Network Working Group C. Perkins
Request for Comment: 2003 IBM
Category: Standards Track October 1996

Инкапсуляция IP в IP

IP Encapsulation within IP

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Тезисы

В этом документе описан метод, с помощью которого дейтаграмма IP может быть инкапсулирована (передана в качестве данных) в другую дейтаграмму IP. Инкапсуляция предложена, как способ изменения обычной маршрутизации IP для дейтаграмм путем их доставки на промежуточный узел, который иначе не может быть выбран на основе (сетевой части) адреса получателя в исходном заголовке IP. Инкапсуляция может применяться для решения разных задач (таких, как доставка дейтаграмм мобильному узлу с использованием Mobile IP).

1. Введение

В этом документе описан метод, с помощью которого дейтаграмма IP может быть инкапсулирована (передана в качестве данных) в другую дейтаграмму IP. Инкапсуляция предложена, как способ изменения обычной маршрутизации IP для дейтаграмм путем их доставки на промежуточный узел, который иначе не может быть выбран на основе (сетевой части) адреса получателя в исходном заголовке IP. После доставки дейтаграммы на промежуточный узел она декапсулируется с восстановлением исходной дейтаграммы IP, которая доставляется адресату, указанному в исходном поле Destination Address. Такое использование инкапсуляции и декапсуляции часто называют «туннелированием» дейтаграмм, а инкапсулятор и декапсулятор - конечными точками туннеля.

В общем случае схема туннелирования имеет вид:

```
отправитель ---> инкапсулятор --------> декапсулятор ---> получатель
```

Отправитель, инкапсулятор, декапсулятор и получатель представляют собой отдельные узлы. Узел инкапсуляции рассматривается, как точка входа в туннель, а узел декапсуляции - как точка выхода из туннеля. В общем случае множество пар отправитель-получатель могут пользоваться одним туннелем между инкапсулятором и декапсулятором.

2. Мотивация

Рабочая группа Mobile IP подготовила спецификацию использования инкапсуляции в качестве способа доставки дейтаграмм из домашней сети мобильного узла агенту, который может доставлять дейтаграммы мобильному узлу в его текущее местоположение за пределами домашней сети [8]. Использование инкапсуляции может также оказаться желательным в тех случаях, когда отправитель дейтаграммы (или промежуточный маршрутизатор) должен влиять на выбор маршрута, по которому дейтаграмма будет доставляться конечному получателю. Другими возможными применениями инкапсуляции являются групповая передача (multicasting), доставка по более дешевому пути, выбор маршрутов с нужными атрибутами безопасности, а также маршрутизация на основе правил.

В общем случае инкапсуляция и опция IP LSR[1] [10] могут оказывать идентичное влияние на маршрутизацию дейтаграммы, но есть несколько причин, по которым инкапсуляция более предпочтительна:

- существуют проблемы безопасности, связанные с использованием опций IP для выбора маршрута;
- современные маршрутизаторы Internet не обеспечивают требуемой производительности при пересылке дейтаграмм с опциями выбора маршрута;
- многие узлы в Internet некорректно обрабатывают опции выбора маршрута;
- межсетевые экраны могут блокировать дейтаграммы IP с заданным отправителем маршрутом;
- вставка опции выбора маршрута может усложнять обработку идентификационной информации отправителем и/или получателем дейтаграммы в зависимости от выбранного способа идентификации;
- изменение дейтаграмм промежуточными маршрутизаторами считается «дурным тоном».

Эти технические преимущества должны сравниваться с недостатками, обусловленными применением инкапсуляции:

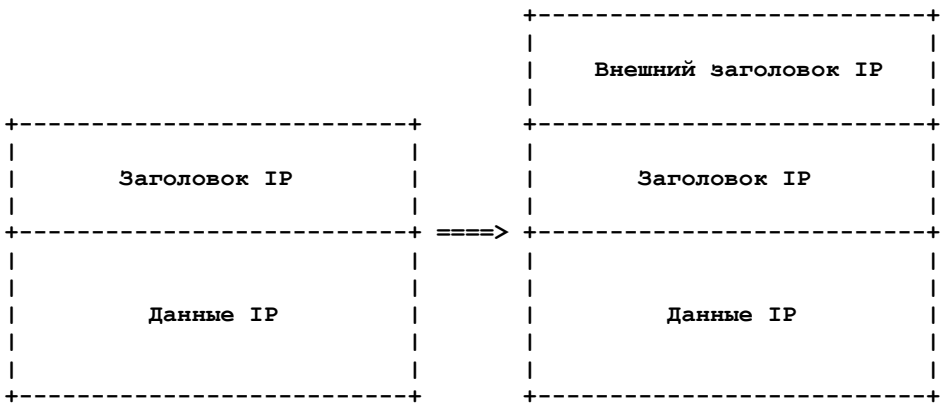
- инкапсулированные дейтаграммы обычно больше дейтаграмм с опциями маршрутизации;
- инкапсуляцию невозможно использовать, если заранее не известно, что узел на другой стороне туннеля может декапсулировать дейтаграмму.

[1] Loose source routing - нестрогое задание маршрута отправителем.

Поскольку большинство современных узлов Internet не обеспечивает качественной обработки опции IP LSR, второй из отмеченных недостатков инкапсуляции не представляется столь же существенным, как первый.

3. Инкапсуляция IP в IP

Для инкапсуляции дейтаграммы IP с использованием схемы IP-in-IP перед существующим заголовком IP добавляется внешний заголовок [10], как показано на рисунке.



Поля Source Address и Destination Address внешнего заголовка идентифцируют конечные точки туннеля. Поля Source Address и Destination Addresses во внутреннем заголовке указывают исходного отправителя и конечного получателя, соответственно. Внутренний заголовок IP не изменяется инкапсулятором за исключением уменьшения на 1 значения поля TTL, как отмечено ниже. Это поле остается неизменным при доставке до конечной точки туннеля. При доставке инкапсулированной дейтаграммы через туннель опции внутреннего заголовка IP не меняются. При необходимости между внешним и внутренним заголовками IP могут помещаться заголовки других протоколов типа IP AH[1] [1]. Отметим, что опции защиты из внутреннего заголовка IP **могут** влиять на выбор опций защиты для инкапсулирующего (внешнего) заголовка IP.

3.1. Поля заголовка IP и их обработка

Ниже перечислены поля внешнего заголовка IP, устанавливаемые инкапсулятором.

Version - версия

4

IHL

Поле IHL[2] указывает размер внешнего заголовка IP в 32-битовых словах [10].

TOS

Поле TOS[3] копируется из внутреннего заголовка IP.

Total Length - общий размер

Это поле показывает полный размер дейтаграммы с учетом внешнего и внутреннего заголовков IP, а также данных.

Identification, Flags, Fragment Offset — идентификация, флаги, смещение фрагмента

Эти три поля устанавливаются, как указано в стандарте [10]. Однако при наличии во внутреннем заголовке флага DF[4], этот флаг **должен** устанавливаться и во внешнем заголовке. Если этот флаг во внутреннем заголовке сброшен, он **может** быть установлен во внешнем заголовке, как описано в параграфе 5.1.

Time to Live - время жизни

Поле TTL[5] во внешнем заголовке устанавливается в соответствии с условиями доставки инкапсулированной дейтаграммы до выходной точки туннеля.

Protocol - протокол

4

Header Checksum - контрольная сумма заголовка

Контрольная сумма внешнего заголовка, вычисленная в соответствии со стандартом [10].

Source Address - адрес отправителя

IP-адрес инкапсулятора (входной точки туннеля).

Destination Address - адрес получателя

IP-адрес декапсулятора (выходной точки туннеля).

Options - опции

Никакие опции из внутреннего заголовка IP в общем случае **не** копируются во внешний заголовок. Однако **могут** добавляться новые опции, относящиеся к туннелю. В частности, любые поддерживаемые опции защиты из внутреннего заголовка **могут** влиять на выбор опций защиты во внешнем заголовке. Прямое отображение опций защиты внутреннего заголовка в опции внешнего заголовка или заголовки защиты для туннеля не предполагается.

При инкапсуляции дейтаграммы значение TTL во внутреннем заголовке IP уменьшается на 1, если туннелирование осуществляется, как часть пересылки дейтаграммы. В противном случае значение TTL во внутреннем заголовке при инкапсуляции не меняется. Если полученное в результате значение TTL во внутреннем заголовке равно 0, дейтаграмма отбрасывается, а отправителю ее **следует** передать сообщение ICMP Time Exceeded. Инкапсулятору **недопустимо** инкапсулировать дейтаграммы с TTL = 0.

Значение TTL во внутреннем заголовке IP не меняется при декапсуляции. Если после декапсуляции во внутреннем заголовке TTL = 0, декапсулятор **должен** отбросить дейтаграмму. Если после декапсуляции дейтаграмма

[1]Authentication header - идентификационный заголовок. Название протокола защиты IP. *Прим. перев.*
[2]Internet Header Length - размер заголовка IP.
[3]Type of Service - тип обслуживания.
[4]Don't Fragment - не фрагментировать.
[5]Time To Live – время жизни.

пересылается через один из сетевых интерфейсов декапсулятора, значение TTL будет уменьшаться в результате обычной пересылки IP. Дополнительная информация о работе с полем TTL приведена в параграфе 4.4.

Инкапсулятор может использовать существующие механизмы IP, подходящие для доставки инкапсулированных данных в выходную точку туннеля. В частности, разрешается использовать опции IP и фрагментацию дейтаграмм, если во внутреннем заголовке не установлен флаг DF. Такое ограничение на использование фрагментации требует от узлов поддержки определения MTU для пути [7].

3.2. Отказы маршрутизации

Маршрутные петли в туннели представляют практическую опасность, когда они заставляют дейтаграммы снова попадать на инкапсулятор. Предположим, что дейтаграмма приходит на маршрутизатор для пересылки и тот определяет, что дейтаграмма была инкапсулирована, еще до ее пересылки. Тогда:

- если Source Address в дейтаграмме соответствует одному из адресов маршрутизатора, для последнего **недопустимо** туннелирование дейтагрраммы и ее **следует** отбросить;
- если Source Address в дейтаграмме соответствует IP-адресу получателя туннеля (точка выхода из туннеля обычно выбирается маршрутизатором на основе значения поля Destination в заголовке IP дейтаграммы), для маршрутизатора **недопустимо** туннелировать дейтаграмму и ее **следует** отбросить.

Дополнительная информация содержится в параграфе 4.4.

4. Сообщения ICMP в туннеле

После передачи инкапсулированной дейтаграммы инкапсулятор может получить сообщение ICMP [9] от любого промежуточного маршрутизатора в туннеле, не являющегося выходом из туннеля. Предпринимаемые инкапсулятором в ответ на такое сообщение меры зависят от типа полученного сообщения ICMP. Когда полученное сообщение содержит достаточно информации, инкапсулятор **может** использовать входящее сообщение для создания похожего сообщения ICMP, которое будет передано отправителю исходной дейтаграммы IP. Этот процесс будем называть трансляцией сообщений ICMP из туннеля.

Сообщения ICMP, говорящие об ошибках при обработке дейтаграммы, включают копию или часть вызвавшей ошибку дейтаграммы. Трансляция сообщения ICMP требует от инкапсулятора вырезать внешний заголовок из полученной копии исходной дейтаграммы. Случай, когда полученное сообщение ICMP не содержит достаточных для его трансляции данных рассматривается в разделе 5.

4.1. Destination Unreachable (тип 3)

Сообщения ICMP Destination Unreachable[1] обрабатываются инкапсулятором в зависимости от кода (поле Code). Предлагаемая здесь модель позволяет с помощью туннелей «расширять» сети для включения в них нелокальных (например, мобильных) узлов. Таким образом, если исходный получатель неинкапсулированной дейтаграммы находится в одной сети с инкапсулятором, некоторые значения кодов Destination Unreachable могут быть изменены в соответствии с предлагаемой моделью.

Network Unreachable[2] (код 0)

Сообщение ICMP Destination Unreachable **следует** возвратить исходному отправителю. Если исходный получатель неинкапсулированной дейтаграммы находится в одной сети с инкапсулятором, создаваемое и передаваемое последним сообщение Destination Unreachable **может** иметь код 1 (Host Unreachable[3]), поскольку дейтаграмма поступила в нужную (корректную) сеть и инкапсулятор пытается создать видимость того, что исходный получатель находится в той же сети, даже если на деле это не так. В остальных случаях, если инкапсулятор возвращает сообщение Destination Unreachable, в нем **должен** быть указан код 0 (сеть недоступна).

Host Unreachable (код 1)

Инкапсулятору **следует** транслировать сообщения Host Unreachable отправителям исходных (неинкапсулированных) дейтаграмм, если это возможно.

Protocol Unreachable[4] (код 2)

Когда инкапсулятор получает сообщение ICMP Protocol Unreachable, ему **следует** передать сообщение Destination Unreachable с кодом 0 или 1 (см. описание для кода 0) отправителю исходной неинкапсулированной дейтаграммы. Поскольку исходный отправитель не использовал протокол 4 в переданной дейтаграмме, бессмысленно возвращать ему код 2.

Port Unreachable[5] (код 3)

Такой код не должен приходить инкапсулятору, поскольку во внешнем заголовке IP не указывается номер порта. **Недопустимо** транслировать такое сообщение отправителю исходной неинкапсулированной дейтаграммы.

Datagram Too Big[6] (код 4)

Инкапсулятор **должен** транслировать сообщения ICMP Datagram Too Big отправителю исходной дейтаграммы.

Source Route Failed[7] (код 5)

Этот код инкапсулятору **следует** обрабатывать самостоятельно. **Недопустимо** транслировать такие сообщения отправителю исходной неинкапсулированной дейтаграммы.

---

[1]Получатель недоступен.
[2]Сеть недоступна.
[3]Хост недоступен.
[4]Протокол недоступен.
[5]Порт недоступен.
[6]Дейтаграмма слишком велика.
[7]Отказ на заданном отправителем маршруте.

4.2. Source Quench[1] (тип 4)

Инкапсулятору **не следует** транслировать сообщения ICMP Source Quench отправителю исходной неинкапсулированной дейтаграммы, а **следует** активировать те или иные поддерживаемые им механизмы контроля насыщения, чтобы помочь в преодолении перегрузки, обнаруженной в туннеле.

4.3. Redirect[2] (тип 5)

Инкапсулятор **может** обрабатывать сообщения ICMP Redirect самостоятельно. **Недопустимо** транслировать сообщения Redirect отправителю исходной неинкапсулированной дейтаграммы.

4.4. Time Exceeded[3] (тип 11)

Сообщения ICMP Time Exceeded говорят о (предполагаемой) маршрутной петле в самом туннеле. При получении сообщения Time Exceeded инкапсулятор **должен** передать отправителю исходной неинкапсулированной дейтаграммы сообщение Host Unreachable (тип 3, код 1). Сообщение Host Unreachable является более предпочтительным, нежели Network Unreachable, поскольку дейтаграмма была обработана инкапсулятором, а он зачастую рассматривается, как находящийся в одной сети с адресатом исходной неинкапсулированной дейтаграммы. Тогда дейтаграмма рассматривается, как направленная в нужную сеть, но не доставленная указанному узлу в этой сети.

4.5. Parameter Problem[4] (тип 12)

Если сообщение Parameter Problem указывает на поле, скопированное из исходной неинкапсулированной дейтаграммы, инкапсулятор **может** транслировать это сообщение ICMP отправителю исходной дейтаграммы. Если проблема связана с опцией IP, добавленной инкапсулятором, ему **недопустимо** транслировать такое сообщение исходному отправителю. Отметим, что в соответствии со сложившейся практикой инкапсуляторам не следует добавлять какие-либо опции IP в инкапсулированные дейтаграммы, за исключением выполнения требований защиты.

4.6. Прочие сообщения ICMP

Остальные сообщения ICMP не связаны с описанной в данной спецификации протокола инкапсуляцией и их следует обрабатывать в соответствии с [9].

5. Управление туннелем

К сожалению протокол ICMP требует от маршрутизаторов IP возврата лишь 8 октетов (64 бита) дейтаграммы в дополнение к заголовку IP. Этого недостаточно для включения копии инкапсулированного (внутреннего) заголовка IP, поэтому инкапсулятор не всегда может транслировать сообщение ICMP от внутренних узлов туннеля отправителю исходного сообщения. Однако за счет аккуратной поддержки информации о состоянии[5] туннеля, в который передаются дейтаграммы, инкапсулятор может во многих случаях возвращать отправителю исходной дейтаграммы нужные сообщения ICMP. Инкапсулятору **следует** поддерживать по крайней мере следующие данные о состоянии каждого туннеля:

- MTU для туннеля (параграф 5.1);
- изменение TTL (длина пути) в туннеле;
- доступность удаленного конца туннеля.

Инкапсулятор использует сообщения ICMP, получаемые от внутренних узлов туннеля, для обновления данных о состоянии этого туннеля. Сообщения ICMP об ошибках, которые могут приходить от маршрутизаторов туннеля, включают:

- Datagram Too Big;
- Time Exceeded;
- Destination Unreachable;
- Source Quench.

При поступлении последующих дейтаграмм для доставки через туннель инкапсулятор проверяет состояние туннеля. Если дейтаграмма не соответствует состоянию туннеля (например, значение TTL в новой дейтаграмме меньше, чем TTL в состоянии туннеля), инкапсулятор передает сообщение ICMP об ошибке отправителю исходной дейтаграммы, но инкапсулирует эту дейтаграмму и пересылает ее в туннель.

При использовании такого метода передаваемые инкапсулятором сообщения ICMP об ошибках не всегда будут точно соответствовать возникающим в туннеле ошибкам, но будут достаточно аккуратно отражать состояние сети.

Метод поддержки информации о состоянии туннеля был предложен в спецификации IPAE[6] [4].

5.1. Определение MTU для туннеля

Когда флаг DF установлен исходным отправителем и копируется во внешний заголовок IP, значение MTU для туннеля можно выяснить из сообщений ICMP Datagram Too Big (тип 3, код 4), передаваемых инкапсулятором. Для поддержки передающих узлов, которые используют механизм Path MTU Discovery, все реализации инкапсуляторов **должны** поддерживать состояние Path MTU Discovery [5, 7] для туннелей. Такие решения обеспечивают ряд преимуществ.

---

[1] Необходимость снизить скорость передачи.
[2] Перенаправление. Говорит о наличии более простого маршрута к адресату. *Прим. перев.*
[3] Время жизни истекло в процессе доставки дейтаграммы.
[4] Проблема с параметрами.
[5] В оригинале используется термин «soft state». *Прим. перев.*
[6] IP Address Encapsulation - инкапсуляция адресов IP.

- Преимущество использования Path MTU Discovery в туннеле выражается в том, что любая фрагментация в результате добавления заголовка инкапсуляции выполняется только один раз после инкапсуляции. Это предотвращает многократную фрагментацию одной дейтаграммы и ведет к росту производительности декапсулятора и маршрутизаторов в туннеле.

- Если отправитель неинкапсулированной дейтаграммы поддерживает Path MTU Discovery, для инкапсулятора желательно знать MTU в туннеле. Все сообщения ICMP Datagram Too Big из туннеля возвращаются инкапсулятору, но, как отмечено в параграфе 5, инкапсулятор не всегда может транслировать сообщения ICMP отправителю исходной неинкапсулированной дейтаграммы. Поддерживая информацию о значении MTU для туннеля, инкапсулятор может возвращать сообщения ICMP Datagram Too Big исходному отправителю неинкапсулированных дейтаграмм для поддержки определения тем значения Path MTU. В этом случае значение MTU, которое передается инкапсулятором исходному отправителю, следует указывать равным значению MTU для туннеля за вычетом размера инкапсулирующего заголовка IP. Это позволяет избежать фрагментации исходных дейтаграмм IP на инкапсуляторе.

- Если отправитель исходной неинкапсулированной дейтаграммы не выполняет Path MTU Discovery, для инкапсулятора все равно желательно знать значение MTU для туннеля. В частности, значительно лучше фрагментировать исходную дейтаграмму при инкапсуляции, нежели допускать фрагментирование инкапсулированной дейтаграммы. Фрагментирование исходной дейтаграммы может быть выполнено инкапсулятором без специальной буферизации и без необходимости поддержки информации о состоянии сборки на декапсуляторе. Если же фрагментировать инкапсулированные дейтаграммы, декапсулятор должен собрать фрагменты (инкапсулированной) дейтаграммы до ее декапсуляции, что требует поддержки информации о состоянии сборки и выделения буферного пространства не декапсуляторе.

Таким образом, инкапсулятору обычно **следует** выполнять операцию Path MTU Discovery, требующую от него передавать в туннель все дейтаграммы с установленным флагом DF во внешнем заголовке IP. Однако в этом случае возникает проблема. Когда флаг DF устанавливает исходный отправитель дейтаграммы, он может быстро отреагировать на любое полученное сообщение ICMP Datagram Too Big, повторно передавая исходную дейтаграмму. Предположим, что инкапсулятор получает из туннеля сообщение ICMP Datagram Too Big. В этом случае, если исходный отправитель не установил для дейтаграммы флаг DF, инкапсулятор ничего не может сделать для того, чтобы исходный отправитель узнал об ошибке. Инкапсулятор **может** сохранить копию переданной дейтаграммы при попытке определения MTU для туннеля, чтобы можно было фрагментировать и повторно передать дейтаграмму в случае получения отклика Datagram Too Big. В другом варианте инкапсулятор **может** не устанавливать бит DF для некоторых типов при отсутствии этого флага в исходной дейтаграмме .

5.2. Насыщение

Инкапсулятор может получать из туннеля сигналы о перегрузке (например, сообщения ICMP Source Quench от узлов внутри туннеля). Кроме того, некоторые протоколы канального уровня и протоколы, не относящиеся к стеку протоколов IP, могут обеспечивать такую сигнализацию в форме флага CE[2] [6]. Инкапсулятору следует отражать перегрузку в поддерживаемом им состоянии туннеля и при пересылке в туннель последующих дейтаграмм инкапсулятору **следует** использовать соответствующие меры по контролю насыщения [3]. Однако инкапсулятору **не следует** передавать сообщений ICMP Source Quench исходному отправителю неинкапсулированных дейтаграмм.

6. Вопросы безопасности

Инкапсуляция IP потенциально снижает уровень безопасности Internet и требуется принятие определенных мер предосторожности при разработке и развертывании инкапсуляции IP. Например, инкапсуляция осложняет фильтрацию дейтаграмм на граничных маршрутизаторах по полям заголовков. В частности, исходные значения полей Source Address, Destination Address и Protocol в заголовке IP, а также номера портов, используемые в любых транспортных заголовках, после инкапуляции дейтаграммы не находятся в своем обычном положении. Поскольку любая дейтаграмма может быть инкапсулирована и передана через туннель, граничным маршрутизаторам нужно аккуратно проверять все дейтаграммы.

6.1. Маршрутизаторы

Маршрутизаторам следует быть внимательными с протоколами инкапсуляции IP для обеспечения корректной фильтрации входящих дейтаграмм. Желательно интегрировать такую фильтрацию с идентификацией IP [1]. При использовании идентификации IP инкапсулированным пакетам может разрешаться вход в сеть организации при условии, что инкапсулирующий (внешний) или инкапсулированный (внутренний) пакет передан идентифицированным, доверенным отправителем. Инкапсулированные пакеты, не содержащие такой идентифкации связаны с потенциально высокой угрозой безопасности.

Инкапсулированные и зашифрованные [2] дейтаграммы также могут создавать проблемы для фильтрующих маршрутизаторов. В этом случае маршрутизатор способен фильтровать дейтаграммы лишь при условии доступа к защищенной связи, используемой для шифрования. Чтобы разрешить такой тип шифрования в средах, где для всех пакетов требуется фильтрация (или хотя бы учет), нужен механизм, который позволит приемному узлу организовать защищенную связь с граничным маршутизатором. Подобная ситуация (хотя и более редко) может возникать и для защищенных связей исходящего направления.

6.2. Хосты

Реализациям хостов, поддерживающим прием инкапсулированных дейтаграмм IP, **следует** принимать лишь дейтаграммы, относящиеся к одной или нескольким из перечисленных ниже категорий.

- Безопасный протокол, для которого не требуется идентификация по адресам.

- Инкапсулирующая (внешняя) дейтаграмма получена от надежно идентифицированного доверенного отправителя. Аутентичность источника может поддерживаться с использованием методов физической защиты

[2]Congestion Experienced - наблюдается насыщение.

в дополнение к настройке граничных маршрутизаторов, но более очевидным представляется использование протокола IP AH [1].

- Инкапсулированная (внутренняя) дейтаграмма включает идентификационный заголовок IP AH.
- Инкапсулированная (внутренняя) дейтаграмма адресована сетевому интерфейсу декапсулятора или узла, с которым у декапсулятора есть специальное соглашение по доставке таких дейтаграмм.

Часть или все из перечисленных выше проверок могут выполняться на граничных маршрутизаторах, а не на приемных узлах, но лучше использовать в этом смысле граничные маршрутизаторы в качестве избыточной, нежели единственной точки контроля.

7. Благодарности

Фрагменты разделов 3 и 5 данного документа заимствованы (с разрешения Bill Simpson) из ранних версий черновых вариантов Mobile IP [8]. Текст раздела 6 (Вопросы безопасности) подготовил Bob Smart. Хорошие идеи, заимствованные из RFC 1853 [11], предложил Bill Simpson. Благодарим также Anders Klemets за обнаруженные ошибки и помощь в переработке черновых вариантов документа. И, наконец, спасибо David Johnson за прочесывание черновика «частым гребнем», нахождение ошибок, устранение несоответствий и внесение множества других улучшений в документ.

Литература

[1] Atkinson, R., "IP Authentication Header", RFC 1826[1], August 1995.

[2] Atkinson, R., "IP Encapsulating Security Payload", RFC 1827[2], August 1995.

[3] Baker, F., Editor, "Requirements for IP Version 4 Routers", RFC 1812[3], June 1995.

[4] Gilligan, R., Nordmark, E., and B. Hinden, "IPAE: The SIPP Interoperability and Transition Mechanism", Work in Progress.

[5] Knowles, S., "IESG Advice from Experience with Path MTU Discovery", RFC 1435, March 1993.

[6] Mankin, A., and K. Ramakrishnan, "Gateway Congestion Control Survey", RFC 1254, August 1991.

[7] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191[3], November 1990.

[8] Perkins, C., Editor, "IP Mobility Support", RFC 2002, October 1996.

[9] Postel, J., Editor, "Internet Control Message Protocol", STD 5, RFC 792[3], September 1981.

[10] Postel, J., Editor, "Internet Protocol", STD 5, RFC 791[3], September 1981.

[11] Simpson, W., "IP in IP Tunneling", RFC 1853, October 1995.

Адреса авторов

Вопросы, связанные с этим документом, следует адресовать:

Charles Perkins

Room H3-D34

T. J. Watson Research Center

IBM Corporation

30 Saw Mill River Rd.

Hawthorne, NY 10532

Work: +1-914-784-7350

Fax: +1-914-784-6205

EMail: perk@watson.ibm.com

С рабочей группой можно связаться через ее текущего руководителя:

Jim Solomon

Motorola, Inc.

1301 E. Algonquin Rd.

Schaumburg, IL 60196

Work: +1-847-576-2753

EMail: solomon@comm.mot.com

Перевод на русский язык
Николай Малых

[1]Этот документ утратил силу и в настоящее время вместо него действует RFC 4302, перевод которого имеется на сайте www.protocols.ru. *Прим. перев.*
[2]Этот документ утратил силу и в настоящее время вместо него действует RFC 4303, перевод которого имеется на сайте www.protocols.ru. *Прим. перев.*
[3]Перевод этого документа имеется на сайте www.protocols.ru. *Прим. перев.*

nmalykh@protocols.ru