РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 363 108** (13) **C2**

(51) МПК
H04L 12/56 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2007103166/09, 29.06.2005**

(24) Дата начала отсчета срока действия патента:
29.06.2005

(30) Конвенционный приоритет:
29.06.2004 US 10/881,451

(43) Дата публикации заявки: **10.08.2008**

(45) Опубликовано: **27.07.2009** Бюл. № **21**

(56) Список документов, цитированных в отчете о поиске: **US 2003126272 A1, 03.07.2003. RU 2204220 C2, 10.05.2003. JP 2002300219 A, 11.10.2002. EP 0743777 A1, 20.11.1996.**

(85) Дата перевода заявки PCT на национальную фазу: **29.01.2007**

(86) Заявка PCT:
US 2005/023660 (29.06.2005)

(87) Публикация PCT:
WO 2006/005029 (12.01.2006)

Адрес для переписки:
129090, Москва, ул. Б.Спасская, 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595

(72) Автор(ы):
БАББАР Уппиндер С. (US),
ЛИОЙ Марчелло (US)

(73) Патентообладатель(и):
КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

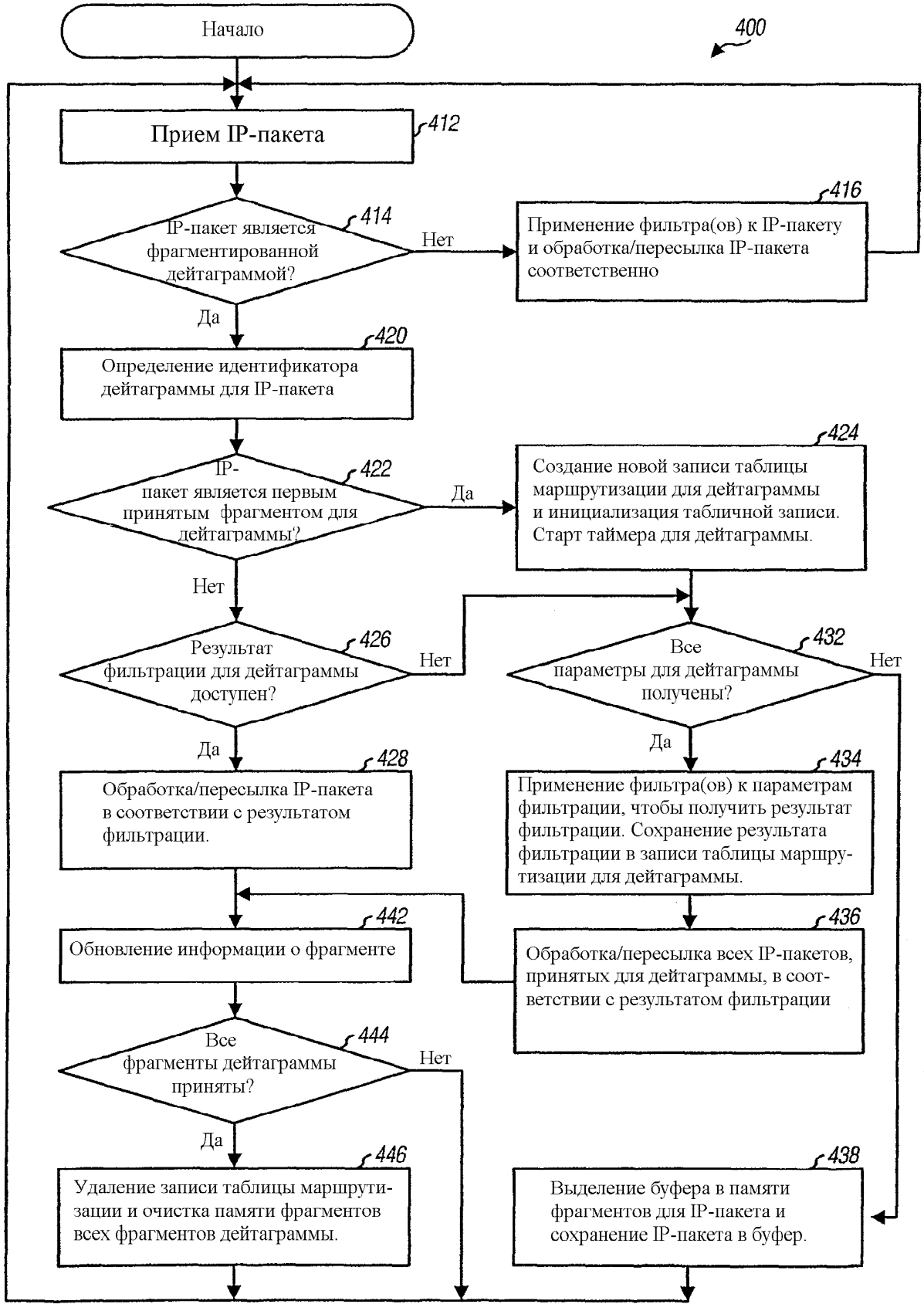
RU 2363108 C2

(54) ФИЛЬТРАЦИЯ И МАРШРУТИЗАЦИЯ ФРАГМЕНТИРОВАННЫХ ДЕЙТАГРАММ В СЕТИ ПЕРЕДАЧИ ДАННЫХ

(57) Реферат:

Изобретение относится к области сетей передачи данных. Технический результат заключается в уменьшении требуемого объема обрабатываемой информации, а также нагрузки на канал связи. Сущность изобретения заключается в том, что для каждой фрагментированной дейтаграммы узел фильтрации получает параметры фильтрации по мере приема фрагментов дейтаграммы. Когда все параметры фильтрации доступны, узел применяет один или более фильтров в отношении параметров фильтрации, чтобы получить результат фильтрации для дейтаграммы, и сохраняет результат фильтрации в записи в таблице маршрутизации. Прежде чем получить результат фильтрации, узел сохраняет все фрагменты, принятые для дейтаграммы, в памяти. Когда результат фильтрации становится доступным, узел обрабатывает все фрагменты, уже принятые для дейтаграммы, в соответствии с результатом фильтрации. По мере приема каждого оставшегося фрагмента для

дейтаграммы узел немедленно обрабатывает фрагмент в соответствии с результатом фильтрации. Когда принимается последний фрагмент, узел очищает память и запись таблицы маршрутизации для дейтаграммы. 5 н. и 26 з.п. ф-лы, 8 ил.



Фиг. 4

RU 2363108 C2

RUSSIAN FEDERATION

(19) **RU** (11) **2 363 108** (13) **C2**

(51) Int. Cl.
H04L 12/56 (2006.01)

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2007103166/09, 29.06.2005**

(24) Effective date for property rights:
29.06.2005

(30) Priority:
29.06.2004 US 10/881,451

(43) Application published: **10.08.2008**

(45) Date of publication: **27.07.2009 Bull. 21**

(85) Commencement of national phase: **29.01.2007**

(86) PCT application:
US 2005/023660 (29.06.2005)

(87) PCT publication:
WO 2006/005029 (12.01.2006)

Mail address:
129090, Moskva, ul. B.Spasskaja, 25, str.3, OOO "Juridicheskaja firma Gorodisskij i Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595

(72) Inventor(s):
BABBAR Uppinder S. (US),
LIOJ Marchello (US)

(73) Proprietor(s):
KVEhLKOMM INKORPOREJTED (US)

RU 2363108 C2

(54) FILTRATION AND ROUTING OF FRAGMENTED DATAGRAMS IN DATA TRANSFER NETWORK

(57) Abstract:

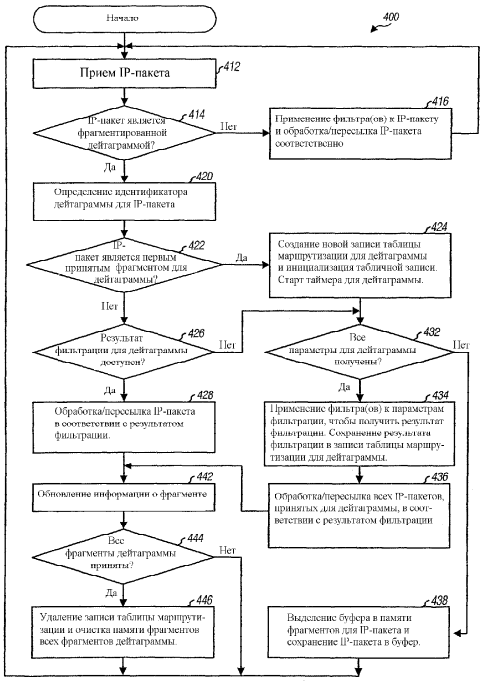
FIELD: communication devices.

SUBSTANCE: invention relates to data transfer networks. For each fragmented datagram the filter unit obtains filtering parametres of at least receiving datagram fragments. When all filtering parametres are accessible, the unit uses one or more filters with respect to filtering parametres, so as obtain a filtering result for the datagram and records the filtering result in a routing table. Before obtaining the filtering result, the unit stores all fragments received for the datagram in memory. When the filtering result becomes accessible, the unit processes all fragments already received for the datagram, in accordance with the filtering result. While receiving each remaining fragment for the datagram, the unit immediately processes the fragment in accordance with the filtering result. When the last fragment is received, the unit clears the memory and entries in the routing table for the datagram.

EFFECT: reduction of required volume of processed information, as well as load on a communication channel.

31 cl, 8 dwg

RU 2363108 C2



Фиг. *4*

RU 2363108 C2

RU 2363108 C2

Область техники, к которой относится изобретение

Настоящее изобретение относится в целом к передаче данных, а более конкретно к технологиям фильтрации и маршрутизации фрагментированных дейтаграмм в сети передачи данных.

Предшествующий уровень техники

Интернет-протокол (IP) является протоколом, который поддерживает передачу блоков данных, называемых дейтаграммами, от источников к получателям в сети передачи данных с коммутацией пакетов. Источники и получатели являются хост-компьютерами, которые идентифицируются посредством IP-адресов фиксированной длины. В терминологии IP “узел” - это устройство, которое реализует IP, “хост-компьютер” - это узел, который является оконечным пунктом IP-пакетов, явно адресованным ему, а “маршрутизатор” - это хост-компьютер, который также пересылает IP-пакеты, явно не адресованные ему. Чтобы передать данные получателю, источник формирует дейтаграмму с IP-заголовком и частью, соответствующей полезной нагрузке. IP-заголовок содержит IP-адреса источника и получателя, так же как и другие поля. Источник затем отправляет дейтаграмму как IP-пакет по направлению к получателю на основе информации о маршрутизации, которую источник имеет для получателя.

Узел может фильтровать IP-пакеты по разным причинам, как описано ниже. В контексте IP “фильтрация” - это процесс, предназначенный для того, чтобы идентифицировать разные типы IP-пакетов на основе определенных характеристик IP-пакетов. Эти характеристики описываются или определяются одним или более параметрами фильтрации, которые могут передаваться в IP-заголовке или части, соответствующей полезной нагрузке. В стеке протоколов IP находится на сетевом уровне, который находится ниже транспортного уровня, который, в свою очередь, находится ниже необязательного сеансового уровня, который находится ниже прикладного уровня. Сеть передачи данных может использовать протокол управления передачей (TCP), протокол пользовательских дейтаграмм (UDP) или некоторый другой протокол транспортного уровня. Параметры фильтрации могут передаваться в IP-заголовке, заголовке транспортного уровня (например, TCP-заголовке), заголовке сеансового уровня, заголовке прикладного уровня, полезной нагрузке прикладного уровня и так далее, или в их комбинации, все из которых инкапсулированы в дейтаграмму.

В контексте IP “фильтр” может быть рассмотрен как ящик, который предоставляет разные выходы для разных значений параметров фильтрации. В качестве примера, фильтр может быть определен на основе IP-адреса получателя и номера порта TCP получателя, равного 23. (TCP-порт относится к логическому каналу для ассоциированных данных.) Этот фильтр может использоваться, чтобы идентифицировать все IP-пакеты, предназначенные для telnet-сервера, работающего на хост-компьютере с этим IP-адресом получателя. Как правило, фильтрация может выполняться для того, чтобы отличить определенные IP-пакеты из потока IP-пакетов на основе характеристик, определенных параметрами фильтрации. Фильтрация позволяет осуществлять индивидуальное управление IP-пакетами, имеющими эти характеристики.

Фильтрация IP-пакетов является еще более перспективной при наличии IP-фрагментации. IP поддерживает фрагментацию дейтаграмм на более мелкие фрагменты. IP-фрагментация может использоваться, например, если дейтаграмма, которая должна быть отправлена, является слишком большой, чтобы быть

переданной протокольным блоком на уровне ниже сетевого уровня. В этом случае большая дейтаграмма может быть разделена на многочисленные небольшие фрагменты, и каждый фрагмент может быть отправлен как отдельный IP-пакет. IP-пакеты для фрагментов будут содержать соответствующую информацию заголовка, которая может использоваться получателем, чтобы повторно собрать эти фрагменты.

Если дейтаграмма делится на многочисленные фрагменты, то параметры фильтрации могут передаваться только в одном фрагменте или поднаборе фрагментов, что впоследствии усложняет фильтрацию IP-пакетов. Например, параметры фильтрации могут быть номерами портов источника и получателя в TCP-заголовке, который в типичном случае передается в первом фрагменте дейтаграммы. Если узел фильтрации выполняет фильтрацию отдельно в отношении каждого IP-пакета, то IP-пакеты, которые не содержат все из параметров фильтрации, не могут быть правильно отфильтрованы. Узел фильтрации - это узел, который выполняет фильтрацию IP-пакетов и может быть хост-компьютером или маршрутизатором.

В одной традиционной схеме фильтрации фрагментированных дейтаграмм узел фильтрации буферизирует все из фрагментов дейтаграммы, повторно собирает дейтаграмму после того, как все из фрагментов были приняты, выполняет фильтрацию по повторно собранной дейтаграмме и (если необходимо) повторно фрагментирует дейтаграмму на фрагменты и отправляет фрагменты как отдельные IP-пакеты. Эта традиционная схема фильтрации имеет несколько недостатков. Первое, длительная буферизация всех фрагментов каждой дейтаграммы прерывает нормальный поток этих фрагментов, привносит лишние задержки в передачу дейтаграммы к ее конечному получателю и может, кроме того, вызвать неравномерную нагрузку канала связи. Второе, повторный сбор и повторная фрагментация каждой дейтаграммы требует дополнительной обработки узлом фильтрации. Если узел фильтрации является маршрутизатором, то повторный сбор и повторная фрагментация сделают маршрутизацию очень эффективной.

Чтобы уменьшить объем обработки, узел фильтрации может выполнять фильтрацию в отношении фрагментированных дейтаграмм без повторной сборки фрагментов дейтаграмм. Однако узлу все еще может потребоваться буферизовать все фрагенты каждой дейтаграммы и применять фильтр только после того, как все фрагменты приняты. Недостатки, связанные с длительной буферизацией, будут все еще иметь место в этом случае.

Поэтому есть необходимость в уровне техники, чтобы эффективно фильтровать и маршрутизировать фрагментированные дейтаграммы.

Сущность изобретения

Технологии для эффективной фильтрации фрагментированных дейтаграмм и маршрутизации фрагментов этих дейтаграмм описываются в данном документе. Для каждой фрагментированной дейтаграммы, которая идентифицируется уникальным идентификатором дейтаграммы, узел фильтрации получает параметры фильтрации для этой дейтаграммы, по мере того как фрагменты дейтаграммы принимаются. Когда все из параметров фильтрации для дейтаграммы доступны, узел применяет один или более фильтров в отношении параметров фильтрации, получает результат фильтрации для дейтаграммы и сохраняет результат фильтрации в информационном элементе в таблице маршрутизации (т.е., в записи таблицы маршрутизации) для дейтаграммы. Результат фильтрации (который может также называться решением по

маршрутизации) указывает обработку, которая должна быть выполнена узлом для всех фрагментов этой дейтаграммы. Например, результат фильтрации может указывать, должен ли узел пропустить фрагменты дейтаграммы на более высокий уровень, пересылать фрагменты к получателю дейтаграммы, пересылать фрагменты по конкретному логическому или физическому каналу связи по направлению к получателю дейтаграммы, удалять все фрагменты дейтаграммы и т. д. Результат фильтрации получается на основе одного или более фильтров, которые в типичном случае предназначены для конкретного приложения.

Прежде чем получить результат фильтрации, узел фильтрации сохраняет все фрагменты, принятые для дейтаграммы, в памяти “фрагментов”, которая может быть любой памятью, подходящей для хранения данных. Когда результат фильтрации становится доступным, узел может обработать все фрагменты, уже принятые для дейтаграммы, в соответствии с результатом фильтрации. По мере последующего приема каждого оставшегося фрагмента для дейтаграммы узел может немедленно обработать фрагмент в соответствии с результатом фильтрации, сохраненным для дейтаграммы. Когда последний фрагмент для дейтаграммы принимается, узел может очистить часть памяти фрагментов, использованную дейтаграммой, и удалить запись таблицы маршрутизации для дейтаграммы.

Далее подробно описаны различные аспекты и варианты осуществления изобретения.

Перечень чертежей

Признаки и сущность настоящего изобретения станут более явными из изложенного ниже подробного описания, рассматриваемого вместе с чертежами, на которых используется сквозная нумерация позиций и на которых:

Фиг.1 - IP-пакет для дейтаграммы;

Фиг.2 - поля идентификатора примерной дейтаграммы;

Фиг.3 - фрагментация дейтаграммы на множество фрагментов;

Фиг.4 - процесс, выполненный узлом фильтрации, чтобы фильтровать фрагментированную дейтаграмму и чтобы маршрутизировать фрагменты дейтаграммы;

Фиг.5 - процесс управления таймерами для фрагментированных дейтаграмм;

Фиг.6 - запись таблицы маршрутизации для фрагментированной дейтаграммы;

Фиг.7 - блок-схема узла фильтрации; и

Фиг.8 - блок-схема устройства беспроводной связи и терминального оборудования.

Подробное описание изобретения

Слово "примерный" используется в данном документе, чтобы обозначать "служащий в качестве примера, отдельного случая или иллюстрации". Любой вариант осуществления или проект, описанный в данном документе как "примерный", необязательно должен быть истолкован как предпочтительный или выгодный по сравнению с другими вариантами осуществления или проектами.

Для ясности в нижеследующем описании используется следующая терминология. Сетевой уровень может принимать данные от более высокого уровня (например, транспортного уровня) и формировать интернет-дейтаграммы (или просто дейтаграммы). Сетевой уровень может также принимать IP-дейтаграммы от других элементов сети через один из его сетевых интерфейсов для обработки и пересылки через один из его оставшихся сетевых интерфейсов. Дейтаграмма может быть обработана и отправлена как один IP-пакет на сетевом уровне. Дейтаграмма может также быть разделена на множество фрагментов, и каждый фрагмент может быть

отправлен как отдельный IP-пакет на сетевом уровне. Каждый IP-пакет может, таким образом, быть для целой нефрагментированной дейтаграммы или для одного фрагмента фрагментированной дейтаграммы. Каждый IP-пакет включает в себя IP-заголовок и часть, соответствующую полезной нагрузке.

Фиг.1 показывает IP-пакет для дейтаграммы. IP-пакет включает в себя некоторое количество полей, и для простоты ниже описываются только те поля, которые отвечают настоящему раскрытию.

Длина Интернет заголовка (IHL) - указывает длину IP-заголовка (в единицах 32-битных слов).

Общая длина - указывает общую длину IP-пакета, включающего в себя IP-заголовок и часть, соответствующую полезной нагрузке (в единицах 8-битных октетов).

Идентификация - несет значение идентификаци, назначенное хост-компьютером, являющимся источником для содействия в повторной сборке фрагментов (если таковая имеет место) для дейтаграммы.

Флаги - содержит три бита, где бит MF является устанавливаемым в “0”, чтобы указать последний фрагмент дейтаграммы, и в “1”, чтобы указать, что один или более фрагментов следуют для дейтаграммы.

Смещение фрагмента - указывает положение фрагмента в дейтаграмме, что позволяет хост-компьютеру правильно повторно собрать фрагменты дейтаграммы.

Время жизни - указывает максимальное время, в течение которого дейтаграмме разрешено оставаться в IP-сети передачи данных.

Протокол - указывает протокол следующего более высокого уровня, используемый в части IP-пакета, соответствующей полезной нагрузке.

Адрес источника - несет IP-адрес хост-компьютера, являющегося источником.

Адрес получателя - несет IP-адрес хост-компьютера, являющегося получателем.

Полезная нагрузка - несет полезную нагрузку IP-пакета и имеет переменную длину.

Формат IP-пакета описан в RFC 791, озаглавленном “Internet Protocol DARPA Internet Program Protocol Specification”, сентябрь 1981. Другие протоколы для других уровней также определяют форматы для их блоков данных. Например, на транспортном уровне формат, использованный протоколом TCP, описывается в RFC 793, озаглавленном “Transmission Control Protocol”, сентябрь 1980, а формат, использованный протоколом UDP, описывается в RFC 768, озаглавленном “User Datagram Protocol”, август 1980.

Фиг.2 показывает поля примерного идентификатора 200 дейтаграммы, используемого, чтобы уникально идентифицировать каждую дейтаграмму. Идентификатор 200 дейтаграммы формируется сцеплением полей идентификации, протокола, адреса источника и адреса получателя IP-заголовка. RFC 791 требует, чтобы хост-компьютер, являющийся источником, устанавливал поле идентификации в значение, которое является уникальным как для (1) заданной комбинации IP-адреса источника, IP-адреса получателя и протокола, используемого для дейтаграммы, так и для (2) времени, в течение которого дейтаграмма будет действительна в сети передачи данных. Одинаковое значение идентификации используется для всех фрагментов заданной дейтаграммы. Таким образом, все IP-пакеты с одинаковым набором значений для этих четырех полей могут быть рассмотрены как принадлежащие одной и той же дейтаграмме. В общей реализации хост-компьютер, являющийся источником, случайным образом выбирает значение для поля идентификации, когда он начинает отправку дейтаграмм, и после этого увеличивает значение идентификации всякий раз,

когда он отправляет новую дейтаграмму (невзирая на протокол). В этом случае идентификатор дейтаграммы может быть задан сцеплением только полей идентификации, адреса источника и адреса получателя.

Фиг.3 показывает фрагментацию дейтаграммы на множество фрагментов. Исходная дейтаграмма содержит IP-заголовок и часть, соответствующую полезной нагрузке, и может превосходить возможности по переносу данных единичного блока данных на более низком уровне (например, канальном уровне). Если часть дейтаграммы, соответствующая полезной нагрузке, содержит M октетов и если каждый блок данных на более низком уровне может нести L октетов плюс IP-заголовок, то дейтаграмма может быть разделена на (M/L) фрагментов, где (x) означает оператор округления до ближайшего целого, который дает в результате ближайшее большее целое число для x. Так как RFC 791 требует, чтобы часть полезной нагрузки дейтаграммы была разделена по 8-октетной границе, для дейтаграммы может потребоваться более чем (M/L) фрагментов. Дейтаграмма может быть разделена на N фрагментов, которые помечаются как фрагменты 1 по N, где N может быть равно или больше, чем (M/L).

Для того чтобы фрагментировать дейтаграмму, IP-заголовок для каждого из N фрагментов генерируется с использованием IP-заголовка дейтаграммы. Новые значения для полей общей длины, смещения фрагмента и флага определяются для IP-заголовка каждого фрагмента. Остающиеся поля IP-заголовка каждого фрагмента копируются из первоначального IP-заголовка дейтаграммы. Опции из первоначального IP-заголовка только копируются в IP-заголовок первого фрагмента. В заключение, пересчитывается контрольная сумма IP-заголовка для каждого фрагмента. Часть дейтаграммы, соответствующая полезной нагрузке, разделяется на N более мелких частей по соответствующей 8-октетной границе. Часть каждого фрагмента, соответствующая полезной нагрузке, заполняется соответствующей одной из N частей. Бит MF в поле флагов IP-заголовка для каждого из фрагментов с 1 по N-1 устанавливается в “1”, чтобы указать, что один или более фрагментов будут следовать для дейтаграммы. Бит MF для фрагмента N устанавливается в “0”, чтобы указать, что этот фрагмент является последним фрагментом для дейтаграммы. Поле смещения фрагмента для каждого фрагмента устанавливается в значение, которое указывает начальное положение части полезной нагрузки в этом фрагменте относительно начала части полезной нагрузки в дейтаграмме. Поле общей длины устанавливается равным длине фрагмента.

Хост-компьютер, являющийся источником, отправляет N фрагментов для дейтаграммы как N отдельных IP-пакетв, в типичном случае по одному IP-пакету, по сети передачи данных. Поскольку эти IP-пакеты могут быть отправлены через разные маршруты, заданный узел может принять эти IP-пакеты не по порядку следования. В последующем описании “фрагмент 1” соответствует первому фрагменту дейтаграммы, а “первый принятый фрагмент” соответствует первому фрагменту, принятому для дейтаграммы, который может быть или может не быть фрагментом 1.

Узел фильтрации может эффективно выполнять фильтрацию фрагментированных дейтаграмм и маршрутизацию фрагментов для этих дейтаграмм следующим образом. Для каждой фрагментированной дейтаграммы узел получает параметры фильтрации для дейтаграммы по мере приема фрагментов дейтаграммы. Как только все из параметров фильтрации для дейтаграммы становятся доступны, узел применяет один или более фильтров в отношении параметров фильтрации, получает результат фильтрации для дейтаграммы и сохраняет результат фильтрации в записи (или строке)

в таблице маршрутизации. Результат фильтрации указывает обработку, которая должна быть выполнена узлом для всех фрагментов дейтаграммы. Прежде чем получить результат фильтрации, узел сохраняет все фрагменты, принятые для дейтаграммы, в памяти фрагментов. Как только результат фильтрации становится доступным, узел может обработать все фрагменты, уже принятые для дейтаграммы, в соответствии с результатом фильтрации. По мере последующего приема каждого оставшегося фрагмента для дейтаграммы узел может немедленно обработать фрагмент в соответствии с результатом фильтрации. Когда последний фрагмент для дейтаграммы принимается, узел может очистить часть памяти фрагментов, использованную дейтаграммой, и удалить запись таблицы маршрутизации для дейтаграммы.

Фрагменты дейтаграммы могут быть сохранены разными способами в памяти фрагментов перед тем, как результат маршрутизации становится доступным. Так как размеры фрагментов могут быть не известны априори и фрагменты могут быть приняты не по порядку, каждый фрагмент может быть сохранен в отдельном буфере, когда фрагмент принимается. Буфер может быть частью памяти фрагментов соответствующего размера, который выделяется, как это необходимо. Это позволяет эффективно использовать память фрагментов, так как буфер соответствующего размера может выделяться, когда и если этот буфер действительно необходим. Может использоваться механизм для того, чтобы (1) определять, какие фрагменты дейтаграммы были приняты, (2) определять буферы, где сохраняются принятые фрагменты, и (3) определять другую подходящую информацию, если таквая имеется для дейтаграммы. Этот механизм может быть реализован, например, с помощью списка очередей, связанного списка и т. д., как известно в области техники. Последующее описание предполагает, что фрагменты сохраняются в отдельных буферах. Однако фрагменты могут также сохраняться другими способами, и это находится в рамках объема изобретения.

Вообще, параметры фильтрации для каждой фрагментированной дейтаграммы могут переноситься в одном или более фрагментах, которые именуются как “целевые” фрагменты. Однако многие приложения используют параметры фильтрации, которые передаются в IP-заголовке и/или заголовке для протокола вышерасположенного уровня, такого как TCP, UCP, ICMP и т. д. Так как минимальный размер фрагмента равен 576 октетам, IP-заголовок и заголовки вышерасположенных уровней в типичном случае переносятся в фрагменте 1 дейтаграммы. Для приложений, которые используют параметры фильтрации, переданные в IP-заголовке и заголовках вышерасположенных уровней, в типичном случае присутствует только один целевой фрагмент, который обычно является фрагментом 1 дейтаграммы. Для этих приложений все из параметров фильтрации для дейтаграммы могут быть получены из фрагмента 1 дейтаграммы, который в типичном случае также является первым принятым фрагментом для дейтаграммы, и результат фильтрации может быть получен для дейтаграммы на основе первого принятого фрагмента. Все последующие фрагменты для дейтаграммы могут быть обработаны по мере их приема без необходимости буферизовать эти фрагменты.

Узел фильтрации может использовать таймер для каждой фрагментированной дейтаграммы, чтобы гарантировать, что фрагменты для “устаревших” дейтаграмм удаляются из памяти фрагментов. Таймер для каждой дейтаграммы может быть установлен в первоначальное значение, когда принимается первый фрагмент для дейтаграммы. Это первоначальное значение может быть, например, значением

времени жизни в IP-заголовке фрагмента. Когда таймер истекает, запись таблицы маршрутизации для дейтаграммы может быть удалена, и все фрагменты, сохраненные в памяти фрагментов, для дейтаграммы могут быть удалены, таким образом, очищая часть памяти фрагментов, используемую дейтаграммой.

Узел фильтрации может выполнять следующую служебную обработку для разных фрагментов фрагментированной дейтаграммы:

- первый принятый фрагмент для дейтаграммы - создание записи для дейтаграммы в таблице маршрутизации и старт таймера для дейтаграммы;
- целевой фрагмент(ы) - получение всех параметров фильтрации для дейтаграммы и применение, по мньшей мере, одного фильтра в отношении параметров фильтрации, чтобы получить результат фильтрации для дейтаграммы; и
- последний фрагмент для дейтаграммы - очистка части памяти фрагментов, использованной дейтаграммой, и удаление записи таблицы маршрутизации для дейтаграммы.

Фиг.4 показывает блок-схему процесса 400, выполняемого узлом фильтрации для фильтрации фрагментированной дейтаграммы и маршрутизации фрагментов этой дейтаграммы. Фрагментированная дейтаграмма составлена из множества фрагментов, и каждый фрагмент отправляется как один IP-пакет. Таким образом, “фрагменты” и “IP-пакеты” являются синонимичными терминами для фрагментированной дейтаграммы. Фиг.4 показывает обработку одного IP-пакета.

Изначально узел фильтрации принимает IP-пакет (этап 412). Узел затем определяет, является ли IP-пакет фрагментированной дейтаграммой (этап 414). Это может быть выполнено посредством проверки бита MF и поля смещения фрагмента в IP-заголовке IP-пакета. IP-пакет является нефрагментированной дейтаграммой, если поле смещения фрагмента установлено в 0 (что является истиной только для фрагмента 1), и бит MF установлен в “0” (указывающий, что никакой другой фрагмент не будет доступен для дейтаграммы). Если IP-пакет является нефрагментированной дейтаграммой (т. е., ответ 'нет' на этапе 414), то узел применяет один или более фильтров к IP-пакету и обрабатывает и/или пересылает IP-пакет на основе результата фильтрации (этап 416). Узел затем возвращается на этап 412, чтобы обработать следующий IP-пакет.

Если текущий IP-пакет является фрагментированной дейтаграммой, как определено на этапе 414, то узел фильтрации определяет, является ли этот IP-пакет первым принятым для дейтаграммы, который может быть или может не быть фрагментом 1 дейтаграммы, так как узел может принять фрагменты не по порядку (этап 422). Это определение может быть сделано посредством (1) получения идентификатора дейтаграммы, например, как сцепление значений идентификации, протокола, адреса источника и адреса получателя в IP-заголовке IP-пакета, (2) поиска записей таблицы маршрутизации для идентификатора этой дейтаграммы и (3) указания текущего IP-пакета в качестве первого принятого фрагмента, если идентификатор дейтаграммы не найден в каком-либо из элементов таблицы маршрутизации.

Если текущий IP-пакет является первым принятым фрагментом, то узел фильтрации создает новую запись таблицы маршрутизации для дейтаграммы, инициализирует поля новой записи таблицы маршрутизации и запускает таймер для дейтаграммы (этап 424). Запись таблицы маршрутизации для дейтаграммы индексируется по идентификатору дейтаграммы. Запись таблицы маршрутизации включает в себя указатель, который указывает на местоположение памяти фрагментов, используемое для того, чтобы хранить фрагменты дейтаграммы. Узел затем переходит на этап 432.

Если текущий IP-пакет не является первым принятым фрагментом, как определено на этапе 422, то узел фильтрации определяет, доступен ли результат фильтрации для дейтаграммы в записи таблицы маршрутизации для дейтаграммы (этап 426). Если результат фильтрации существует, то узел обрабатывает/пересылает IP-пакет в соответствии с результатом фильтрации (этап 428) и затем переходит на этап 442. Иначе, если результат фильтрации недоступен, то узел определяет, были ли получены все параметры фильтрации для дейтаграммы из текущего IP-пакета и приняты ли уже все IP-пакеты, которые также являются фрагментами этой дейтаграммы (этап 432). Если все параметры фильтрации не были получены, то узел выделяет буфер в памяти фрагментов для текущего IP-пакета и сохраняет IP-пакет в буфере (этап 438). Узел затем возвращается на этап 412, чтобы обработать следующий IP-пакет. Иначе, если все параметры фильтрации были получены, то узел применяет фильтр(ы) в отношении параметров фильтрации, получает результат фильтрации для дейтаграммы и сохраняет результат фильтрации в записи таблицы маршрутизации, созданной для дейтаграммы (этап 434). Узел затем обрабатывает/пересылает текущий IP-пакет, так же как и все уже принятые IP-пакеты, которые также являются фрагментом этой дейтаграммы и сохранены в памяти фрагментов, в соответствии с результатом фильтрации (этап 436). Узел затем переходит к этапу 442.

Для варианта осуществления, показанного на фиг.4, узел фильтрации также отслеживает то, какие фрагменты были приняты для дейтаграммы, и обновляет эту информацию всякий раз, когда для дейтаграммы принимается новый фрагмент (этап 442). Узел может выполнять эту регистрацию на основе смещения фрагмента каждого фрагмента (которое получается из поля смещения фрагмента) и размера полезной нагрузки каждого фрагмента (который получается на основе полей IHL и общей длины). Узел затем определяет, были ли приняты все фрагменты дейтаграммы, на основе обновленной информации о фрагменте (этап 444). Целая дейтаграмма принимается, если (1) был принят последний фрагмент дейтаграммы (последний фрагмент имеет бит MF, установленный в “0”), и (2) все другие фрагменты дейтаграммы были также приняты (что может быть определено на основе смещения фрагмента и размера полезной нагрузки каждого принятого фрагмента). Если целая дейтаграмма была принята, то узел удаляет запись таблицы маршрутизации для дейтаграммы и удаляет все фрагменты дейтаграммы из памяти фрагментов (этап 446). С этапов 444 и 446 узел возвращается на этап 412, чтобы обработать следующий IP-пакет.

В другом варианте осуществления вместо удаления элемента таблицы маршрутизации для дейтаграммы и очистки памяти фрагментов, используемой дейтаграммой, на этапе 446, узел фильтрации просто предписывает истечение таймера для дейтаграммы на этапе 446. Механизм управления таймером затем удалит запись таблицы маршрутизации и очистит память фрагментов, используемую этой дейтаграммой, так как ее таймер истек, как описано ниже. В еще одном варианте осуществления узел фильтрации не отслеживает информацию информации о фрагменте и не выполняет обработку, показанную на этапах 442, 444 и 446. Для этого варианта осуществления узел фильтрации полагается на механизм управления таймером, чтобы удалить запись таблицы маршрутизации и удалить фрагменты в памяти фрагментов для старых дейтаграмм, когда их таймеры естественным образом истекли.

Как показано на фиг.4, фрагменты дейтаграммы временно хранятся в буферах в памяти фрагментов до тех пор, пока все параметры фильтрации для дейтаграммы не

будут приняты и результат фильтрации не сможет быть получен для дейтаграммы. Если все из параметров фильтрации передаются в фрагменте 1 дейтаграммы, то результат фильтрации может быть получен, как только будет принят фрагмент 1, который в типичном случае является первым принятым фрагментом для дейтаграммы, если фрагменты принимаются не по порядку. В этом случае как задержка, связанная с маршрутизацией, так и требования касаемо буферизации минимизируются для дейтаграммы. Кроме того, фрагмент 1 может быть без труда идентифицирован значением, равным 0, для поля смещения фрагмента.

Если все из параметров фильтрации не переносятся в фрагменте 1, то и задержка, связанная с маршрутизацией, и требования касаемо буферизации увеличиваются в пропорции к задержке, требуемой узлом, чтобы получить все из требуемых параметров фильтрации. Кроме того, если множество фрагментов требуется для получения всех параметров фильтрации, то параметры фильтрации также нужно будет временно сохранять по мере их приема, до того времени, когда фильтр(ы) может быть применен. Соответствующая обработка выполняется, чтобы обнаружить целевой фрагмент(ы), который переносит требуемые параметры фильтрации.

Эффективная фильтрация может быть выполнена посредством проектирования таких фильтров, чтобы использовались параметры, которые предполагаются быть доступными только в одном фрагменте (чтобы избежать буферизации параметров фильтрации) и предпочтительно в фрагменте 1 (чтобы избежать размещения в буфер фрагментов). Если хост-компьютер, являющийся источником, знает о фильтрации, применяемой узлом фильтрации, то хост-компьютер, являющийся источником, может отправить фрагменты таким способом, что узел фильтрации может принять все параметры фильтрации так скоро, насколько возможно. Например, узел-источник может отправить фрагменты не по порядку, так что фрагменты, переносящие параметры фильтрации, отправляются первыми. Вообще, фильтры могут быть заданы так и/или фрагменты могут быть отправлены так, что задержка, связанная с маршрутизацией, и требование касаемо буферизации могут быть минимизированы до возможной степени.

Узел фильтрации буферизирует фрагменты дейтаграммы в буферы до тех пор, пока результат фильтрации для дейтаграммы не будет получен. Некоторые фрагменты могут быть сброшены или испорчены при прохождении через сеть передачи данных (по любым причинам) и могут быть не приняты узлом. Если фрагмент, содержащий требуемый параметр фильтрации, сброшен, то результат фильтрации для дейтаграммы не может быть получен, и все фрагменты, принятые для дейтаграммы, могут находиться в памяти неопределенное время до тех пор, пока память фрагментов не будет очищена каким-либо механизмом. Даже если результат фильтрации для дейтаграммы получен, любой сброшенный фрагмент приведет к сбою в приеме целой дейтаграммы, и запись таблицы маршрутизации для дейтаграммы может храниться в памяти неопределенное время, если запись таблицы маршрутизации не будет удалена каким-либо механизмом.

Таймер может использоваться для каждой дейтаграммы, чтобы избежать ситуаций, в которых запись таблицы маршрутизации и/или буферы для дейтаграммы сохраняются неопределенное время, например, так как фрагмент сброшен сетью передачи данных. Таймер устанавливается в первоначальное значение, когда принимается первый фрагмент для дейтаграммы. Это первоначальное значение должно быть достаточно большим, так чтобы узел преждевременно не удалил запись таблицы маршрутизации или какие-либо ожидающие фрагменты, уже принятые для

дейтаграммы. Это первоначальное значение должно также быть равно или меньше, чем время, требуемое для полного цикла значения в поле идентификации, так чтобы результат фильтрации, полученный для дейтаграммы, ошибочно не был применен к другой дейтаграмме с таким же значением идентификации. Первоначальное значение может быть установлено в значение в поле времени жизни IP-заголовка, например, или в некоторое другое значение времени.

Фиг.5 показывает блок-схему процесса 500 управления таймерами для фрагментированных дейтаграмм. Таймер для фрагментированной дейтаграммы устанавливается, когда принимается первый фрагмент для дейтаграммы, как показано на фиг. 4. После этого, таймер обновляется (например, уменьшается) на количество времени, которое прошло после последнего обновления. После истечения таймера дейтаграмма считается “устаревшей” и все фрагменты, принятые для дейтаграммы, и ее запись таблицы маршрутизации могут быть удалены. Это предотвращает накопление старых фрагментированных дейтаграмм в памяти фрагментов.

Узел обновляет таймер для каждой фрагментированной дейтаграммы, например, периодически или когда он запущен некоторым событием (этап 512). Затем узел определяет, истек ли таймер для какой-либо дейтаграммы (этап 514). Если ответ - 'да', то узел получает идентификатор дейтаграммы для каждой устаревшей дейтаграммы с истекшим таймером (этап 516). Узел затем удаляет запись таблицы маршрутизации для каждой устаревшей дейтаграммы и сбрасывает все фрагменты, сохраненные в памяти фрагментов для устаревшей дейтаграммы (этап 518). Сбрасывание устаревших и/или неполных дейтаграмм не является катастрофическим, так как протокол на более высоком уровне будет, по всей вероятности, выполнять соответствующее корректирующее действие, например инициировать повторную передачу отсутствующих дейтаграмм.

Процесс 500 может выполняться периодически, например, каждое предварительно определенное число секунд. Процесс 500 может также выполняться будучи инициированным, например, после обработки каждого принятого IP-пакета.

Фиг.6 показывает вариант осуществления табличной записи 600 в таблице маршрутизации для фрагментированной дейтаграммы. Для этого варианта осуществления поле 612 хранит идентификатор дейтаграммы, который может быть сформирован из полей в IP-заголовке первого принятого фрагмента дейтаграммы, как описано выше. Запись таблицы маршрутизации индексируется идентификатором дейтаграммы. Поле 614 хранит таймер для дейтаграммы, который устанавливается в предварительно определенное значение, когда создается запись таблицы маршрутизации, и после этого обновляется, например, периодически или будучи инициированным некоторым событием.

Поле 616 хранит параметры фильтрации, уже принятые для дейтаграммы. Если все из параметров фильтрации передаются в одном фрагменте, то фильтр(ы) может быть применен, и результат фильтрации может быть получен, когда принимается этот фрагмент. В этом случае поле 616 не используется. Однако, если все из параметров фильтрации для дейтаграммы переносятся в множестве фрагментов, то параметры фильтрации могут сохраняться в поле 616 по мере их приема, и фильтр(ы) может быть применен после того, как все из параметров фильтрации приняты. Поле 616 может приниматься во внимание для определения по этапу 432 на фиг.4.

Поле 618 хранит результат фильтрации для дейтаграммы, который получается от применения фильтра(ов) к параметрам фильтрации для дейтаграммы. Поле 618

устанавливается в известное состояние, когда создается запись таблицы маршрутизации, чтобы указать, что результат фильтрации еще недоступен. Поле 618 может проверяться, чтобы определить, доступен ли результат фильтрации, для этапа 426 на фиг.4. Результат фильтрации, если доступен, используется, чтобы обработать/переслать все фрагменты дейтаграммы. Результат фильтрации может включать в себя разные типы информации, такие как, например, (1) касаемо того, пропустить ли фрагменты дейтаграммы на вышерасположенный уровень, удалить фрагменты или переслать фрагменты хост-компьютеру, являющемуся получателем, (2) информацию маршрутизации для хост-компьютера, являющегося получателем, такую как адрес интерфейса или шлюз, если применимо, (3) идентификатор канала связи для многоканальных интерфейсов, (4) метки для классификации качества обслуживания (QoS) и т. д.

Поле 620 хранит информацию для фрагментов дейтаграммы, например смещение фрагмента и размер полезной нагрузки каждого фрагмента. Эта информация о фрагменте используется, чтобы определить, была ли принята целая дейтаграмма, для этапа 444 на фиг.4. Поле 620 может быть очищено, когда запись таблицы маршрутизации создается и после этого заполняется битом MF, смещением фрагмента и размером полезной нагрузки для каждого принятого фрагмента. Поле 622 хранит указатель на местоположение в памяти фрагментов, в которой хранятся фрагменты дейтаграммы.

Фиг. 7 показывает блок-схему варианта осуществления узла 700, выполненного с возможностью фильтровать дейтаграммы и маршрутизировать фрагменты. В узле 700 процессор 720 принимает IP-пакеты и обрабатывает каждый принятый IP-пакет (например, как показано на фиг. 4). Для каждого принятого IP-пакета процессор 720 хранит IP-пакет в памяти 730 фрагментов, если результат фильтрации недоступен, и обрабатывает/пересылает IP-пакет, если результат фильтрации доступен. Главная память 722 хранит данные и программные коды, используемые процессором 720.

Память 730 фрагментов хранит IP-пакеты для фрагментов дейтаграмм, для которых результаты фильтрации недоступны. Таблица 732 маршрутизации хранит табличную запись для каждой фрагментированной дейтаграммы, обрабатываемой в текущий момент процессором 720. Процессор 720 создает новую запись в таблице 732 маршрутизации всякий раз, когда он принимает новую фрагментированную дейтаграмму, и удаляет табличную запись, если целая дейтаграмма была принята или если таймер дейтаграммы истекает. Источник 734 времени предоставляет информацию хронирования, используемую для обновления таймеров для фрагментированных дейтаграмм.

Узел 700 может быть устройством или аппаратурой в базовой сети, например маршрутизатором в сети передачи данных. Узел 700 может также быть устройством или аппаратурой, соединенным(ой) с сетью передачи данных. Сеть передачи данных может быть проводной сетью, сетью беспроводной связи или комбинацией обоих типов сети.

Фиг.8 показывает блок-схему устройства беспроводной связи 810 и терминального оборудования 850. Устройство беспроводной связи 810 может осуществлять связь с одной или более системами беспроводной связи, такими как система множественного доступа с кодовым разделением каналов (CDMA), глобальная система мобильной связи (GSM) и т. д. В системе CDMA может быть реализован один или более стандартов CDMA, таких как IS-2000, IS-856, IS-95, Wideband-CDMA (W-CDMA) и т. д. Устройство беспроводной связи 810 выполнено с возможностью обеспечения

двунаправленной связи через тракт передачи и тракт приема.

Для тракта передачи процессор 840 модема обрабатывает (например, кодирует и модулирует) данные, которые должны быть переданы устройством беспроводной связи 810, и предоставляет символы данных блоку передатчика (TMTR) 842. Блок 842 передатчика приводит к определенному состоянию (например, преобразует в аналоговый сигнал, фильтрует, усиливает и преобразует с повышением частоты) элементарные сигналы данных и генерирует модулированный сигнал, который передается через антенну 844. Для тракта приема сигналы, переданные базовыми станциями в одной или более системах, принимаются антенной 844 и предоставляются блоку приемника (RCVR) 846. Блок 846 приемника приводит к определенному состоянию (например, фильтрует, усиливает и преобразует с понижением частоты) принятый сигнал, оцифровывает преобразованный сигнал и предоставляет выборки данных процессору 840 модема для демодуляции и декодирования.

Основной процессор 820 выполняет разные функции и управляет работой блоков обработки в устройстве беспроводной связи 810. Блок 822 главной памяти хранит данные и программные коды, используемые основным процессором 820. Блок 836 ввода/вывода (I/O) обеспечивает интерфейс с внешними объектами, например терминальным оборудованием 850. Шина 838 связывает разные блоки в устройстве беспроводной связи 810.

Устройство беспроводной связи 810 может также быть узлом фильтрации. В этом случае память 830 фрагментов хранит фрагменты дейтаграмм, для которых результаты фильтрации недоступны. Таблица 832 маршрутизации хранит элементы для обрабатываемых фрагментированных дейтаграмм. Источник 834 времени предоставляет информацию о времени, используемую, чтобы поддерживать таймеры для фрагментированных дейтаграмм.

Терминальное оборудование 850 может быть, например, портативным компьютером, персональным цифровым информационным устройством (PDA) или неким другим электронным устройством. Терминальное оборудование 850 включает в себя процессор 860, который выполняет обработку для терминального оборудования, память 862, которая хранит данные и программные коды, используемые процессором 860, и блок 864 связи, который поддерживает связь с другими объектами, например устройством беспроводной связи 810.

Технологии, описанные в данном документе, могут эффективно фильтровать фрагментированные дейтаграммы. Это достигается посредством сбора параметров фильтрации для каждой фрагментированной дейтаграммы по мере их приема и применения одного или более фильтров для дейтаграммы, как только все параметры фильтрации для дейтаграммы приняты. Фильтр(ы) может, таким образом, быть применен быстро (например, по первому фрагменту, принятому для дейтаграммы) без необходимости ждать, чтобы все фрагменты были приняты. Технологии, описанные в данном документе, могут также эффективно маршрутизировать фрагменты. Это достигается получением результата фильтрации для каждой фрагментированной дейтаграммы, как только все параметры фильтрации для дейтаграммы приняты, сохранением результата фильтрации в записи таблицы маршрутизации и применением результата фильтрации к каждому фрагменту, принимаемому впоследствии для дейтаграммы.

Технологии, описанные в данном документе, гарантируют, что каждый фрагмент претерпевает минимальную задержку при прохождении через узел фильтрации. Дополнительная задержка, испытываемая фрагментом дейтаграммы, может быть

такой же, как и количество времени, которое требуется узлу для приема всех параметров фильтрации для дейтаграммы. Последующие фрагменты для дейтаграммы могут быть обработаны/пересланы, как только принят каждый фрагмент.

Если все из параметров фильтрации для дейтаграммы передаются в фрагменте 1 и этот фрагмент принимаеся узлом первым, то узлу не нужно хранить какие-либо фрагменты для дейтаграммы. Если требуемый параметр фильтрации передается в принятом позже фрагменте (например, так как фрагмент 1 принимается не по порядку или потому что параметр фильтрации передается в фрагменте, отличном от фрагмента 1), то узлу может быть необходимо буферизовать некоторые фрагменты на короткую продолжительность времени, пока все параметры фильтрации не будут приняты. При надлежащей конструкции фильтра и для общего случая, в котором фрагменты принимаются по порядку, фрагменты могут быть обработаны/пересланы, как только они приняты узлом. В наихудшем случае требуемый параметр фильтрации переносится в последнем фрагменте, принятом для дейтаграммы, и все фрагменты дейтаграммы нужно буферизовать и задержать. Однако этот сценарий наихудшего случая для технологий, описанных в данном документе, будет обычным случаем для традиционной схемы фильтрации, которая применяет фильтр(ы) только после того, как все фрагменты дейтаграммы были приняты.

Фильтрация IP-пакетов может использоваться для разных вариантов применения в средах передачи данных, где маршрутизация, основанная исключительно на IP-адресе получателя, может не быть подходящей. Некоторые примерные варианты применения фильтрации описываются ниже.

Фильтрация IP-пакетов может использоваться для того, чтобы поддерживать на устройстве беспроводной связи приложения, основывающиеся на локальных сокетах. Так, например, устройство беспроводной связи (например, сотовый телефон, беспроводная карта данных или беспроводной модуль, выполненный с возможностью предоставлять услуги пакетов данных) может быть соединено с терминальным оборудованием (например, портативным компьютером), как показано на фиг.8. Терминальное оборудование использует устройство беспроводной связи, чтобы получить возможность установления IP-соединений по сети беспроводной связи, и устройство беспроводной связи пересылает терминальному оборудованию IP-пакеты, принятые из сети беспроводной связи. Многие сети беспроводной связи передачи данных (например, сеть CDMA) в типичном случае назначают один IP-адрес устройству беспроводной связи. Устройство беспроводной связи может затем предоставить этот IP-адрес терминальному оборудованию для использования для передачи данных. В этом случае, если устройство беспроводной связи принимает IP-пакеты с этим IP-адресом в качестве IP-адреса получателя, устройство беспроводной связи будет пересылать эти IP-пакеты терминальному оборудованию.

Простая основанная на IP-адресе маршрутизация, описанная выше, не поддерживает приложения, основывающиеся на локальных сокетах, в устройстве беспроводной связи. Такие приложения могут включать в себя, например, Mobile IP, используемый для обеспечения возможности установления IP-соединений для устройства беспроводной связи в мобильной среде, GPS для определения местоположения и т. д. Эти приложения могут требовать взаимодействия между устройством беспроводной связи и сетью беспроводной связи для обмена подходящей информацией. Таким образом, из-за сущности этих приложений может быть более подходящим запустить приложения на устройстве беспроводной связи вместо терминального оборудования.

Технологии фильтрации и маршрутизации, описанные в данном документе, могут использоваться, чтобы одновременно поддерживать, с помощью одного IP-адреса, и (1) системные приложения, работающие по локальным сокетам в устройстве беспроводной связи, и (2) приложения конечных пользователей, работающие на терминальном оборудовании. Один или более фильтров могут быть определены с помощью параметров фильтрации, которые являются применимыми для системных приложений, но не для приложений конечных пользователей. Например, параметр фильтрации может быть определен для номеров TCP-портов, используемых системными приложениями, но не приложениями конечных пользователей. Тогда фильтр(ы) будет применяться к каждой дейтаграмме, принятой устройством беспроводной связи из сети беспроводной связи, чтобы получить результат фильтрации для дейтаграммы. Результат фильтрации будет указывать, должна ли быть дейтаграмма передана терминальному оборудованию или отправлена вверх по стеку протоколов в устройстве беспроводной связи. Фильтрация IP-пакетов может, таким образом, использоваться, чтобы отличать IP-пакеты для устройства беспроводной связи от IP-пакетов для терминального оборудования.

Фильтрация IP-пакетов может также использоваться, чтобы поддерживать разные степени качества обслуживания (QoS) в сети передачи данных. Разные типы данных трафика (например, для системных приложений и/или приложений конечных пользователей) могут быть назначены разным степеням QoS и могут быть отправлены по разным логическим каналам. Желаемое QoS каждой дейтаграммы может быть отправлено узлом-источником и определено набором параметров фильтрации. IP-маршрутизаторы в сети передачи данных могут использовать предварительно определенные фильтры, которые работают в отношении параметров фильтрации для каждой дейтаграммы, чтобы определить, какой логический канал должен использоваться для дейтаграммы для обеспечения желаемого QoS. Если все IP-пакеты предназначаются устройству, которое имеет множественные логические связи по одному и тому же IP-интерфейсу, то маршрутизация на основе IP-адреса не будет достаточной, чтобы определить, какой логический канал использовать для каждой дейтаграммы. Одно или более дополнительных полей в дейтаграмме могут использоваться в качестве параметра(ов) фильтра для маршрутизации дейтаграмм, требующих разные степени QoS, в соответствующие логические каналы.

Может также использоваться фильтрация IP-пакетов, чтобы эффективно поддерживать широковещательную передачу и многоадресную передачу. Для поддержки многоадресной передачи узел объединяет группу многоадресной передачи, которая идентифицируется IP-адресом многоадресной передачи. В типичном все IP-пакеты с IP-адресом многоадресной передачи отправляются узлу невзирая на номера портов транспортного уровня, для которых эти IP-пакеты предназначены. Узел затем может обработать эти IP-пакеты и пропустить все из IP-пакетов вверх на транспортный уровень, который может затем сбросить ненужные IP-пакеты. Передача ненужных IP-пакетов нежелательна для сети беспроводной связи передачи данных, так как используются дорогостоящие ресурсы эфирной связи для того, чтобы передать ненужные IP-пакеты, которые впоследствии сбрасываются устройством беспроводной связи. IP-маршрутизатор в сети беспроводной связи передачи данных может использовать фильтрацию IP-пакетов, чтобы более эффективно поддерживать многоадресную передачу. Фильтр может быть определен на основе номеров портов транспортного уровня или некоторых других параметров фильтрации. IP-маршрутизатор затем применит фильтр ко всем дейтаграммам и выборочно

перешлет только дейтаграммы, востребованные узлом. Фильтрация IP-пакетов может применяться подобным образом, чтобы эффективно поддерживать широковещательную передачу.

Фильтрация IP-пакетов может также использоваться инструментальными средствами регистрации пакетов, чтобы выборочно регистрировать IP-пакеты. Определенные IP-пакеты могут быть выбраны на основе одного или более параметров фильтрации среди многочисленных IP-пакетов, принятых узлом. Выбранные IP-пакеты могут регистрироваться для последующей оценки. Фильтрация IP-пакетов может также использоваться, чтобы реализовать сетевой эмулятор, имитируя различные условия сети, выборочно дублируя, разрушая и/или сбрасывая IP-пакеты.

Фильтрация IP-пакетов может, таким образом, использоваться для разных вариантов применения фильтрации, чтобы выборочно обработать и/или переслать дейтаграммы. Для всех из этих вариантов применеия фильтрации технологии, описанные в данном документе, могут использоваться, чтобы эффективно фильтровать IP-пакеты и маршрутизировать IP-пакеты для фрагментированных дейтаграмм.

Фильтрация IP-пакетов может выполняться различными устройствами в сети. Например, фильтрация IP-пакетов может выполняться устройством беспроводной связи, маршрутизатором в сети (например, чтобы обеспечить QoS), автономным устройством (которое может быть связано или не связано с терминальным оборудованием) и т. д.

Разные приложения фильтрации используют разные фильтры, которые могут работать в отношении одинаковых или разных наборов параметров фильтрации. Параметры фильтрации могут переноситься в IP-заголовке, заголовке протокола более высокого уровня (например, TCP-заголовке), полезной нагрузке более высокого уровня и т. д., как упомянуто выше. В качестве некоторых примеров следующее может быть использовано как параметры фильтрации: (1) из IP-заголовка - IP-адрес источника, IP-адрес получателя, маска подсети, номер протокола, информация безопасности, такая как индекс параметра безопасности IPSec (SPI), тип обслуживания (например, в IPv4), класс трафика (например, в IPv6), метка потока (например, в IPv6) и т. д., и (2) из заголовка транспортного уровня - порт источника, порт получателя, диапазон портов источника и диапазон портов получателя. Маска подсети указывает диапазон IP-адресов, которые могут использоваться для подсети в сети передачи данных. Вообще, любое поле любого заголовка или полезной нагрузки на любом уровне может использоваться для параметров фильтрации.

Технологии фильтрации и маршрутизации, описанные в данном документе, могут быть реализованы различными средствами. Например, эти технологии могут быть реализованы в аппаратных средствах, программных средствах или их комбинации. Для аппаратной реализации модули обработки, используемые для того, чтобы выполнить фильтрацию и маршрутизацию, могут быть реализованы в одной или более специализированных интегральных схемах (ASIC), процессорах цифровых сигналов (DSP), устройствах обработки цифровых сигналов (DSPD), программируемых логических устройствах (PLD), программируемых пользователем матричных БИС (FPGA), процессорах, контроллерах, микроконтроллерах, микропроцессорах, других электронных устройствах, предназначенных выполнять функции, описанные в данном документе, или их комбинации.

Для программной реализации технологии фильтрации и маршрутизации могут быть

реализованы в модулях (например, процедурах, функциях и т. д.), которые выполняют функции, описанные в данном документе. Программные коды могут храниться в модуле памяти (например, блоке 722 памяти на фиг.7 или блоке 822 памяти на фиг.8) и исполняться процессором (например, процессором 720 на фиг.7 или главным процессором 820 на фиг.8). Модуль памяти может быть реализован внутренним или внешним по отношению к процессору образом, в последнем случае он может быть подсоединен с возможностью осуществления связи к процессору через различные средства, как известно в области техники.

Предшествующее описание раскрытых вариантов осуществления предоставлено, чтобы дать возможность любому специалисту в данной области техники создавать или использовать настоящее изобретение. Различные модификации в этих вариантах осуществления будут очевидны для специалистов в данной области техники, а описанные в данном документе общие принципы могут быть применены к другим вариантам осуществления без отступления от сущности и объема изобретения. Таким образом, не подразумевается, что настоящее изобретение ограничено показанными в данном документе вариантами осуществления, напротив, ему соответствует наиболее широкий объем, согласующийся с принципами и новыми признаками, раскрытыми в данном документе.

Формула изобретения

1. Способ обработки фрагментированных дейтаграмм в сети передачи данных, содержащий этапы, на которых

идентифицируют по меньшей мере один параметр фильтрации для дейтаграммы, отправленной как множество фрагментов;

сохраняют один или более принятых фрагментов дейтаграммы в качестве сохраненных фрагментов;

получают упомянутый по меньшей мере один параметр фильтрации из упомянутых одного или более принятых фрагментов;

определяют результат фильтрации для дейтаграммы на основе упомянутого по меньшей мере одного параметра фильтрации, причем результат фильтрации определяют до того, как будут приняты все фрагменты дейтаграммы;

обрабатывают сохраненные фрагменты на основе результата фильтрации; и

обрабатывают дополнительные фрагменты, принятые после того, как результат фильтрации определен на основе упомянутого по меньшей мере одного параметра фильтрации.

2. Способ по п.1, в котором дейтаграмма является дейтаграммой Интернет-протокола (IP).

3. Способ по п.1, дополнительно содержащий этап, на котором получают упомянутый по меньшей мере один параметр фильтрации из по меньшей мере одного из заголовка протокола и полезной нагрузки, ассоциированных с упомянутыми одним или более сохраненными фрагментами данных.

4. Способ по п.1, дополнительно содержащий этапы, на которых

поддерживают таймер для дейтаграммы; и

вычищают сохраненные фрагменты для дейтаграммы после истечения таймера.

5. Способ по п.1, дополнительно содержащий этап, на котором идентифицируют каждый фрагмент дейтаграммы на основе идентификатора дейтаграммы.

6. Способ по п.5, в котором идентификатор дейтаграммы содержит адрес источника, адрес получателя, значение идентификации и номер протокола,

включенные в IP-заголовок для фрагмента.

7. Способ по п.1, дополнительно содержащий этапы, на которых

принимают фрагмент для дейтаграммы; и

если фрагмент является первым фрагментом, принятым для дейтаграммы, создают табличную запись для дейтаграммы в первой памяти, причем данная табличная запись включает в себя поле для хранения результата фильтрации.

8. Способ по п.7, в котором табличная запись для дейтаграммы дополнительно включает в себя таймер, используемый для отсчета времени, прошедшего с тех пор, как был принят первый фрагмент для дейтаграммы.

9. Способ по п.7, дополнительно содержащий этап, на котором, если результат фильтрации для дейтаграммы недоступен, сохраняют принятый фрагмент во второй памяти.

10. Способ по п.9, в котором табличная запись для дейтаграммы дополнительно включает в себя указатель на местоположение во второй памяти, используемой для хранения фрагментов, принятых для дейтаграммы.

11. Способ по п.9, дополнительно содержащий этапы, на которых

определяют, было ли принято упомянутое множество фрагментов для дейтаграммы; и

если упомянутое множество фрагментов было принято, удаляют табличную запись в первой памяти и очищают вторую память, используемую для хранения принятых фрагментов дейтаграммы.

12. Способ по п.1, в котором определение результата фильтрации для дейтаграммы содержит этапы, на которых

определяют, доступны ли все из упомянутого по меньшей мере одного параметра фильтрации, и

применяют по меньшей мере один фильтр к упомянутому по меньшей мере одному параметру фильтрации, если таковой доступен, чтобы получить результат фильтрации.

13. Способ по п.1, дополнительно содержащий этап, на котором обнаруживают один или более сохраненных фрагментов, несущих упомянутый по меньшей мере один параметр фильтрации для дейтаграммы.

14. Способ по п.13, в котором упомянутый по меньшей мере один параметр фильтрации переносится в первом фрагменте дейтаграммы, при этом обнаружение выполняют на основе известного значения для поля в заголовке протокола для первого фрагмента дейтаграммы.

15. Способ по п.1, в котором упомянутый по меньшей мере один параметр фильтрации содержит адрес источника, адрес получателя, маску подсети, порт источника, порт получателя, диапазон портов источника, диапазон портов получателя, информацию о качестве обслуживания (QoS), тип обслуживания, класс трафика, метку потока, информацию о безопасности или их комбинацию.

16. Способ по п.2, в котором упомянутый по меньшей мере один параметр фильтрации для IP-дейтаграммы содержит IP-адрес источника, IP-адрес получателя, маску подсети, порт источника, порт получателя, диапазон портов источника, диапазон портов получателя, номер протокола, тип обслуживания для IP версии 4 (IPv4), класс трафика для IP версии 6 (IPv6), метку потока для IPv6, индекс параметра безопасности IPSec (SPI) или их комбинацию.

17. Способ обработки фрагментированных дейтаграмм в сети передачи данных, содержащий этапы, на которых

принимают фрагмент для дейтаграммы, отправленной как множество фрагментов;

если фрагмент является первым фрагментом, принятым для дейтаграммы, создают для дейтаграммы запись в таблице маршрутизации;

определяют, были ли получены все из по меньшей мере одного параметра фильтрации для дейтаграммы;

применяют по меньшей мере один фильтр к упомянутому по меньшей мере одному параметру фильтрации, если таковой доступен, чтобы получить результат фильтрации для дейтаграммы;

сохраняют результат фильтрации, если он получен, в упомянутой табличной записи;

сохраняют фрагмент в памяти, если результат фильтрации еще недоступен; и

обрабатывают фрагмент в соответствии с результатом фильтрации, если таковой доступен.

18. Устройство, выполненное с возможностью обработки фрагментированных дейтаграмм в сети передачи данных и содержащее

процессор, выполненный с возможностью

идентифицировать по меньшей мере один параметр фильтрации для дейтаграммы, отправленной как множество фрагментов,

сохранять один или более принятых фрагментов дейтаграммы в качестве сохраненных фрагментов,

получать упомянутый по меньшей мере один параметр фильтрации из упомянутых одного или более принятых фрагментов,

определять результат фильтрации для дейтаграммы на основе упомянутого по меньшей мере одного параметра фильтрации, причем результат фильтрации определяют до того, как будут приняты все фрагменты дейтаграммы,

обрабатывать сохраненные фрагменты на основе результата фильтрации, и

обрабатывать дополнительные фрагменты, принятые после того, как результат фильтрации определен на основе упомянутого по меньшей мере одного параметра фильтрации; и

первую память, выполненную с возможостью сохранять результат фильтрации для дейтаграммы.

19. Устройство по п.18, дополнительно содержащее

вторую память, выполненную с возможностью сохранять фрагменты, принятые для дейтаграммы, перед тем, как результат фильтрации станет доступным.

20. Устройство по п.19, в котором процессор дополнительно выполнен с возможностью получать упомянутый по меньшей мере один параметр фильтрации из по меньшей мере одного из заголовка протокола и полезной нагрузки, ассоциированных с упомянутыми одним или более сохраненными фрагментами данных.

21. Устройство по п.18, в котором процессор дополнительно выполнен с возможностью

принимать фрагмент для дейтаграммы, и

если фрагмент является первым фрагментом, принятым для дейтаграммы, создавать табличную запись для дейтаграммы в первой памяти, причем данная табличная запись включает в себя поле для хранения результата фильтрации.

22. Устройство по п.21, в котором процессор дополнительно выполнен с возможностью

определять, было ли принято упомянутое множество фрагментов для дейтаграммы, и

вычищать табличную запись в первой памяти для дейтаграммы, если упомянутое

множество фрагментов для дейтаграммы было принято.

23. Устройство по п.19, в котором процессор дополнительно выполнен с возможностью

определять, было ли принято упомянутое множество фрагментов для дейтаграммы, и

вычищать фрагменты, сохраненные во второй памяти для дейтаграммы, если упомянутое множество фрагментов для дейтаграммы было принято.

24. Устройство по п.19, в котором процессор дополнительно выполнен с возможностью

поддерживать таймер для дейтаграммы; и

вычищать фрагменты, сохраненные для дейтаграммы, после истечения таймера.

25. Устройство беспроводной связи, содержащее устройство по п.18.

26. Устройство, выполненное с возможностью обрабатывать фрагментированные дейтаграммы в сети передачи данных и содержащее

средство для идентификации по меньшей мере одного параметра фильтрации для дейтаграммы, отправленной как множество фрагментов;

средство для сохранения одного или более принятых фрагментов дейтаграммы в качестве сохраненных фрагментов;

средство для получения упомянутого по меньшей мере одного параметра фильтрации из упомянутых одного или более принятых фрагментов;

средство для определения результата фильтрации для дейтаграммы на основе упомянутого по меньшей мере одного параметра фильтрации, причем результат фильтрации определяют до того, как будут приняты все фрагменты дейтаграммы;

средство для обработки сохраненных фрагментов на основе результата фильтрации; и

средство для обработки дополнительных фрагментов, принятых после того, как результат фильтрации определен на основе упомянутого по меньшей мере одного параметра фильтрации.

27. Устройство по п.26, дополнительно содержащее средство для приема фрагмента для дейтаграммы, и

средство для создания табличной записи для дейтаграммы в памяти, если фрагмент является первым фрагментом, принятым для дейтаграммы, причем упомянутая табличная запись включает в себя поле для хранения результата фильтрации.

28. Устройство по п.26, дополнительно содержащее средство для получения упомянутого по меньшей мере одного параметра фильтрации из по меньшей мере одного из заголовка протокола и полезной нагрузки, ассоциированных с упомянутыми одним или более сохраненными фрагментами данных.

29. Устройство по п.26, дополнительно содержащее средство для поддержки таймера для дейтаграммы, и

средство для удаления сохраненных фрагментов для дейтаграммы после истечения таймера.

30. Машиночитаемый носитель для хранения инструкций, исполняемых в устройстве, чтобы:

идентифицировать по меньшей мере один параметр фильтрации для дейтаграммы, отправленной как множество фрагментов;

сохранять один или более принятых фрагментов дейтаграммы в качестве сохраненных фрагментов;

получать упомянутый по меньшей мере один параметр фильтрации из упомянутых

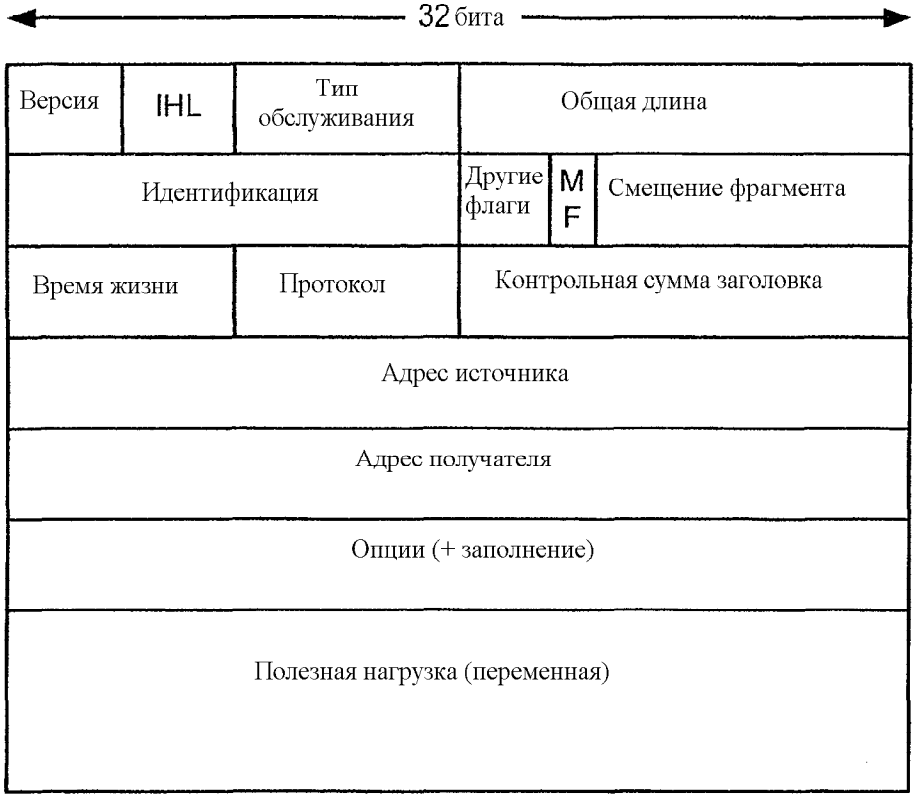
одного или более принятых фрагментов;

определять результат фильтрации для дейтаграммы на основе упомянутого по меньшей мере одного параметра фильтрации, причем результат фильтрации определяют до того, как будут приняты все фрагменты дейтаграммы;

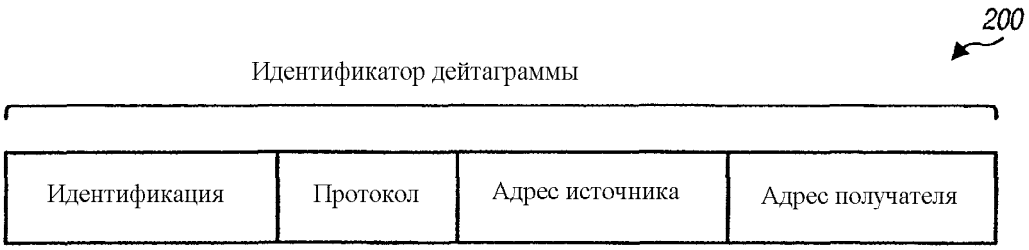
обрабатывать сохраненные фрагменты на основе результата фильтрации; и

обрабатывать дополнительные фрагменты, принятые после того, как результат фильтрации определен на основе упомянутого по меньшей мере одного параметра фильтрации.

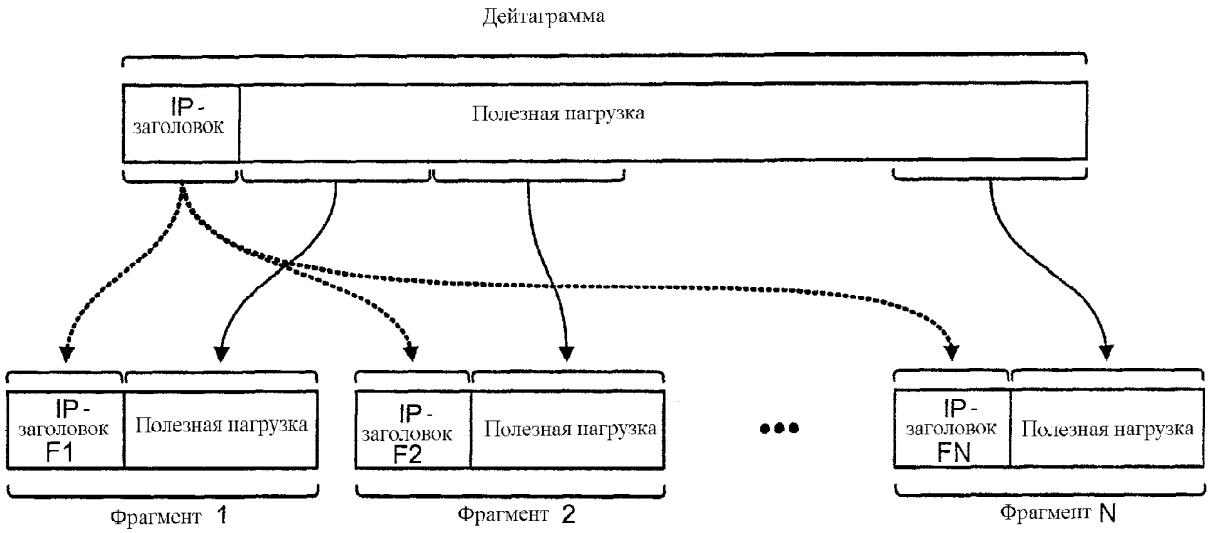
31. Машиночитаемый носитель по п.30, дополнительно хранящий инструкции, предназначенные для того, чтобы получать упомянутый по меньшей мере один параметр фильтрации из по меньшей мере одного из заголовка протокола и полезной нагрузки, ассоциированных с упомянутыми одним или более сохраненными фрагментами данных.



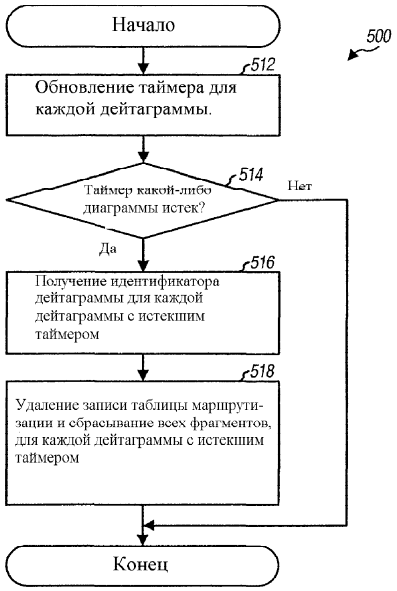
Фиг. 1



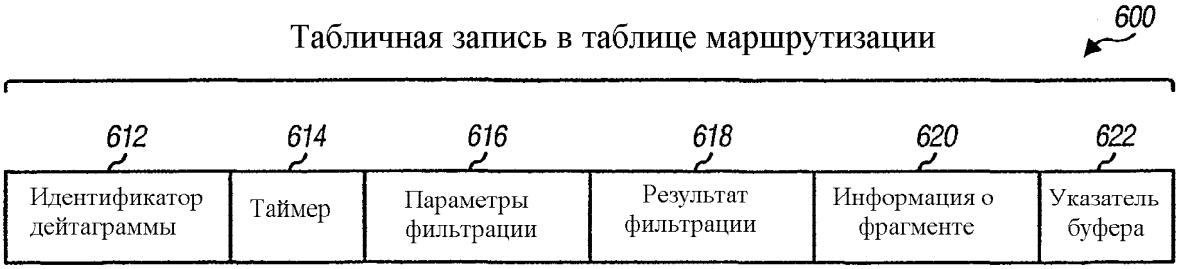
Фиг. 2



Фиг. 3

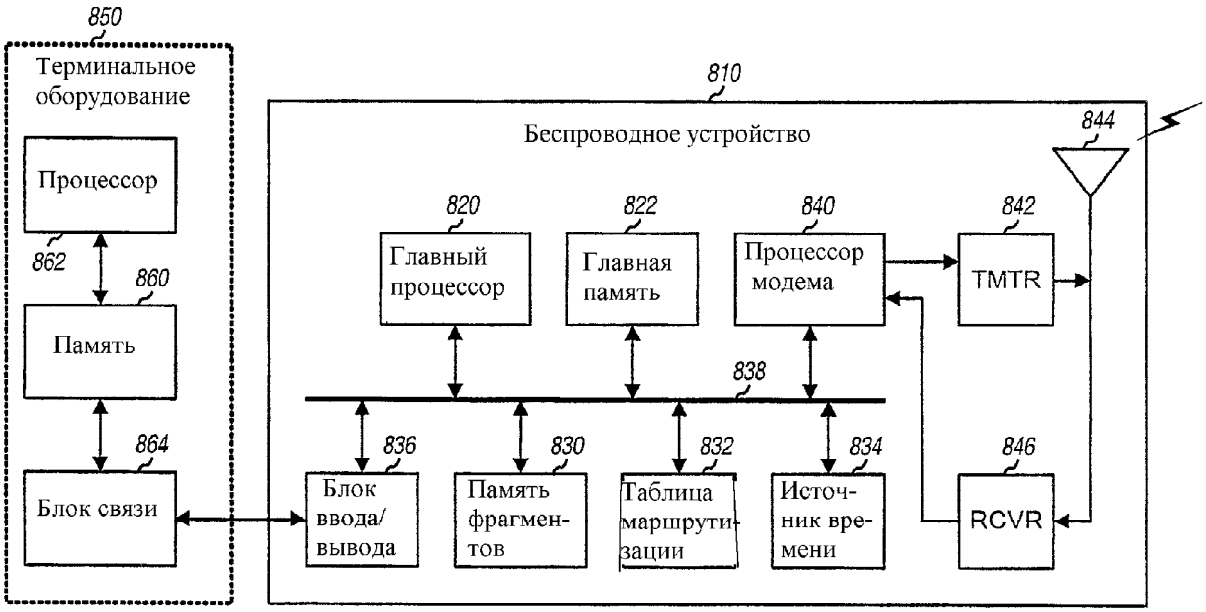


Фиг. 5



Фиг. 6

Фиг. 7


850
Терминальное оборудование
Процессор
862
860
Память
864
Блок связи
810
Беспроводное устройство
844
820
Главный процессор
822
Главная память
840
Процессор модема
842
TMTR
838
836
Блок ввода/вывода
830
Память фрагментов
832
Таблица маршрутизации
834
Источник времени
846
RCVR

Фиг. 8