одноуровневого определения MTU). Рассмотрим случай, когда хост IPv6 подключен к маршрутизатору IPv4/IPv6, который соединен с сетью, где генерируется сообщение ICMPv4 об избыточном размере пакета. Сначала маршрутизатору нужно определить значение MTU (IPv4) для туннеля, что вызовет потерю по крайней мере одного пакета, а потом хосту нужно узнать значение MTU (IPv6) от маршрутизатора, что приведет к потере еще по крайней мере одного пакета. Тем не менее во всех случаях может теряться более одного пакета, если одновременно приходит множество избыточно больших пакетов.

3.5. Создание заголовка IPv4

При инкапсуляции пакета IPv6 в дейтаграмму IPv4 поля заголовка IPv4 устанавливаются следующим образом:

Version

4

IP Header Length (в 32-битовых словах)

5 (в инкапсулирующем заголовке нет опций IPv4).

Type of Service

0, если явно не задано иное (см. [RFC2983] и параграф 9.1 [RFC3168], где рассматривается использование поля ToS и туннелирование).

Total Length

Размер данных из заголовка IPv6 плюс размер заголовков IPv6 и IPv4 (т. е., размер данных IPv6 + 60 байтов).

Identification

Генерируется, как для прочих пакетов IPv4, передаваемых системой.

Flags

Флаг DF устанавливается в соответствии с параграфом 3.2. При фрагментировании в пакетах, не содержащих последний фрагмент устанавливается флаг MF³.

Fragment Offset

Устанавливается в соответствии с фрагментированием.

Time to Live

Устанавливается реализацией, как описано в параграфе 3.3.

Protocol

41 (номер протокола для IPv6).

Header Checksum

Контрольная сумма заголовка IPv4 [RFC791].

Source Address

Адрес IPv4 для инкапсулятора (адрес, заданный администратором, или адрес выходного интерфейса).

Destination Address

Адрес IPv4 для конечной точки туннеля.

При инкапсуляции пакетов узел должен гарантировать использование корректного адреса отправителя, чтобы пакеты были приемлемы для декапсулятора, как описано в параграфе 3.6. Настройка адреса отправителя приемлема, в частности, для случаев, когда автоматическое определение адреса отправителя может давать результаты, меняющиеся с течением времени. Это часто возникает при наличии множества адресов и множества интерфейсов или при частом изменении маршрутов. Поэтому в таких случаях **следует** обеспечивать возможность административно задавать адрес отправителя для туннеля.

3.6. Декапсуляция

Когда хост или маршрутизатор IPv6/IPv4 получает дейтаграмму IPv4, направленную по одному из его собственных адресов IPv4 или адресу связанной с ним multicast-группы и поле протокола имеет значение 41, это говорит о том, что пакет может относиться к туннелю и нужно проверить его принадлежность к одному из настроенных туннельных интерфейсов (путем просмотра адресов отправителя и получателя), собрать фрагменты (если использовалась фрагментация IPv4), удалить заголовок IPv4 и передать полученную в результате дейтаграмму IPv6 на уровень IPv6 этого узла.

Декапсулятор **должен** убедиться в корректности адреса источника туннеля до начала обработки пакета, чтобы предотвратить проблемы, связанные с подменой адресов (см. раздел 4). Такая проверка выполняется также для пакетов, доставленных транспортным протоколам на декапсуляторе. Проверка выполняется путем сравнения адреса отправителя IPv4 с адресом инкапсулятора, заданном на декапсуляторе. Пакеты, для которых адреса не соответствуют, **должны** отбрасываться; сообщения ICMP при этом генерировать **не следует**. Однако, если реализация обычно передает сообщения ICMP при получении пакетов неизвестных протоколов, она **может** передать сообщение (например, ICMPv4 Protocol 41 Unreachable).

Побочным эффектом такой проверки адресов является отбрасывание узлом (без уведомления) пакетов с некорректным адресом отправителя и пакетов, которые были получены узлом но не адресованы ему напрямую (например, широковещательных пакетов).

Независимо от других форм входной фильтрации IPv4, которые администратор может задать, реализация **может** выполнять фильтрацию входящих пакетов (т. е., проверять, что пакеты, приходят с интерфейса в направлении маршрута пересылки к конечной точке туннеля, подобно проверке RPF⁴ [RFC3704]). Поскольку такая проверка может

¹Недоступен.

²Адрес недоступен.

³More Fragments - есть еще фрагменты.

⁴Strict Reverse Path Forwarding - строгая проверка обратного пути пересылки.

создавать проблемы в туннелях, маршрутизируемых по нескольким каналам, **рекомендуется** по умолчанию отключать эту проверку. Пакеты, не прошедшие проверку, **следует** отбрасывать; генерировать сообщения ICMP при этом **не следует**.

Декапсулятор **должен** быть способен поддерживать на туннельных интерфейсах значение IPv6 MRU не менее 1500 байтов и не менее самого большого из значений MTU (IPv6) на этом декапсуляторе.

Декапсулятор **должен** быть способен собирать фрагменты IPv4 пакетов размером (после сборки) до 1500 байтов и не менее наибольшего значения MTU (IPv4) для интерфейсов декапсулятора. Значение 1500 байтов обусловлено результатом инкапсуляции с использованием статического MTU (параграф 3.2.1), тогда, как инкапсуляторы с динамическим выбором (параграф 3.2.2) могут создавать пакеты размер которых определяется наибольшим значением MTU на декапсуляторе (отметим, что это значение строго совпадает с MTU интерфейса последнего маршрутизатора IPv4 перед данным декапсулятором, но для большинства каналов значения MTU одинаковы для всех соседей).

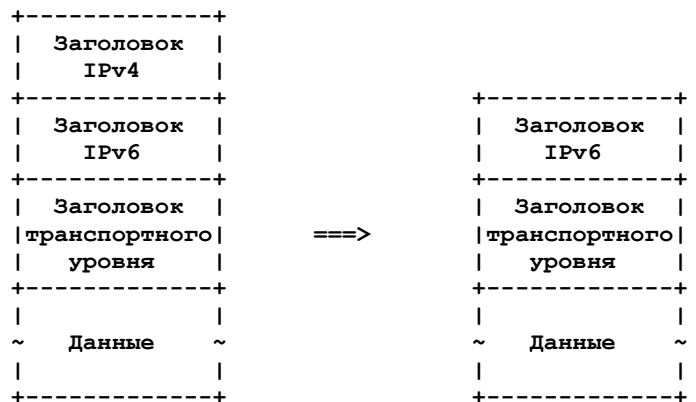
Ограничение на размер собираемых пакетов позволяет инкапсулятору динамически определять значение MTU и использовать преимущества больших значений MTU для пути IPv4. Реализация **может** иметь конфигурационный параметр, позволяющий установить больший размер буфера сборки для туннеля, нежели указано выше, но использовать меньшие размеры буферов **недопустимо**.

Декапсуляция показана на рисунке справа.

Декапсулятор выполняет сборку фрагментов IPv4 перед декапсуляцией пакета IPv6.

При декапсуляции пакета заголовок IPv6 не меняется (см. [RFC2983] и параграф 9.1 [RFC3168] в части поля ToS и туннелирования). Если пакет будет пересылаться дальше, значение hop limit уменьшается на 1.

Инкапсулирующий заголовок IPv4 отбрасывается и корректность полученного пакета проверяется при передаче уровню IPv6. При реконструкции пакета IPv6 размер **должен** определяться по значению поля IPv6 payload length, поскольку в пакете IPv4 могло использоваться заполнение (т. е., после отбрасывания заголовка остаток может превышать размер пакета IPv6).



Декапсуляция IPv6 из IPv4

После декапсуляции узел **должен** без уведомления отбрасывать пакеты с некорректным адресом отправителя IPv6. В список некорректных адресов отправителей **следует** включать по крайней мере:

- все групповые адреса (FF00::/8);
- все loopback-адреса (::1);
- все совместимые с IPv4 адреса IPv6 [RFC3513] (::/96), исключая незаданный адрес для детектирования дубликатов¹ (::/128);
- все отображаемые на IPv4 адреса IPv6 (::ffff:0:0/96).

В дополнение к этому на узле следует настроить входную фильтрацию [RFC2827][RFC3704] по адресам отправителей IPv6, подобным адресам любого из интерфейсов узла. Например,

- 1) если туннель направлен в Internet, узел следует настроить на проверку того, что его префиксы не используются в адресах отправителей;
- 2) если туннель направлен в периферийную сеть, на узле следует настроить проверку того, что адреса отправителей относятся к данной периферийной сети.

В первом случае список префиксов обычно требуется задавать вручную, а во втором проверка может выполняться автоматически (например, с использованием строгой проверки RPF), если интерфейс обозначен, как работающий в направлении периферийной сети.

Рекомендуется обеспечивать администратору возможность управления строгой фильтрацией на входе в направлении периферийных сетей с помощью одной конфигурационной опции.

3.7. Адреса Link-Local

Настраиваемые туннели имеют интерфейсы IPv6 (через «канальный уровень» IPv4) и поэтому **должны** иметь адреса link-local. Эти адреса используются, например, протоколами маршрутизации, работающими через туннели.

Идентификатор интерфейса [RFC3513] в таких случаях может быть основан на 32-битовом адресе IPv4 соответствующего интерфейса или сформирован иным способом, обеспечивающим разумную вероятность уникальности идентификаторов на разных концах туннеля.

Отметим, что может оказаться желательным формирование адреса link-local так, чтобы минимизировать вероятность и влияние смены этого адреса при изменении топологии или замене оборудования.

Если для формирования адреса IPv6 link-local используется адрес IPv4, идентификатором интерфейса будет адрес IPv4, дополненный слева нулями (prepend). Отметим, что бит Universal/Local имеет значение 0, показывающее, что идентификатор интерфейса не является уникальным в глобальном масштабе. Адрес link-local формируется путем добавления идентификатора интерфейса в конец адресного префикса FE80::/64.

¹The unspecified address for Duplicate Address Detection.

Когда хост имеет более одного адреса IPv4 на рассматриваемом физическом интерфейсе, выбор одного из этих адресов IPv4 для формирования адреса link-local осуществляется администратором или разработчиком.

FE	80	00	00	00	00	00	00	00
00	00	00	00					Адрес IPv4

3.8. Обнаружение соседей через туннели

Реализации с настраиваемыми туннелями **должны** по крайней мере воспринимать (и отвечать на них) пакеты зондирования, используемые для определения недоступности соседа (NUD¹) [RFC2461]. Реализациям **следует** также передавать тестовые пакеты NUD для обнаружения отказов в сконфигурированных туннелях, чтобы можно было перейти на использование другого пути к адресату. Отметим, что механизм Neighbor Discovery позволяет отказаться от передачи пакетов NUD на каналах между маршрутизаторами, если протокол маршрутизации отслеживает доступность в обоих направлениях.

Предполагается, что настраиваемые туннели **не** имеют адресов link-layer для обнаружения соседей даже при наличии адреса link-layer (IPv4). Это означает, что:

- отправителям пакетов Neighbor Discovery **не следует** включать опцию Source Link Layer Address или Target Link Layer Address на туннельном канале;
- получатель **должен** в любом случае обрабатывать пакет Neighbor Discovery, отбрасывая без уведомления опции Source Link Layer Address и Target Link Layer Address, полученные в туннельном канале.

Отказ от использования опций адресов link-layer согласуется с практикой использования механизма Neighbor Discovery на других каналах «точка-точка».

4. Угрозы, связанные с подменой адресов отправителей

Приведенная выше спецификация включает правила проверки адресов отправителя для туннелей (в частности входная фильтрация [RFC2827][RFC3704]), выполняемой в общем случае до декапсуляции пакетов. При использовании туннелей IP-in-IP (независимо от версии IP) важно, чтобы это не использовалось для обхода любых входных фильтров, применяемых для пакетов, не относящихся к туннелям. Таким образом, приведенные в этом документе правила построены таким образом, чтобы использование входной фильтрации для IPv4 и IPv6 не давало простого пути обхода фильтров.

В этом случае без специфической входной фильтрации на декапсуляторе атакующий может использовать вставку пакетов со следующими адресами:

- внешний адрес отправителя IPv4 - реальный адрес IPv4 атакующего;
- внешний адрес получателя IPv4 - адрес IPv4 декапсулятора;
- внутренний адрес отправителя IPv6 - Alice (декапсулятор или узел вблизи его);
- внутренний адрес получателя IPv6 - Bob.

Даже при реализации входных фильтров (IPv4) на всех маршрутизаторах IPv4 между атакующим и декапсулятором, а также входных фильтров (IPv6) на всех маршрутизаторах IPv6 между декапсулятором и хостом Bob указанные выше пакеты не будут отфильтрованы. В результате Bob будет получать пакеты, которые похожи на пакеты от хоста Alice, хотя реальный отправитель является совсем другим.

Решением этой проблемы будет восприятие инкапсулятором исключительно тех пакетов, адреса отправителей которых явно указаны в числе других сторон туннелей, как описано в параграфе 3.6. Хотя и это решение не обеспечивает полной защиты в случаях отсутствия входных фильтров, оно, тем не менее, существенно повышает уровень безопасности. Более подробно проблемы угроз рассмотрены в следующем разделе документа.

5. Вопросы безопасности

Базовые вопросы безопасности при использовании протокола IPv6 рассмотрены в работе [V6SEC].

Разработчики туннельных решений должны осознавать, что туннель, хоть и является каналом (как определено в [RFC2460]), модель угроз для туннеля может существенно отличаться от моделей угроз для других типов каналов, поскольку туннель потенциально может включать весь Internet.

Некоторые механизмы (например, Neighbor Discovery) опираются на значение Hop Count = 255 и/или адреса link-local, как некую гарантию происхождения пакета на другой стороне канала в полудоверенной среде. Туннели более уязвимы к нарушению таких допущений, нежели физические каналы, поскольку атакующий из любой точки Internet может отправлять пакеты IPv6-in-IPv4 декапсулятору туннеля, что приведет к попаданию подставных инкапсулированных пакетов IPv6 на интерфейс настраиваемого туннеля, если проверки при декапсуляции не способны обнаруживать такие подставные пакеты.

По этой причине в данном документе указывается, что декапсуляторы выполняют перечисленные ниже проверки (см. также параграф 3.6) для снижения уровня угроз:

- адрес отправителя IPv4 в пакет **должен** совпадать с заданным в конфигурации адресом другой стороны туннеля;
- независимо от имеющейся входной фильтрации IPv4 администратор может настроить, а реализация **может** выполнять входную фильтрацию IPv4 для проверки получения пакетов IPv4 с ожидаемого интерфейса (по причине возможных проблем такая проверка по умолчанию может быть отключена);

¹Neighbor Unreachability Detection.

- пакеты IPv6 с некоторыми обычно некорректными адресами отправителя IPv6 **должны** отбрасываться (см. параграф 3.6);
- следует выполнять входную фильтрацию IPv6 (обычно требует настройки со стороны оператора) для проверки получения пакетов IPv6 с ожидаемого интерфейса.

Первая из проверок особо важна. Для обхода этой проверки атакующему потребуется узнать адрес другой стороны туннеля (это весьма сложно) и суметь подставить его в пакеты (достаточно просто).

Если остаточные угрозы после проверки адресов отправителей в туннеле представляются значимыми, следует использовать туннелирование с аутентификацией, например, IPsec [RFC2401] (предпочтительно) или GRE¹ с заранее созданным секретным ключом [RFC2890]. Поскольку настраиваемые туннели организуются в той или иной степени вручную, организация ключей не должна создать существенных проблем. Организация защищенных IPsec туннелей IPv6-in-IPv4 описана в отдельном документе [V64IPSEC].

Если туннелирование осуществляется в рамках одного административного домена, подобающая входная фильтрация на краю домена может избавить от внешних угроз. Следовательно, короткие туннели являются более предпочтительными, чем длинные, которые могут проходить через Internet.

В дополнение к сказанному реализация **должна** трактовать интерфейсы к разным каналам, как отдельные (например, обеспечить, чтобы пакеты Neighbor Discovery, прибывающие по одному каналу, не оказывали влияния на другие каналы). Это особенно важно для туннельных каналов.

При отбрасывании пакетов, не прошедших проверку по адресам отправителя IPv4 для туннеля узлу не следует «подтверждать» существование туннеля, поскольку эта информация может использоваться для проверки доступности адресов конечных точек туннелей. По этой причине в данной спецификации сказано, что такие пакеты **должны** отбрасываться, а сообщения ICMP об ошибках генерировать **не следует**, если реализация обычно не передает сообщений ICMP о недоступности получателя для неизвестных протоколов (в этом случае **можно** передавать тот же код). Как должно быть очевидно, возврат другого кода ICMP может показать, что включен стек IPv6 (или обработка туннелей для протокола 41). Поведение реализации должно быть согласованным, иначе она становится прозрачной для зондирования.

6. Благодарности

Авторы благодарят членов рабочих групп IPv6, ngtrans² и v6ops за их предложения и рецензирование этого документа. Особо следует отметить (в алфавитном порядке) Jim Bound, Ross Callon, Tim Chown, Alex Conta, Bob Hinden, Bill Manning, John Moy, Mohan Parthasarathy, Chirayu Patel, Pekka Savola и Fred Templin за множество полезных предложений. Pekka Savola помог при редактировании окончательной версии спецификации.

7. Литература

7.1. Нормативные документы

- [RFC791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", [RFC 1191](#), November 1990.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

7.2. Дополнительная литература

- [ASSIGNED] IANA, "Assigned numbers online database", <http://www.iana.org/numbers.html>
- [DNSOPV6] Durand, A., Ihren, J., and Savola P., "Operational Considerations and Issues with IPv6 DNS", Work in Progress³, October 2004.
- [KM97] Kent, C., and J. Mogul, "Fragmentation Considered Harmful". In Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology. August 1987.
- [V6SEC] Savola, P., "IPv6 Transition/Co-existence Security Considerations", Work in Progress⁴, October 2004.
- [V64IPSEC] Graveman, R., et al., "Using IPsec to Secure IPv6-over-IPv4 Tunnels", Work in Progress⁵, December 2004.
- [RFC1435] Knowles, S., "IESG Advice from Experience with Path MTU Discovery", RFC 1435, March 1993.
- [RFC1812] Baker, F., "Requirements for IP Version 4 Routers", [RFC 1812](#), June 1995.
- [RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401⁶, November 1998.
- [RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

¹Generic Routing Encapsulation - базовая инкапсуляция маршрутных данных.

²Next Generation Transition - переход к следующему поколению (IP).

³Работа завершена и опубликована в RFC 4472. Прим. перев.

⁴Работа завершена и опубликована в RFC 4942. Прим. перев.

⁵Работа завершена и опубликована в RFC 4891. Прим. перев.

⁶Документ признан устаревшим и заменен [RFC 4301](#). Прим. перев.

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, [RFC 2827](#), May 2000.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.
- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.
- [RFC3232] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", [RFC 3232](#), January 2002.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.
- [RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, [RFC 3704](#), March 2004.
- [RFC4087] Thaler, D., "IP Tunnel MIB", RFC 4087, June 2005.

8. Отличия от RFC 2893

Основной причиной внесения большого числа изменений послужило стремление упростить документ, сохранив в нем только широко используемые механизмы.

RFC 2893 описывает механизм автоматического туннелирования. Однако в RFC 3056 [RFC3056] описан механизм более общего назначения, который дает каждому узлу с (глобальным) адресом IPv4 префикс /48 IPv6, которого достаточно для всего сайта.

Ниже перечислены отличия от RFC 2893:

- удалены ссылки на записи A6 и сохранены ссылки на AAAA;
- удалено автоматическое туннелирование и совместимые с IPv4 адреса;
- удален используемый по умолчанию настраиваемый туннель с адресом IPv4 Anycast Address;
- удален раздел Source Address Selection¹, поскольку он был включен в другой документ ([RFC3484]);
- удалено упоминание 6over4;
- список литературы разделен на две части (нормативные документы и дополнительная литература);
- удалены слова «or equal» в выражении «if (IPv4 path MTU - 20) is less than or equal to 1280»²;
- Удален текст: «However, IPv6 may be used in some environments where interoperability with IPv4 is not required. IPv6 nodes that are designed to be used in such environments need not use or even implement these mechanisms.»³
- Раздельно описаны классы со статическим (Static MTU) и динамическим (Dynamic MTU) определением MTU; указано, что динамический механизм является **необязательным**, но при его реализации следует выполнять правила параграфа 3.2.2;
- Указано, что по умолчанию статическое значение MTU лежит в диапазоне от 1280 до 1480 байтов и может быть настраиваемым; рассмотрено использование больших значений для Static MTU;
- заданы минимальные правила сборки фрагментом IPv4 и IPv6 MRU для повышения уровня интероперабельности и минимизации «черных дыр»;
- явно указаны ссылки на [RFC2983] и [RFC3168] в описании поля ToS;
- исправлена ссылка на реестр Assigned Numbers (online-версия) с указанием RFC⁴ Assigned Numbers is obsolete;
- исправлен текст о входной фильтрации; в частности, указано, что фильтрация применима к пакетам, доставленным транспортным протоколам на декапсуляторе, а также к пересылаемым декапсулятором пакетам, а также указано, как проверка на декапсуляторах помогает при наличии входной фильтрации IPv4 и IPv6;
- удалено одностороннее туннелирование; предполагается, что все туннели являются двухсторонними и организуются между адресами конечных точек (а не узлами);
- удалены рекомендации по анонсированию адресов в DNS, поскольку они не относятся к данной спецификации, и даны ссылки на соответствующие документы;

¹Выбор адреса отправителя.

²< вместо ≤. Прим. перев.

³Однако IPv6 может использоваться в некоторых средах, где интероперабельность с IPv4 не требуется. Узлам IPv6, предназначенным для таких сред, не требуется использовать и даже поддерживать этот механизм.

⁴RFC3232. Прим. перев.

- удалено требование **следует** (SHOULD) для формирования адресов link-local на базе адресов IPv4;
- добавлено требование **следует** для реализации опции установки адреса отправителя в туннеле, а также обсуждена польза этой опции;
- добавлено более строгое описание проверки адреса отправителя - оба адреса (IPv4 и IPv6) **должны** проверяться, а фильтрация типа RPF является опциональной;
- переписан раздел «Вопросы безопасности» с уточнением угроз при туннелировании;
- добавлено примечание об использовании TTL=255 при инкапсуляции;
- в параграфе 3.2 расширено обсуждение использования «бесконечного» значения IPv6 MTU, явно ведущего к проблемам интероперабельности;
- добавлено явное требование при использовании обоих методов определения MTU выбирать один метод для каждого канала независимо;
- отмечено, что обработка сообщений ICMPv4 об ошибках применима только при динамическом определении MTU;
- разъяснено описание фильтрации записей DNS; API **следует** использовать и при его отсутствии фильтрация **недопустима**; упорядочение выходит за пределы спецификации и описано в RFC3484;
- отмечено, что адрес получателя IPv4 может быть групповым;
- **рекомендовано** обеспечивать опцию включения строгой фильтрации на входе для каждого интерфейса;
- обобщен текст о данных в сообщениях ICMPv4;
- внесено множество редакционных правок.

Адреса авторов

Erik Nordmark

Sun Microsystems

17 Network Circle

Menlo Park, CA 94025

USA

Phone: +1 650 786 2921

EMail: erik.nordmark@sun.com

Robert E. Gilligan

Intransa, Inc.

2870 Zanker Rd., Suite 100

San Jose, CA 95134 USA

Phone : +1 408 678 8600

Fax : +1 408 678 8800

EMail: bob.gilligan@acm.org

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

Полное заявление авторских прав

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Интеллектуальная собственность

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license

under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.