РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 380 746** (13) **C2**

(51) МПК
G06F 15/163 (2006.01)
G06F 9/50 (2006.01)
G06F 15/177 (2006.01)
G11B 23/00 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2004117220/09, 07.06.2004**

(24) Дата начала отсчета срока действия патента:
07.06.2004

(30) Конвенционный приоритет:
30.06.2003 US 10/610,519

(43) Дата публикации заявки: **10.01.2006**

(45) Опубликовано: **27.01.2010** Бюл. № **3**

(56) Список документов, цитированных в отчете о поиске: **US 6330605 B1, 11.12.2001. RU 2156546 C2, 20.09.2000. WO 00/22526 A1, 20.04.2000. US 6546423 B1, 08.04.2003.**

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):
ДАРЛИНГ Кристофер Л. (US),
ДЖОЙ Джозеф М. (US),
ШРИВАСТАВА Сунита (US),
СУББАРАМАН Читтур (US)

(73) Патентообладатель(и):
МАЙКРОСОФТ КОРПОРЕЙШН (US)

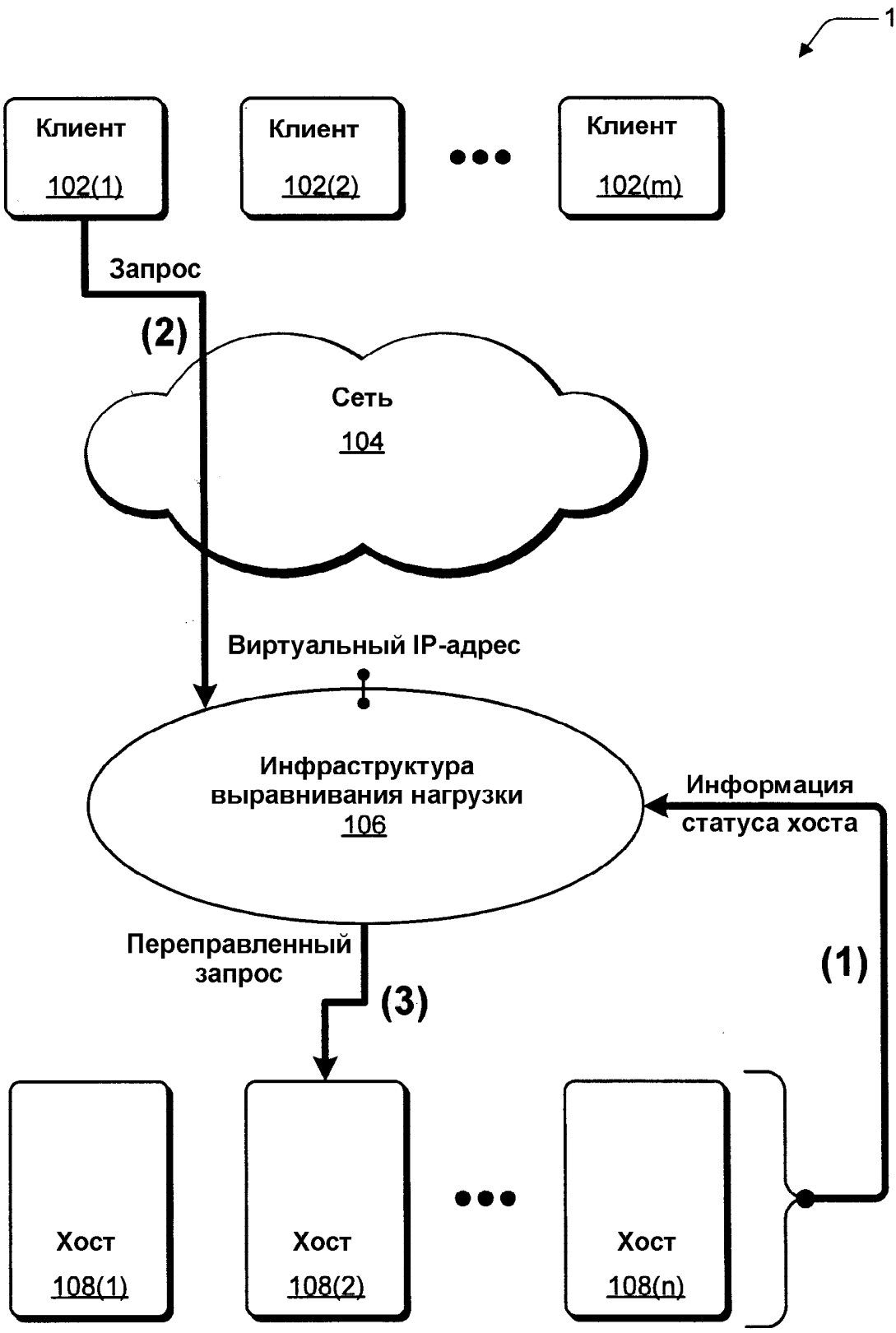
(54) ВЫРАВНИВАНИЕ СЕТЕВОЙ НАГРУЗКИ С ПОМОЩЬЮ ИНФОРМАЦИИ СТАТУСА ХОСТА

(57) Реферат:

Изобретение относится к выравниванию сетевой нагрузки. Техническим результатом является расширение функциональных возможностей. В первой реализации носитель, доступный процессору, содержит команды, которые при выполнении процессором предписывают системе осуществлять накопление информации статуса хоста на множественных хостах и отправку накопленной информации статуса хоста с множественных хостов. Во второй реализации носитель, доступный процессору, содержит команды, выполняемые процессором, которые предписывают системе осуществлять прием информации статуса хоста от множественных хостов и принятие решений по выравниванию нагрузки в соответствии с принятой информацией статуса хоста. В третьей реализации носитель, доступный процессору, содержит команды, выполняемые процессором, которые предписывают системе осуществлять действия: определение информации работоспособности и нагрузки для каждого приложения и выбор приложения из множественных в соответствии с информацией работоспособности и нагрузки. 4 н. и 72 з.п. ф-лы, 45 ил.

RU 2380746 C2

RU 2380746 C2



Фиг. 1

RU 2380746 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(19) **RU** (11) **2 380 746** (13) **C2**

(51) Int. Cl.
G06F 15/163 (2006.01)
G06F 9/50 (2006.01)
G06F 15/177 (2006.01)
G11B 23/00 (2006.01)

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2004117220/09, 07.06.2004**

(24) Effective date for property rights:
07.06.2004

(30) Priority:
30.06.2003 US 10/610,519

(43) Application published: **10.01.2006**

(45) Date of publication: **27.01.2010 Bull. 3**

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):
DARLING Kristofer L. (US),
DZhOJ Dzhozef M. (US),
ShRIVASTAVA Sunita (US),
SUBBARAMAN Chittur (US)

(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

RU 2380746 C2

(54) NETWORK LOAD BALANCING USING HOST STATUS INFORMATION

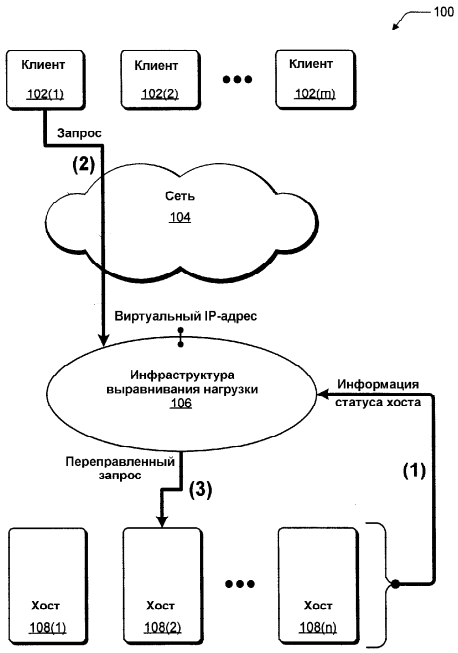
(57) Abstract:

FIELD: information technologies.

SUBSTANCE: in the first implementation, a media available for processor contains commands which being executed by processor instruct system to perform: accumulation of host status information on multiple hosts and sending accumulated host status information from multiple hosts. In the second implementation, a media available for processor contains commands executed by processor which instruct system to perform: receiving host status information from multiple hosts and making decisions on load balancing in accordance with received host status information. In the third implementation, a media available for processor contains commands executed by processor which instruct system to perform actions: determination of operability and load information for each application; and selection of application from multiple applications according to operability and load information.

EFFECT: functionality enhancement.

76 cl, 45 dwg



Фиг. 1

RU 2380746 C2

Область техники, к которой относится изобретение

Данное изобретение относится, в целом, к выравниванию сетевой нагрузки и, в частности, в порядке примера, но не ограничения, к выравниванию сетевой нагрузки с помощью информации статуса хоста.

Уровень техники

Интернет оказал большое влияние на связь и многие аспекты жизнедеятельности с использованием связи. Интернет обеспечивает быстрый и сравнительно простой обмен данными между двумя людьми или сущностями. Интернет включает в себя многочисленные сетевые узлы, связанные между собой, что обеспечивает возможность переноса информации между ними. Некоторые сетевые узлы могут представлять собой маршрутизаторы, которые переправляют пакет с одной линии связи на другую, могут представлять собой отдельные компьютеры-клиенты, могут представлять собой частные сети для различных сущностей (например, интрасети на предприятиях) и т.д.

В случае частной сети, а также в других случаях, пакеты, поступающие на узел или узлы Интернета, распределяются на другие узлы частной сети. Такая частная сеть может быть сформирована, например, из группы серверов, каждый из которых может обрабатывать пакеты, поступающие в частную сеть. В частные сети предприятий, университетов, государственных учреждений и т.д. в течение короткого временного интервала могут поступать многочисленные пакеты. Чтобы своевременно отвечать и снизить вероятность отказа или потери поступающих пакетов, в частной сети можно использовать множественные серверы, способные одновременно обрабатывать поступающие пакеты.

Поступающие пакеты часто являются запросами, относящимися к определенной информации, например, документу, элементу каталога, веб-странице и т.д. Поступающие пакеты также могут относиться к финансовой транзакции между покупателем и продавцом. В пакетной связи пакеты можно использовать и в других целях. В любом случае поступающие пакеты распределяются между различными серверами из группы серверов для обеспечения быстрого получения пакетов и/или организации сложной связи.

Распределение поступающих пакетов между серверами из группы серверов часто называют выравниванием сетевой нагрузки. Иными словами, операцию выравнивания нагрузки можно осуществлять над пакетами по мере их поступления на узел или узлы Интернета, когда узел или узлы составляют частную сеть и/или когда они соединяют частную сеть с Интернетом.

Такая операция выравнивания нагрузки осуществляется с использованием специального оборудования, которое находится «перед» частной сетью на узле или узлах, соединяющих частную сеть с Интернетом, и/или которое обеспечивает присутствие частной сети в Интернете. Физическое оборудование, которое осуществляет операцию выравнивания нагрузки, обычно полностью дублируется, чтобы реализовать избыточность и повысить надежность операции выравнивания нагрузки. Для повышения емкости операций выравнивания нагрузки прежнее оборудование выравнивания нагрузки заменяют более мощным оборудованием, которое полностью повторяет прежнее оборудование выравнивания нагрузки. Таким образом, расширение выравнивания нагрузки ограничивается увеличением мощности оборудования при его замене.

Для реализации операции выравнивания нагрузки оборудование обычно осуществляет циклическое распределение поступающих запросов соединения. Другими

словами, поступающие запросы соединения распределяются на серверы из группы серверов в линейном, повторяющемся порядке, при этом единичный запрос соединения распределяется на каждый сервер. Это циклическое распределение нагрузки для соединений обычно используется безотносительно к состоянию частной сети или характеру поступающих запросов соединения. Если операция выравнивания нагрузки не распространяется за пределы циклического распределения, эти другие факторы учитываются лишь постольку, поскольку их можно вывести из сетевого трафика и/или из уровня перегрузки частной сети.

Соответственно, требуются схемы и/или способы усовершенствования выравнивания сетевой нагрузки и/или связанных с этим операций.

Раскрытие изобретения

В первой проиллюстрированной реализации носителя один или несколько носителей, доступных процессору, содержат команды, выполняемые процессором, которые, при выполнении, предписывают системе осуществлять действия, которые включают в себя накопление информации статуса хоста на множественных хостах и отправку накопленной информации статуса хоста с множественных хостов. Во второй проиллюстрированной реализации носителя один или несколько носителей, доступных процессору, содержат команды, выполняемые процессором, которые, при выполнении, предписывают системе осуществлять действия, которые включают в себя прием информации статуса хоста от множественных хостов и принятие решений по выравниванию нагрузки в соответствии с принятой информацией статуса хоста. В третьей проиллюстрированной реализации носителя один или несколько носителей, доступных процессору, содержат команды, выполняемые процессором, которые, при выполнении, предписывают системе осуществлять действия, которые включают в себя определение информации работоспособности и нагрузки для каждого приложения и выбор приложения из множественных приложений в соответствии с информацией работоспособности и нагрузки.

В четвертой проиллюстрированной реализации носителя один или несколько носителей, доступных процессору, содержат команды, выполняемые процессором, которые, при выполнении, предписывают системе осуществлять действия, которые включают в себя анализ информации работоспособности и/или нагрузки для множественных конечных точек приложения и выявление маркерного распределения для множественных конечных точек приложения в соответствии с анализом.

В пятой проиллюстрированной реализации носителя один или несколько носителей, доступных процессору, содержат команды, выполняемые процессором, которые, при выполнении, позволяют системе реализовать протокол обмена сообщениями, используемый для передачи информации работоспособности и/или нагрузки между, по меньшей мере, одним хостом и одним или несколькими модулями выравнивания нагрузки.

В проиллюстрированной реализации системы система содержит по меньшей мере, одно устройство, хостирующее одно или несколько приложений, причем, по меньшей мере, одно устройство содержит таблицу работоспособности и нагрузки, содержащую множественные элементы, каждый элемент связан из множественных соединений с приложением из одного или нескольких приложений; каждый элемент из множественных элементов содержит идентификатор приложения для конкретного приложения из одного или нескольких приложений; информацию, характеризующую, по меньшей мере, один статус конкретного приложения; и, по меньшей мере, одну директиву выравнивания нагрузки, касающуюся конкретного приложения.

Здесь описаны и другие реализации способа, системы, подхода, аппарата, программного интерфейса приложения (API), устройства, носителя, процедуры, конфигурации и т.д.

Краткое описание чертежей

Для обозначения аналогичных и/или соответствующих аспектов, особенностей и компонентов на чертежах используются одинаковые позиции.

Фиг.1 - схема, иллюстрирующая выравнивание сетевой нагрузки, где показаны инфраструктура выравнивания нагрузки и множественные хосты.

Фиг.2 - схема, иллюстрирующая выравнивание сетевой нагрузки, где показаны множественные модули выравнивания нагрузки и множественные хосты.

Фиг.3 - схематическое иллюстративное изображение модуля выравнивания нагрузки, имеющего разделенные функции, и иллюстративного хоста.

Фиг.4 - схема, иллюстрирующая инфраструктуру выравнивания сетевой нагрузки с разделенными функциями классификации и пересылки.

Фиг.5 - логическая иллюстративная блок-схема способа расширения инфраструктуры выравнивания сетевой нагрузки в различные конфигурации.

Фиг.6 - иллюстративная схема первой конфигурации инфраструктуры выравнивания сетевой нагрузки в отношении устройств.

Фиг.7 - иллюстративная схема второй конфигурации инфраструктуры выравнивания сетевой нагрузки в отношении устройств.

Фиг.8А и 8В - иллюстративная схема первой и второй конфигураций инфраструктуры выравнивания сетевой нагрузки в отношении компонентов.

Фиг.9А и 9В - иллюстративная схема первой и второй конфигураций инфраструктуры выравнивания сетевой нагрузки с точки зрения ресурсов.

Фиг.10 - иллюстративная схема подхода к выравниванию сетевой нагрузки с использованием информации статуса хоста.

Фиг.11 - логическая иллюстративная блок-схема способа выравнивания сетевой нагрузки с использованием информации статуса хоста.

Фиг.12 - иллюстративная схема подхода к выравниванию сетевой нагрузки с использованием информации работоспособности и нагрузки.

Фиг.13А - иллюстративная таблица работоспособности и нагрузки, обозначенная на фиг.12.

Фиг.13В - иллюстративный объединенный кэш работоспособности и нагрузки, обозначенный на фиг.12.

Фиг.14 - логическая иллюстративная блок-схема способа выравнивания сетевой нагрузки с использованием информации работоспособности и нагрузки.

Фиг.15 - иллюстративная схема протокола обмена сообщениями для передач, показанных на фиг.12, между хостами и модулями выравнивания нагрузки.

Фиг.16 - иллюстративная схема передачи сообщения для передач, показанных на фиг.12, между хостами и модулями выравнивания нагрузки.

Фиг.17А и 17В - иллюстративные схемы сценариев хранения на посреднике информации работоспособности и нагрузки для таблиц работоспособности и нагрузки, показанных на фиг.13А, и для объединенных кэшей работоспособности и нагрузки, показанных на фиг.13 В, соответственно.

Фиг.18 - иллюстративная схема процедуры распределения для целевого хоста, в которой используется информация работоспособности и нагрузки.

Фиг.19 - иллюстративная схема подхода к выравниванию сетевой нагрузки с использованием информации сеанса.

Фиг.20 - иллюстративная схема подхода к выравниванию сетевой нагрузки с использованием передачи информации сеанса посредством извещений и сообщений.

Фиг.21 - логическая иллюстративная блок-схема способа выравнивания сетевой нагрузки с использованием передачи информации сеанса посредством извещений и сообщений.

Фиг.22 - иллюстративная схема подхода к управлению информацией сеанса на множественных модулях выравнивания нагрузки.

Фиг.23A - иллюстративная таблица сеансов, указанная на фиг.20.

Фиг.23B - иллюстративная таблица (TPДА) распределенного диспетчера атомов (РДА), указанная на фиг.22.

Фиг.24 - логическая иллюстративная блок-схема способа управления информацией сеанса на множественных модулях выравнивания нагрузки.

Фиг.25 - иллюстративная схема инфраструктуры выравнивания сетевой нагрузки с функцией маршрутизации запросов.

Фиг.26 - логическая иллюстративная блок-схема способа маршрутизации входящих пакетов в соответствии с (i) информацией сеанса и (ii) информацией работоспособности и нагрузки.

Фиг.27 - иллюстративная схема последовательности действий по маршрутизации трафика в отсутствие сбоев.

Фиг.28 - иллюстративная схема последовательности действий по маршрутизации трафика при наличии сбоя(ев).

Фиг.29 - иллюстративная схема дополнительных процедур преодоления сбоя для повышения надежности инфраструктуры выравнивания сетевой нагрузки.

Фиг.30 - иллюстрация эксплуатационной реализации взаимодействия маршрутизации трафика с информацией работоспособности и нагрузки.

Фиг.31 - иллюстрация механизмов обеспечения высокой надежности инфраструктуры выравнивания сетевой нагрузки.

Фиг.32 - иллюстрация подхода к выравниванию сетевой нагрузки с переносом соединений.

Фиг.33 - логическая иллюстративная блок-схема способа переноса соединения с первого устройства на второе устройство.

Фиг.34 - иллюстрация подхода к переносу соединений с точки зрения устройства-отправителя.

Фиг.35 - иллюстрация подхода к переносу соединений с точки зрения устройства-адресата.

Фиг.36 - иллюстрация подхода к процедуре выгрузки для переноса соединения.

Фиг.37 - иллюстрация подхода к процедуре загрузки для переноса соединения.

Фиг.38 - иллюстрация подхода к туннелированию пакетов между блоком пересылки и хостом.

Фиг.39 - логическая иллюстративная блок-схема способа туннелирования пакетов между первым устройством и вторым устройством.

Фиг.40 - иллюстративная схема операционной среды компьютера (или устройства общего назначения), которая способна (полностью или частично) реализовать, по меньшей мере, один аспект описанного здесь выравнивания сетевой нагрузки.

Осуществление изобретения

Схемы выравнивания сетевой нагрузки

В этом разделе описаны схемы выравнивания сетевой нагрузки, и он используется для обеспечения принципов, сред, контекстов и т.д. для описаний в следующих

разделах. В этом разделе ссылки, в основном, идут на фиг.1-3.

На фиг.1 изображена схема 100 выравнивания сетевой нагрузки, где показаны инфраструктура 106 выравнивания нагрузки и множественные хосты 108. Иллюстративная схема 100 выравнивания сетевой нагрузки включает в себя множественные клиенты 102(1), 102(2)... 102(m) и множественные хосты 108(1), 108(2)... 108(n), а также сеть 104 и инфраструктуру 106 выравнивания нагрузки.

Каждый клиент 102 может представлять собой устройство, способное к сетевой связи, например, компьютер, мобильную станцию, развлекательное устройство, другую сеть и т.д. Клиенты 102 могут также относиться к лицу или сущности, эксплуатирующему клиентское устройство. Иными словами, клиенты 102 могут представлять собой логические клиенты, которые являются пользователями, и/или машины. Сеть 104 может быть сформирована из одной или более сетей, например, Интернета, интрасети, проводной или беспроводной телефонной сети и т.д. Дополнительные примеры устройств для клиентов 102 и сетевые типы/топологии для сети 104 описаны ниже со ссылкой на фиг.40 в разделе «Операционная среда для компьютера или другого устройства».

Отдельные клиенты 102 способны связываться с одним или несколькими хостами 108 и, наоборот, по сети 104 через инфраструктуру 106 выравнивания нагрузки. Хосты 108 содержат одно или несколько приложений для взаимодействия/связи с клиентами 102, для использования клиентами 102 и т.д. Каждый хост 108 может соответствовать серверу и/или устройству, множественным серверам и/или множественным устройствам, части сервера и/или части устройства, некоторым их комбинациям и т.д. Частичные реализации хостов 108 описаны ниже применительно к различным случаям выравнивания сетевой нагрузки. (Однако поддержка прикладной части для хостов 108, для сохранения ясности существа изобретения не показана.) Дополнительные примеры устройств для хостов 108 описаны ниже со ссылкой на фиг.40 в разделе, озаглавленном «Операционная среда для компьютера или другого устройства».

Инфраструктура 106 выравнивания нагрузки достижима или обнаружима через сеть 104 по одному или нескольким виртуальным адресам Интернет-протокола (IP). Передачи с клиентов 102 (или других узлов), направленные по виртуальному IP-адресу инфраструктуры 106 выравнивания нагрузки, принимаются здесь и пересылаются на хост 108. Инфраструктура 106 выравнивания нагрузки состоит из аппаратных и/или программных компонентов (не показанных явно на фиг.1).

Хотя инфраструктура 106 показана как единый эллипс, инфраструктура для осуществления выравнивания нагрузки может также быть распределена на другие аспекты иллюстративной схемы 100 распределения сетевой нагрузки. Например, программный(е) компонент(ы) инфраструктуры 106 выравнивания нагрузки может/могут размещаться на одном или нескольких хостах 108 согласно описанному ниже. Примеры архитектур инфраструктуры 106 выравнивания нагрузки описаны ниже со ссылкой на фиг.40 в разделе, озаглавленном «Иллюстративная операционная среда для компьютера или другого устройства».

Согласно указанному позицией (1) один или несколько хостов 108 могут предоставлять информацию статуса хоста от хостов 108 инфраструктуре 106 выравнивания нагрузки. Эта информация статуса хоста может зависеть от приложения. Примеры такой информации статуса хоста описаны ниже и включают в себя информацию работоспособности и/или нагрузки, информацию сеанса и т.д. для хостов 108. Конкретная реализация, включающая в себя предоставление информации

работоспособности и/или нагрузки от хостов 108 инфраструктуре 106 выравнивания нагрузки, описана ниже в разделе, озаглавленном «Иллюстративное управление работоспособностью и нагрузкой».

Позицией (2) обозначен запрос, передаваемый от клиента 102(1) по сети 104 на инфраструктуру 106 выравнивания нагрузки по ее виртуальному IP-адресу. Содержимое, формат и т.д. запроса от клиента 102 может зависеть от приложения, которому адресован запрос, и термин «запрос» может неявно включать в себя ответ или ответы хоста(ов) 108 в зависимости от контекста. Виды клиентских запросов включают в себя, но не исключительно:

1. Запросы GET протокола передачи гипертекстовых файлов (HTTP) от клиента с использованием программы обозревателя. В зависимости от приложения (в частности, от унифицированного указателя ресурса (URL) запросов) может быть лучше обслуживать запросы разными группами хостов, и существование клиентского состояния «сеанса» на хостах может препятствовать маршрутизации этих запросов от конкретных клиентов на конкретные хосты. Запросы можно посылать посредством соединения, работающего по протоколу защищенных сокетов (SSL) (или другого зашифрованного соединения).

2. Соединения виртуальной частной сети (ВЧС) (например, хосты являются группой серверов ВЧС). В этом случае «запрос» можно рассматривать как «соединение» по протоколу туннелирования второго уровня (L2TP) или протоколу двухточечного туннелирования (PPTP) (последнее является комбинацией соединения по протоколу управления передачей (TCP) и соответствующих данных общей маршрутизации с инкапсуляцией (GRE).

3. Соединения терминального сервера (например, хосты являются группой терминальных серверов).

4. Собственнические запросы в виде индивидуальных TCP-соединений (по одному на запрос), использующие собственнический протокол, зависящий от приложения.

5. Запросы по простому протоколу доступа к объектам (SOAP).

6. Запросы на связь в режиме реального времени, предусматривающие управляющую информацию по TCP-соединению и потоковую передачу мультимедиа, чувствительную к задержке, по протоколу передачи в реальном времени (RTF).

Таким образом, запросы могут принимать разнообразные формы в зависимости от приложений. В некоторых описанных вариантах осуществления инфраструктура 106 выравнивания нагрузки может принимать решения на пересылку в зависимости от приложений.

Позиция (3) обозначает, что инфраструктура выравнивания нагрузки пересылает запрос от 102(1) на хост 108(2) (в данном примере). Инфраструктура 106 выравнивания нагрузки может учитывать один или несколько факторов при выборе хоста 108 для пересылки запроса в зависимости от того, какое/какие из описанных здесь реализаций используются. Например, инфраструктура 106 выравнивания нагрузки может учитывать: информацию работоспособности и нагрузки приложения для каждого хоста 108, информацию сеанса, относящуюся к клиенту 102(1), хранящуюся на хосте 108, и т.д.

На фиг.2 изображена схема 200 выравнивания сетевой нагрузки, где показаны множественные модули 106 выравнивания нагрузки и множественные хосты 108. В частности, в иллюстративной схеме 200 выравнивания сетевой нагрузки инфраструктура 106 выравнивания нагрузки показана в виде множественных модулей 106(1), 106(2),..., 106(u) выравнивания нагрузки. Кроме того, показаны два

маршрутизатора и/или коммутатора 202(1) и 202(2).

Маршрутизаторы/коммутаторы 202, если присутствуют, можно рассматривать в составе инфраструктуры 106 выравнивания нагрузки (фиг.1) или отдельно от нее. Маршрутизаторы/коммутаторы 202 отвечают за направление всех запросов и индивидуальных пакетов, поступающих из сети 104, по совместно используемому(ым) виртуальному(ым) IP-адресу(ам) (VIP) модулей 106 выравнивания нагрузки. В случае сбоя первого маршрутизатора/коммутатора 202 его функции берет на себя второй маршрутизатор/коммутатор 202. Хотя показаны два маршрутизатора/коммутатора 202, альтернативно можно использовать один или более двух маршрутизаторов/коммутаторов 202.

Маршрутизаторы/коммутаторы 202 могут не знать об инфраструктуре выравнивания нагрузки или могут знать о выравнивании нагрузки. Если маршрутизаторы/коммутаторы 202 не знают о выравнивании нагрузки, то можно использовать одну из двух опций. Первая из них состоит в том, что одному модулю 106 выравнивания нагрузки «присваивается» совместно используемый VIP-адрес, и весь сетевой трафик переправляется на него. Тогда этот один модуль 106 выравнивания нагрузки равномерно перераспределяет трафик по другим модулям 106 выравнивания нагрузки. Однако в связи с первой опцией возникают проблемы и вопросы преодоления сбоя (которые можно смягчить при наличии множественных совестно используемых VIP-адресов и разделения между множественными модулями 106 выравнивания нагрузки). Вторая опция заключается в том, что маршрутизаторы/коммутаторы 202 «обманываются» в направлении сетевого трафика на все модули 106 выравнивания нагрузки, каждый из которых самостоятельно решает, какой трафик принять для выравнивания нагрузки. Однако эта вторая опция приводит к недостаточному дублированию усилий и к проблемам производительности/совместимости коммутаторов.

Если же маршрутизаторы/коммутаторы 202 знают о выравнивании нагрузки, то маршрутизаторы/коммутаторы 202 можно заставить распределять входящий сетевой трафик между множественными модулями 106 выравнивания нагрузки (например, в циклическом режиме). Следует понимать, что такие маршрутизаторы/коммутаторы 202, знающие о выравнивании нагрузки, способны выполнять функции выравнивания нагрузки на рудиментарном уровне (например, аппаратно). Например, маршрутизаторы/коммутаторы 202, знающие о выравнивании нагрузки, могут осуществлять простое выявление сродства к сеансу на основе IP-адресов, чтобы все пакеты с конкретного IP-адреса источника направлялись на один и тот же модуль 106 выравнивания нагрузки.

Каждый отдельно показанный модуль 106 выравнивания нагрузки из модулей 106 выравнивания нагрузки может представлять одно физическое устройство, множественные физические устройства или часть единого физического устройства. Например, модуль 106(1) выравнивания нагрузки может соответствовать одному серверу, двум серверам или более. Альтернативно, модуль 106(1) выравнивания нагрузки и модуль 106(2) выравнивания нагрузки могут совместно соответствовать одному серверу. Иллюстративный модуль 106 выравнивания нагрузки описан ниже с точки зрения его функций со ссылкой на фиг.3.

На фиг.2 проиллюстрированы два пути [1] и [2] запроса. Что касается пути [1] запроса, клиент 102(2) передает запрос по сети 104, и этот запрос поступает на маршрутизатор/коммутатор 202(1). Маршрутизатор/коммутатор 202(1) направляет пакет(ы) запроса, поступающие от клиента 102(2), на модуль 106(1) выравнивания

нагрузки. Затем модуль 106(1) выравнивания нагрузки пересылает пакет(ы) запроса на хост 108(1) в соответствии с некоторыми функциями выравнивания нагрузки (например, политикой). Что касается пути [2] запроса, клиент 102(m) передает запрос по сети 104, и этот запрос поступает на маршрутизатор/коммутатор 202(2). Маршрутизатор/коммутатор 202(2) направляет пакет(ы) запроса, поступающие от клиента 102(m), на модуль 106(u) выравнивания нагрузки. Затем модуль 106(u) выравнивания нагрузки пересылает пакет(ы) запроса на хост 108(n) в соответствии с некоторыми функциями выравнивания нагрузки. Иллюстративные функции выравнивания нагрузки описаны ниже со ссылкой на фиг.3.

На фиг.3 проиллюстрирован модуль 106 выравнивания нагрузки с разделенными функциями и иллюстративный хост 108. Модуль 106 выравнивания нагрузки содержит семь (7) функциональных блоков 302-314. Эти функциональные блоки модуля 106 выравнивания нагрузки могут быть реализованы, по меньшей мере частично, программными средствами. Хост 108 включает в себя одно или несколько приложений 316. В описываемом варианте осуществления модуль 106 выравнивания нагрузки содержит блок 302 пересылки, классификатор 304, маршрутизатор 306 запросов, блок 308 отслеживания сеансов, блок 310 переноса соединений, блок 312 туннелирования и обработчик 314 работоспособности и нагрузки.

Обработчик 314 работоспособности и нагрузки размещен частично на хостах 108 и частично на устройствах модулей 106 выравнивания нагрузки. Обработчик 314 работоспособности и нагрузки отслеживает работоспособность и/или нагрузку (или, более обще, статус) хостов 108, чтобы его информацию работоспособности и/или нагрузки можно было использовать для функций выравнивания нагрузки (например, при принятии решений на выравнивание нагрузки). Иллюстративные реализации обработчика 314 работоспособности и нагрузки описаны ниже, в частности, в разделе, озаглавленном «Обработка работоспособности и нагрузки».

Блок 308 отслеживания сеансов также может размещаться частично на хостах 108 и частично на устройствах модулей 106 выравнивания нагрузки. Блок 308 отслеживания сеансов отслеживает сеансы, установленные клиентами 102, чтобы облегчить функциям выравнивания нагрузки восстановление/продолжение ранее установленных сеансов. Например, некоторые приложения поддерживают на хостах зависящие от приложения данные клиентского сеанса (которые также являются своего рода информацией статуса). Эти приложения обычно ожидают, что клиенты используют один и тот же хост на протяжении любого данного сеанса. Иллюстративные типы сеансов включают в себя: (i) TCP-соединение (которое, строго говоря, является сеансом); (ii) SSL-сеанс; (iii) защищенный IP-сеанс (IPsec); (iv) сеанс на основе куки HTTP и т.д.

Хотя блок 308 отслеживания сеансов показан в виде отдельного блока в модуле 106 выравнивания нагрузки, функция отслеживания сеансов блока 308 отслеживания сеансов, в действительности, может быть реализована на глобальном уровне. Иными словами, сродство к сеансу поддерживается по множественным модулям 106 выравнивания нагрузки. Блок 308 отслеживания сеансов включает в себя централизованную базу данных и/или распределенную базу данных информации сеансов для сохранения сродства к сеансу. Иллюстративные реализации блока 308 отслеживания сеансов, с упором на подход распределенной базы данных, описаны ниже, в частности, в разделе, озаглавленном «Иллюстративное отслеживание сеансов».

Классификатор 304 использует данные, собираемые и поддерживаемые обработчиком 314 работоспособности и нагрузки и/или блоком 308 отслеживания

сеансов, возможно, совместно с другими факторами, для классификации входящих запросов. Другими словами, классификатор 304 выделяет хост 108 назначения для каждого входящего запроса от клиента 102. Блок 302 пересылки пересылает запросы клиента (и/или его пакеты) в соответствии с хостом 108 назначения, выбранным классификатором 304. Блок 302 пересылки и классификатор 304 описаны ниже, в частности, в разделах, озаглавленных «Иллюстративный подход к гибкому выравниванию сетевой нагрузки» и «Иллюстративные классификация, пересылка и маршрутизация запросов».

Маршрутизатор 306 запросов, в отличие от пакетно-ориентированных реализаций блока 302 пересылки и классификатора 304, может действовать как посредник для приложения, действующего на хосте 108. Например, маршрутизатор 306 запросов может заканчивать TCP-соединения, анализировать (возможно, частично) каждый логический запрос от клиента 102 и перенаправлять каждый логический запрос на хост 108 назначения. Следовательно, каждый логический запрос от клиента 102 можно направлять на тот или иной хост 108 в зависимости от решений, принятых маршрутизатором 306 запросов. Кроме того, маршрутизатор 306 запросов может осуществлять предварительную обработку на соединении (например, дешифровку SSL), может выбирать поглощение определенных запросов (например, потому что маршрутизатор 306 запросов поддерживает кэш ответов), может по своему усмотрению изменять запросы прежде, чем пересылать их на хосты 108, и т.д. Иллюстративные реализации маршрутизатора 306 запросов также описаны ниже, в частности, в разделах, озаглавленных «Иллюстративный подход к гибкому выравниванию сетевой нагрузки» и «Иллюстративные классификация, пересылка и маршрутизация запросов».

Блок 310 переноса соединений обеспечивает сначала окончание соединения на модуле 106 выравнивания нагрузки, а затем его перенос, чтобы соединение впоследствии заканчивалось на хосте 108. Этот перенос соединений может облегчать выравнивание нагрузки на уровне приложений. Блок 310 переноса соединений способен переносить соединение от модуля 106 выравнивания нагрузки на хост 108 таким образом, чтобы первоначальное окончание на модуле 106 выравнивания нагрузки было прозрачно для запрашивающего клиента 102 и приложениям 316 вновь оканчивающего хоста 108. Блок 312 туннелирования может использовать схему инкапсуляции для туннелирования пакетов, которая не вносит избыточной нагрузки в каждый туннелируемый пакет.

Функции блока 312 туннелирования можно также использовать в случаях, которые не предусматривают перенос соединения. Кроме того, блок 310 переноса соединения и/или блок 312 туннелирования можно дополнительно использовать в реализациях без выравнивания нагрузки. Иллюстративные реализации блока 310 переноса соединения, а также блока 312 туннелирования описаны ниже, в частности, в разделе, озаглавленном «Перенос соединения с необязательным туннелированием и/или выравниванием нагрузки на уровне приложений».

Любая данная реализация модуля 106 выравнивания нагрузки может включать в себя одну или несколько из проиллюстрированных функций. Хотя каждая из функций блоков 302-314 показана по отдельности, она в действительности, может быть взаимосвязана с, перекрываться с другими функциями и/или входить в их состав. Например, классификатор 304 может использовать информацию работоспособности и/или нагрузки обработчика 314 работоспособности и нагрузки. Кроме того, блок 310 переноса соединения и блока 312 туннелирования работают совместно с блоком 302

пересылки и классификатором 304. Ниже описаны определенные другие иллюстративные перекрытия и взаимодействия.

В описанной реализации на хосте 108 работает одно или несколько приложений 316 и предоставляется доступ к ним. В целом, приложения 316 включают в себя программы доставки файлов, программы управления/сервера веб-сайтов, программы удаленного доступа, программы электронной почты, программы доступа к базам данных и т.п. В частности, приложения 316 могут включать в себя, но не исключительно, Internet Information Server® (IIS) от корпорации Microsoft®, терминальные серверы, например, Terminal Server™ фирмы Microsoft® и продукты брандмауэра и посредника, например, Internet Security and Acceleration Server™ (ISA). Хотя конкретные примеры приложений 316 в предыдущем предложении относятся к продуктам Майкрософт, описанное здесь выравнивание сетевой нагрузки не ограничивается каким-либо конкретным поставщиком, приложением или операционной системой.

Подход к гибкому выравниванию сетевой нагрузки

В этом разделе показано, как реализуется выравнивание сетевой нагрузки, описанное в этом и других разделах, обеспечивающее гибкий подход к выравниванию сетевой нагрузки. В этом разделе ссылки идут, в основном, на фиг.4-9B.

Как отмечено выше, функции выравнивания сетевой нагрузки можно расширить, заменив первый выравниватель сетевой нагрузки вторым, большим и более мощным выравнивателем сетевой нагрузки. Аппаратные возможности второго выравнивателя сетевой нагрузки повторяют все аппаратные возможности первого выравнивателя сетевой нагрузки за исключением обеспечения более высокой емкости. Это негибкий подход, который может быть весьма неэффективным, особенно, когда единственным признаком выравнивания сетевой нагрузки является ограничение производительности и обусловливание обновления выравнивателя сетевой нагрузки.

На фиг.4 показана инфраструктура выравнивания сетевой нагрузки, имеющая раздельные функции классификации и пересылки. Разделенные функция классификации и функция пересылки представлены классификатором 304 и блоком 302 пересылки соответственно. Хотя функции классификации и пересылки описаны ниже, особенно в разделе, озаглавленном «Иллюстративные классификация, пересылка и маршрутизация запросов», первичное описание представлено здесь как пример взаимодействия между функциональными блоками инфраструктуры выравнивания сетевой нагрузки и хостами 108.

В описанной реализации блок 302 пересылки соответствует виртуальному IP-адресу (VIP) (или адресам) и является конечной точкой сети для него (них). Блок 302 пересылки является относительно низкоуровневым компонентом, который принимает упрощенные и/или элементарные политические решения, если принимает, при маршрутизации пакетов на дальнейший или окончательный пункт назначения. Блок 302 пересылки сверяется с таблицей маршрутизации для определения этого пункта назначения. Классификатор 304 заполняет таблицу маршрутизации на основании одного или нескольких факторов (например, информации статуса хоста), которые описаны в других разделах.

Клиенты 102 и хосты 108 также соответствуют указанным сетевым адресам. В частности, клиент 102(1) соответствует адресу C1, клиент 102(2) соответствует адресу C2, …, клиент 102(m) соответствует адресу Cm. Кроме того, хост 108(1) соответствует адресу H1, хост 108(2) соответствует адресу H2, …, хост 108(n) соответствует адресу Hn.

На фиг.4 показано пять путей связи (1)-(5). Путь (1) связи проходит между клиентом 102(1) и блоком 302 пересылки, а путь (5) связи проходит между блоком 302 пересылки и хостом 108(1). Пути (2)-(5) связи проходят между блоком 302 пересылки и классификатором 304. Для простоты в этом примере соединение, связанное с путями (1)-(5) связи, будем считать TCP-соединениями HTTP. Кроме того, выравнивание нагрузки в этом примере относится к маршрутизации входящих соединений с, по меньшей мере, загруженным хостом 108, по меньшей мере, без какого бы то ни было явного учета выравнивания нагрузки на уровне приложений.

Пути (1)-(5) связи указывают, как блок 302 пересылки и классификатор 304 выравнивают нагрузку одного TCP-соединения HTTP от клиента 102(1). На пути (1) клиент 102(1) инициирует TCP-соединение, отправляя пакет SYN TCP, адресованный по VIP-адресу. Маршрутизирующая инфраструктура сети 104 маршрутизирует этот пакет на блок 302 пересылки через маршрутизатор/коммутатор 202(1), который является «ближайшим» маршрутизатором/коммутатором 202 к блоку 302 пересылки.

На пути (2) блок 302 пересылки сверяется с таблицей маршрутизации, которая может быть внутренней по отношению к блоку 302 пересылки или иным образом доступной для него, чтобы найти это соединение. Это соединение может быть идентифицировано в таблице маршрутизации упорядоченной четверкой TCP/IP (т.е. IP-адрес источника, TCP-порт источника, IP-адрес назначения, TCP-порт назначения). Поскольку это первый пакет соединения, в таблице маршрутизации нет ни одного элемента. Поэтому блок 302 пересылки применяет действие «маршрут по умолчанию», согласно которому этот пакет нужно переслать на классификатор 304.

На пути (3) классификатор 304 сверяется со своим (например, объединенным) кэшем информации статуса хоста для хостов 108(1), 108(2), …, 108(n). Классификатор 304 приходит к выводу, что хост 108(1) имеется в наличии и является последним нагруженным хостом 108 в этот момент в этом примере. Классификатор 304 также «прокладывает» маршрут в таблице маршрутизации, к которой обращается блок 302 пересылки на предмет этого TCP-соединения. Например, классификатор 304 добавляет элемент маршрута или предписывает блоку 302 пересылки добавить элемент маршрута в таблицу маршрутизации, который отображает TCP-соединение (например, идентифицированное упорядоченной четверкой TCP) с конкретным хостом 108 назначения, который в этом примере является хостом 108(1). В частности, элемент маршрута указывает сетевой адрес H1 хоста 108(1).

На пути (4) классификатор 304 отправляет пакет SYN TCP обратно на блок 302 пересылки. Альтернативно, классификатор 304 может пересылать этот начальный пакет SYN TCP на хост 108(1) без использования блока 302 пересылки. Другие доступные опции классификатора 304 описаны ниже.

На пути (5) блок 302 пересылки может осуществлять доступ к элементу маршрута для соединения, представленного пакетам SYN, чтобы переслать пакет на хост 108(1) по адресу H1. Блок 302 пересылки также пересылает все последующие пакеты от клиента 102(1) для этого соединения непосредственно на хост 108(1). Другими словами, блок 302 пересылки может избежать дальнейшего взаимодействия с классификатором 304 для этого соединения. Для удаления элемента маршрута по завершении соединения можно использовать один или несколько механизмов, описанных ниже.

Что касается пути (5), во многих средах протоколов блок 302 пересылки не может просто отправлять пакеты от клиента 102(1) как есть на хост 108(1) по сетевому

адресу H1, поскольку эти пакеты адресованы по VIP-адресу, который хостирован самим блоком 302 пересылки. Вместо этого блок 302 пересылки может использовать одну или несколько из следующих иллюстративных опций.

1. Блок 302 пересылки осуществляет трансляцию сетевых адресов (TCA) (i), записывая вместо IP-адреса (C1) (клиента 102(1)) и номера порта источника IP-адрес и номер порта, генерированный путем TCA, блока 302 пересылки и записывая вместо IP-адреса (VIP) назначения IP-адрес (H1) хоста (108(1)).

2. Блок 302 пересылки осуществляет «полу-TCA», записывая вместо IP-адреса (VIP) назначения IP-адрес (H1) хоста (108(1)), что позволяет сохранить IP-адрес (C1) и номер порта источника (клиента 102(1)).

3. Блок 302 пересылки «туннелирует» пакеты, полученные от клиента 102(1) с блока 302 пересылки, на хост 108(1). В частности, в этом примере туннелирование можно осуществлять путем инкапсуляции каждого пакета в новом IP-пакете, который адресован хосту 108(1). Программное обеспечение, знающее о выравнивании сетевой нагрузки на хосте 108(1), реконструирует исходный пакет при получении на блоке 302 пересылки от клиента 102(1). Затем этот исходный пакет указывается на виртуальном интерфейсе на хосте 108(1) (например, VIP-адрес, соответствующий блоку 102 пересылки, привязывается к этому виртуальному интерфейсу на хосте 108(1)). Иллюстративные реализации такого туннелирования описаны ниже со ссылкой на блок 312 туннелирования, особенно для сценариев переноса соединения и, в частности, в разделе, озаглавленном «Перенос соединений с необязательным туннелированием и/или выравниванием нагрузки на уровне приложений».

Хотя на фиг.4-9B показаны две конкретные разделенные функции, а именно классификация и пересылка, следует понимать, что и другие функции, например, функции маршрутизатора 306 запросов, блока 308 отслеживания сеансов, блока 310 переноса соединений и обработчика 314 работоспособности и нагрузки, также можно расширять независимо (например, независимо пропорционально увеличить), как описано ниже. Кроме того, следует заметить, что одну или более двух функций можно отделять и расширять независимо в разные моменты времени и/или одновременно. Кроме того, хотя для ясности во многих примерах в этом и других разделах используется TCP/IP, описанные здесь принципы выравнивания сетевой нагрузки применимы к другим протоколам передачи и/или связи.

Согласно примеру, показанному на фиг.4, функции выравнивания сетевой нагрузки (например, показанные на фиг.3) можно отделять друг от друга в целях расширения. Их также можно разделять и дублировать в различные конфигурации для повышения надежности. Иллюстративные конфигурации для масштабируемости и/или надежности описаны ниже со ссылкой на фиг.6-9B после описания способа со ссылкой на фиг.5.

На фиг.5 показана логическая блок-схема 500 способа расширения инфраструктуры выравнивания сетевой нагрузки в разные конфигурации. Логическая блок-схема 500 содержит три блока 502-506. Хотя действия логической блок-схемы 500 могут выполняться в других средах посредством разнообразных программных схем, фиг.1-4 и 6-9B используются, в частности, для иллюстрации определенных аспектов и примеров способа.

На блоке 502 инфраструктура выравнивания сетевой нагрузки эксплуатируется в первой конфигурации. Например, каждая конфигурация может относиться к одной или нескольким из выделения, пропорции и/или взаимоотношения различных функций выравнивания нагрузки; к ряду и/или типу(ам) различных устройств; к организации

и/или схеме различных компонентов; к распределению и/или выделению ресурсов; и т.п. На блоке 504 осуществляется расширение инфраструктуры выравнивания сетевой нагрузки. Например, отдельные функции выравнивания нагрузки могут расширяться и/или согласованно сокращаться на индивидуальной и/или независимой основе. На блоке 506 расширенная инфраструктура выравнивания сетевой нагрузки эксплуатируется во второй конфигурации.

Как отмечено выше, монолитный выравниватель сетевой нагрузки можно расширить, увеличив все функции выравнивания сетевой нагрузки путем замены прежнего оборудования выравнивания нагрузки более мощным оборудованием выравнивания сетевой нагрузки. Напротив, расширение инфраструктуры выравнивания сетевой нагрузки может позволить индивидуально и/или независимо расширять (под-)функции выравнивания сетевой нагрузки. Это также может позволить расширять функции выравнивания сетевой нагрузки совместно или по отдельности между разными количествами устройств. Ниже приведены примеры расширения в отношении устройств, компонентов и ресурсов.

На фиг.6 показана первая конфигурация инфраструктуры выравнивания сетевой нагрузки в отношении устройств. В этой первой конфигурации инфраструктуры выравнивания сетевой нагрузки, ориентированной на устройство, показаны три устройства 602(1), 602(2) и 602(3). Однако, альтернативно, можно использовать одно, два или более трех устройств 602.

Показано, что на устройстве 602(1) присутствуют и выполняются блок 302(1) пересылки, классификатор 304(1) и хост 108(1). На устройстве 602(2) присутствуют блок 302(2) пересылки, классификатор 304(2) и хост 108(2). Кроме того, на устройстве 602(3) присутствуют блок 302(3) пересылки, классификатор 304(3) и хост 108(3). Таким образом, в этой первой конфигурации инфраструктуры выравнивания сетевой нагрузки, ориентированной на устройство, соответствующие блок 302 пересылки, классификатор 304 и хост 108 совместно используют ресурсы каждого соответствующего устройства 602.

В ходе эксплуатации блоки 302 пересылки являются конечными точками для VIP-адреса(ов). Любой классификатор 304 может прокладывать маршрут для соединения с любым хостом 108 в зависимости от информации статуса хоста. Например, классификатор 304(2) может прокладывать маршрут для нового входящего соединения с хостом 108(3). В соответствии с новым элементом маршрута для этого соединения блок 302(2) пересылки пересылает следующие пакеты на хост 108(3).

В одной альтернативной конфигурации инфраструктуры выравнивания сетевой нагрузки в отношении устройств, в которую может быть расширена иллюстративная первая конфигурация, может быть добавлено четвертое устройство 602(4) (не показанное явно на фиг.6), которое содержит блок 302(4) пересылки, классификатор 304(4) и хост 108(4). Если же достаточные функции классификации уже присутствуют с классификаторами 304(1-3), но дополнительные функции могут способствовать обработке запросов на хостах 108, может быть добавлено четвертое устройство 602(4), которое содержит блок 302(4) пересылки и, в необязательном порядке, хост 108(4). Для этой расширенной конфигурации другой классификатор 304(1, 2 или 3) может прокладывать маршруты для блока 302(4) пересылки к любому из хостов 108(1, 2 или 3) и хосту 108(4), если имеется.

Первая проиллюстрированная конфигурация инфраструктуры выравнивания сетевой нагрузки, ориентированная на устройство, изображенная на фиг.6, может

быть особенно пригодна для ситуаций меньшего хостинга, в которых раздельные устройства для инфраструктуры выравнивания сетевой нагрузки не являются технически и/или экономически дающими надлежащие результаты или жизнеспособными. Однако когда нагрузка хостинга расширяется до большего количества (и/или большей потребности в том же количестве) хостов 108 или, если сетевая нагрузка на хосты 108 значительна, первая иллюстративная конфигурация инфраструктуры выравнивания сетевой нагрузки, ориентированная на устройство, может быть преобразована для приспособления к этому расширению, представленному второй иллюстративной конфигурацией инфраструктуры выравнивания сетевой нагрузки, ориентированной на устройство, показанной на фиг.7.

На фиг.7 показана вторая конфигурация инфраструктуры выравнивания сетевой нагрузки в отношении устройств. В этой второй конфигурации инфраструктуры выравнивания сетевой нагрузки, ориентированной на устройство, также показаны три устройства 602(1), 602(2) и 602(3). Опять же, иллюстративно, можно использовать одно, два или более трех устройств 602.

Показано, что на устройстве 602(1) присутствуют и выполняются блок 302(1) пересылки и классификатор 304(1). На устройстве 602(2) присутствуют блок 302(2) пересылки и классификатор 304(2). Кроме того, на устройстве 602(3) присутствуют блок 302(3) пересылки и классификатор 304(3). Таким образом, в этой второй конфигурации инфраструктуры выравнивания сетевой нагрузки, ориентированной на устройство, каждый соответствующий блок 302 пересылки и классификатор 304 не используют совместно ресурсы каждого соответствующего устройства 602 с хостом 108. Кроме того, инфраструктура выравнивания сетевой нагрузки может обслуживать любое количество хостов 108.

В ходе эксплуатации блоки 302 пересылки опять же являются конечными точками для VIP-адреса(ов). Кроме того, любой классификатор 304 может прокладывать маршрут для соединения с любым хостом 108 в зависимости от информации статуса хоста. Например, классификатор 304(3) может прокладывать маршрут для нового входящего соединения с хостом 108(2). В соответствии с новым элементом маршрута для этого соединения блок 302(3) пересылки пересылает последующие пакеты данных на хост 108(2).

Поэтому инфраструктуру выравнивания сетевой нагрузки, реализованную программными средствами, можно расширять, перемещая инфраструктуру выравнивания сетевой нагрузки (или ее часть) с устройств, используемых совместно с хостами 108, на устройства, не используемые совместно с хостами 108. Кроме того, согласно упомянутому выше со ссылкой на фиг.6 к инфраструктуре выравнивания сетевой нагрузки можно добавить еще одно устройство 602(4) для обеспечения дополнительных функций пересылки, дополнительных функций классификации, дополнительных функций обоих типов и т.д.

На фиг.8А и 8В показаны первая и вторая конфигурации инфраструктуры выравнивания сетевой нагрузки в отношении компонентов. Изображено, что первая компонентно-ориентированная иллюстративная конфигурация 800 инфраструктуры выравнивания сетевой нагрузки содержит четыре компонента. Вторая компонентно-ориентированная иллюстративная конфигурация 850 инфраструктуры выравнивания сетевой нагрузки содержит шесть компонентов. Альтернативная вторая конфигурация 850 содержит седьмой компонент, обозначенный блоком, изображенным пунктирной линией, который будет описан ниже.

В частности, первая компонентно-ориентированная конфигурация 800 инфраструктуры выравнивания сетевой нагрузки (или первая конфигурация 800) содержит (i) два блока 302(1) и 302(2) пересылки и (ii) два классификатора 304(1) и 304(2). Вторая компонентно-ориентированная иллюстративная конфигурация 850 инфраструктуры выравнивания сетевой нагрузки (или вторая конфигурация 850) содержит (i) четыре блока 302(1), 302(2), 302(3) и 302(4) пересылки и (ii) два классификатора 304(1) и 304(2). Таким образом, первая конфигурация 800 расширяется во вторую конфигурацию 850 путем добавления двух компонентов, которые в данном примере являются компонентами пересылки.

В описанной реализации каждый функциональный компонент, относящийся к выравниванию сетевой нагрузки, соответствует соответствующему устройству (не показанному явно на фиг.8A и 8B); однако каждый компонент может альтернативно соответствовать части устройства или более чем одному устройству. Например, блоки 302(1) и 302(2) пересылки могут быть распределены по трем устройствам. В другом случае блок 302(1) пересылки и классификатор 304(1) может соответствовать первому устройству, а блок 302(2) пересылки и классификатор 304(2) может соответствовать второму устройству.

Для расширения первой конфигурации 800 во вторую конфигурацию 850 добавляются два функциональных компонента, относящихся к выравниванию сетевой нагрузки. Однако для расширения инфраструктуры выравнивания сетевой нагрузки можно альтернативно добавлять один компонент (или более двух). Кроме того, можно «одновременно» расширять два или более функциональных компонентов разных типов. Например, при расширении первой конфигурации 800 во вторую конфигурацию 850 можно также добавить еще один компонент классификации (например, классификатор 304(3)), обозначенный блоком, изображенным пунктирной линией.

Кроме того, расширение двух или более функциональных компонентов разных типов можно осуществлять в подобных (например, эквивалентных) или неподобных пропорциях друг к другу. Показано, что добавление компонентов 302(3) и 302(4) без добавления какого-либо компонента 304 классификации или с добавлением одного компонента 304(3) классификации представляет расширение в неподобных пропорциях. Однако для расширения в подобных пропорциях при добавлении двух компонентов 302(3) и 302(4) пересылки можно добавлять два компонента 304(3) и 304(4) классификации (последний случай не показан явно на фиг.8B). Тем не менее, разные функциональные компоненты, относящиеся к выравниванию сетевой нагрузки, могут потреблять разные объемы доступных ресурсов инфраструктуры выравнивания сетевой нагрузки, что описано со ссылкой на фиг.9A и 9B.

На фиг.9A и 9B показаны первая и вторая иллюстративные конфигурации инфраструктуры выравнивания сетевой нагрузки с точки зрения ресурсов. Первая ресурсо-ориентированная иллюстративная конфигурация 900 инфраструктуры выравнивания сетевой нагрузки (или первая конфигурация 900) содержит первое распределение или выделение ресурсов для модуля 106 выравнивания нагрузки. Вторая ресурсо-ориентированная иллюстративная конфигурация 950 инфраструктуры выравнивания сетевой нагрузки (или первая конфигурация 950) содержит второе распределение или выделение ресурсов для модуля 106 выравнивания нагрузки.

Показано, что первая конфигурация 900 содержит распределение ресурсов 70%/30%, а вторая конфигурация 950 содержит распределение ресурсов 40%/60%. Такие ресурсы могут включать в себя ресурсы всего устройства (например, количество устройств),

ресурсы обработки (например, количество циклов процессора), ресурсы памяти (например, участок кэша, главной памяти и т.д.), ресурсы пропускной способности и/или интерфейса сети (например, биты в секунду и/или физические сетевые адаптеры (СА) и т.п.).

В частности, для первой конфигурации 900 блок 302 пересылки потребляет 70% ресурсов модуля 106 выравнивания нагрузки, а классификатор 304 потребляет 30% этих ресурсов. После повторного выделения в ходе процедуры расширения для создания второй конфигурации 950 блок 302 пересылки потребляет 40% ресурсов модуля 106 выравнивания нагрузки, а классификатор 304 потребляет 60% этих ресурсов.

В иллюстративном случае первая конфигурация 900 призвана повышать производительность выравнивания сетевой нагрузки, когда соответствующие хосты (не показанные на фиг.9А и 9В) обрабатывают меньшее количество более длительных транзакций, поскольку функции классификации используются после первоначальной связи для соединения, а функции пересылки используются потом. Вторая конфигурация 950, напротив, призвана повышать производительность выравнивания сетевой нагрузки, когда соответствующие хосты обрабатывают большее количество более коротких транзакций, поскольку функции классификации используются для большего процента полного количества пакетов, туннелируемых через инфраструктуру выравнивания сетевой нагрузки. В этом случае, если также используются функции маршрутизации запросов, то маршрутизатору(ам) 306 запросов также выделяется процент совокупных вычислительных ресурсов. Распределение ресурсов между тремя функциями можно регулировать в процессе обработки соединений (например, регулировать «на лету») в зависимости от текущего потребления и/или дефицита ресурсов.

Согласно фиг.2 и 3 каждый модуль 106 выравнивания нагрузки может соответствовать всей полной инфраструктуре 106 выравнивания сетевой нагрузки или ее части. Для каждого данного физически, логически, произвольно и т.д. заданного или оговоренного модуля 106 выравнивания нагрузки его ресурсы можно повторно выделять посредством процедуры расширения. В частности, распределение ресурсов между разными разделенными функциями, относящимися к выравниванию сетевой нагрузки, модуля 106 выравнивания нагрузки можно изменять посредством процедуры расширения. Кроме того, более чем двум разным функциям, а также другим функциям, относящимся к выравниванию сетевой нагрузки, которые конкретно не показаны на фиг.9А и 9В, можно выделять изменяющиеся проценты ресурсов.

Процент совокупных системных ресурсов, выделяемый всем функциям выравнивания нагрузки, также можно изменять посредством процедуры расширения. В качестве примера общей вычислительной мощности, процент полной вычислительной мощности, выделяемый для выравнивания нагрузки, можно постепенно увеличивать по мере увеличения объема трафика, в отношении которого нужно выполнять выравнивание нагрузки.

Программное обеспечение выравнивания сетевой нагрузки может, в необязательном порядке, осуществлять мониторинг, чтобы анализировать и определять, следует ли перевыделять ресурсы. Например, программное обеспечение выравнивания сетевой нагрузки может отслеживать использование процессора разными функциями, относящимися к выравниванию сетевой нагрузки. Фактическое перевыделение может также, в необязательном порядке, осуществляться

автоматически программным обеспечением выравнивания сетевой нагрузки в автономном или оперативном режиме.

Следует понимать, что расширенные способности описанной здесь инфраструктуры выравнивания сетевой нагрузки (например, реализованной, по меньшей мере частично, программными средствами) может относиться к разным установкам и не обязательно к изменению одной установки. В ресурсо-ориентированном примере описанная здесь инфраструктура выравнивания сетевой нагрузки может быть настроена в соответствии с одним распределением ресурсов в одной среде установки и может быть настроена в соответствии с другим распределением ресурсов в другой среде установки, имеющей другие рабочие параметры. Дополнительно, возможности, признаки, опции и пр., описанные выше в отношении расширения, также применимы к сужению. Иными словами, ресурсы, выделяемые инфраструктуре выравнивания сетевой нагрузки (или ее подфункциям), можно также уменьшать.

Иллюстративная обработка работоспособности и нагрузки

В этом разделе описано, как информацию статуса хоста, например, информацию работоспособности и/или нагрузки, можно собирать для выравнивания сетевой нагрузки и использовать в нем. В этом разделе ссылки идут, главным образом, на фиг.10-18 и рассматриваются функции работоспособности и нагрузки, например, обеспечиваемые обработчиком 314 работоспособности и нагрузки (фиг.3). Согласно описанному выше со ссылкой на фиг.3 каждый хост 108 хостирует одно или несколько приложений 316. Обработчик 314 работоспособности и нагрузки использует информацию работоспособности и/или нагрузки, относящуюся к приложениям 316 и/или хостам 108 для определенных описанных реализаций выравнивания сетевой нагрузки.

На фиг.10 показан иллюстративный подход к выравниванию сетевой нагрузки с использованием информации 1006 статуса хоста (ИСХ). Каждый хост 108(1), 108(2),..., 108(n) включает в себя одно или несколько приложений 316(1), 316(2),..., 316(n) соответственно. Эти хосты 108, в целом, и эти приложения 316, в частности, могут время от времени изменять статус.

Например, хосты 108 и приложения 316 могут принимать новые соединения или не принимать новые соединения. Кроме того, они могут быстро обрабатывать клиентские запросы или медленно обрабатывать клиентские запросы. Кроме того, они могут иметь много ресурсов в резерве или мало неиспользованных ресурсов. Все, или часть таких данных, или другие данные могут содержать информацию 1006 статуса хоста. В целом, информация 1006 статуса хоста обеспечивает указание статуса некоторого аспекта хостов 108 и/или работающих на них приложений 316.

В описанной реализации каждый хост 108(1), 108(2),..., 108(n) содержит определитель 1002(1), 1002(2),... и 1002(n) соответственно информации статуса хоста (ИСХ). Каждый хост 108(1), 108(2),..., 108(n) также содержит распространитель 1004(1), 1004(2),... и 1004(n) соответственно информации статуса хоста (ИСХ). Каждый определитель 1002 информации статуса хоста и/или распространитель 1004 информации статуса хоста может быть частью инфраструктуры 106 выравнивания нагрузки (ИВН).

Каждый определитель 1002 информации статуса хоста определяет информацию 1006 статуса хоста для соответствующего хоста 108 и/или действующих на нем приложений 316. Иллюстративные методы определения такой информации 1006 статуса хоста описаны ниже со ссылкой на фиг.12-14 и, в частности, на фиг.13А.

Каждый распространитель 1004 информации статуса хоста распространяет информацию 1006 статуса хоста для соответствующего хоста и/или приложений 316 в инфраструктуру 106 выравнивания нагрузки (например, ту/те часть/части инфраструктуры 106 выравнивания нагрузки, которые не располагаются на хостах 108). Иллюстративные методы распространения такой информации 1006 статуса хоста описаны ниже со ссылкой на фиг.12-17 и, в частности, на фиг.13B и 15-17.

В частности, каждый распространитель 1004 информации состояния хоста распространяет информацию 1006 статуса хоста (прямо или косвенно) в каждый модуль 106 выравнивания нагрузки (MBH) инфраструктуры 106 выравнивания нагрузки, который включает в себя, по меньшей мере, один обработчик 314 работоспособности и нагрузки и/или классификатор 304. Инфраструктура 106 выравнивания нагрузки ссылается на информацию 1006 статуса хоста при реализации выравнивания сетевой нагрузки. Например, как указано логикой 1008, инфраструктура 106 выравнивания нагрузки способна принимать решения по выравниванию нагрузки в соответствии с информацией 1006 статуса хоста.

На этапе (1) работы определители 1002 информации статуса определяют информацию 1006 статуса хоста для соответствующих хостов 108 и/или приложений 316. На этапах (1) и (2) распространители 1004 информации статуса хоста распространяют информацию 1006 статуса хоста из хостов 108 в инфраструктуру 106 выравнивания нагрузки. Например, информация 1006 статуса хоста может быть распространена в индивидуальные модули 106 выравнивания нагрузки. На этапе (3) логика 1008 принимает решения по выравниванию сетевой нагрузки в соответствии с информацией 1006 статуса хоста. На этапе (4) соединения пересылаются на хосты 108 назначения на основании этих решений по выравниванию сетевой нагрузки.

На фиг.11 показана логическая блок-схема 1100 иллюстративного способа выравнивания сетевой нагрузки с использованием информации статуса хоста. Логическая блок-схема 1100 содержит три блока 1102-1106. Хотя действия логической блок-схемы 1100 могут осуществляться в других средах и посредством различных программных схем, фиг.1-3 и 10 используются, в частности, для иллюстрации определенных аспектов и примеров способа.

На блоке 1102 информация статуса хоста отправляется с хостов на модули выравнивания нагрузки. Например, информация 1006 статуса хоста может быть отправлена с хостов 108 на модули 106 выравнивания нагрузки. На блоке 1104, информация статуса хоста принимается от хостов на модулях выравнивания нагрузки. Например, модули 106 выравнивания нагрузки получают информацию 1006 статуса хоста от хостов 108. На блоке 1106 принимаются решения по выравниванию нагрузки в соответствии с полученной информацией статуса хоста. Например, логика 1008 на модулях 106 выравнивания нагрузки может принимать решения по выравниванию сетевой нагрузки в соответствии с информацией 1006 статуса хоста.

Таким образом, на фиг.10 инфраструктура 106 выравнивания нагрузки собирает информацию 1006 статуса хоста от хостов 108 (и/или их приложений 316) и входящие запросы на выравнивание нагрузки, адресованные хостам 108 в соответствии с информацией 1006 статуса хоста. Согласно описанному ниже со ссылкой на фиг.12-18 эта информация 1006 статуса хоста может зависеть от приложения. Также согласно описанному ниже примеры информации 1006 статуса хоста включают в себя информацию работоспособности и/или нагрузки.

На фиг.12 показан подход к выравниванию сетевой нагрузки с использованием информации работоспособности и/или нагрузки (ИРН) 1206. Хосты 108(1),

108(2),..., 108(n) подключены к модулям 106(1), 106(2),..., 106(u) выравнивания нагрузки через средство связи 1210, например, сеть.

Показано, что хосты 108 передают информацию 1206 работоспособности и нагрузки на модули 106 выравнивания нагрузки с использованием средства связи 1210. Двусторонняя передача информации 1206 работоспособности и нагрузки, обозначенная двусторонней стрелкой, относится к двусторонней связи между модулями 106 выравнивания нагрузки и хостами 108, которая обеспечивает определенную полноту, когерентность, точность и т.д., в результате чего хосты 108 и/или модули 106 выравнивания нагрузки могут отказывать независимо друг от друга. Такие двусторонние связи между модулями 106 выравнивания нагрузки и хостами описаны ниже с конкретной ссылкой на фиг.15.

Информация работоспособности отражает, способен(но) ли данный(ое) хост и/или приложение обрабатывать клиентские запросы. Информация нагрузки отражает количество, величину и/или уровень клиентских запросов, которые в конкретный момент времени способен(но) обрабатывать данный(ое) хост и/или приложение. Иными словами, нагрузка отражает прямо или обратно доступные количество, величину и/или уровень полной емкости данного хоста и/или приложения. Как отмечено выше, реализации, описанные со ссылкой на фиг.12-18, фокусируются на информации работоспособности и/или нагрузки; однако эти реализации также применимы к общей информации статуса для хостов (включая их приложения).

В описанной реализации каждый хост 108(1), 108(2),..., 108(n) содержит соответствующий компонент 1202(1), 1202(2),..., 1202(n) инфраструктуры работоспособности и нагрузки (ИРиН). Каждый компонент 1202 инфраструктуры работоспособности и нагрузки может, в необязательном порядке, быть частью инфраструктуры 106 выравнивания нагрузки, которая располагается и выполняется на каждом хосте 108. Информацию 1206 работоспособности и нагрузки можно реализовать программными средствами. При функционировании каждая инфраструктура 1202(1), 1202(2),..., 1202(n) создает и поддерживает таблицу 1204(1), 1204(2),..., 1204(n) работоспособности и нагрузки (РиН).

Таблицы 1204 работоспособности и нагрузки могут включать в себя элементы, зависящие от приложения. Информация 1206 работоспособности и нагрузки, хранящаяся в таблицах 1204 работоспособности и нагрузки, может быть независимой от инфраструктуры 106 выравнивания нагрузки. Например, администраторы, конструкторы и т.д. могут задавать критерии для информации 1206 работоспособности и нагрузки во время настройки. Дополнительно, элементы, внешние по отношению к устройству, которое является хостом 108 или содержит его, могут способствовать определению информации 1206 работоспособности и нагрузки для приложений 316 на устройстве. Иллюстративная таблица 1204 работоспособности и нагрузки описана ниже со ссылкой на фиг.13А.

Каждый модуль 106(1), 106(2),... 106(u) выравнивания нагрузки включает в себя соответствующий объединенный кэш 1208(1), 1208(2),... 1208(u) работоспособности и нагрузки (РиН). Каждый объединенный кэш 1208 работоспособности и нагрузки содержит информацию из каждой таблицы 1204(1), 1204(2),..., 1204(n) работоспособности и нагрузки. Поэтому каждому модулю 106 выравнивания нагрузки предоставляется быстрый (например, кэшированный) доступ к информации 1206 работоспособности и нагрузки для каждого хоста 108, для которого модули 106 выравнивания нагрузки выравнивают нагрузку сетевого трафика.

В ходе эксплуатации инфраструктуры 1202 работоспособности и нагрузки

переносят информацию 1206 работоспособности и нагрузки из таблиц 1204 работоспособности и нагрузки в объединенные кэши 1208 работоспособности и нагрузки. Механизм для предоставления информации 1206 работоспособности и нагрузки является управляемым событиями, в результате чего изменения в таблицах 1204 работоспособности и нагрузки поступают в объединенные кэши 1208 работоспособности и нагрузки своевременно и масштабируемо.

На фиг.13A показана таблица 1204 работоспособности и нагрузки, обозначенная на фиг.12. В описанной реализации таблица 1204 работоспособности и нагрузки содержит множественные элементы 1302, каждый из которых связан с отдельным приложением 316. Каждый элемент 1302 может соответствовать строке в таблице 1204 работоспособности и нагрузки, которая имеет три столбца. Эти столбцы соответствуют идентификатору (ИД) 1302(A) приложения, характеристике 1302(B) статуса приложения и директиве 1302(C) выравнивателя нагрузки.

Поскольку каждый элемент 1302 связан с конкретным приложением 316, всякий раз при запуске (например, администратором) приложения происходит добавление строки. Аналогично, всякий раз при закрытии приложения происходит уничтожение/удаление строки. Аналогично, отдельные поля в столбцах 1302(A), 1302(B) и/или 1302(C) изменяются/обновляются при изменении их значений. Например, при изменении значения характеристики статуса для данного приложения 316 значение поля характеристики 1302(B) статуса приложения для элемента 1302 данного приложения 316 обновляется.

Добавления и удаления элементов 1302 для приложений 316 могут осуществляться с помощью входной информации от диспетчера управления на хосте 108. Например, диспетчер управления операционной системы знает, когда приложение 316 начинает и прекращает работать, поскольку он активно участвует в запуске и остановке приложений 316. Поэтому диспетчер управления может указать, что он, по меньшей мере, частично, запустил приложение 316, и диспетчер управления может установить, что он, по меньшей мере, частично, остановил приложение 316. Таким образом, диспетчер управления может информировать инфраструктуру 1202 работоспособности и нагрузки о запуске и остановке приложений 316. Поэтому не требуется обеспечивать явной передачи данных от приложений 316 на инфраструктуру 1202 работоспособности и нагрузки. Примером диспетчера управления является диспетчер управления службами (ДУС) операционной системы Windows® от корпорации Microsoft®.

Идентификатор 1302(A) приложения содержит информацию, используемую для однозначной идентификации приложения 316, с которым связан элемент 1302. Идентификатор 1302(A) приложения может содержать одно или более из следующего для соответствующего приложения 316: виртуальный IP-адрес и порт, физический IP-адрес и порт, используемый протокол и любую информацию, связанную с протоколом. В качестве протокола может выступать HTTP, IPsec, SOAP и т.д. В качестве информации, связанной с протоколом, может выступать шаблон или строка URL для дополнительного указания приложения, связанного с элементом 1302. Таким образом, идентификатор 1302(A) приложения более конкретно относится к конечной точке данного приложения на конкретном хосте 108.

Альтернативно, можно использовать другие идентификаторы приложения. Например, для снижения нагрузки на линию связи, идентификатор 1302(A) приложения может представлять собой 32-разрядное двоичное число, соответствующее вышеупомянутой иллюстративной информации на

инфраструктуре 1202 работоспособности и нагрузки и модулях 106 выравнивания нагрузки. Кроме того, любое из полей в элементе 1302 может фактически содержать глобально уникальный идентификатор (GUID), используемый в качестве ключа для поиска истинной информации для поля.

Характеристика 1302(B) статуса приложения содержит информацию, отражающую статус приложения 316, с которым связан элемент 1302. Характеристика 1302(B) статуса приложения содержит следующее для соответствующего приложения 316: работоспособность приложения, нагрузку приложения и емкость приложения. Работоспособность приложения - это квазибулево значение, указывающее, работает ли приложение. Работоспособность приложения может принимать значения «работоспособно», «неработоспособно» и «неизвестно». Работоспособность приложения - это относительно мгновенное значение, которое передается со сравнительно небольшой задержкой (например, порядка секунды или нескольких секунд) на модули 106 выравнивания нагрузки при изменении значения работоспособности приложения.

Нагрузка приложения - это значение, указывающее, насколько нагружено или занято приложение, и, таким образом, прямо или обратно, сколько дополнительной нагрузки может обрабатывать данное приложение. Нагрузка приложения - это относительно медленно изменяющееся или усредненное значение, которое, при желании, можно сглаживать с помощью механизма гистерезисного возбуждения для устранения переходных пиков возрастающей или убывающей нагрузки. Она передается на модули 106 выравнивания нагрузки сравнительно редко (например, от одного до четырех раз в минуту). Значение нагрузки приложения приобретает смысл в отношении емкости приложения.

Емкость приложения - это значение, указывающее максимальную емкость приложения. Его выбирают общим способом, чтобы оно имело смысл для данного контекста, но все же было достаточно гибким для других контекстов. Емкость приложения - это безразмерная величина, имеющая ограниченный диапазон значений (например, 0-99), которую можно определить во время настройки. Она может зависеть от вычислительной мощности, размера/скорости памяти, сетевого доступа, некоторых их комбинаций и т.п. Емкость приложения выражает относительные емкости между другими приложениями того же типа в группе хостов 108(1, 2,..., n).

Таким образом, нагрузка приложения приобретает смысл по отношению к емкости приложения. Нагрузка приложения для данного приложения это процент емкости приложения для данного приложения. Альтернативно, нагрузка приложения может выражаться безразмерной величиной, из которой можно получить процент с привлечением значения емкости приложения.

Директива 1302(C) выравнивателя нагрузки содержит информацию, отражающую желаемое и/или ожидаемое состояние директивы, установленной инфраструктурой 1202 работоспособности и нагрузки для модулей 106 выравнивания нагрузки в отношении приложения 316, с которым связан элемент 1302. Директива 1302(C) выравнивателя нагрузки содержит следующее для соответствующего приложения 316: целевое состояние выравнивания нагрузки и текущее состояние выравнивания нагрузки.

Целевое состояние выравнивания нагрузки отражает состояние директивы для модулей 106 выравнивания нагрузки в соответствии с указанием инфраструктуры 1202 работоспособности и нагрузки. Текущее состояние выравнивания нагрузки отражает то, что инфраструктура 1202 работоспособности и нагрузки понимает, каким должно

быть текущее состояние директивы для модулей 106 выравнивания нагрузки, записываемое на модулях 106 выравнивания нагрузки. Текущее состояние выравнивания нагрузки, таким образом, отражает директиву выравнивания нагрузки, которую, как ожидает инфраструктура 1202 работоспособности и нагрузки, в данный момент выполняют модули 106 выравнивания нагрузки в соответствии с требованиями используемого протокола связи. Такой иллюстративный протокол связи описан ниже со ссылкой на фиг.15. Взаимодействие и соотношение между целевым состоянием выравнивания нагрузки и текущим состоянием выравнивания нагрузки также дополнительно поясняются в описании фиг.15.

Целевое состояние выравнивания нагрузки, как и текущее состояние выравнивания нагрузки, может принимать значения «активное», «неактивное» или «истощающееся». Директива «активное» указывает, что новые запросы/соединения могут поступать и направляться приложению, связанному с элементом 1302. Директива «неактивное» указывает, что никакие дополнительные пакеты не подлежат пересылке соответствующему приложению. Директива «истощающееся» указывает, что никакие пакеты для новых запросов/соединений не подлежат отправке соответствующему приложению, но пакеты существующих запросов/соединений следует продолжать пересылать соответствующему приложению.

В описанной реализации окончательная версия соответствующей информации 1206 работоспособности и нагрузки сохраняется в таблицах 1204 работоспособности и нагрузки, размещенных на каждом соответствующем хосте 108 из множественных хостов 108. В этой реализации, если хост 108 ломается, информация 1206 работоспособности и нагрузки, которая теряется, относится к тем приложениям 316, которые также уничтожаются. Таким образом, высокая степень надежности достигается автоматически без дублирования данных. Однако окончательная версия информации 1206 работоспособности и нагрузки может альтернативно храниться в другом месте. Другие такие опции хранения включают в себя сами модули 106 выравнивания нагрузки, хост 108, который (в качестве самостоятельной задачи или совместно с задачами хостинга) хранит и поддерживает информацию 1206 работоспособности и нагрузки для множественных других (включая все остальные) хостов 108, других отдельных и/или внешних устройств и т.д.

Если окончательная версия информации 1206 работоспособности и нагрузки хранится и поддерживается в другом месте, помимо того, что она распределена между хостами 108(1, 2, …, n), то такая информация 1206 работоспособности и нагрузки может храниться избыточно (например, также храниться в дублирующем устройстве, резервно скопированы и т.д.) в целях повышения надежности. Иллюстративные сценарии посредника для сохранения информации 1206 работоспособности и нагрузки описаны ниже со ссылкой на фиг.17A и 17B. На фиг.17A изображен сценарий посредника для таблиц 1204 работоспособности и нагрузки, а на фиг.17B изображен сценарий посредника для объединенных кэшей работоспособности и нагрузки.

На фиг.13B изображен объединенный кэш 1208 работоспособности и нагрузки, обозначенный на фиг.12. В описанной реализации каждый объединенный кэш 1208 работоспособности и нагрузки в каждом модуле 106 выравнивания нагрузки содержит, по меньшей мере, часть информации, хранящейся в каждой таблице 1204 работоспособности и нагрузки для каждой инфраструктуры 1202 работоспособности и нагрузки для каждого хоста 108. Кэшированная информация работоспособности и нагрузки может быть организована любым образом в объединенном кэше 1208 работоспособности и нагрузки.

Показано, что объединенный кэш 1208 работоспособности и нагрузки содержит кэш для каждого хоста 108(1), 108(2),..., 108(n), который частично или полностью дублирует информацию таблицы 1204 работоспособности и нагрузки каждого соответствующего хоста 108(1, 2,..., n). В частности, объединенный кэш 1208 работоспособности и нагрузки содержит кэш для хоста №1 1304(1), кэш для хоста №2 1304(2), …, кэш для хоста №n 1304(n). Таким образом, иллюстрируемый объединенный кэш 1208 работоспособности и нагрузки организован на широком уровне по хосту 108(1, 2,..., n), причем каждый отдельный кэш 1304 содержит элементы, зависящие от приложения для соответствующего хоста 108(1, 2,..., n). Альтернативно, объединенный кэш 1208 работоспособности и нагрузки может быть организован на широком уровне по типу приложения 316 с отдельными блоками, направленными на конкретный тип приложения, дополнительно деленными по хосту 108(1, 2,..., n). Можно также использовать другие форматы структуры данных.

На фиг.14 показана логическая блок-схема способа выравнивания сетевой нагрузки с использованием информации работоспособности и нагрузки. Логическая блок-схема 1400 содержит восемь блоков 1402-1416. Хотя действия, указанные на логической блок-схеме 1400, могут осуществляться в других средах и посредством различных программных схем, фиг.1-3 и 12-13В используются, в частности, для иллюстрации определенных аспектов и примеров способа. Например, действия двух блоков 1402-1404 осуществляются хостом 108, а действия шести блоков 1406-1416 осуществляются модулем 106 выравнивания нагрузки.

На блоке 1402 происходит определение информации работоспособности и нагрузки на хосте. Например, инфраструктура 1202(2) работоспособности и нагрузки может получать информацию 1206 работоспособности и нагрузки для приложений 316(2) и сохранять в таблице 1204(2) на хосте 108(2). На блоке 1404 определенная информация работоспособности и нагрузки сеется в модули выравнивания нагрузки. Например, инфраструктура 1202(2) работоспособности и нагрузки может отправить информацию 1206 работоспособности и нагрузки для приложений 316(2) на модули 106(1, 2... u) выравнивания нагрузки. Стрелка 1418 указывает, что действия блоков 1402 и 1404 повторяются, что позволяет непрерывно отслеживать работоспособность и нагрузку (приложения) и обновлять их, когда происходят изменения.

На блоке 1406 происходит получение информации работоспособности и нагрузки от хостов. Например, модуль 106(1) выравнивания нагрузки может получать информацию 1206 работоспособности и нагрузки от множественных хостов 108(1, 2,..., n), которая содержит информацию 1206 работоспособности и нагрузки для приложений 316(2) хоста 108(2). На блоке 1408 полученная информация работоспособности и нагрузки кэшируется. Например, модуль 106(1) выравнивания нагрузки может сохранять информацию 1206 работоспособности и нагрузки от хостов 108(1, 2,..., n) в объединенном кэше 1208(1) работоспособности и нагрузки. Согласно реализации объединенного кэша 1208(1) работоспособности и нагрузки, изображенного на фиг.13 В, информация 1206 работоспособности и нагрузки для приложений 316(2) от хоста 108(2) может храниться в кэше 1304(2) для хоста №2. Стрелка 1420 указывает, что действия блоков 1406 и 1408 повторяются, что позволяет непрерывно отслеживать работоспособность и нагрузку (приложения) и обновлять их, когда происходят изменения.

Пунктирная стрелка 1422 указывает, что модули 106 выравнивания нагрузки также обрабатывают передачи от клиентов 102, одновременно обрабатывая информацию

работоспособности и нагрузки. На блоке 1410 происходит получение пакета, запрашивающего новое соединение. Например, модуль 106(1) выравнивания нагрузки может получить пакет SYN TCP от клиента 102(2) по сети 104. На блоке 1412 осуществляется сверка с кэшированной информацией работоспособности и нагрузки. Например, модуль 106(1) выравнивания нагрузки может сверяться с объединенным кэшем 1208(1) работоспособности и нагрузки. В частности, модуль 106(1) выравнивания нагрузки может сверяться с элементами, связанными с приложением, которому адресован пакет SYN TCP, по кэшам 1304(1, 2,..., n) для хостов №1, №2, …, №n.

На блоке 1414 происходит выбор хоста в соответствии с кэшированной информацией работоспособности и нагрузки. Например, модуль 106(1) выравнивания нагрузки может выбрать хост 108(2), имеющий приложение(я) 316(2), в соответствии с информацией 1206 работоспособности и нагрузки, кэшированной в объединенном кэше 1208(1) работоспособности и нагрузки. Выбранное приложение 316 (и хост 108) должно быть работоспособным и способным принимать дополнительную нагрузку (например, по возможности, наименее нагруженным приложением среди приложений того типа приложения, которому адресован пакет SYN TCP).

Сверка с кэшированной информацией работоспособности и нагрузки (на блоке 1412) и выбор хоста в соответствии с кэшированной информацией работоспособности и нагрузки (на блоке 1414) может осуществляться до получения конкретного пакета запроса нового соединения и/или с использованием групповой схемы. Кроме того, выбор можно осуществлять в соответствии с любой из многочисленных схем. Например, можно применять маркерную или круговую схемы. При любой схеме выбор может предусматривать взвешивание относительных нагрузок среди опций приложения. Эти сверка и выбор, совместно с маркерной и круговой схемами, описаны ниже со ссылкой на фиг.18 в разделе, озаглавленном «Иллюстративные классификация, пересылка и маршрутизация запросов», особенно в связи с функциями классификации.

После выбора хоста назначения на блоке 1414 на него можно отправить пакет запроса нового соединения. На блоке 1416 пакет, полученный от клиента, пересылается на выбранный хост. Например, пакет SYN TCP пересылается с модуля 106(1) выравнивания нагрузки на выбранный хост 108(2). Пересылка этого начального пакета может осуществляться непосредственно классификатором 304 или блоком 302 пересылки, что также описано ниже в разделе, озаглавленном «Иллюстративные классификация, пересылка и маршрутизация запросов».

В рассмотренной реализации инфраструктура 1202 работоспособности и нагрузки размещена на множественных хостах 108 и распределена между ними, а также размещена на модулях 106 выравнивания нагрузки (будучи представлена обработчиком 314 работоспособности и нагрузки). Инфраструктура 1202 работоспособности и нагрузки имеет три функции. Во-первых, она открывает точку(и) прослушивания для получения обновлений статуса приложения для характеристик 1302(B) статуса приложения таблиц 1204 работоспособности и нагрузки. Во-вторых, он синтезирует информацию статуса приложения, чтобы определить, что должны делать модули 106 выравнивания нагрузки, что воплощено в директиве 1302(C) выравнивателя нагрузки. В-третьих, инфраструктура 1202 работоспособности и нагрузки передает эту директиву от хостов 108 на модули 106 выравнивания нагрузки.

Директивное содержимое директивы 1302(C) выравнивателя нагрузки является

эффективно классифицированной версией информации для характеристик 1302(B) статуса приложения. Однако модули 106 выравнивания нагрузки также могут принимать необработанную информацию характеристик 1302(B) статуса приложения, а также эту обработанную директиву. Передача содержимого этих и других полей таблиц 1204 работоспособности и нагрузки осуществляется с использованием протокола обмена сообщениями, который описан ниже со ссылкой на фиг.15.

На фиг.15 проиллюстрирован протокол 1500 обмена сообщениями для передач, относящихся к информации работоспособности и нагрузки, обозначенных на фиг.12, между хостами 108 и модулями 106 выравнивания нагрузки. В общем случае механизм, управляемый событиями, используется для передачи изменений в таблицах 1204 работоспособности и нагрузки с хостов 108 на модули 106 выравнивания нагрузки. Иными словами, для описываемой реализации информация передается с хостов 108 на модули 106 выравнивания нагрузки при обновлении таблиц 1204 работоспособности и нагрузки. Это позволяет избегать отправки моментальных снимков всех таблиц 1204 работоспособности и нагрузки и, таким образом, снизить потребление полосы пропускания сети инфраструктурой 1202 работоспособности и нагрузки.

Протокол 1500 обмена сообщениями может быть реализован с использованием имеющегося механизма передачи сообщений. Такие механизмы включают в себя надежный широковещательный механизм, двухточечную передачу (например, протокол пользовательских датаграмм (UDP)) и т.д. Показано, что протокол 1500 обмена сообщениями содержит семь типов сообщений 1502-1514: сообщение 1502 «пульс», сообщение 1504 «прощание», сообщение 1506 «изменение строки», сообщение 1508 «получить мгновенный снимок таблицы», сообщение 1510 «отправить мгновенный снимок таблицы», сообщение 1512 «постулировать состояние таблицы» и сообщение 1514 «постулировать ошибку».

Следует понимать, что, за исключением стрелок 1516 и 1518, данная иллюстрация не устанавливает никакой взаимосвязи по времени между разными типами сообщений 1502-1514. Например, сообщение 1506 «изменение строки» не обязано следовать за сообщением 1504 «прощание».

Сообщение 1502 «пульс» указывает, что конкретный хост 108 действует и обеспечивает некоторый контроль ошибок для содержимого соответствующей таблицы 1204 работоспособности и нагрузки в отношении соответствующего конкретного кэша для конкретного хоста 1304 в объединенном кэше 1208 работоспособности и нагрузки. Каждая инфраструктура 1202 работоспособности и нагрузки на каждом хосте 108 отправляет сообщение «пульс», прямо или косвенно, в каждый кэш 1208 работоспособности и нагрузки на каждом модуле 106 выравнивания нагрузки.

Сообщения 1502 «пульс» решают проблему устаревания данных в объединенных кэшах 1208 работоспособности и нагрузки, которая возникает, отчасти, из-за того, что мгновенный снимок всей таблицы 1204 работоспособности и нагрузки не периодически передается каждому модулю 106 выравнивания нагрузки. Схема передачи сообщений 1502 «пульс» описана ниже со ссылкой на фиг.16.

Сообщения 1502 «пульс» включают в себя идентификатор для хоста, данные контроля ошибок и, в необязательном порядке, имя DNS. Идентификатор хоста может представлять собой уникальное (32-разрядное) двоичное число, выбранное во время настройки. Данные контроля ошибок могут представлять собой контрольную сумму, порядковый номер смены состояния, номер поколения, значение CRC и т.д., что позволяет принимающему модулю 106 выравнивания нагрузки удостоверяться, что

содержимое его объединенного кэша 1208 работоспособности и нагрузки согласуется с содержимым таблицы 1204 работоспособности и нагрузки передающего хоста 108. В случае использования номера поколения можно использовать множественные ИД поколения, причем каждый ИД поколения присваивается «куску» приложений. Тогда сообщения могут ссылаться на номер куска или пару номер куска/ИД поколения в зависимости от контекста.

Данные контроля ошибок (или, более обще, индикатор содержимого) могут представлять собой одно значение для всей таблицы 1204 работоспособности и нагрузки, а могут представлять собой множественные значения, определенные для каждого элемента 1302. В необязательном порядке может передаваться имя DNS (например, каждые «x» «ударов пульса») для проверки или обновления текущего правильного сетевого адреса хоста.

Сообщение 1504 «прощание» передается с конкретного хоста 108 на модули 106 выравнивания нагрузки, чтобы указывать, что планируется отключение конкретного хоста 108. Сообщение 1504 «прощание» включает в себя идентификатор хоста, который может индексироваться/отображаться на сетевой адрес для конкретного хоста 108. Сообщение 1504 «прощание» используется для чистых, преднамеренных отключений хостами 108, чтобы обусловливать «быструю очистку». Однако в случае потери сообщения 1504 «прощание» кэши, в конце концов, отстают от элементов конкретного хоста 108, поскольку сообщения 1502 «пульс» больше не передаются.

Сообщение 1506 «изменение строки» передается с конкретного хоста 108 на модули 106 выравнивания нагрузки, чтобы указывать, что работоспособность и нагрузка данного приложения 316 конкретного хоста 108 изменились. Сообщение 1506 «изменение строки» включает в себя идентификатор хоста, идентификатор приложения, операцию и данные для операции. Иллюстративные идентификаторы хостов описаны выше в связи с сообщениями 1502 «пульс» и сообщениями 1504 «прощание». Иллюстративные идентификаторы приложений описаны выше в связи с идентификатором 1302(A) приложения в элементе 1302, связанном с приложением, таблиц 1204 работоспособности и нагрузки.

Операция смены строки может представлять собой добавление, удаление или обновление. Иными словами, данные для операции могут добавляться (для операции добавления) или заменять (для операции обновления) информацию, уже присутствующую в объединенных кэшах 1208 работоспособности и нагрузки на модулях 106 выравнивания нагрузки. Для операции удаления никаких данных не предусмотрено. Протокол 1500 обмена сообщениями задан так, что для одного сообщения 1506 «изменение строки» может быть предусмотрено выполнение множественных операций. Поэтому для конкретного идентификатора хоста наборы из идентификатора приложения, операции и данных операции могут повторяться для множественных приложений 316 хоста 108, идентифицированного конкретным идентификатором хоста.

Сообщение 1508 «получить мгновенный снимок таблицы» передается с конкретного модуля 106 выравнивания нагрузки для конкретного объединенного кэша 1208 работоспособности и нагрузки на отдельный хост 108 или хосты 108. Это сообщение 1508 «получить мгновенный снимок таблицы» запрашивает у инфраструктуры 1202 работоспособности и нагрузки на хостах 108 предоставление соответствующей таблицы 1204 работоспособности и нагрузки для соответствующего хоста 108. Это сообщение включает в себя идентификацию запрашивающего модуля 106 выравнивания нагрузки и может использоваться модулем 106

выравнивания нагрузки (i) после его сбоя и последующего восстановления; (ii) после того, как хост 108 испытал сбой, восстановился и вновь начал отправлять сообщения 1502 «пульс»; (iii) если сообщение 1506 «изменение строки» отправлено на модуль 106 выравнивания нагрузки, но сообщение пропало, в результате чего произошла рассинхронизация между кэшем 1208 работоспособности и нагрузки и соответствующей таблицей 1204 работоспособности и нагрузки для соответствующего хоста 108; и (iv) т.д.

Для случая (iii) потеря синхронизации между кэшем 1208 работоспособности и нагрузки и соответствующей таблицей 1204 работоспособности и нагрузки для соответствующего хоста 108 обнаруживается посредством последующего сообщения 1502 «пульс» от соответствующего хоста 108, поскольку «контроль ошибок» указывает, что объединенный кэш 1208 работоспособности и нагрузки устарел. Тогда модуль 106 выравнивания нагрузки отправляет сообщение 1508 «получить мгновенный снимок таблицы», что позволяет ему обновить свой объединенный кэш 1208 работоспособности и нагрузки. Таким образом, в любом из трех иллюстративных случаев (i, ii, iii) модуль 106 выравнивания нагрузки впоследствии восстанавливает свой объединенный кэш 1208 работоспособности и нагрузки с использованием сообщения 1508 «получить мгновенный снимок таблицы». Сообщение 1508 «получить мгновенный снимок таблицы» может многократно отправляться на каждый хост 108 в двухточечном режиме или может отправляться однократно на множественные хосты 108 в широковещательном режиме.

Сообщение 1510 «отправить мгновенный снимок таблицы» отправляется с отдельного хоста 108 на конкретный модуль 106 выравнивания нагрузки после того, как отдельный хост 108 получил сообщение 1508 «получить мгновенный снимок таблицы» от конкретного модуля 106 выравнивания нагрузки, что указано стрелкой 1516. Содержимое сообщения 1510 «отправить мгновенный снимок таблицы» подготавливается инфраструктурой 1202 работоспособности и нагрузки и может включать в себя все или, по меньшей мере, множественные строки таблицы 1204 работоспособности и нагрузки для отдельных хостов 108, что позволяет конкретному модулю 106 выравнивания нагрузки восстанавливать свой объединенный кэш 1208 работоспособности и нагрузки. Сообщение 1510 «отправить мгновенный снимок таблицы» может быть отдельно построенным сообщением или может быть эквивалентно последовательности операций добавления в сообщении 1506 «изменение строки».

Сообщение 1512 «постулировать состояние таблицы» и сообщение 1514 «постулировать ошибку» относятся к целевому состоянию выравнивания нагрузки и текущему состоянию выравнивания нагрузки директивы 1302(C) выравнивателя нагрузки элемента 1302 таблицы 1204 работоспособности и нагрузки. Целевое состояние выравнивания нагрузки это директива, в соответствии с которой должны работать модули 106 выравнивания нагрузки согласно желанию инфраструктуры 1202 работоспособности и нагрузки. Текущее состояние выравнивания нагрузки это директива, в соответствии с которой в данный момент работают модули 106 выравнивания нагрузки согласно ожиданию инфраструктуры 1202 работоспособности и нагрузки. В общем случае два состояния выравнивания нагрузки идентичны.

Однако целевое состояние выравнивания нагрузки может отличаться от текущего состояния выравнивания нагрузки в переходный период изменения директивы состояния. Пусть, например, целевое состояние выравнивания нагрузки и текущее состояние выравнивания нагрузки первоначально установлены активными. При

обнаружении проблемы с хостом 108 и/или его приложением 316 директива состояния выравнивания нагрузки меняется на «истощающееся». Эта директива «истощающееся» передается на модули 106 выравнивания нагрузки с использованием сообщения 1506 «изменение строки».

Пока это изменение директивы будет отмечено во всех объединенных кэшах 1208 работоспособности и нагрузки всех модулей 106 выравнивания нагрузки, имеет место задержка. В течение этого переходного периода целевое состояние выравнивания нагрузки является «истощающимся», тогда как текущее состояние выравнивания нагрузки по-прежнему является «активным» в таблице 1204 работоспособности и нагрузки хоста 108. Прежде чем текущее состояние выравнивания нагрузки сменится на «истощающееся», инфраструктура 1202 работоспособности и нагрузки хочет удостовериться, что объединенные кэши 1208 работоспособности и нагрузки действительно были обновлены до нового директивного состояния «истощения».

Чтобы проверить, что объединенные кэши 1208 работоспособности и нагрузки модулей 106 выравнивания нагрузки были обновлены в соответствии с новой директивой состояния, инфраструктура 1202 работоспособности и нагрузки отправляет сообщение 1512 «постулировать состояние таблицы» на модули 106 выравнивания нагрузки. Сообщение 1512 «постулировать состояние таблицы» передается через некоторое время (например, с заранее определенным периодом задержки) после передачи сообщения 1506 «изменение строки», указывая, что директива состояния должна измениться. В данном примере сообщение 1512 «постулировать состояние таблицы» указывает, что состояние таблицы должно быть «истощающееся». Пунктирная стрелка 1518 указывает, что модуль 106 выравнивания нагрузки отвечает на это сообщение 1512 «постулировать состояние таблицы», если его объединенный кэш 1208 работоспособности и нагрузки отличается от постулированной директивы состояния.

Если директива в объединенном кэше 1208 работоспособности и нагрузки не отличается от постулированной директивы состояния, то модуль 106 выравнивания нагрузки отправляет сообщение 1514 «постулировать ошибку» на инфраструктуру 1202 работоспособности и нагрузки хоста 108, который выдал сообщение 1512 «постулировать состояние таблицы». Затем эта инфраструктура 1202 работоспособности и нагрузки периодически повторно отправляет сообщение 1512 «постулировать состояние таблицы», пока от объединенных кэшей 1208 работоспособности и нагрузки не перестанут поступать сообщения 1514 «постулировать ошибку». В этот момент инфраструктура 1202 работоспособности и нагрузки отправляет новое сообщение 1506 «изменение строки» с новым состоянием выравнивания нагрузки. В этом смысле объединенные кэши 1208 работоспособности и нагрузки являются окончательными определителями текущего состояния выравнивания нагрузки и инфраструктура 1202 работоспособности и нагрузки является окончательным определителем целевого состояния выравнивания нагрузки.

На фиг.16 изображена схема передачи сообщений для передач, показанных на фиг.12, между хостами 108 и модулями 106 выравнивания нагрузки. Иллюстративная схема передачи сообщений может снижать полосу пропускания, потребляемую сообщениями 1502 «пульс» на средстве 1210 связи. Схема передачи сообщений, показанная на фиг.16, конкретно приспособлена к сообщениями 1502 «пульс», но может использоваться для других сообщений протокола 150 обмена сообщениями.

Множественные хосты 108(1), 108(2), 108(3),..., 108(11) и 108(12) показаны совместно с модулями 106(1), 106(2),..., 106(u) выравнивания нагрузки. Каждая линия

представляет связь между членами или принадлежность к группе хостов 108(1, 2, …, 12). Группа хостов 108(1, 2, …, 12) образует множество узлов, которые работают совместно для распространения информации «пульса» на модули 106 выравнивания нагрузки. Хотя показано двенадцать хостов, в каждую данную группу хостов может входить больше или меньше хостов. Кроме того, полная группа хостов 108, обслуживаемых инфраструктурой 106 выравнивания нагрузки, может делиться на одну, две, три или более групп хостов.

В описываемой реализации узлы, входящие в группу хостов 108(1, 2, …, 12), избирают лидера, отвечающего на передачу сообщений 1502 «пульс» на модули 106 выравнивания нагрузки. Каждый (нелидирующий) хост 108 из группы хостов 108(1, 2, …, 12) отправляет свои сообщения 1502 «пульс» избранному лидеру. В данном примере избранным лидером является хост 108(4).

Информация «пульса» каждого хоста 108 из группы хостов 108(1, 2, …, 12) через узлы-члены распространяется на лидирующий хост 108(4) группы. Хост 108(4) собирает информацию «пульса» и объединяет ее в объединенном сообщении 1602 «пульс». Объединенные сообщения 1602(1), 1602(2),..., 1602(u) «пульс» отправляются на соответствующие модули 1602(1), 1602(2),..., 1602(u). Эти объединенные сообщения 1602 «пульс» можно, в необязательном порядке, сжимать для дальнейшего снижения потребления полосы пропускания.

В качестве другой альтернативы лидирующий хост 108(4) может только пересылать изменения состава группы в объединенные кэши 1208 работоспособности и нагрузки. Иными словами, в этом режиме объединенные кэши 1208 работоспособности и нагрузки имеют дело главным образом, если не исключительно, с изменениями состояния в отношении принадлежности. Лидирующий хост 108(4) призван гарантировать пересылку первого приветствия, когда хост 108 выходит на связь, и отправку сообщения 1504 «прощание» при отключении хоста 108. Дополнительно, хост 108 может периодически указывать необходимость «пересылки» сообщения 1502 «пульс». Таким образом лидирующий хост 108(4) получает предписание отправить его в объединенные кэши 1208 работоспособности и нагрузки, даже если оно не выражает изменение принадлежности.

Сообщения 1502 «пульс» (в том числе объединенные сообщения 1602 «пульс») используются модулями 106 выравнивания нагрузки, когда их объединенные кэши 1208 работоспособности и нагрузки рассинхронизированы с таблицами 1204 работоспособности и нагрузки. Эта рассинхронизация может возникать, например, вследствие повреждения или иного сбоя объединенного кэша 1208 работоспособности и нагрузки и/или модуля 106 выравнивания нагрузки. Согласно описанному выше каждое сообщение 1502 «пульс» содержит данные контроля ошибок, которые формируются для проверки эквивалентности между объединенным кэшем 1208 работоспособности и нагрузки и таблицами 1204 работоспособности и нагрузки. В случае обнаружения неэквивалентности в отношении конкретного хоста 108 и/или его приложения 316 из сообщений 1502 «пульс» извлекается имя DNS конкретного хоста 108.

Объединенный кэш 1208 работоспособности и нагрузки использует имя DNS для отправки сообщения 1508 «получить мгновенный снимок таблицы» на конкретный хост 108 в целях получения обновленной информации 1206 работоспособности и нагрузки в виде сообщения 1510 «отправить мгновенный снимок таблицы». Другое или такое же сообщение 1508 «получить мгновенный снимок таблицы» отправляется на каждый хост 108, для которого обнаружена неэквивалентность. В конце концов,

информация 1206 работоспособности и нагрузки в объединенном кэше 1208 работоспособности и нагрузки оказывается эквивалентной информации 1206 работоспособности и нагрузки в таблицах 1204 работоспособности и нагрузки, что проверяется посредством новых сообщений 1502 «пульс». Таким образом, работа сбойного кэша 1208 работоспособности и нагрузки может восстанавливаться без вмешательства извне с использованием протокола 1500 обмена сообщениями и схемы контроля эквивалентности.

На фиг.17A и 17B изображены сценарии хранения на посреднике информации работоспособности и нагрузки для таблиц 1204 работоспособности и нагрузки и для объединенных кэшей 1208 работоспособности и нагрузки соответственно. В реализациях, описанных выше со ссылками на фиг.12-16, хосты 108 содержат инфраструктуру 1202 работоспособности и нагрузки. Однако в других реализациях хосты могут не содержать инфраструктуру 1202 работоспособности и нагрузки.

Например, на хосте может работать версия приложения(й) и/или операционная система, для которой инфраструктура работоспособности и нагрузки либо не реализована, либо, по соображениям политики, не установлена на хосте. В результате на этих типах хостов инфраструктура 1202 работоспособности и нагрузки не выполняется. Хост 1702 является таким хостом, на котором не выполняется инфраструктура 1202 работоспособности и нагрузки. Тем не менее, хост 1702 может использовать инфраструктуру 1202 работоспособности и нагрузки, которая выполняется на одном или нескольких посредниках, например, на посреднике 1704.

На посреднике 1704 размещается и выполняется инфраструктура 1202 работоспособности и нагрузки, включающая в себя таблицу 1204 работоспособности и нагрузки для приложений, работающих на хосте 1702. Альтернативно, посредник 1704 может выводить работоспособность и нагрузку на хосте 1702, выполняя действия внешнего слежения. Посредник 1704 показан как посредник 1704(1) и 1704(2) для избыточности и, следовательно, высокой надежности.

В реализациях, описанных выше со ссылками на фиг.12-16 и ниже со ссылками на фиг.8, выравнивание нагрузки осуществляется с помощью модулей 106 выравнивания нагрузки, содержащих объединенные кэши 1208 работоспособности и нагрузки. Однако другие реализации могут предусматривать выравнивание нагрузки без применения объединенных кэшей 1208 работоспособности и нагрузки.

Например, выравнивание нагрузки можно осуществлять посредством монолитного оборудования выравнивания нагрузки или другой инфраструктуры выравнивания нагрузки, которая не хранит и/или не может хранить или иным образом не содержит объединенный кэш 1208 работоспособности и нагрузки. Выравниватель 1706 нагрузки отражает такое(ие) устройство или устройства выравнивания нагрузки, которые не имеют объединенного кэша 1208 работоспособности и нагрузки. Тем не менее, выравниватель 1706 нагрузки может использовать объединенный кэш 1208 работоспособности и нагрузки, существующий на одном или нескольких посредниках, например, на посреднике 1708.

Посредник 1708 содержит объединенный кэш 1208 работоспособности и нагрузки, в котором хранится информация 1206 работоспособности и нагрузки для хостированных приложений, обслуживаемых выравнивателем 1706 нагрузки. Выравниватель 1706 нагрузки может использовать информацию 1206 работоспособности и нагрузки объединенного кэша 1208 работоспособности и нагрузки при осуществлении функций выравнивания нагрузки, осуществляя доступ к такой информации с использованием программных интерфейсов приложения (API),

«родных» для выравнивателя 1706 нагрузки и поддерживаемых им. Альтернативно, объединенный кэш 1208 работоспособности и нагрузки может вызывать API для передачи информации 1206 работоспособности и нагрузки, включая директивы, на выравниватель 1706 нагрузки. Посредник 1708 показан как посредник 1708(1) и 1708(2) избыточности и, следовательно, высокой надежности.

На фиг.18 показана процедура распределения целевых конечных точек приложения с использованием классификатора 304 и обработчика 314 работоспособности и нагрузки, входящих в состав модуля 106 выравнивания нагрузки. После того, как обработчик 314 работоспособности и нагрузки получает доступ к объединенному кэшу 1208 работоспособности и нагрузки, содержащаяся в нем информация 1206 работоспособности и нагрузки используется при выборе конечных точек приложения для новых запросов/соединений.

Согласно описанному выше со ссылкой на фиг.13B объединенный кэш 1208 работоспособности и нагрузки содержит кэшированную информацию 1206 работоспособности и нагрузки, полученную от множественных хостов 108, причем информация 1206 работоспособности и нагрузки организована в нем так, что она доступна по идентификатору каждого хоста 108. Однако информация 1206 работоспособности и нагрузки также организована в нем так, что она доступна по типу приложения 316, с целью облегчения выбора конечной точки приложения.

Другими словами, обработчик 314 работоспособности и нагрузки способен обращаться к информации 1206 работоспособности и нагрузки для каждого приложения 316 в отдельности по информации 1206 работоспособности и нагрузки для множественных хостов 108. После получения информации 1206 работоспособности и нагрузки для данного приложения 316 для каждого хоста 108 выделение входящих запросов соединения может осуществляться в соответствии с этой информацией 1206 работоспособности и нагрузки. Например, возможные конечные точки для данного приложения 316 могут выделяться входящим запросам соединения путем выбора конечных точек данного приложения 316 с учетом имеющейся относительной емкости нагрузки среди работоспособных конечных точек для данного приложения 316.

В описываемой реализации классификатор 304 делает запрос 1802 распределения целевых конечных точек приложения обработчику 314 работоспособности и нагрузки. Показано, что запрос 1802 распределения целевых конечных точек приложения содержит (i) виртуальный IP-адрес и порт, (ii) протокол и (iii) информацию, связанную с протоколом. Поэтому запрос 1802 распределения целевых конечных точек приложения идентифицирует тип приложения 316, которому адресованы входящие запросы соединения.

Обработчик 314 работоспособности и нагрузки получает запрос 1802 распределения целевых конечных точек приложения и выбирает, по меньшей мере, одну физическую конечную точку, соответствующую идентифицированному типу приложения 316, с использованием любого одного или более из многочисленных механизмов выбора. Для уменьшения задержки обработчик 314 работоспособности и нагрузки выбирает распределение конечных точек, подлежащее использованию по ряду входящих запросов соединения. Это распределение поступает от обработчика 314 работоспособности и нагрузки на классификатор 304 с использованием ответа 1804 по распределению конечных точек приложения. Показано, что ответ 1804 по распределению конечных точек приложения содержит распределение физических IP-адресов и портов (например, конечных точек IP1, IP2 и IP3) для

идентифицированного типа приложения 316.

Ответ 1804 по распределению конечных точек приложения может быть выполнен с использованием одной или нескольких схем распределения. Для примера показаны маркерная схема 1806 распределения и процентная схема 1808 распределения. Маркерная схема 1806 распределения является модульной схемой выделения, а процентная схема 1808 распределения является временной схемой распределения.

Маркерная схема 1806 распределения выделяет маркеры для каждой работоспособной конечной точки IP1, IP2 и IP3 в соответствии с их относительной нагрузкой и отношениями емкостей. В иллюстрируемом примере из полной имеющейся емкости IP1 имеет 40% имеющейся емкости, IP2 имеет 35% имеющейся емкости и IP3 имеет 25% имеющейся емкости. Таким образом, общее количество маркеров делится в соответствии с этими процентами. Общее количество маркеров может обеспечиваться как часть запроса 1802 распределения целевых конечных точек приложения или определяться обработчиком 314 работоспособности и нагрузки.

Можно использовать любое значение общего количества маркеров, например, 10, 45, 100, 250, 637, 1000 и т.д. Это значение может устанавливаться в зависимости от количества запросов соединения в секунду и скорости/частоты изменения работоспособности и/или нагрузки приложения. Классификатор 304 «тратит»/потребляет один маркер, отвечая на каждый запрос соединения при выделении конечных точек приложения, пока не закончатся маркеры; затем классификатор 304 запрашивает другое маркерное распределение с использованием запроса 1802 распределения целевых конечных точек приложения.

Процентная схема 1808 распределения определяет имеющуюся относительную емкость аналогичным образом. Однако вместо маркеров эти определенные имеющиеся относительные емкости на конечную точку приложения предоставляются классификатору 304 совместно с таймером 1810 длительности. Классификатор 304 выделяет целевые конечные точки приложения входящим запросам соединения в соответствии с этими имеющимися процентами относительной емкости до истечения таймера 1810 длительности.

Согласно процентной схеме 1808 распределения классификатор 304 поддерживает непрерывную запись выделений конечных точек приложения для присоединения к распределенным процентам и отслеживает время таймера 1810 длительности. По истечении таймера классификатор 304 запрашивает другое процентное распределение с использованием запроса 1802 распределения целевых конечных точек приложения.

Заметим, что маркерная схема 1806 распределения также может использовать предел времени. Если распределенные маркеры слишком стары, их можно отбросить и получить новые. В противном случае классификатор 304 может потреблять несвежие маркеры, которые были ранее выделены на основании информации работоспособности и нагрузки, которая в данный момент слишком устарела. Использование распределений конечных точек приложения классификатором 304 описано ниже в разделе, озаглавленном «Классификация, пересылка и маршрутизация запросов».

Отслеживание сеансов

В этом разделе описано, как информацию статуса хоста, например, информацию сеанса, можно собирать и использовать для выравнивания сетевой нагрузки. В этом разделе ссылки идут, главным образом, на фиг.19-24 и освещены функции сохранения сродства к сеансу, например, обеспечиваемые блоком 308 отслеживания сеансов (фиг.3). Согласно описанному выше со ссылкой на фиг.1-3 каждый хост 108 хостирует

одно или несколько приложений 316, которые предоставляют услугу(и) клиентам 102. Блок 308 отслеживания сеансов использует информацию сеанса, относящуюся к контекстам для соединений, установленных между приложениями 316 и клиентами 102 для определенных описанных реализаций выравнивания сетевой нагрузки.

На фиг.19 показан подход к выравниванию сетевой нагрузки с использованием информации 1902 сеанса. Соединение [1] обозначает, что клиент 102(1) создает новое соединение с хостом 108(2) через инфраструктуру 106 выравнивания нагрузки. Инфраструктура 106 выравнивания нагрузки может состоять из одного или нескольких модулей 106 выравнивания нагрузки. При поступлении запроса соединения на инфраструктуру 106 выравнивания нагрузки запрос обычно маршрутизируется на хост 108 с использованием функций выравнивания сетевой нагрузки в соответствии с информацией работоспособности и/или нагрузки хостов 108 и/или их приложений 316 (явно не показанной на фиг.19).

При установлении соединения [1] между клиентом 102(1) и обслуживающим приложением 316, которое, в данном примере, размещается на хосте 108(2), устанавливается сеанс. Сеанс обеспечивает контекст для двусторонней связи между клиентом 102(1) и хостом 108(2). Информация для контекста сеанса хранится на хосте 108(2). По завершении соединения [1] контекст сеанса может не использоваться вновь. С другой стороны, контекст сеанса может вновь быть полезен, если клиент 102(1) пытается инициировать другое соединение с хостами 108 для услуг, предоставляемых приложением 316. Если это другое соединение не маршрутизируется на тот же хост 108(2), где хранится этот контекст сеанса, то клиенту 102(1) приходится устанавливать новый контекст сеанса, что может требовать затрат времени, больших объемов данных/обработки, и/или расстраивать человека-пользователя клиента 102(1). При выравнивании сетевой нагрузки на основе информации работоспособности и/или нагрузки вероятность того, что второе соединение будет маршрутизировано на 108(2), будет не больше, чем случайный шанс.

Если же инфраструктура 106 выравнивания нагрузки имеет доступ к отображению между информацией сеанса и хостами 108, то инфраструктура 106 выравнивания нагрузки может маршрутизировать запросы соединения, относящиеся к ранее установленным сеансам, на соответствующий хост 108. Некоторую информацию сеанса можно вывести из содержимого пакетов, переносимых через инфраструктуру 106 выравнивания нагрузки. Однако этот подход неточен и нецеленаправлен по ряду причин. Во-первых, установление и окончание сеанса всего лишь предполагается. Во-вторых, некоторые сеансы не заканчиваются «официально» с надлежащим указанием, включаемым в пакет. Например, некоторые сеансы просто приостанавливаются. В-третьих, пакеты, передаваемые от хоста 108(2) на клиент 102(1), могут перемещаться по пути, который не включает в себя инфраструктуру 106 выравнивания нагрузки, что мешает инфраструктуре 106 выравнивания нагрузки отслеживать такие пакеты на предмет информации сеанса.

Согласно фиг.19 хосты 108 предоставляют информацию 1902 сеанса (ИС) инфраструктуре 106 выравнивания нагрузки. Используя информацию 1902 сеанса от хостов 108, блок 1904 сохранения сродства к сеансу может сохранять сродство между установленным сеансом и хостом 108, на котором был установлен сеанс. Информация 1902 сеанса включает в себя связь или отображение между каждым сеансом, установленным между клиентом 102 и конкретным хостом 108, и этим конкретным хостом 108. Это отображение доступно блоку 1902 сохранения сродства к сеансу как часть отображения 1906 хост - информация сеанса. Более конкретные

примеры информации 1902 сеанса приведены ниже, особенно со ссылкой на фиг.20, 22, 23A и 23B.

В определенных описанных реализациях отслеживания сеансов логическая природа клиентов 102 является подходящей. Согласно указанному выше со ссылкой на фиг.1 клиент 102 может быть конкретным устройством и/или конкретным пользователем устройства. Поэтому сродство к сеансу для пользовательского клиента 102, который осуществляет доступ к хостам 108 с различных устройств, все еще может сохраняться. Поэтому продолжения сеансов с использованием информации 1902 сеансов могут все еще осуществляться в сценариях посредника (например, как у некоторых поставщиков услуг интернета (ПУИ)).

Согласно примеру соединения [1] сеанс, установленный на хосте 1802(2), предоставляется инфраструктуре 106 выравнивания нагрузки как информация 1902 сеанса. В частности, связь/отображение между (i) контекстом сеанса клиента 102(1) и хостом 108(2) и (ii) идентификатор для хоста 108(2) создаются на отображении 1906 хост - информация сеанса. Когда впоследствии поступает запрос соединения для соединения [2] для того же контекста сеанса, блок 1904 сохранения сродства к сеансу размещает этот контекст сеанса в отображении 1906 хост - информация сеанса и выясняет, что хост 108(2) связан с контекстом сеанса из связи/отображения.

В соответствии с отображением хоста 108(2) на запрашиваемый контекст сеанса выявленным блоком 1904 сохранения сродства к сеансу из отображения 1906 хост - информация сеанса соединение [2] маршрутизируется на хост 108(2). В этом смысле сохранение сродства к сеансу имеет более высокий приоритет для инфраструктуры 106 выравнивания нагрузки, чем решения по выравниванию сетевой нагрузки на основании работоспособности и нагрузки приложения. Однако работоспособность и нагрузка могут быть более важными факторами выравнивания сетевой нагрузки, чем отслеживание сеанса, например, в случае чрезвычайно высокой нагрузки или сбойного состояния приложения, относящегося к сеансу, и/или хоста.

Многие типы соединений могут быть связанными с сеансом. Примеры включают в себя: TCP-соединение, сеанс по протоколу защиты транспортного уровня (TLS)/SSL, сеанс PPTP, сеанс IPSec/L2TP, сеанс ISA, сеанс на основе куки HTTP, сеанс терминального сервера, сеанс, заданный администратором, и т.д. Для наглядности TCP-соединение рассматривается как сеанс TCP-пакетов. Кроме того, может перечисляться и поддерживаться модель задания сеансов администратором. Кроме того, могут также поддерживаться сеансы на основе IP-адреса клиента, разграниченные простоями. Это сравнительно неинтеллектуальная поддержка сеанса, но ожидается некоторыми пользователями.

Запрос соединения от клиента 102 различается по типу нужного сеанса. Например, для сеансов типа «TCP-соединение» запрос соединения содержит TCP-пакет. Для сеансов типа «сеанс SSL» запрос соединения содержит TCP-соединение. Другие такие запросы соединения соответствуют другим типам сеанса. Эти примеры также показывают, какие могут быть уровни сеанса. На нижнем уровне сеанса контекст сеанса для TCP-соединения может включать в себя упорядоченную четверку TCP, номер сеанса, количество переданных/принятых битов и т.д. На более высоком уровне сеанса контекст сеанса для сеанса SSL может включать в себя 32-битовый ИД сеанса, открытый ключ клиента 102, предоставляемый хосту 108, и т.д.

На фиг.20 изображен подход к выравниванию сетевой нагрузки с использованием передачи информации сеанса посредством извещений 2006 и сообщений 2008. Показаны множественные модули 106(1), 106(2),..., 106(u) выравнивания нагрузки и

множественные хосты 108(1), 108(2),..., 108(n). Каждый соответствующий хост 108(1), 108(2),..., 108(n) включает в себя одно или несколько соответствующих приложений 316(1), 316(2),..., 316(n), размещенные и выполняющиеся на нем. Извещения 2006 используются для предоставления информации сеанса от приложений 316, и сообщения 2008 используются для предоставления информации сеанса от хостов 108 на модули 106 выравнивания нагрузки.

Показано, что каждый соответствующий хост 108(1), 108(2),..., 108(n) содержит инфраструктуру 2002(1), 2002(2),..., 2002(n) отслеживания сеанса (ИОС). Каждая соответствующая инфраструктура 2002(1), 2002(2),..., 2002(n) отслеживания сеанса содержит соответствующую таблицу 2014(1), 2014(2),... 2014(n) сеансов (хотя на фиг.19 явно показана только одна таблица 2014(1) сеансов).

Каждый соответствующий модуль 106(1), 106(2),..., 106(u) выравнивания нагрузки содержит соответствующую функцию 2012(1), 2012(2),..., 2012(u) маршрутизации трафика (ФМТ). Функция 2012 маршрутизации трафика может содержать, например, функцию классификации и/или запрашивания маршрутизации, например, обеспечиваемую классификатором 304 и маршрутизатором 306 запросов соответственно. Распределенным по модулям 106(1), 106(2),..., 106(u) является распределенный диспетчер 2010 отслеживания сеансов.

В описываемой реализации функция 2012 маршрутизации трафика и распределенный диспетчер 2010 отслеживания сеансов являются частью инфраструктуры 106 выравнивания нагрузки. Инфраструктура 2002 отслеживания сеансов также может быть (например, удаленной) частью инфраструктуры 106 выравнивания нагрузки.

API 2004 используется для предоставления информации сеанса от приложений 316 на инфраструктуру 2002 отслеживания сеансов. Использование API 2004 позволяет приложениям 316 сообщать инфраструктуре 2002 отслеживания сеансов информацию сеанса, включая различные ее изменения. В частности, каждое приложение 316 способно выдавать, а инфраструктура 2002 отслеживания сеансов способна получать, извещения 2006.

При новом установлении или открытии сеанса приложение 316 выдает извещение об установлении сеанса (или извещение 2006(E) об установлении сеанса). Извещение 2006(E) об установлении сеанса содержит идентификатор сеанса и, в необязательном порядке, идентификатор приложения 316. При окончании или закрытии сеанса приложение 316 выдает извещение об окончании сеанса (или извещение 2006(T) об окончании сеанса). Извещение 2006(T) об окончании сеанса также содержит идентификатор сеанса и, в необязательном порядке, идентификатор приложения 316.

Получив извещение 2006(E) об установлении сеанса, инфраструктура 2002 отслеживания сеансов вставляет в таблицу 2014 сеансов элемент для нового сеанса. Иллюстративная таблица 2014 сеансов описана ниже со ссылкой на фиг.23A. Получив извещение 2006(T) об окончании сеанса, инфраструктура 2002 отслеживания сеансов удаляет из таблицы 2014 сеансов элемент для старого сеанса.

Таблица 2014(1) сеансов является официальным источником информации 1902 сеансов в отношении приложений 316(1) на хосте 108(1). Однако при запрашивании функции 2012 маршрутизации трафика на контакт с хостами 108 для доступа к таблицам 2014 сеансов по получении каждого входящего запроса соединения, имеющего ссылку на сеанс, имеет место слишком большая задержка. Поэтому информация 1902 сеансов кэшируется на модулях 106 выравнивания нагрузки.

На модулях 106 выравнивания нагрузки распределенный диспетчер 2010 отслеживания сеансов кэширует информацию 1902 сеанса, что входит в его обязанности по управлению отслеживанием сеансов. В общем случае распределенный диспетчер 2010 отслеживания сеансов является распределенным приложением и/или виртуальной службой, размещенной частично на каждом модуле 106 выравнивания нагрузки. Для каждого логического сеанса распределенный диспетчер 2010 отслеживания сеансов сохраняет, по меньшей мере, одну кэшированную копию информации сеанса для него надежным и масштабируемым образом, которую можно быстро использовать для маршрутизации трафика при поступлении входящих запросов соединения, имеющих ссылку на сеанс, на инфраструктуру 106 выравнивания нагрузки.

Связь между хостами 108 и модулями 106 выравнивания нагрузки осуществляется с помощью надежного протокола, который гарантирует, что сообщения 2008, отправленные с хоста 108, поступают на нужный модуль 106 выравнивания нагрузки. Каждый хост 108 привязан к, по меньшей мере, одному конкретному модулю 106 выравнивания нагрузки, который является назначенным модулем 106 выравнивания нагрузки для сообщений 2008. Эта привязка создается путем присваивания IP-адреса конкретного модуля 106 выравнивания нагрузки каждому хосту 108 для передачи сообщений 2008 отслеживания сеансов между инфраструктурой 2002 отслеживания сеанса и распределенным диспетчером 2010 отслеживания сеанса. Для повышения надежности инфраструктуры 106 выравнивания нагрузки, в случае отказа модуля 106 выравнивания нагрузки, другой модуль 106 выравнивания нагрузки принимает на себя IP-адрес сбойного модуля 106 выравнивания нагрузки. Обнаружение сбоев для принятия IP-адреса может осуществляться с использованием «пульса» или другой схемы отслеживания работоспособности.

Таким образом, сообщения 2008 переносят информацию 1902 сеанса от инфраструктуры 2002 отслеживания сеанса на распределенный диспетчер 2010 отслеживания сеансов. Например, когда инфраструктура 2002 отслеживания сеанса принимает извещение 2006(E) об установлении сеанса, она также отправляет сообщение 2008(U) начала сеанса на распределенный диспетчер 2010 отслеживания сеансов. Сообщение 2008(U) начала сеанса содержит идентификатор сеанса, идентификатор хоста и, в необязательном порядке, другую информацию. Содержимое сообщения 2008(U) начала сеанса описано ниже со ссылкой на фиг.23B в отношении информации, которая может сохраняться для каждого сеанса посредством реализации распределенного диспетчера 2010 отслеживания сеансов. Когда инфраструктура 2002 отслеживания сеанса принимает извещение 2006(T) об окончании сеанса, она также отправляет сообщение 2008(D) окончания сеанса на распределенный диспетчер 2010 отслеживания сеансов. Передача сообщений 2008 может осуществляться до, во время или после того, как инфраструктура 2002 отслеживания сеанса надлежащим образом изменит таблицу 2014 сеансов в соответствии с извещениями 2006.

На фиг.21 показана логическая блок-схема 2100 способа выравнивания сетевой нагрузки с использованием передачи информации сеанса посредством извещений и сообщений. Логическая блок-схема 2100 содержит пятнадцать блоков 2102-2130. Хотя действия логической блок-схемы 2100 могут осуществляться в других средах и с помощью различных других программных схем, фиг.1-3 и 19-20 используются, в частности, для иллюстрации определенных аспектов и примеров способа.

Например, действия четырех блоков 2102-2104 и 2118-2120 осуществляются приложением 316, действия шести блоков 2106-2110 и 2122-2126 осуществляются

инфраструктурой 2002 отслеживания сеансов, и действия пяти блоков 2112-2116 и 2128-2130 осуществляются распределенным диспетчером 2010 отслеживания сеансов. Действия восьми из этих блоков 2102-2116, главным образом, направлены на открытие сеанса, и действия семи из этих блоков 2118-2130, главным образом, направлены на закрытие сеанса.

На блоке 2102 происходит открытие сеанса. Например, приложение 316 может открывать сеанс с клиентом 102. На блоке 2104 предоставляется извещение об установлении сеанса. Например, приложение 316 может предоставлять извещение 2006(E) об установлении сеанса инфраструктуре 2002 отслеживания сеансов с использованием API 2004 в результате открытия сеанса и/или совместно с ним.

На блоке 2106 происходит получение извещения об установлении сеанса. Например, инфраструктура 2002 отслеживания сеансов может получать извещение 2006(E) об установлении сеанса от приложения 316 в соответствии с API 2004. На блоке 2108 в таблицу сеансов вставляется элемент. Например, инфраструктура 2002 отслеживания сеансов может вставлять элемент в таблицу 2014 сеансов для открытого сеанса. Примеры такого вставления описаны ниже, в особенности со ссылкой на фиг.23A. На блоке 2110 происходит отправка сообщения начала сеанса. Например, инфраструктура 2002 отслеживания сеансов может отправлять сообщение 2008(U) начала сеанса на распределенный диспетчер 2010 отслеживания сеансов с использованием надежного протокола связи.

На блоке 2112 происходит получение сообщения начала сеанса. Например, распределенный диспетчер 2010 отслеживания сеансов может получать сообщение 2008(U) начала сеанса от инфраструктуры 2002 отслеживания сеансов в соответствии с надежным протоколом связи. На блоке 2114 создается элемент информации сеанса. Например, распределенный диспетчер 2010 отслеживания сеансов может создавать элемент информации сеанса для кэшированной информации 1902 сеанса на одном или нескольких модулях 106 выравнивания нагрузки. Примеры такого создания и последующего добавления описаны ниже, в особенности, со ссылкой на фиг.22 и 23B.

На блоке 2116 сетевой трафик маршрутизируется с помощью информации сеанса. Например, функция 2012 маршрутизации трафика совместно с распределенным диспетчером 2010 отслеживания сеансов может использовать кэшированную информацию 1902 сеанса, включая созданный элемент информации сеанса, для маршрутизации входящих запросов соединения, имеющих ссылку на сеанс. Пример такой маршрутизации трафика описан ниже, в особенности, со ссылкой на фиг.24. Дополнительные примеры описаны ниже в разделе, озаглавленном «Классификация, пересылка и маршрутизация запросов».

На блоке 2118 сеанс закрывается. Например, приложение 316 может закрыть сеанс с клиентом 102. На блоке 2120 предоставляется извещение об окончании сеанса. Например, приложение 316 может предоставлять извещение 2006(T) об окончании сеанса инфраструктуре 2002 отслеживания сеансов с использованием API 2004 в результате закрытия сеанса и/или совместно с ним.

На блоке 2120 происходит получение извещения об окончании сеанса. Например, инфраструктура 2002 отслеживания сеансов может получать извещение 2006(T) об окончании сеанса от приложения 316 в соответствии с API 2004. На блоке 2124 происходит удаление элемента из таблицы сеансов. Например, инфраструктура 2002 отслеживания сеансов может удалять элемент из таблицы 2014 сеансов для закрытого сеанса. На блоке 2126 происходит отправка сообщения окончания сеанса. Например,

инфраструктура 2002 отслеживания сеансов может отправлять сообщение 2008(D) окончания сеанса на распределенный диспетчер 2010 отслеживания сеансов с использованием надежного протокола связи.

На блоке 2128 происходит получение сообщения окончания сеанса. Например, распределенный диспетчер 2010 отслеживания сеансов может получать сообщение 2008(D) окончания сеанса от инфраструктуры 2002 отслеживания сеансов в соответствии с надежным протоколом связи. На блоке 2130 элемент информации сеанса уничтожается. Например, распределенный диспетчер 2010 отслеживания сеансов может уничтожать элемент информации сеанса в кэшированной информации 1902 сеанса на любых модулях 106 выравнивания нагрузки, имеющих элемент информации сеанса. Примеры такого уничтожения и последующего удаления описаны ниже, в особенности со ссылками на фиг.22 и 23B.

На фиг.22 показан подход к управлению информацией сеанса на множественных модулях 106 выравнивания нагрузки. Каждый соответствующий модуль 106(1), 106(2),..., 106(u) выравнивания нагрузки содержит соответствующую часть 2202(1), 2202(2),..., 2202(u) распределенного диспетчера 2202 атомов (РДА). РДА 2202 является реализацией распределенного диспетчера 2010 отслеживания сеансов. Каждая соответствующая часть 2202(1), 2202(2),..., 2202(u) РДА содержит соответствующую часть 2206(1), 2206(2),..., 2206(u) таблицы РДА (ТРДА) 2206.

РДА 2202 является распределенным приложением или виртуальной службой, которая управляет информацией 1902 сеанса надежным и масштабируемым образом, что позволяет функции 2012 маршрутизации трафика использовать его для сохранения сродства к сеансу. Например, функция 2012 маршрутизации трафика может осуществлять доступ к РДА 2202 с использованием API (конкретно не показан) для поиска в ТРДА 2206. Функциональные вызовы 2204, работа РДА 2202 и другие аспекты фиг.22 описаны ниже после описания фиг.23A и 23B.

На фиг.23A показана таблица 2014 сеансов, указанная на фиг.20. Таблица 2014 сеансов содержит «v» элементов 2302(1), 2302(2),..., 2302(v). Каждый элемент 2302 вставляется инфраструктурой 2002 отслеживания сеансов в соответствии с извещением 2006(E) об установлении сеанса, полученным от приложения 316. Каждый элемент 2302 удаляется инфраструктурой 2002 отслеживания сеансов в соответствии с извещением 2006(T) об окончании сеанса, полученным от приложения 316.

Согласно описанному выше каждое извещение 2006(E) об установлении сеанса содержит идентификатор сеанса и, в необязательном порядке, идентификатор приложения 316. Каждый соответствующий элемент 2302(1), 2302(2),..., 2302(v) таблицы 2014 сеансов содержит соответствующие поля (i) идентификатора 2302(1I), 2302(2I),..., 2302(vI) сеанса и (ii) тип сеанса и/или приложение 2302(1T), 2302(2T),..., 2302(vT).

Тип сеанса и/или приложение 2302(T) может быть "TCP", "IPSEC", «Терминальный сервер», «куки HTTP», тип приложения, как отмечено выше, и т.д. Идентификатор 2302(I) сеанса может представлять собой "<IP-адрес источника, TCP-порт источника, IP-адрес назначения, TCP-порт назначения>", "IP клиента=172.30.189.122", "Пользователь='joe_user'", "Куки='{b7595cc9-e68b-4eb0-9bfl-bb717b31d447}'", другой, например, идентификатор сеанса, связанный с приложением, и т.д. Для типов TCP-соединение/сеанс, идентификатор 2302(I) сеанса может, альтернативно, представлять собой каноническую версию упорядоченной четверки TCP (для IPv4 или IPv6). Альтернативно, можно также использовать другие значения для полей идентификатора 2302(I) сеанса и приложения/типа сеанса 2302(T).

На фиг.23B показана таблица 2206 (TРДА) распределенного диспетчера атомов (РДА), обозначенная на фиг.22. Таблица 2206 РДА содержит «w» элементов 2304(1), 2304(2),..., 2304(w). Каждый элемент 2304 информации сеанса создается РДА 2202 в соответствии с сообщением 2008(U) начала сеанса, полученным от инфраструктуры 2002 отслеживания сеансов. Каждый элемент 2304 информации сеанса уничтожается в соответствии с сообщением 2008(D) окончания сеанса, полученным от инфраструктуры 2002 отслеживания сеансов. Согласно описанному ниже элементами 2304 информации сеанса из таблиц 2206 РДА можно реально управлять посредством РДА 2202 с использованием функциональных вызовов 2204.

Согласно описанному выше сообщение 2008(U) начала сеанса содержит идентификатор сеанса, идентификатор хоста и, в необязательном порядке, другую информацию. Каждый соответствующий элемент 2304(1), 2304(2),..., 2304(w) информации сеанса в таблице 2206 РДА содержит соответствующие поля (i) ключа 2304(1K), 2304(2K),..., 2304(wK), (ii) данных 2304(1D), 2304(2D),..., 2304(wD) и (iii) метаданных 2304(1M), 2304(2M),..., 2304(wM). Например, значения полей 2304(K) ключа могут представлять собой буквенно-цифровые строки, а значения полей 2304(D) данных могут представлять собой биты. Значения ключа 2304(K) также могут представлять собой биты.

Ключ 2304(K) может соответствовать идентификатору 2302(I) сеанса. Данные 2304(D) могут соответствовать идентификатору хоста, например, сетевому адресу хоста 108, на котором существует контекст сеанса. Метаданные 2304(M) могут соответствовать другой, необязательной, информации. Примером таких метаданных 2304(M) могут служить данные, внутренне используемые РДА 2202 для разрешения конфликтов атомов и отслеживания работоспособности атомов (например, посредством механизма приостановки). (Это представление элементов 2304 как атомарных более подробно описано в следующем абзаце.) В частности, метаданные 2304(M) включают в себя, помимо прочего, идентификацию сущности (например, экземпляра функции 2012 маршрутизации трафика), добавившей элемент 2304 информации сеанса в таблицу 2206 РДА.

В описываемой реализации каждый элемент 2304 информации сеанса является атомарным в том смысле, что РДА 2202 может добавлять, удалять, копировать и т.д. элементы 2304 как целое, но РДА 2202 обычно не изменяет часть любого элемента 2304 как целого. Таким образом, атомарные элементы 2304 добавляются, удаляются, копируются, иным образом обрабатываются и т.д. по таблицам 2206 РДА посредством РДА 2202 для реализации надежности и масштабируемости для реализации сохранения сродства к сеансу.

Функциональные вызовы 2204 (фиг.22) используются РДА 2202 для манипулирования атомарными элементами 2304 таблицы 2206 РДА. Функциональные вызовы 2204 могут поступать от одного модуля 106 выравнивания нагрузки на один или несколько модулей 106 выравнивания нагрузки в двухточечном или широковещательном режиме. Эти функциональные вызовы включают в себя «добавить атом» 2204(A), «удалить атом» 2204(D), «запросить атом» 2204(Q) и «возвратить атом» 2204(R).

«Добавить атом» 2204(A) имеет вид AddAtom(ключ, данные) и используется для добавления атомарного элемента 2304 в одну или несколько таблиц 2206. Поэтому функциональный вызов «добавить атом» 2204(A) можно выразить как AddAtom(<идентификатор сеанса>IP-адрес хоста). «Удалить атом» 2204(D) имеет вид DeleteAtom(ключ) и используется для удаления атомарного элемента 2304 из одной

или нескольких таблиц 2206 РДА. Функциональные вызовы «удалить атом» 2204(D) могут быть направлены на те таблицы 2206 РДА, про которые известно, что они имеют копию сеанса, идентифицированного ключом 2304(K), или могут рассылаться на все таблицы 2206 РДА, что позволяет гарантировать удаление любых копий.

«Запросить атом» 2204(Q) имеет вид QueryAtom(ключ) и используется конкретной частью 2202 РДА, когда идентификатор сеанса, на который ссылается входящий запрос соединения, не находится в конкретной локальной таблице 2206 РДА или конкретной части 2202 РДА. Функциональные вызовы «запросить атом» 2204(Q) отправляются на одну или несколько (возможно, все) другие части 2202 РДА. В ответ каждая часть 2202 РДА проверяет свою локальную таблицу 2206 РДА на предмет ключа/идентификатора сеанса. Если ключ обнаруживается другой частью 2202 РДА, то эта другая часть 2202 РДА отвечает посредством «возвратить атом» 2204(R).

«Возвратить атом» 2204(R) имеет вид ReturnAtom(ключ, данные) и используется для ответа на функциональный вызов «запросить атом» 2204(Q). Функциональные вызовы «возвратить атом» 2204(R) используются, когда часть 2202 РДА имеет запрошенный атомарный элемент 2304 в своей локальной таблице 2206 РДА, идентифицированный ключом 2304(K), указанным в функциональном вызове «запросить атом» 2204(Q). Функциональные вызовы «возвратить атом» 2204(R) могут быть направлены обратно на часть 2202 РДА, выдавшую функциональный вызов «запросить атом» 2204(Q).

Функциональные вызовы «добавить атом» 2204(A) используются в ответ на сообщения 2008(U) начала сеанса и/или для дублирования атомарного элемента 2304 в одну или более других таблиц 2206 РДА. Такое дублирование может выполняться для избыточности и/или масштабируемости.

Функциональные вызовы «удалить атом» 2204(D) используются в ответ на сообщения 2008(D) окончания сеанса и также могут отправляться на одну или несколько таблиц 2206 РДА. После удаления атомарного элемента 2304 атомарный элемент 2304 может войти в состояние «зомби», при котором он остается в РДА 2202 и, в необязательном порядке, по-прежнему хранится в таблице 2206 РДА с указателем «зомби» в поле метаданных 2304(M) атомарного элемента 2304.

Таким образом, после удаления атомарного элемента 2304 он может оставаться в РДА 2202 и таблице 2206 РДА в состоянии «зомби», в результате чего пакеты для этого (теперь неработающего и закрытого) сеанса направляются на хост 108 контекста сеанса для надлежащей, в зависимости от протокола, обработки. Например, TCP-пакеты, полученные после устранения TCP-соединения, направляются на хост 108, который прекратил соединение. Этот хост 108 может надлежащим образом ответить, возможно, послав RST или повторно послав FIN-ACK. Время, которое атомарный элемент 2304 проводит в этом состоянии «зомби», совпадает (по возможности точно) с зависящим от протокола временем простоя для используемого надежного протокола связи.

Функциональный вызов «запросить атом» 2204(Q) используется для доступа к атомарному элементу 2304, когда первый модуль 106 выравнивания нагрузки принимает входящий запрос соединения, ссылающийся на сеанс, который не хранится в локальной таблице 2206 РДА для РДА 2202 первого модуля 106 выравнивания нагрузки. Заметим, что другие части 2202 РДА можно одновременно запрашивать посредством широковещательного функционального вызова «запросить атом» 2204(Q) или последовательно, пока не будет получен положительный функциональный вызов «возвратить атом» 2204(R).

Функциональный вызов «возвратить атом» 2204(R) используется частью 2202 РДА

второго модуля 106 выравнивания нагрузки для предоставления атомарного элемента 2304 части 2202 РДА первого модуля 106 выравнивания нагрузки, когда атомарный элемент 2304 имеет ключ 2304(K), заданный ключом/идентификатором сеанса в функциональном вызове «запросить атом» 2204(Q), ранее выданном частью 2202 РДА первого модуля 106 выравнивания нагрузки. Заметим, что другие компоненты, например, функция 2012 маршрутизации трафика, также может пользоваться функциональными вызовами 2204, в особенности функциональным вызовом «запросить атом» 2204(Q), в соответствии с API или подобным.

Части 2202 РДА и таблицы 2206 РДА можно организовывать и администрировать всевозможными способами. Иллюстративные способы относятся к дублированию/избыточности, локальному кэшированию после получения, хэшированию для выбора местоположения и т.д., можно использовать нулевой, первый, второй или более высокий уровень дублирования вплоть до полного дублирования. При нулевом уровне дублирования каждый атомарный элемент 2304 сохраняется в РДА 2202, принимающем сообщение 2008(U) начала сеанса для него, без дублирования в другие части 2202 РДА.

При первом уровне дублирования каждый атомарный элемент 2304 сохраняется в РДА 2202, принимающем сообщение 2008(U) начала сеанса для него, и также добавляется (копируется) в другую часть 2202 РДА с использованием функционального вызова «добавить атом» 2204(A). Это позволяет справиться с одним уровнем сбоя для модуля 106 выравнивания нагрузки. Аналогично, при втором уровне дублирования каждый атомарный элемент 2304 сохраняется в РДА 2202, принимающем сообщение 2008(U) начала сеанса для него, и также добавляется в две другие части 2202 РДА. В целом, одна, две и т.д. части 2202 РДА, в которые данная часть 2202 РДА копирует атомарные элементы 2304, заранее определены или выбираются произвольно. Можно также применять третий, четвертый и т.д. уровни дублирования.

Кроме того, можно использовать полное дублирование, при котором каждый атомарный элемент 2304, который сохраняется на РДА 2202, принимающем сообщение 2008(U) начала сеанса, также добавляется в каждую другую часть 2202 РДА. На выбор уровня дублирования влияют несколько факторов. По мере возрастания уровня дублирования надежность повышается, а задержка уменьшается. С другой стороны, сетевой трафик и использование памяти возрастают с повышением уровня дублирования.

Когда полное дублирование не используется, возможно локальное кэширование после получения. Например, когда часть 2202 РДА не обнаруживает идентификатора сеанса, на который указывает ссылка, в своей части таблицы 2206 РДА, часть 2202 РДА выдает функциональный запрос «запросить атом» 2204(Q), чтобы получить доступ к атомарному элементу 2304, связанному с идентификатором сеанса, на который указывает ссылка, посредством функционального вызова «возвратить атом» 2204(R). Вместо того чтобы отбрасывать атомарный элемент 2304 после его использования, часть 2202 РДА кэширует полученный атомарный элемент 2304 в своей части таблицы 2206 РДА. Эта опция дает возможность выбора между вышеперечисленными факторами.

Еще одна опция в отсутствие использования полного дублирования может состоять в кэшировании для выбора местоположения. Первый атомарный элемент 2304 для сеанса сохраняется в части 2202 РДА, которая принимает сообщение 2008(U) начала сеанса. Дублированная копия или копии отправляются посредством функциональных

вызовов «добавить атом» 2204(A) на конкретную(ые) часть(и) 2202 РДА с использованием хэш-функции. Из множества всевозможных хэш-значений каждой части 2202 РДА присваивается его подмножество. Каждый идентификатор сеанса хэшируется с использованием некоторой хэш-функции для получения хэширующего значения. Это хэширующее значение отображается на присвоенные части 2202 РДА. Часть 2202 РДА, которая первая добавила атомарный элемент 2304, затем дублирует атомарный элемент 2304 в присвоенную(ые) часть(и) 2202 РДА.

Благодаря хэшированию для выбора местоположения, по меньшей мере, одна часть 2202 РДА, которая локально кэшировала нужный атомарный элемент 2304 в своей таблице 2206 РДА, доступна по идентификатору сеанса. Поэтому функциональный вызов «запросить атом» 2204(Q) может быть направлен на известную(ые) часть(и) 2202 РДА. Это обычно снижает сетевой трафик и/или задержку.

Это хэширование для выбора местоположения можно использовать на первом, втором, третьем или более уровнях дублирования, причем каждый диапазон хэширующих значений отображается на одну, две, три и т.д. разные части 2202 РДА, соответственно. Дополнительно, хэширование для выбора местоположения можно использовать совместно с локальным кэшированием после получения.

На фиг.24 показана логическая блок-схема 2400 способа управления информацией сеанса на множественных модулях выравнивания нагрузки. Логическая блок-схема 2400 содержит восемь блоков 2402-2416. Хотя действия логической блок-схемы 2400 могут осуществляться в других средах и с помощью различных программных схем, фиг.1-3, 19, 20, 22 и 23B используются, в частности, для иллюстрации определенных аспектов и примеров способа.

На блоке 2402 анализируется входящий запрос соединения со ссылкой на сеанс. Например, функция 2012 маршрутизации трафика может получить входящий запрос соединения, который ссылается на ранее открытый/установленный сеанс определенного типа. На блоке 2404 осуществляется поиск в локальной таблице РДА с использованием ссылки на сеанс. Например, для данных модуля 106 выравнивания нагрузки и функции 2012 маршрутизации трафика его часть 2202 РДА может осуществлять поиск в своей соответствующей таблице 2206 РДА по ссылке на сеанс.

На блоке 2406 производится определение, совпадает ли ссылка на сеанс с ключом локальной таблицы РДА. Например, часть 2202 РДА может производить поиск в полях 2304(K) ключа множественных элементов 2304 таблицы 2206 РДА, чтобы определить, совпадает ли ссылка на сеанс с какими-либо значениями полей 2304(K) ключа. Если да, то логическая блок-схема 2400 переходит к блоку 2412.

Если же ссылка на сеанс не совпадает ни с каким ключом, то логическая блок-схема 2400 переходит к блоку 2408. На блоке 2408 осуществляется функциональный вызов «запросить атом». Например, часть 2202 РДА может осуществлять функциональный вызов «запросить атом» 2204(Q), содержащий в качестве ключа ссылку на сеанс/идентификатор сеанса. Функциональный вызов «запросить атом» 2204(Q) может быть отправлен на, по меньшей мере, одну другую часть 2202 РДА. Количество, выбор, порядок и т.д. возможных частей 2202 РДА назначения для «запросить атом» 2204(Q) может зависеть от опций (например, уровня дублирования, хэширования для выбора местоположения, локального кэширования после получения, двухточечного/широковещательного режима и т.д.), применяемых в РДА 2202.

На блоке 2410 происходит получение возвращенного атома. Например, может быть получена информация от функционального вызова «возвратить атом» 2204(R),

выданного другой частью 2202 РДА. Другая часть 2202 РДА успешно обнаружила атомарный элемент 2304 в своей соответствующей таблице 2206 РДА, причем обнаруженный атомарный элемент 2304 имеет ключ, совпадающий со ссылкой на сеанс. Информация от функционального вызова «возвратить атом» 2204(R) содержит значения из поля 2304(K) и поля 2304(D) данных для обнаруженного атомарного элемента 2304. Эти значения соответствуют идентификатору сеанса для сеанса и сетевому адресу хоста 108, который имеет сродство к сеансу.

На блоке 2412 происходит извлечение атомарного элемента. Атомарный элемент извлекается из локальной таблицы РДА, если совпадение найдено локально (на блоках 2404 и 2406) или из возвращенного атома, если совпадение найдено в другом месте (на блоках 2408 и 2410). Например, атомарный элемент 2304 может быть извлечен из таблицы 2206 РДА части 2202 РДА или из информации, полученной посредством функционального вызова «возвратить атом» 2204(R). Извлеченный атомарный элемент 2304 может быть кэширован в локальной таблице 2206 РДА, если он получен в результате функционального вызова «возвратить атом» 2204(R).

На блоке 2414 из атомарного элемента выявляют хост, имеющий сродство к сеансу, на который указывает ссылка. Например, значение поля 2304(D) данных извлеченного атомарного элемента 2304 можно узнать, чтобы, таким образом, узнать сетевой адрес хоста 108, имеющего сродство к сеансу. На блоке 2416 входящий запрос соединения маршрутизируется на выявленный хост, например, функция 2012 маршрутизации трафика и/или функция пересылки может маршрутизировать входящий запрос соединения, имеющий ссылку на сеанс на выявленный хост 108, имеющий сродство к сеансу. В следующем разделе описаны иллюстративные функции классификации, маршрутизации запросов и пересылки.

Классификация, пересылка и маршрутизация запроса

В этом разделе описано, как можно реализовать маршрутизацию трафика для выравнивания сетевой нагрузки, в том числе в отношении высокой надежности такой функции маршрутизации трафика. Функция маршрутизации трафика может включать в себя функцию классификации и/или запрашивания маршрутизации, особенно в связи с функцией пересылки. В этом разделе ссылки идут, главным образом, на фиг.25-31. Он освещает функцию маршрутизатора 306 запросов (фиг.3), взаимоотношения между отслеживанием сеансов и использованием информации работоспособности и нагрузки при маршрутизации трафика, эксплуатационных реализаций взаимодействий маршрутизации трафика с информацией сеанса и/или информацией работоспособности и нагрузки, процедур преодоления сбоя для высокой надежности инфраструктуры выравнивания сетевой нагрузки (включая обработку сбоев компонентов классификации, пересылки и/или маршрутизации запросов), дополнительных конфигураций инфраструктуры выравнивания сетевой нагрузки и т.д.

На фиг.25 показана инфраструктура выравнивания сетевой нагрузки, имеющая функцию маршрутизации запросов, реализованную маршрутизатором 306(H/S) запросов. Согласно отмеченному выше со ссылкой на функцию 2012 маршрутизации трафика маршрутизация трафика может опираться на классификацию (например, с пересылкой) и/или запрашивание маршрутизации. Классификация на уровне пакетов, совместно с пересылкой, описана выше, в частности, со ссылкой на фиг.4. Маршрутизация запросов описана здесь, в частности, со ссылкой на фиг.25.

Маршрутизация на уровне запросов происходит на более высоком уровне, чем маршрутизация на уровне пакетов. В общем случае маршрутизатор 306 запросов действует как посредник для приложения 316, работающего на хосте 108.

Маршрутизатор 306 запросов заканчивает TCP-соединения, анализирует (возможно, частично) каждый запрос от клиента 102 и перенаправляет каждый запрос на хост 108. Маршрутизатор 306 запросов может осуществлять предварительную обработку на соединении, например, SSL-дешифрование. Кроме того, маршрутизатор 306 запросов может выбирать поглощение определенных запросов (например, маршрутизатор запросов поддерживает кэш ответов) и может по своему усмотрению изменять запросы прежде, чем пересылать их на хосты 108.

Маршрутизаторы 306 запросов обычно зависят от приложения и могут быть достаточно расширяемыми в отношении того, что они могут делать. Исключительно в порядке примера в нижеследующем описании рассмотрен единственный класс маршрутизаторов 306 запросов - маршрутизаторы 306(H/S) запросов HTTP/SSL. Показано, что клиент 102, имеющий сетевой адрес C1, связывается по сети 104 с хостами 108(1) и 108(2), имеющими сетевые адреса H1 и H2 соответственно. Связь осуществляется посредством инфраструктуры выравнивания нагрузки, которая содержит маршрутизатор 306(H/S) запросов HTTP/SSL.

Маршрутизатор 306(H/S) запросов HTTP/SSL заканчивает трафик HTTP и SSL, расшифровывает трафик SSL, проверяет каждый запрос HTTP от клиента 102, применяет зависящие от приложения правила для классификации каждого запроса и для определения «наилучшей» конечной точки для этого запроса, в то же время учитывая информацию работоспособности и нагрузки конечной точки приложения, и направляет запрос на конечную точку. Для подачи запроса на конечную точку используется отдельное TCP-соединение, отличное от инициированного клиентом 102 (последнее соединение заканчивается на маршрутизаторе 306(H/S) запросов HTTP/SSL).

Эти действия можно рассматривать как логически эквивалентные действиям, производимым классификатором 304, но с той разницей, что эти действия в маршрутизаторе 306(H/S) запросов HTTP/SSL осуществляются на уровне логических запросов для каждого запроса в TCP-соединении. Маршрутизатор 306(H/S) запросов HTTP/SSL и, вообще, маршрутизаторы 306 запросов могут использовать те же инфраструктуры (i) работоспособности и нагрузки и (ii) отслеживания сеансов, которые используются классификаторами 304.

Маршрутизатор 306(H/S) запросов HTTP/SSL действует как посредник между клиентом 102 и двумя хостами 108(1) и 108(2). Он обрабатывает два запроса от клиента 102 по одному TCP-соединению. В описываемой реализации окончательная маршрутизация запросов предусматривает ряд действий. Во-первых, клиент 102 устанавливает HTTP- или HTTPS-соединение [1] с маршрутизатором 306(H/S) запросов HTTP/SSL и посылает запрос №1 2502(1).

Во-вторых, маршрутизатор 306(H/S) запросов HTTP/SSL заканчивает сеанс SSL (если трафик зашифрован посредством SSL), анализирует запрос №1 2502(1) и проверяет содержимое запроса №1 2502(1). С учетом информации работоспособности и нагрузки, а также сеанса маршрутизатор 306(H/S) запросов HTTP/SSL определяет, что хост 108(1) является «наилучшим» хостом для этого конкретного запроса №1 2502(1) в этом примере.

В-третьих, маршрутизатор 306(H/S) запросов HTTP/SSL устанавливает вторичное TCP-соединение [2] с хостом 108(1). Альтернативно, можно использовать существующее соединение [2] с хостом 108(1). Затем маршрутизатор 306(H/S) запросов HTTP/SSL посылает, например, незашифрованную версию запроса №1 2502(1) на хост 108(1). В-четвертых, хост 108(1) отвечает посредством ответа №1 2504(1). В-пятых, маршрутизатор 306(H/S) запросов HTTP/SSL шифрует этот ответ №1

2504(1) и отправляет его обратно на клиент 102 по TCP-соединению [1].

В-шестых, клиент 102 посылает другой запрос, запрос №2 2502(2). Запрос №2 2502(2) обрабатывается аналогично запросу №1 2502(1), за исключением того, что маршрутизатор 306(H/S) запросов HTTP/SSL выбирает хост 108(2). Причина другого выбора может быть в том, что хост 108(1) в данный момент находится в нерабочем состоянии или перегружен, поскольку запрос №2 2502(2) направлен по другому URL, чем запрос №1 2502(1), и т.д. Так или иначе маршрутизатор 306(H/S) запросов HTTP/SSL устанавливает другое вторичное TCP-соединение, но это вторичное TCP-соединение [3] организовано с хостом 108(2). Незашифрованный запрос №2 2502(2) маршрутизируется на хост 108(2), и в результате ответ №2 2504(2) поступает оттуда. Затем, зашифрованная версия ответа №2 2504(2) отправляется на клиент 102 от маршрутизатора 306(H/S) запросов HTTP/SSL.

В-седьмых, клиент 102 закрывает TCP-соединение [1] с маршрутизатором 306(H/S) запросов HTTP/SSL. Маршрутизатор 306(H/S) запросов HTTP/SSL (в некоторый момент в будущем) закрывает соединения [2] и [3], установленные с хостами 108(1) и 108(2) соответственно, со стороны клиента 102. TCP-соединение [2] можно, альтернативно, закрывать после того, как маршрутизатор 306(H/S) запросов HTTP/SSL решит открыть/использовать TCP-соединение [3] для запроса №2 2502(2).

Поскольку маршрутизатор 306(H/S) запросов HTTP/SSL заканчивает HTTP/HTTPS-соединение, то маршрутизатор 306(H/S) запросов HTTP/SSL может не только маршрутизировать запросы. Например, маршрутизатор 306(H/S) запросов HTTP/SSL может, в принципе, поддерживать собственный кэш ответов (например, с помощью внеполосного механизма, чтобы сделать кэш недействительным). Согласно отмеченному в вышеприведенном примере маршрутизатор 306(H/S) запросов HTTP/SSL также может, в принципе, маршрутизировать запросы других видов на другие группы хостов 108 на основании, например, запрашиваемого URL. Опять же, маршрутизатор 306(H/S) запросов HTTP/SSL может, в принципе, агрегировать запросы от многочисленных короткоживущих клиентских соединений и передавать их по немногочисленным долгоживущим TCP-соединениям на хосты 108. Такое агрегирование соединений может снижать служебную нагрузку обработки соединений на хостах 108.

Маршрутизаторы запросов других классов могут соответствовать другим иллюстративным протоколам помимо HTTP. Например, маршрутизатор запросов может представлять собой маршрутизатор запросов SOAP. Маршрутизатор запросов SOAP действует аналогично маршрутизатору 306(H/S) запросов HTTP/SSL. Однако маршрутизатор запросов SOAP предназначен именно для маршрутизации трафика SOAP. Маршрутизаторы запросов SOAP понимают заголовки SOAP и принимают решения по маршрутизации на основании заголовков SOAP, а также работоспособности и нагрузки приложений.

Классификация и пересылка уровня пакетов (или маршрутизация уровня пакетов) и маршрутизация уровня запросов могут обеспечивать некоторый вид выравнивания нагрузки уровня 7. Выравнивание нагрузки уровня 7 описано ниже в разделе, озаглавленном «Перенос соединений с необязательным туннелированием и/или выравниванием нагрузки на уровне приложений». Маршрутизация на уровне пакетов обеспечивает доступ только для чтения к начальной части данных TCP-соединения клиента, а маршрутизация на уровне запросов обеспечивает доступ с возможностью чтения и изменения ко всему потоку данных.

Маршрутизация на уровне пакетов имеет несколько преимуществ над

маршрутизацией на уровне запросов. Эти преимущества включают в себя прозрачность (клиентские пакеты доставляются на хосты в первозданном виде, сохраняя IP-адреса и номера портов источника и назначения), низкую служебную нагрузку обработки (в целом, пересылка трафика предусматривает поиск маршрута), низкую задержку (отдельные пакеты пересылаются, и пакеты не ставятся в очередь после определения пункта назначения TCP-соединения) и высокая надежность (в целом, сбой на блоке пересылки не прекращает TCP-соединение). С другой стороны, маршрутизация на уровне запросов обычно имеет следующие преимущества над маршрутизацией на уровне пакетов: возможность проверять весь поток данных, идущий на и от клиента, и возможность преобразовывать поток данных и даже расщеплять поток данных между множественными хостами или агрегировать потоки данных от множественных клиентов.

На фиг.26 показана логическая блок-схема 2600 способа маршрутизации входящих пакетов в соответствии с (i) информацией сеанса и (ii) информацией работоспособности и нагрузки. Логическая блок-схема содержит восемь блоков 2602-2616. Хотя действия логической блок-схемы 2400 могут осуществляться в других средах и с помощью различных программных схем, фиг.1-3, 12, 18-20, 22 и 23B используются, в частности, для иллюстрации определенных аспектов и примеров способа.

На блоке 2602 происходит прием входящего пакета. Например, пакет от клиента 102 может поступать на блок пересылки 302 модуля 106 выравнивания нагрузки. На блоке 2604 происходит определение, относится ли полученный пакет к уже существующему сеансу. Например, блок 302 пересылки может свериться с локальной таблицей 2206() РДА, чтобы определить, что принятый пакет уже является частью сеанса TCP/IP.

Кроме того, блок 302 пересылки может свериться с локальной таблицей 2206() РДА и определить, что полученный пакет уже не является частью сеанса TCP/IP. В этом случае блок 302 пересылки подает полученный пакет на классификатор 304, который проверяет сродство к сеансу на более высоком уровне для полученного пакета, если он имеет ссылку на сеанс. Примеры таких действий описаны выше, в частности, со ссылкой на фиг.24 и ниже, в частности, со ссылками на фиг.27 и 28.

Если полученный пакет относится к уже существующему сеансу (что определено на блоке 2604), то происходит переход к блоку 2606. На блоке 2606 происходит обнаружение хоста, имеющего сродство к уже существующему сеансу. Например, хост 108, имеющий сродство к сеансу, может быть выявлен из локальной таблицы 2206() РДА и/или общей распределенной таблицы 2206 РДА блоком 302 пересылки или классификатором 304.

На блоке 2608 производится определение, работоспособен ли хост, имеющий сродство к сеансу. Например, классификатор 304 может свериться с объединенным кэшем 1208 работоспособности и нагрузки, чтобы определить, работоспособен ли хост 108, имеющий сродство к сеансу, в особенности для тех принятых пакетов, которые являются частью сеансов, которые находятся на более высоком логическом уровне, чем сеансы TCP/IP. Действие(я) этого блока могут осуществляться совместно с обработчиком 314 работоспособности и нагрузки.

Если хост, имеющий сродство к сеансу, работоспособен (что определено на блоке 2608), то происходит переход к блоку 2610. На блоке 2610 полученный пакет маршрутизируется на хост, имеющий сродство к сеансу. Например, блок 302 пересылки (для сеансов TCP/IP) или классификатор 304 (для сеансов более высокого

уровня) может маршрутизировать пакет на хост 108, имеющий сродство к сеансу. В альтернативной реализации классификатор 304 может возвращать принятый пакет на блок 302 пересылки для маршрутизации на хост 108, имеющий сродство к сеансу, даже для принятых пакетов, которые являются частью сеансов более высокого уровня.

Если же хост, имеющий сродство к сеансу, неработоспособен (что определено на блоке 2608), то происходит переход к блоку 2612. Кроме того, если, с другой стороны, полученный пакет не относится к уже существующему сеансу (что определено на блоке 2604), то происходит переход к блоку 2612. На блоке 2612 осуществляется выбор хоста в соответствии с информацией работоспособности и нагрузки. Например, классификатор 304 может выбирать хост 108 из и/или с использованием распределения для приложения на основании работоспособности и нагрузки (например, из ответа 1804 по распределению целевых конечных точек приложения), полученного от обработчика 314 работоспособности и нагрузки. Примеры этих действий описаны выше, в частности, со ссылкой на фиг.19 и 18 и ниже, в частности, со ссылкой на фиг.30.

На блоке 2614 полученный пакет маршрутизируется на выбранный хост. Например, классификатор 304 может маршрутизировать (в необязательном порядке, через блок 302 пересылки) пакет на выбранный хост 108. На блоке 2616 прокладывается маршрут для пути соединения к выбранному хосту. Например, классификатор 304 может добавить элемент информации сеанса в таблицу 2206 РДА, в особенности в таблицу 2206() РДА, которая является локальной по отношению к блоку 302 пересылки, который передал полученный пакет классификатору 304. Этот элемент информации сеанса можно дублировать в соответствии с установленной политикой избыточности для РДА 2202 (например, блока 308 отслеживания сеансов).

Действия блоков 2614 и 2616 могут осуществляться в конкретно показанном порядке, когда блок 2616 выполняется до блока 2614, когда действия частично или полностью перекрываются в любом порядке и т.д. Заметим, что вышеописанные действия, осуществляемые классификатором 304, могут альтернативно выполняться маршрутизатором 306 запросов (или, в целом, функцией 2012 маршрутизации трафика).

На фиг.27 показана последовательность действий по маршрутизации трафика в отсутствие сбоев. Показано, что перед остальной частью инфраструктуры 106 выравнивания нагрузки (отдельно не указана) имеются один или несколько коммутаторов 202(LBA), знающих о выравнивании нагрузки. Функции пересылки и классификации распределены по трем устройствам или узлам. Первое устройство содержит блок 302(1) пересылки и классификатор 304(1). Второе устройство содержит классификатор 304(2). Третье устройство содержит блок 302(2) пересылки.

С помощью классификатора 304(2), работающего на втором устройстве, и блока 302(2) пересылки, работающего на третьем устройстве, каждое устройство можно конкретно настраивать на его соответствующие функции. Например, аппаратное, программное, аппаратно-программное обеспечение, некоторые их комбинации и т.д. второго устройства и третьего устройства можно приспособить для поддержки нужных функций, не привлекая дополнительных средств. Таким образом, третье устройство, содержащее блок 302(2) пересылки, по своим аппаратным возможностям может быть сродни коммутатору и/или маршрутизатору, а второе устройство, содержащее классификатор 304(2), по своим аппаратным возможностям может больше походить на сервер и/или персональный компьютер.

Хотя показаны три устройства, обеспечивающие функциональные возможности по четырем компонентам, альтернативные логические и/или аппаратные конфигурации

для функций пересылки и классификации применимы к иллюстративной последовательности действий по маршрутизации трафика, описанной на фиг.27. Кроме того, хотя пункты назначения маршрутизации показаны как хосты 108, описанные здесь реализации маршрутизации можно альтернативно применять, в целом, к следующему узлу-адресату для пакета и не обязательно к окончательному узлу, потребляющему пакет.

Реализация РДА 2202 блока 308 отслеживания сеансов используется для реализации таблицы 2206 РДА. Однако блоки 1904 сохранения сродства к сеансу, в общем случае, также применимы к иллюстративной последовательности действий по маршрутизации трафика, показанной на фиг.27. Блок 302(1) пересылки содержит часть 2206(1) таблицы РДА, а блок 302(2) пересылки содержит часть 2206(2) таблицы РДА. Входящие пакеты маршрутизируются на хост 108(1) или хост 108(2).

В описанной реализации РДА 2202 является распределенной, размещенной в памяти, таблицей «атомов» 2304 (например, пар ключевое слово/значение, с необязательными метаданными), имеющих информацию сеанса. РДА 2202, и таблица 2206 РДА описаны выше, в частности, со ссылкой на фиг.22-24. Любой узел в группе классификаторов 304 может добавлять, запрашивать и удалять атомы 2304. РДА 2202 поддерживает высоконадежную таблицу 2206 РДА, которая содержит информацию о действующих маршрутизаторах (например, уровня TCP/IP), а также сеансах более высокого уровня.

Позиция (1) обозначает, что коммутаторы 202(LBA), знающие о выравнивании нагрузки, направляют входящий пакет на блок 302(1) пересылки. Позиция (2) обозначает, что блок 302(1) сверяется со своей внутренней таблицей маршрутизации, таблицей 2206(1) РДА. Когда блок 302(1) пересылки не находит атомарный элемент 2304 для этого пакета, он пересылает пакет на присвоенный ему и/или связанный с ним классификатор, классификатор 304(1).

Позиция (3) обозначает, что классификатор 304(1) распознает, что пакет в данном примере является пакетом TCP-SYN. Поэтому классификатор 304(1) обрабатывает пакет как начало нового TCP-соединения от клиента 102. Используя информацию работоспособности и нагрузки от обработчика 314 (явно не показан) работоспособности и нагрузки (не показан в явном виде), классификатор 304(1) определяет, что хост 108(1) должен принимать этот сеанс. Классификатор 304(1) обновляет таблицу 2206(1) РДА, которая служит локальной таблицей маршрутизации для блока 302(1) пересылки, а также вставляет атомарный элемент 2304, представляющий маршрут, в общую таблицу 2206 РДА. Это могут быть отдельные операции, единая операция, в которой сеансы уровня TCP/IP таблицы 2206 РДА размещаются на блоках 302 пересылки, и т.д. РДА 2202 внутренне дублирует этот маршрут в один или несколько членов группы классификаторов 304 в соответствии со своей оговоренной политикой избыточности.

Позиция (4) обозначает, что блок 302(1) пересылки напрямую пересылает последующие пакеты для этого соединения на хост 108(1), не взаимодействуя с классификатором 304(1). РДА 2202 можно использовать для маскировки, по меньшей мере, отчасти, сбоя блока 302 пересылки, классификатора 304 или пары 302/304 блок пересылки/классификатор. РДА 2202 также можно использовать, по меньшей мере, частично, для сохранения способности клиента к соединению, если коммутаторы 202(LBA), знающие о выравнивании нагрузки, по случайности, начинают передавать пакеты для установленного соединения на другой блок 302 пересылки.

На фиг.28 показана последовательность действий по маршрутизации трафика при наличии сбоя(ев). В противоположность иллюстративной последовательности действий по маршрутизации трафика в отсутствие сбоев, показанной на фиг.27, на фиг.29 показан сбой, произошедший в части инфраструктуры 106 выравнивания сетевой нагрузки (конкретно не указанной). В частности, первое устройство, на котором размещены и работают блок 302(1), пересылки и классификатор 304(1) переходит в нерабочее состояние после установления соединения, показанного на фиг.27. РДА 2202, по меньшей мере, частично маскирует этот сбой.

Позиция (1) обозначает, что коммутаторы 202(LBA), знающие о выравнивании нагрузки, обнаруживают сбой блока 302(1) пересылки и начинают передавать пакеты для соединения с каким-либо другим блоком 302 пересылки в группе. В этом примере другой блок 302 пересылки представляет собой блок 302(2) пересылки. Хотя на фиг.28 показан случай сбоя, коммутаторы 202(LBA), знающие о выравнивании нагрузки, могут также направлять этот трафик на блок 302(2) пересылки, даже если блок 302(1) пересылки все еще действует. Эта смена блоков 302 пересылки, инициированная в отсутствие сбоя, происходит, например, потому, что коммутаторы 202(LBA), знающие о выравнивании нагрузки, «забывают» о сродстве этого трафика с блоком 302(1) пересылки. Действия, обозначенные позициями (2)-(5), применяются как в случаях сбоя, так и в случаях «забытого сродства».

Позиция (2) обозначает, что блок 302(2) пересылки сверяется со своей таблицей маршрутизации, таблицей 2206(2) РДА. Не найдя маршрут для этого пакета, он пересылает этот пакет на свой классификатор 304(2). Позиция (3) обозначает, что классификатор распознает, что этот пакет является пакетом «середины сеанса», и классификатор 304(2) запрашивает у РДА 2202 маршрут для этого пакета. РДА 2202 отвечает посредством маршрута для соединения из связанного с ним атомарного элемента 2304.

Позиция (4) обозначает, что классификатор 304(2) прокладывает маршрут в блоке 302(2) пересылки. Иллюстративный протокол прокладки маршрутов описан ниже. Позиция (5) обозначает, что последующие пакеты для этого соединения, направленные на блок 302(2) пересылки, маршрутизируются непосредственно на надлежащий хост, который в данном примере является хостом 108(1) без сверки с классификатором 304(2).

В целом, протокол прокладки маршрутов для связи между классификаторами 304 и блоками 302 пересылки включает в себя команды добавления и удаления маршрутов. В частности, классификатор 304 отправляет на блок 302 пересылки команду добавления маршрута для данного соединения. Например, классификатор 304(2) может подать на блок 302(2) пересылки команду добавления маршрута, указанную на фиг.28 позицией (4). Маршрут (например, ключ и соответствующее значение) добавляется в локальную таблицу 2206(2) РДА для быстрого доступа к блоку 302(2) пересылки в будущем. В этом примере, классификатор 304(2) является устройством, отдельным от блока 302(2) пересылки, вследствие чего протокол прокладки маршрутов может представлять собой протокол между устройствами. Однако, протокол прокладки маршрутов можно также использовать для связи внутри устройства.

В описанной реализации классификатор 304(2) содержит реестр 2802 соединений. С помощью реестра 2802 соединений классификатор 304(2) отслеживает соединения любых блоков 302 пересылки (например, блока 302(2) пересылки), для которых классификатор 304(2) прокладывает маршруты. Чтобы классификатор 304(2) мог

отслеживать соединения, включая их прекращения, блок 302(2) пересылки пересылает окончательные пакеты для соединений (например, пакет FIN TCP) на классификатор 304(2). Затем классификатор 304(2) удаляет из реестра 2802 соединений элемент, который соответствует соединению и отправляет на блок 302(2) пересылки команду удаления маршрута. Получив команду удаления маршрута, блок 302(2) пересылки удаляет соответствующий маршрут из таблицы 2206(2) РДА. Таким образом, функция классификации совместно с функцией отслеживания сеансов может управлять таблицами маршрутизации и их маршрутами, которые используются функцией пересылки. Поэтому функция пересылки, которая разделена на другое устройство, может осуществляться с использованием высокоскоростного, но сравнительно простого оборудования.

На фиг.29 показаны дополнительные процедуры преодоления сбоя для повышения надежности инфраструктуры 106 выравнивания сетевой нагрузки. Опишем процедуры преодоления сбоя для двух разных сбоев, сбоя 2902 и сбоя 2906. Показано, что инфраструктура 106 выравнивания сетевой нагрузки (отдельно не указана) содержит пять компонентов: блок 302(1) пересылки, блок 302(2) пересылки, блок 302(3) пересылки, классификатор 304(1) и классификатор 304(2).

В описываемой реализации каждый из этих пяти компонентов 302(1), 302(2), 302(3), 304(1) и 304(2) соответствует отдельному устройству. Однако подобные или аналогичные процедуры преодоления сбоя можно применять к средам, в которых другие компоненты выравнивания нагрузки совместно используют устройства.

Первоначально согласно позиции [1] маршрутизатор(ы)/коммутатор(ы) 202 направляют входящий пакет, который, как оказалось, относится к новому соединению, на блок 302(1) пересылки. Поскольку блок 302(1) пересылки не имеет маршрут для этого соединения в своей локальной таблице маршрутизации, он отправляет пакет на классификатор 304(1), что указано пунктирной двойной стрелкой, обозначенной (1). Классификатор 304(1) сначала проверяет информацию сеанса со ссылкой на блок 308 отслеживания сеансов на предмет возможного сродства к сеансу более высокого уровня. В этом примере пакет не имеет сродства к существующему сеансу, поэтому классификатор 304(1) выбирает хост 108 со ссылкой на информацию работоспособности и нагрузки со ссылкой на обработчик 314 работоспособности и нагрузки.

В частности, в данном примере классификатор 304(1) выбирает хост 108(1). Предполагая, что пакет относится к сеансу TCP/IP, классификатор 304(1) добавляет этот сеанс TCP/IP, привязанный к хосту 108(1), к РДА 2202 с использованием функционального вызова «добавить атом» 2204(A). Классификатор 304(1) или блок 302(1) пересылки пересылает начальный пакет на хост 108(1). Классификатор 304(1) также прокладывает маршрут в локальной таблице маршрутизации блока 302(1) пересылки. Последующие пакеты блок 302(1) пересылки пересылает уже без взаимодействия с классификатором 304(1).

В некоторый момент в ходе соединения [1] на блоке 302(1) пересылки происходит сбой. Этот сбой 2902 обнаруживается с помощью маршрутизатора(ов)/коммутатора(ов) 202(LBA), знающих о выравнивании нагрузки. В результате в точке 2904 маршрутизатор(ы)/коммутатор(ы) 202 направляют дальнейшие пакеты, которые должны были быть направлены на блок 302(1) пересылки по соединению [1], на другой блок 302 пересылки, в данном примере блок 302(2) пересылки.

Таким образом, блок 302(2) пересылки получает дальнейшие пакеты по

соединению [2]. Поскольку блок 302(2) пересылки не имеет элемент в своей локальной таблице маршрутизации для пакетов, которые ранее были направлены на блок 302(1) пересылки, блок 302(2) пересылки направляет первый полученный пакет соединения [2] на классификатор, которому он присвоен/с которым он связан. В этом примере блок 302(2) пересылки присвоен классификатору 304(2), что указано пунктирной двойной стрелкой (2).

Классификатор 304(2) использует функциональный вызов «запросить атом» 2204(Q) для получения атомарного элемента 2304 (явно не показан) из РДА 2202, который связан с существующим соединением TCP/IP. Этот атомарный элемент 2304 обеспечивается через РДА 2202 блока 308 отслеживания сеансов посредством функционального вызова «возвратить атом» 2204(R). Классификатор 304(2) извлекает хост 108(1), который имеет сродство к этому соединению TCP/IP, из возвращенного атомарного элемента 2304. Классификатор 304(2) пересылает этот первый полученный пакет для соединения [2] на хост 108(1) и также прокладывает маршрут в локальной таблице маршрутизации блока 302(2) пересылки. Последующие пакеты блок 302(2) пересылки пересылает уже без взаимодействия с классификатором 304(2).

Вышеприведенные описания сосредоточены, главным образом, на сбоях отдельных компонентов блока 302 пересылки. Однако компоненты классификатора 304 также могут испытать сбой. Например, в некоторый момент происходит сбой 2906 на классификаторе 304(2). Блок 302(2) пересылки обнаруживает сбой 2906 при попытке потребить услуги классификации или посредством извещения об отсутствии некоторого указания работоспособности, например, указателя типа «пульс». Для обработки сбоя 2906 блок 302(2) пересылки переназначается другому классификатору 304 или повторно связывается с ним, который в данном примере представляет собой классификатор 304(1). Классификатор 304(1) предоставляет блоку 302(2) пересылки дополнительные функции классификации, что указано пунктирной двойной стрелкой (3).

На фиг.30 проиллюстрирована эксплуатационная реализация взаимодействия маршрутизации трафика с информацией работоспособности и нагрузки. Блок 302 пересылки и классификатор 304 взаимодействуют с обработчиком 314 работоспособности и нагрузки для маршрутизации пакетов на хосты 108(1), 108(2),..., 108(n). Хотя показаны блок 302 пересылки и классификатор 304, иллюстративная эксплуатационная реализация также применима к маршрутизатору 306 запросов (или, в целом, к функции 2012 маршрутизации трафика).

Показано, что хост 108(1) содержит конечные точки IP1, IP3, и IP4 приложения для приложения №1, приложения №1 и приложения №2, соответственно. Хост 108(2) содержит конечные точки IP2 и IP6 приложения для приложения №1 и приложения №2, соответственно. Хост 108(n) содержит конечную точку №5 приложения для приложения №2. Обработчик 314 работоспособности и нагрузки отслеживает эти хосты 108(1), 108(2),..., 108(n) и конечные точки IP1, IP2, IP3, IP4, IP5 и IP6 приложения (например, с использованием инфраструктуры 1202 работоспособности и нагрузки, объединенного кэша 1208 работоспособности и нагрузки и т.д.).

В описываемой реализации (1) обозначает, что классификатор 304 запрашивает одно или несколько распределений конечных точек приложения (например, посредством, по меньшей мере, одного запроса 1802 распределения конечных точек приложения) в среде, где используется маркерная схема 1806 распределения. Обработчик 314 работоспособности и нагрузки в этом примере отвечает, предоставляя маркерные распределения 3002 (например, посредством, по меньшей

мере, одного ответа 1804 по распределению конечных точек приложения).

В частности, маркерное распределение для приложения №1 3002(1) и маркерное распределение для приложения №2 3002(2) доступны классификатору 304. Маркерное распределение для приложения №1 3002(1) первоначально предоставляет 40 маркеров для IP1, 35 маркеров для IP2 и 25 маркеров для IP3. Маркерное распределение для приложения №2 3002(2) предоставляет 10 маркеров для IP4, 72 маркера для IP5 и 18 маркеров для IP6. Для каждого нового соединения, которому классификатор 304 выделил маршрутизацию на конечную точку приложения, классификатор 304 потребляет маркер.

Позиция (2) обозначает, что блок 302 пересылки получает начальный входящий пакет для нового соединения. Поскольку никакой маршрутизации для этого нового соединения в локальной части таблицы 2206 РДА блока 302 пересылки нет, то блок 302 пересылки пересылает начальный пакет на классификатор 304, что обозначено позицией (3).

Позиция (4) обозначает, что классификатор 304 (например, определив, что начальный пакет не содержит ссылку на сеанс для сеанса более высокого уровня) выбирает конечную точку приложения (и, таким образом, хост 108) в соответствии с информацией работоспособности и нагрузки. В частности, для нового соединения, которое должно обслуживаться приложением №1, классификатор 304 может выбрать любой из IP1, IP2 и IP3, если маркер для соответствующей конечной точки все еще существует.

Классификатор 304 может потреблять маркеры любым из многих возможных способов. Например, классификатор 304 может использовать круговой подход вне зависимости от количества маркеров на конечную точку. Альтернативно, согласно линейному подходу классификатор 304 может просто начать с IP1 и продвигаться через IP3, потребляя все маркеры для каждой конечной точки прежде, чем перейти к следующей конечной точке. Кроме того, классификатор 304 может потреблять, в любой момент времени, маркер из группы маркеров, зависящей от конечной точки, которая в данный момент имеет наибольшее количество маркеров. Согласно последнему подходу классификатор 304 выбирает IP1. Также можно использовать и другие подходы.

Показано, что классификатор 304 потребляет маркер для конечной точки IP2 приложения. Следовательно, при потреблении маркера группа маркеров для IP2 уменьшается с 35 маркеров до 34 маркеров. Кроме того, начальный пакет для нового соединения должен маршрутизироваться на конечную точку IP2 приложения.

Позиция (5А) обозначает пересылку начального пакета с классификатора 304 на конечную точку IP2 приложения для хоста 108(2). До, во время или после этой пересылки классификатор 304, что обозначено (5 В), прокладывает маршрут для этого соединения в локальной части таблицы 2206 РДА. Классификатор 304 может также добавить атомарный элемент 304 для этого сеанса в таблицу 2206 РДА в целях распределения и дублирования. Позиция (6) обозначает пересылку дальнейших пакетов для этого соединения/сеанса с блока 302 пересылки на конечную точку IP2 приложения для хоста 108(2) с использованием локальной таблицы маршрутизации блока 302 пересылки, реализованной как локальная часть таблицы 2206 РДА на фиг.30.

На фиг.31 показаны механизмы обеспечения высокой надежности инфраструктуры 106 выравнивания сетевой нагрузки. В частности, показаны иллюстративное обнаружение 3104 сбоя, иллюстративная обработка 3106 сбоя и

иллюстративное исправление 3108 сбоя. Эти иллюстративные механизмы обеспечения высокой надежности описаны применительно к разным компонентам инфраструктуры 106 выравнивания сетевой нагрузки. Компоненты инфраструктуры 106 выравнивания сетевой нагрузки включают в себя блок 302 пересылки, классификатор 304, маршрутизатор 306 запросов, блок 308 отслеживания сеансов и обработчик 314 работоспособности и нагрузки.

Позиция 3102(A) обозначает локальный сбой блока 302 пересылки. Позиция 3104(A) обозначает, что сбой обнаружен, по меньшей мере, одним коммутатором, знающим о выравнивании нагрузки. Для обработки локального сбоя 3102(A) коммутатор, знающий о выравнивании нагрузки, перенаправляет пакеты на другой(ие) блок(и) пересылки, что обозначено как 3106(A). Для исправления сбоя блока 302 пересылки маршруты, локально хранившиеся на блоке 302 пересылки, повторно строятся, что обозначено как 3108(A) на блоке(ах) пересылки, на которые перенаправлены пакеты, с использованием распределенного диспетчера отслеживания сеансов и его таблицы, в частности, РДА и его таблицы. Таким образом, распределенный диспетчер отслеживания сеансов может обеспечивать избыточные данные на одном или нескольких уровнях.

Позиция 3102(B) обозначает локальный сбой классификатора 304. Позиция 3104(B) обозначает, что сбой обнаружен, по меньшей мере, одним блоком пересылки. Для обработки локального сбоя 3102(B) блок пересылки, обнаруживший сбой, перенаправляет пакеты на другой(ие) классификатор(ы), что обозначено как 3106(B). Для исправления сбоя классификатора 304 информация сеанса, локально хранившаяся на классификаторе 304, повторно строится, что обозначено как 3108(B), на классификаторе(ах), на которые перенаправлены пакеты, с использованием РДА. Эта информация сеанса может представлять собой, например, информацию сеанса более высокого уровня, чем основные соединения TCP/IP. Кроме того, эту информацию сеанса можно рассматривать как часть инфраструктуры отслеживания сеансов, размещенную на том же устройстве, что и классификатор 304.

Позиция 3102(C) обозначает локальный сбой маршрутизатора 306 запросов. Позиция 3104(C) обозначает, что сбой обнаружен, по меньшей мере, одним блоком пересылки и/или коммутатором, знающими о выравнивании нагрузки. Для обработки локального сбоя 3102(C) блок пересылки и/или коммутатор, знающий о выравнивании нагрузки, перенаправляет пакеты на другой(ие) маршрутизатор(ы) запросов, что обозначено как 3106(C). Отдельные текущие логические запросы, на которых работает маршрутизатор 306 запросов после возникновения локального сбоя 3102(C), могут быть потеряны, если дублировать каждый такой отдельный логический запрос, пока запрос обрабатывается. Для исправления сбоя маршрутизатора 306 запроса информация сеанса и/или маршруты, локально хранившиеся на маршрутизаторе 306 запросов, повторно строятся, что обозначено как 3108(C) на маршрутизаторе(ах) запросов, на которые перенаправляются пакеты (и, таким образом, новые логические запросы). Повторное построение информации сеанса может осуществляться с использованием РДА. Опять же такую информацию сеанса можно рассматривать как часть инфраструктуры отслеживания сеансов, размещенную на том же устройстве, что и маршрутизатор 306 запросов.

Позиция 3102(D) обозначает локальный сбой блока 308 отслеживания сеанса. Позиция 3104(D) обозначает, что сбой обнаружен, по меньшей мере, одним блоком пересылки и/или классификатором. Например, если блок 308 отслеживания сеансов размещен на том же устройстве, что и классификатор, то сбой может обнаружить

блок пересылки или другой классификатор. Если блок 308 отслеживания сеансов размещен на отдельном устройстве, то сбой может обнаружить классификатор. Для обработки локального сбоя 3102(D) для информации отслеженного сеанса устанавливаются избыточность данных одного или нескольких уровней и распределение по множественным устройствам, что обозначено как 3106(D). Для исправления сбоя блока 308 отслеживания сеансов информация сеанса из таблиц РДА может перераспределяться и повторно дублироваться на, по меньшей мере, двух устройствах (если она уже так не распределена и не дублирована в достаточной степени), что обозначено как 3108(D), для обработки второго уровня сбоя.

Позиция 3102(E) обозначает локальный сбой обработчика 316 работоспособности и нагрузки. Позиция 3104(E) обозначает, что сбой обнаружен, по меньшей мере, одним классификатором и/или маршрутизатором запросов. Например, сбой может обнаружить компонент, принимающий информацию работоспособности и нагрузки от обработчика 314 работоспособности и нагрузки, если обработчик 314 работоспособности и нагрузки перестает отвечать, особенно, если обработчик 314 работоспособности и нагрузки размещен не на том устройстве, где размещен запрашивающий компонент. Для обработки локального сбоя 3102(E) для информации работоспособности и нагрузки используются избыточность данных работоспособности и нагрузки и внутренняя обработка сбоя, что обозначено как 3106(E).

Например, каждый обработчик 314 работоспособности и нагрузки может содержать объединенный кэш 1208 работоспособности и нагрузки, который дублирует информацию в таблицах 1204 работоспособности и нагрузки на множественных хостах 108. Кроме того, потребители информации 1206 работоспособности и нагрузки данного обработчика 314 работоспособности и нагрузки могут располагаться на том же устройстве, что и обработчик 314 работоспособности и нагрузки, вследствие чего этот сбой будет внутренне допустим. Аналогично, официальная версия соответствующей части информации 1206 работоспособности и нагрузки находится на соответствующем хосте 108, из-за чего сбой хоста 108 делает потерю соответствующей части информации работоспособности и нагрузки допустимой.

Для исправления сбоя обработчика 314 работоспособности и нагрузки данный компонент выравнивания сетевой нагрузки, который потребляет информацию работоспособности и нагрузки, может запросить другой обработчик работоспособности и нагрузки, потому что каждый такой обработчик 314 работоспособности и нагрузки содержит объединенный кэш информации обработчика работоспособности и нагрузки. Кроме того, когда обработчик 314 работоспособности и нагрузки вновь становится доступным, можно использовать протокол 1500 обмена сообщениями, что обозначено как 3108(E), для повторного построения его объединенного кэша информации работоспособности и нагрузки. Использование этих иллюстративных механизмов обеспечения высокой надежности можно обнаруживать, обрабатывать и исправлять сбойные компоненты инфраструктуры 106 выравнивания сетевой нагрузки, чтобы маскировать такие сбои для клиентов 102.

Перенос соединений с необязательным туннелированием и/или выравниванием нагрузки на уровне приложений

В этом разделе описано, как манипуляции соединениями, например, перенос соединений, можно использовать для выравнивания сетевой нагрузки. В этом разделе ссылки идут, главным образом, на фиг.32-39 и описана функция переноса соединений,

например, обеспечиваемая блоком 310 переноса соединений (фиг.3). Согласно описанному выше со ссылкой на фиг.3 и 4, каждое входящее соединение на инфраструктуре 106 выравнивания нагрузки может заканчиваться на ней. После этого соединение может быть перенесено на хост 108, так что соединение затем оканчивается на хосте 108. Блок 310 переноса соединений способен осуществлять перенос соединений и может размещаться частично на хостах 108 для осуществления переноса. Такой перенос соединений может осуществляться совместно с выравниванием нагрузки на уровне приложений классификатором 304 и/или с использованием туннелирования через блок 312 туннелирования.

На фиг.32 показан иллюстративный подход к выравниванию сетевой нагрузки на уровне приложений с переносом соединений. Выравнивание нагрузки на уровне приложений или на уровне 7 связано с принятием решений в отношении приложения по обработке соединения. Для осуществления выравнивания нагрузки на уровне приложений инфраструктура 106 выравнивания нагрузки обычно учитывает часть данных соединения. Пока не используется маршрутизация запросов, классификатор 304 обычно считывает начальную часть соединения, а затем переносит соединение, совместно с блоком 310 переноса соединений, на выбранный хост 318.

Для выравнивания нагрузки на уровне приложений в среде на основе TCP, в целом, классификаторы 304 считывают начальную часть TCP-данных клиента при принятии решения, куда переслать TCP-соединение клиента. Таким образом, логика уровня приложений проверяет данные клиента и принимает решения по выравниванию нагрузки на основании этих данных. Например, если соединение является (незашифрованным) соединением HTTP, то классификатор 304 может считывать HTTP-заголовок первого HTTP-запроса в соединении и может принимать решения на основании некоторой части содержимого заголовка (например, URL, куки и т.д.). Хотя выравнивание нагрузки на уровне приложений, перенос соединений и туннелирование применимы к другим протоколам, в приведенных здесь примерах используется, в основном, TCP/IP.

Показано, что инфраструктура 106 выравнивания нагрузки (конкретно не указана) содержит блок 302 пересылки, классификатор 304, блок 312 туннелирования и блок 310 переноса соединений (и, возможно, например, маршрутизаторы/коммутаторы 202(LBA), знающие о выравнивании нагрузки). Блок 302 пересылки соответствует виртуальному IP-адресу и пересылает пакеты на хосты 108, выбранные классификатором 304. Хотя это, для ясности, и не показано конкретно на фиг.32, хосты 108 также содержат функцию 310 переноса 310 соединений и функцию туннелирования 312.

В описываемой реализации блок 302 пересылки, классификатор 304 и блок 310 переноса соединений (на классификаторе 304 и на хостах 108), совместно с программным обеспечением протокола TCP на классификаторе 304 и хостах 108, действуют совместно для обеспечения переноса соединений. Перенос соединений, показанный на фиг.32, относится к соединению от клиента 102(1), которое обычно оканчивается на классификаторе 304. После переноса соединения соединение от клиента 102(1) оканчивается на хосте 108(1). Когда соединение оканчивается на хосте 108(1), пакеты для соединения могут туннелировать с использованием блока 312 туннелирования (на блоке 302 пересылки и хосте 108(1)).

Позиция (1) обозначает, что клиент 102(1) отправляет пакет SYN на блок 302 пересылки, чтобы сигнализировать о начале нового TCP-соединения. Позиция (2) обозначает, что блок 302 пересылки пересылает этот пакет на классификатор 304.

Позиция (3) обозначает, что классификатор 304 принимает TCP-соединение со стороны хоста 108 (чей идентификатор еще не известен, фактический хост 108() назначения еще предстоит выбрать). Применительно к протоколу TCP классификатор 304 отправляет пакет SYN-ACK клиенту 102(1).

Позиция (4) обозначает, что клиент 102(1) начинает передавать данные. (Начальный пакет SYN также может содержать данные.) Данные обрабатываются классификатором 304, который может сверяться со специализированной логикой. Специализированная логика может действовать в зависимости от того, какой хост 108 способен обрабатывать или лучше всех обрабатывать какие типы запросов или соединений. Поэтому классификатор 304 использует данные, а также информацию работоспособности и нагрузки приложения из обработчика 314 работоспособности и нагрузки, чтобы определить хост 108, который лучше или лучше всех подходит для обработки этого соединения от клиента 102(1). В данном примере выбран хост 108(1).

Позиция (5) обозначает, что классификатор 304 отправляет «большой двоичный объект» (блоб), представляющий состояние TCP-соединения хосту 108(1). Это состояние соединения агрегируется блоком 310 переноса соединений во взаимодействии со стеком TCP на классификаторе 304. Двоичный блоб содержит данные от клиента 102(1), квитированные классификатором 304, и параметры TCP, например, упорядоченная четверка TCP/IP, начальные порядковые номера и т.д.

Позиция (6) обозначает, что компонент блока 310 переноса соединений на хосте 108(1) (не показанный явно на фиг.32) «вставляет» это соединение в стек TCP на хосте 108(1). Эта вставка состояния соединения осуществляется во взаимодействии со стеком TCP на хосте 108(1), в результате чего приложениям 316 на хосте 108(1) кажется, что это соединение изначально принято самим хостом 108(1). Клиенту 102(1) не известно о переносе соединения.

Позиция (7) обозначает, что классификатор 304, во взаимодействии со стеком TCP на классификаторе 304, «молча» очищает внутреннее состояние, поддерживаемое для этого соединения. Классификатор 304 также добавляет маршрут в локальную таблицу маршрутизации блока 302 пересылки, который указывает, что хост 108(1) является пунктом назначения для пакетов этого соединения.

Позиция (8) обозначает, что последующие пакеты для соединения маршрутизируются блоком 302 пересылки на хост 108(1). Эти пакеты могут обрабатываться тем же блоком 302 пересылки, что и пакеты для соединений, которые классифицируются и маршрутизируются без использования переноса соединений. Эти последующие пакеты можно, в необязательном порядке, туннелировать с блока 302 пересылки на хост 108(1) с использованием блока 312 туннелирования. Блок 312 туннелирования также обозначен (пунктирными линиями) на блоке 310 переноса соединений на классификаторе 304, поскольку определенный(е) параметр(ы), используемые блоком 312 туннелирования, могут быть определены при переносе соединения и/или связаны с переносимым соединением. Иллюстративные реализации блока 312 туннелирования описаны ниже, в частности, со ссылкой на фиг.38 и 39.

На фиг.33 изображена логическая блок-схема 3300 способа переноса соединения с первого устройства на второе устройство. Логическая блок-схема 3300 содержит семь блоков 3302-3314. Хотя фиг.32 и 34-37 посвящены, главным образом, переносу соединения в среде выравнивания сетевой нагрузки, описанный здесь перенос соединения можно осуществлять между двумя устройствами общего вида, каждое из которых имеет функцию переноса соединения, например, как у блока 310 переноса соединений.

На блоке 3302 первое устройство принимает соединение. Например, первое устройство может заканчивать входящее соединение в соответствии с одним или несколькими протоколами части стека протоколов сетевого стека. На блоке 3304 происходит получение данных для соединения на первом устройстве. Например, эти данные можно принимать в начальном пакете, который запрашивает соединение, или в одном или нескольких пакетах, которые поступают после принятия соединения.

На блоке 3306 происходит агрегирование состояния соединения для принятого соединения из стека протоколов (в более общем случае, из сетевого стека) на первом устройстве. Например, состояние протокола для одного или нескольких протоколов из стека протоколов можно компилировать и агрегировать с помощью любых полученных данных, которые были квитированы. На блоке 3308 состояние соединения отправляется с первого устройства на второе устройство. Например, агрегированная информация первого состояния может быть отправлена с использованием надежного протокола на второе устройство.

На блоке 3310 состояние соединения для переносимого соединения поступает с первого устройства на второе устройство. На блоке 3312 состояние соединения вставляется в стек протоколов (в более общем случае, в сетевой стек) второго устройства. Например, соединение можно «регидратировать» с использованием протоколов стека протоколов второго устройства, чтобы программы, находящиеся выше уровня стека протоколов, не знали, что соединение является перемещенным соединением. В частности, состояние протокола можно внедрить в стек протоколов. Агрегированные данные состояния соединения также часто включаются на втором устройстве. На блоке 3314 соединение продолжается на втором устройстве. Например, соединение может продолжаться на втором устройстве, как если бы соединение не оканчивалось раньше в другом месте.

На фиг.34 иллюстративно показан подход к переносу соединений с точки зрения устройства-отправителя 3400. Перенос соединения на устройстве-отправителе 3400 осуществляется, по меньшей мере, блоком 310 переноса соединений. В описанной реализации устройство-отправитель 3400 - это устройство, которое является частью инфраструктуры 106 выравнивания сетевой нагрузки. Например, устройство-отправитель 3400 может содержать классификатор 304, возможно, совместно с блоком 302 пересылки, маршрутизатором 306 запросов и т.д.

Показано, что устройство-отправитель 3400 содержит в качестве части своего сетевого стека физический сетевой интерфейс (ФСИ) 3410, мини- порт 3408 ФСИ, протокольно-аппаратный интерфейс 3406, стек 3404 протоколов и уровень 3402 сокетов. Устройство-отправитель 3400 также содержит функцию 106 выравнивания нагрузки, например, классификатор 304 на уровне приложений и блок 310 переноса соединений. В частности, блок 310 переноса соединений содержит промежуточный драйвер 3411 переноса и «прокладку» 3412 блока переноса. Блок 310 переноса соединений способен выгружать соединение из устройства-отправителя 3400.

В описываемой реализации физический сетевой интерфейс 3410 может представлять собой сетевой адаптер (СА) (например, СА Ethernet), беспроводной интерфейс и т.д. Хотя показан только один физический сетевой интерфейс 3410, данное устройство может иметь множество таких физических сетевых интерфейсов 3410 (т.е. устройство-отправитель 3410 может быть подключено к нескольким линиям передачи данных). Каждый физический сетевой интерфейс 3410 обычно соответствует одному или нескольким физическим сетевым адресам.

Мини-порт 3408 ФСИ - это программный модуль, который понимает и

обеспечивает сопряжение с конкретной аппаратной реализацией физического сетевого интерфейса 3410. Протокольно-аппаратный интерфейс 3406 - это уровень, содержащий один или несколько соответствующих интерфейсов между двумя или более соответствующими протоколами и мини-портом 3408 ФСИ.

Стек 3404 протоколов включает в себя один или несколько соответствующих модулей, каждый из которых предназначен для одного или нескольких соответствующих протоколов. Примеры таких протоколов описаны ниже со ссылкой на фиг.36 и 37. В переходном контексте стек 3404 протоколов содержит состояние 3420 протокола для каждого соединения, существующего на устройстве-отправителе 3400. Уровень 3402 сокетов лежит между программой, например, функцией 106 выравнивания нагрузки, и стеком 3404 протоколов. Уровень 3402 сокетов обеспечивает API между функцией 106 выравнивания нагрузки и стеком 3404 протоколов и позволяет, помимо прочего, программам регистрироваться для соединений.

Промежуточный драйвер 3414 переноса или, более обще, драйвер 3414 переноса, размещается на уровне 3406 протокольно-аппаратного интерфейса. Прокладка 3412 блока переноса размещается прозрачно между стеком 3404 протоколов и уровнем 3402 сокетов.

Когда начальный пакет (не показан), запрашивающий новое соединение, представляется устройству-отправителю 3400, пакет направляется вверх от физического сетевого интерфейса 3410, на мини-порт 3408 ФСИ, через уровень 3406 протокольно-аппаратного интерфейса, и на стек 3404 протоколов. Когда пакет пересекает один или несколько протоколов стека 3404 протоколов, на нем создается состояние 3420 протокола. Кроме того, в результате этого начального пакета или вследствие того, что функция 106 выравнивания нагрузки принимает соединение, чтобы считать запрос, на устройство-отправитель 3400 поступают данные 3416. В ходе работы промежуточный драйвер 3414 переноса отводит копию данных 3416 на логику блока 310 переноса соединений. Когда функция 106 выравнивания нагрузки выдает функциональный вызов «перенести соединение», функциональный вызов переноса поступает на самый верхний уровень стека 3404 протоколов, чтобы могло начаться агрегирование 3418 состояния соединения. Состояние 3420 протокола компилируется из одного или нескольких протоколов стека 3404 протоколов. В реализации TCP/IP состояние 3420 протокола может включать в себя (i) TCP-порты и IP-адреса назначения и источника (например, упорядоченную четверку TCP/IP), (ii) состояние окна TCP, (iii) начальные порядковые номера, (iv) информацию приостановки, (v) ИД фрагмента IP, (vi) информацию маршрутизации и (vii) т.д.

Агрегирование 3418 состояния соединения также агрегирует данные 3416, отведенные на блок 310 переноса соединений и уже квитированные из устройства-отправителя 3400 (например, функцией 106 выравнивания нагрузки). Это агрегированное состояние 3418 соединения включает в себя состояние 3420 протокола и данные 3416 (и, в необязательном порядке, другую информацию, относящуюся к соединению). Затем агрегированное состояние 3418 соединения передается в виде двоичного блоба 3422 с устройства-отправителя 3400 на устройство-адресат с использованием надежного протокола. Этот двоичный блоб 3422 можно также пакетировать с идентификатором потока, если соединение подлежит в дальнейшем туннелированию посредством блока 312 туннелирования. Идентификаторы потока описаны ниже, в частности, со ссылкой на фиг.38 и 39.

На фиг.35 иллюстративно показан подход к переносу соединений с точки зрения

устройства-адресата 3500. Устройство-адресат 3500 подобно устройству-отправителю 3400 в отношении различных показанных уровней/модулей, включая блок 310 переноса соединений. Однако показано, что, по меньшей мере, одно приложение 316 на уровне приложений сопрягается с уровнем 3402 сокетов. Поэтому устройство-адресат 3500 может содержать хост 108. Кроме того, блок 310 переноса соединения способен загружать соединение от устройства-отправителя 3400.

В описываемой реализации приложение 316 является пунктом назначения пакета, инициирующего соединение, принятого на устройстве-отправителе 3400. Устройство-адресат 3500 принимает двоичный блоб 3422 от устройства-отправителя 3400. Двоичный блоб 3422 включает в себя состояние соединения, связанное с соединением, переносимым на устройство-адресат 3500 и, в необязательном порядке, идентификатор потока. Это состояние соединения включает в себя состояние 3420 протокола и квитированные данные 3416 (и, возможно, другую информацию, относящуюся к соединению).

В ходе работы, когда двоичный блоб 3422 достигает уровня 3406 протокольно-аппаратного интерфейса, промежуточный драйвер 3411 переноса распознает его как блоб для переноса соединения и отводит его. Состояние соединения вставляется в состояние 3502, чтобы создать видимость для приложения 316, что соединение изначально заканчивалось на устройстве-адресате 3500.

В частности, состояние 3420 протокола вставленного состояния 3502 соединения внедряется в стек 3404 протоколов. В описываемой реализации состояние 3420 протокола внедряется первым на протоколах высшего уровня, а затем на протоколах более низкого уровня стека 3404 протоколов. После внедрения состояния 3420 протокола в стек 3404 протоколов данные 3416 можно указывать приложению 316. Эти данные 3416 можно предоставлять приложению 316, как если бы они были частью вновь и локально оконченного соединения.

По завершении вставки 3502 состояния соединения соединение, инициированное пакетом, полученным на устройстве-отправителе 3400, успешно переносится оттуда на устройство-адресат 3500. Последующие пакеты для соединения можно пересылать непосредственно на устройство-адресат 3500 без прохождения через устройство-отправитель 3400 или, по меньшей мере, только с простой маршрутизацией и без применения к ним анализа уровня приложений. В необязательном порядке, эти пакеты могут туннелировать так, что промежуточный драйвер 3414 переноса эффективно работает как виртуальный CA на программной основе, который привязан к виртуальному IP-адресу.

На фиг.36 показан подход к процедуре 3600 выгрузки для переноса соединения. Процедура 3600 выгрузки переноса демонстрирует дополнительные иллюстративные подробности переноса соединения со стороны устройства-отправителя 3400. Показано, что общий стек 3404 протоколов содержит стек 3404(T) TCP, стек 3404(I) IP и стек 3404(A) протокола разрешения адресов (ARP). Однако, альтернативно, можно использовать другие стеки 3404() конкретных протоколов.

Для примера, уровень 3406 протокольно-аппаратного интерфейса может быть реализован как уровень на основе спецификации стандартного интерфейса сетевых адаптеров (NDIS) в среде операционной системы (OS) Microsoft® Windows®. Кроме того, уровень 3402 сокетов может быть реализован как уровень Winsock™ в среде (OS) Microsoft® Windows®.

В описываемой реализации промежуточный драйвер 3414 переноса включает в себя

протокольно-аппаратные интерфейсы 3406 на стыках со стеком 3404(A) ARP и с мини-портом 3408 ФСИ. Драйвер 3414 переноса служит в качестве цели выгрузки в процедуре 3600 выгрузки переноса. Цель выгрузки - это мини-порт протокольно-аппаратного интерфейса 3406, показанного в этом примере. В процедуре 3700 загрузки переноса (на фиг.37) промежуточный драйвер 3414 переноса служит в качестве отводчика загрузки.

В частности, промежуточный драйвер 3414 переноса привязан к каждому физическому сетевому интерфейсу 3410, через который можно переносить TCP-соединение. Промежуточный драйвер 3414 переноса обычно действует как транзитный драйвер, пропуская пакеты вверх или вниз по сетевому стеку и не взаимодействуя с пакетами иным образом. Однако промежуточный драйвер 3414 переноса не взаимодействует с пакетами, относящимися к переносу соединения (в необязательном порядке, включающими в себя впоследствии туннелированные пакеты).

Промежуточный драйвер 3414 переноса отвечает за следующее: (i) принятие запросов выгрузки переноса; (ii) агрегирование информации состояния протокола, относящейся к переносимому TCP-соединения, скомпилированной из стеков 3404() конкретных протоколов, совместно с квитированными данными, для получения информации состояния соединения; и (iii) передача агрегированного состояния соединения на устройство-адресат 3500 для процедуры 3700 загрузки переноса. Надежный протокол проводной связи для такой передачи может совместно использоваться с тем, который используется компонентами 2002 и 2010 отслеживания сеансов, для отправки и получения сообщений 2008 информации сеанса (например, описанных выше со ссылкой на фиг.20).

Другой задачей промежуточного драйвера 3414 переноса (например, в процедуре 3700 загрузки переноса) является инициирование загрузки переносимых соединений, которые он получает от других устройств, и буферизация любых входящих пакетов, относящихся к переносу соединения, когда оно находится в процессе загрузки. Чтобы загрузить соединение, промежуточный драйвер 3414 блока переноса посылает запрос загрузки на прокладку 3412 блока переноса. Прокладка 3412 блока переноса выдает вызов вставки в стек 3404 протоколов на стеке 3404(A) TCP, чтобы обрабатывать соединение в участке стека 3404 протоколов сетевого стека.

Прокладка 3412 блока переноса открывает интерфейс клиента стеку 3404(T) TCP и открывает интерфейс поставщика транспортного уровня уровню 3402 сокетов. Прокладка 3412 блока переноса играет две роли: (i) инициируют процедуру 3600 выгрузки переноса соединения на устройстве-отправителе 3400, а затем процедуру 3700 загрузки переноса на устройстве-адресате 3500 и (ii) посредничает в процессе классификации между прикладной программой 316 хоста, программой классификации 304 выравнивания нагрузки и уровнем 3402 сокетов. Прокладка 3412 блока пересылки и промежуточный драйвер 3414 блока пересылки описаны ниже со ссылкой на фиг.36 и 37.

Для процедуры 3600 выгрузки переноса перенос TCP-соединения осуществляется после того, как классификатор 304 классифицирует входящее TCP-соединение с использованием одного, двух или более его пакетов. Процедура 3600 выгрузки переноса описана посредством позиций<1>-<7>.

Позиция<1>обозначает инициализацию, которая осуществляется до операций классификации. Стек 3404 протоколов делает запросы на уровне 3406

протокольно-аппаратного интерфейса, чтобы определить, какие имеются возможности выгрузки, если вообще имеются. Промежуточный драйвер 3414 блока переноса указывает, что выгрузка переноса соединения доступна и распространяет запрос вниз на мини-порт 3408 ФСИ. Если возможность выгрузки TCP обеспечена физическим сетевым интерфейсом 3410, то мини-порт 3408 ФСИ также это указывает. Выгрузка TCP позволяет выгружать некоторую обработку TCP/IP на оборудование физического сетевого интерфейса 3410 и предусматривает некоторое компилирование состояния 3420 протокола. Поэтому некоторая логика компиляции и агрегирования может совместно использоваться двумя механизмами выгрузки.

Позиция<2>обозначает, что, после того как TCP-соединение классифицировано, классификатор 304 инициирует перенос TCP-соединения на выбранный хост 108. В частности, через уровень 3402 сокетов на прокладку 3412 блока переноса поступает команда переноса, указывающая устройство-адресат 3500.

Позиция<3>обозначает, что прокладка 3412 блока переноса инициирует перенос TCP-соединения для компиляции состояния протокола TCP. В частности, прокладка 3412 блока переноса вызывает API начальной выгрузки переноса TCP (или, в более общем случае, функциональный вызов «перенести соединение» или команду переноса соединения). Эта процедура компилирует соответствующее состояние для указанного TCP-соединения, которое используется для восстановления соединения на устройстве-адресате 3500. Скомпилированное состояние 3420 протокола включает в себя состояние из промежуточных уровней стека, в том числе стека 3404(T) TCP, стека 3404(I) IP и стека 3404(A) ARP.

Позиция<4>обозначает, что, после того как стек 3404 протоколов скомпилировал состояние 3420 протокола для переносимого TCP-соединения, он вызывает API начальной выгрузки переноса на мини-порте, к которому он привязан; в этом примере, этим мини-портом является промежуточный драйвер 3414 блока переноса. Однако на практике, между стеком 3404 протоколов и промежуточным драйвером 3414 блока переноса могут быть вставлены другие промежуточные драйверы, например, QoS IP. В этом случае эти промежуточные драйверы блока переноса могут участвовать в переносе, если относятся, компилируя/агрегируя свое состояние в информацию состояния соединения для переносимого соединения. Промежуточные драйверы продолжают распространять вызов «начать выгрузку переноса» вниз на сетевой стек, что, в конце концов, приводит к выполнению обработчика выгрузки переноса на промежуточном драйвере 3414 блока переноса. При этом промежуточный драйвер 3414 блока переноса также агрегирует любые квитированные данные с остальным состоянием соединения для переноса TCP-соединения на устройство-адресат 3500.

Позиция<5>обозначает, что после сохранения/копирования информации состояния соединения для переносимого TCP-соединения промежуточный драйвер 3414 блока переноса извещает сетевой стек о том, что перенос находится на своих окончательных стадиях, вызывая API завершения начальной выгрузки переноса. Этот API завершения начальной выгрузки переноса идет по обратному пути вверх от сетевого стека, через те же промежуточные драйверы, если имеются, и, наконец, к стеку 3404 протоколов. По мере того как каждый уровень обрабатывает этот вызов, информация состояния, связанная с переносимым соединением, может освобождаться. Пока обработка этого вызова не завершится, каждый уровень может посылать извещения об обновлении вниз на сетевой стек, чтобы обновить любую часть состояния соединения, которая изменилась с тех пор, как начался перенос.

Позиция<6>обозначает, что, когда процедура завершения начальной выгрузки переноса достигает стека 3404(T) TCP, TCP молча (т.е. отправляя команду перезагрузки на клиент 102), закрывает соединение, сбрасывая все состояние, связанное с переносимым соединением, и распространяет вызов «начальная выгрузка переноса завершена» на прокладку 3412 блока переноса. При этом сетевой стек освобождается от любой остаточной информации переносимого TCP-соединения.

Позиция<7>обозначает, что, когда вызов «начальная выгрузка переноса завершена» возвращается на промежуточный драйвер 3414 блока переноса (через часть прокладки 3412 блока переноса блока 310 переноса соединения), перенос TCP-соединения от устройства-отправителя 3400 на устройство-адресат 3500 может начинаться с переноса на него состояния соединения. Состояние соединения можно передавать асинхронно и надежно.

После начала переноса устройство-отправитель 3400 также должно гарантировать, что последующие данные от клиента 102 пересылаются на устройство-адресат 3500. Поэтому даже после успешного переноса соединения на адресат отправитель сохраняет некоторый объем состояния для соединения (например, элемент таблицы маршрутизации), чтобы правильно маршрутизировать последующие пакеты на адресат. Когда соединение заканчивается, адресат извещает отправителя, чтобы позволить ему очистить все оставшееся состояние для перенесенного соединения.

Кроме того, вследствие асинхронной природы переноса соединения пакеты данных для переноса соединения, которые пересылаются устройством-отправителем 3400 (или блоком пересылки, рассматриваемым как отдельное устройство), могут начинать поступать на устройство-адресат 3500 до того, как устройство-адресат 3500 получит состояние переносимого соединения. Промежуточный драйвер 3414 блока переноса на устройстве-адресате 3500 отвечает за буферизацию этих пакетов, пока соответствующее переносимое соединение не будет установлено на устройстве-адресате 3500.

На фиг.37 показан подход к процедуре 3700 загрузки для переноса соединения. Процедура 3700 загрузки переноса демонстрирует иллюстративные подробности для переноса соединения со стороны устройства-адресата 3500.

Когда переносимое соединение поступает на устройство-адресат 3500, оно опирается на промежуточный драйвер 3414 блока переноса для обработки. После расширения и объединения состояния переносимого соединения промежуточный драйвер 3414 блока переноса, в сочетании с прокладкой 3412 блока переноса, вставляет переносимое соединение в локальный сетевой стек, прозрачно для приложения 316. Для иллюстративной процедуры 3700 загрузки переноса описан перенос TCP-соединения в позициях<1>-<8>.

Позиция<1>, как описано выше со ссылкой на процедуру 3600 выгрузки переноса, обозначает инициализацию, осуществляемую до операций хостирования приложений. В частности, стек 3404 протоколов делает запросы относительно того, какие имеются возможности выгрузки, если вообще имеются. Промежуточный драйвер 3414 блока переноса заполняет запрос поддержки переноса TCP-соединения, чтобы указать, что загрузка переноса соединения доступна, а также распространяет запрос вниз на мини-порт 3408 ФСИ для возможных возможностей выгрузки TCP.

Позиция<2>обозначает, что, когда данные переноса соединения поступают на устройство-адресат 3500, информация переноса соединения (например, пакетированный двоичный блоб 3422) доставляется на промежуточный драйвер 3414 блока переноса. Промежуточный драйвер 3414 блока переноса повторно собирает

состояние соединения, согласует его с любыми соответствующими данными, поступившими в ходе переноса, и подготавливается к загрузке на сетевой стек. Любые данные от клиента 102, поступающие в процессе загрузки переносимого соединения, буферизуются промежуточным драйвером 3414 блока переноса. После успешного завершения переноса данные будут доставлены на приложение 316.

Согласно позиции<3>, чтобы инициировать загрузку переносимого соединения в локальный сетевой стек, промежуточный драйвер 3414 блока переноса извещает прокладку 3412 блока переноса о поступлении запроса переносимого соединения. Промежуточный драйвер 3414 блока переноса также доставляет состояние соединения (или, по меньшей мере, состояние 3420 протокола) на прокладку 3412 блока переноса.

Позиция<4>обозначает, что прокладка 3412 блока переноса инициирует загрузку переносимого соединения, вызывая процедуру начальной вставки TCP (или, в более общем случае, процедуру внедрения состояния протокола) и выдавая состояние 3420 переносимого протокола на стек 3404(T) TCP. Позиция<5>обозначает, что TCP/IP восстанавливает переносимое соединение посредством стека 3404 протоколов с использованием предоставленного состояния 3420 протокола. Это состояние 3420 протокола может включать в себя одно или несколько из состояния транспорта (TCP), состояния пути (IP), состояния соседа и следующего скачка (ARP) и т.д.

Позиция<6>обозначает, что, если переносимое соединение успешно повторно установлено на устройстве-адресате 3500, то TCP инициирует событие соединения с клиентской частью прокладки 3412 блока переноса, чтобы указать установление нового соединения. Имеется множество возможных причин для сбоя, но общие причины могут включать в себя недостаток соответствующего слушателя, сбой маршрутизации и т.д. В этих случаях, когда сетевой стек не способен повторно установить переносимое соединение, не указывается ни одно событие соединения, и в вызове «начальная вставка завершена» задается статус «сбой». Блок 310 переноса соединений отвечает за очистку переноса и отправку извещения о перезагрузке обратно на клиент 102, чтобы прекратить соединение.

Позиция<7>обозначает, что прокладка 3412 блока переноса действует, как поставщик, распространяющий событие соединения на уровень 3402 сокетов, чтобы указывать слушающему приложению 316 об установлении нового соединения. Если приложение 316 принимает соединение, оно обрабатывает запросы и ответы посредством нормальных операций чтения и записи сокета; приложению 316 может быть неизвестно, что соединение было перемещено. Если приложение 316 не приняло соединение, то TCP заканчивает соединение, но не посылает извещение о перезагрузке обратно на клиент 102. Опять же в вызове «начальная вставка завершена» устанавливается статус «сбой», и блок 310 переноса соединения отвечает за очистку переноса и отправку извещения о перезапуске обратно на клиент 102, чтобы прекратить соединение.

Особая ситуация возникает, когда приложение 316 и классификатор 304 совместно расположены на одном и том же устройстве: прокладка 3412 блока переноса может быть судьей между ними. Когда на одном и том же хосте 108 размещены оба класса программ, они оба могут слушать один/одни и тот/те же IP-адрес(а) и порт(ы). Однако TCP обычно имеет одного слушателя на уникальный IP-адрес и порт. Поэтому прокладка 3412 блока переноса может затемнять конфигурацию, когда две программы слушают на одном и том же IP-адресе и порте, мультиплексируя два сокета в единый слушатель на уровне TCP.

В этом случае, когда события соединения поступают на клиентскую часть прокладки 3412 блока переноса, прокладка 3412 блока переноса, в качестве поставщика, определяет, на какой слушающий сокет доставлять извещение о соединении на уровне 3402 сокетов. При наличии только одного сокета, слушающего соответствующий IP-адрес и порт, то этот сокет принимает событие соединения. При наличии более одного слушающего сокета получатель зависит от контекста, в котором указано событие соединения. Если событие соединения является фирменным новым соединением для виртуального IP-адреса, то событие соединения доставляется на классификатор 304; если событие соединения относится к выделенному IP-адресу (IP-адресу без выравнивания нагрузки) или является результатом загрузки переносимого соединения, то событие соединения доставляется на целевое приложение 316.

Позиция<8>обозначает, что по завершении переносимого соединения TCP извещает прокладку 3412 блока переноса, вызывая предоставленный обработчик завершения начальной вставки. Код статуса предусмотрен для извещения прокладки 3412 блока переноса, было ли соединение успешно обновлено. Если загрузка переносимого соединения не удается, то блок 310 переноса соединения отвечает за очистку переноса и за извещение клиента 102 о прекращении соединения путем отправки ему команды перезапуска. Если переносимое соединение было успешно вставлено в локальный сетевой стек, то промежуточный драйвер 3414 блока переноса может начать доставлять буферизованные данные с клиента 102, передавая полученный(е) пакет(ы) через путь приема пакетов протокольно-аппаратного интерфейса 3406.

Когда переносимое соединение оканчивается (по причине неудачной загрузки, из-за того, что перенесенное соединение впоследствии закрыто нормальными средствами, и т.д.), устройство-адресат 3500 извещает устройство-отправитель 3400. Устройство-отправитель 3400 использует эти извещения, чтобы более эффективно и надежно очищать неактивное состояние для переносимых соединений, включая элементы таблицы маршрутизации. Поэтому для учета успешно перенесенных соединений, которые произвольно заканчиваются в будущем, прокладка 3412 блока переноса может отслеживать их деятельность и извещать промежуточный драйвер 3414 блока переноса, когда сокеты из-за этого закрываются.

На фиг.38 иллюстративно показан подход к туннелированию пакетов между блоком 302 пересылки и хостом 108. Инкапсулированные пакеты 3808 могут туннелировать из блока 302 пересылки на хост 108 без привлечения служебной нагрузки для каждого переданного пакета. Согласно описанному ниже туннелирование осуществляется с использованием идентификатора 3814 потока и таблиц 3806 и 3810 отображения инкапсуляции блоков 312(F) и 312(H) туннелирования, соответственно, блока 302 пересылки и хоста 108, соответственно. Идентификатор 3814 потока вставляется в инкапсулированные пакеты 3808.

Как отмечено выше со ссылкой фиг.32, пакеты для соединения, которые поступают после переноса соединения, могут маршрутизироваться блоком 302 пересылки на хост 108(1) с использованием туннелирования с помощью блока 312 туннелирования. Согласно позиции (8) (фиг.32) блок 302 пересылки пересылает эти последующие пакеты с блока 302 пересылки, имеющего сетевой адрес «F», на хост 108(1), имеющий сетевой адрес «H1». Как описано выше со ссылкой на фиг.4, блок 302 пересылки может осуществлять TCA [трансляцию сетевых адресов], полу-TCA, туннелирование и т.д., чтобы маршрутизировать входящие пакеты на хост 108(1).

Такие входящие пакеты содержат IP-адрес назначения виртуального IP-адреса (VIP)

и IP-адрес источника «C1» для пакетов, поступающих от клиента 102(1). Пакеты, маршрутизируемые на хост 108(1), имеют IP-адрес назначения и адрес источника C1 (для полу-TCA) или «F» (для полной TCA). Это переписывание адресов может мешать некоторым протоколам, которые ожидают, что клиент 102(1) и хост 108(1) имеют идентичные виды адресов источника и назначения.

Поэтому, по меньшей мере, в отношении полной TCA, обратные пути от хоста 108(1) к клиенту 102(1), которые не проходят через блок 302 пересылки, запрещены, поскольку хост 108(1) не знает адрес клиента 102(1). Прямые пути от хоста 108(1) к клиенту 102(1) желательны в ситуациях, когда трафик от хоста 108(1) к клиенту 102(1) особенно высок и/или значительно превышают трафик в обратном направлении (например, когда хост 108(1) обеспечивает потоковую передачу мультимедийных данных на клиент 102(1)).

Туннелирование посредством блоков 312 туннелирования, согласно описанному здесь, может обеспечивать идентичные виды в отношении адресов (и портов) источника и назначения для клиентов 102 и приложений 316 на хостах 108. Для примера и со ссылками на фиг.34 и 35, блок 312 туннелирования на каждом из блока 302 пересылки и хоста 108 могут действовать как часть промежуточного драйвера 3414 переноса блока 310 переноса или совместно с ним.

В реализации, описанной со ссылкой на фиг.38, блок 310 переноса соединений обеспечивает отображение 3812 инкапсуляции между идентификатором 3814 потока и упорядоченной четверки 3804 TCP/IP. Блок 310 переноса соединений может быть связан с классификатором 304, и блок 310 переноса соединений (в необязательном порядке, совместно с таким классификатором 304) может располагаться на том же устройстве, что и блок 302 пересылки. Альтернативно, блок 310 переноса соединений (а также классификатор 304) может размещаться на другом устройстве, чем блок 302 пересылки. Отображение 3812 инкапсуляции может, альтернативно, обеспечиваться посредством функции блока 312 туннелирования или совместно с ней, т.е., например, размещаться на классификаторе 304 или быть связанным с ним.

Будучи отображен в упорядоченную четверку 3804 TCP/IP посредством отображения 3812 инкапсуляции, идентификатор 3814 потока служит для идентификации потока инкапсулированных пакетов 3808 для конкретного соединения. Упорядоченная четверка 3804 TCP/IP содержит сетевой адрес (и порты и пр.) для источника и назначения для конкретного соединения в соответствии с протоколом TCP/IP или любого подобного или аналогичного протокола. Идентификатор 3814 потока является 32-битовым в описываемой реализации, поскольку для соединений, установленных в соответствии с интернет-протоколом IPv4, доступно 32 бита. Однако, альтернативно, можно использовать идентификаторы 3814 потока другой длины, в особенности для других протоколов, таких, как IPv6, UDP и т.д. интернета и т.д.

Идентификаторы 3814 потока можно генерировать с использованием любого подходящего механизма, например, возрастающего счетчика соединений. Кроме того, упорядоченная четверка 3804 TCP/IP является, в более общем случае, парой источник/пункт назначения. Каждое значение источника и значение пункта назначения отдельной пары источник/пункт назначения может включать в себя идентификатор сетевого узла (например, сетевой адрес, порт, некоторые их комбинации и т.д.) для источника и пункта назначения, соответственно, данного пакета, распространяющегося по конкретному соединению.

Блок 310 переноса соединений предоставляет отображение 3812 инкапсуляции

хосту 108. Блок 312(H) туннелирования на хосте 108 сохраняет отображение 3812 инкапсуляции в таблице 3810 отображения инкапсуляции как элемент 3810(1) отображения инкапсуляции. Блок 312(H) туннелирования может затем использовать идентификатор 3814 потока для отображения и идентификации конкретного соединения, соответствующего упорядоченной четверке 3804 TCP/IP. Отображение 3812 инкапсуляции может, в необязательном порядке, сообщаться на хост 108 как часть пакетированного двоичного блоба 3422 в операции переноса соединения.

Блок 302 пересылки также содержит компонент блока 312(F) туннелирования с таблицей 3806 отображения инкапсуляции. В таблице 3806 отображения инкапсуляции хранится элемент 3806(1) отображения инкапсуляции, который связывает/отображает упорядоченную четверку 3804 TCP/IP для конкретного соединения в идентификатор 3814 потока. Блок 312(F) туннелирования также получает информацию отображения для элемента 3814 отображения инкапсуляции от блока 310 переноса соединений (например, отображение 3812 инкапсуляции).

Хотя показан только один элемент 3806(1) и 3810(1) отображения инкапсуляции, таблица 3806 отображения инкапсуляции и таблица 3810 отображения инкапсуляции могут иметь много таких элементов. Эти таблицы 3806 и 3810 отображения инкапсуляции можно комбинировать с другой информацией, например, таблицами для информации сеанса блока 308 отслеживания сеансов.

Когда устройство (например, блок 302 пересылки), передающее и устройство (например, хост 108), принимающее инкапсулированные пакеты 3808, только обеспечивают туннелирование между собой, их таблицы отображения инкапсуляции, вероятно, имеют одинаковые элементы отображения инкапсуляции. В противном случае таблица 3806 отображения инкапсуляции и таблица 3810 отображения инкапсуляции, вероятно, имеют разные полные наборы элементов 3806() отображения инкапсуляции и элементов 3810() отображения инкапсуляции соответственно.

В ходе работы входящий пакет 3802 для конкретного соединения поступает на блок 302 пересылки. Конкретное соединение связано с упорядоченной четверкой 3804 TCP/IP. Входящий пакет 3802 содержит упорядоченную четверку 3804 TCP/IP с IP-адресом источника (клиента 102), IP-адресом назначения (виртуальным IP), TCP-портом источника (клиента 102) и TCP-портом назначения.

Блок 312(F) туннелирования принимает входящий пакет 3802 для туннелирования на хост 108. Используя упорядоченную четверку 3804 TCP/IP, блок 312(F) туннелирования обращается к таблице 3806 отображения инкапсуляции, чтобы отыскать элемент 3806(1) отображения инкапсуляции. Идентификатор 3814 потока извлекается из элемента 3806(1) отображения инкапсуляции как связанный/отображенный на упорядоченную четверку 3804 TCP/IP.

Для создания инкапсулированного пакета 3808 блок 312(F) туннелирования вставляет идентификатор 3814 потока в части порта источника и назначения упорядоченной четверки TCP/IP. Для реализации интернет-протокола IPv4 эти две части TCP-порта обеспечивают суммарное пространство в 32 бита. Кроме того, для части IP-адреса источника заголовка упорядоченной четверки TCP/IP блок 312(F) туннелирования вставляет IP-адрес «F» блока 302 пересылки. Для части IP-адреса назначения заголовка упорядоченной четверки TCP/IP блок 312(F) туннелирования вставляет IP-адрес «H» хоста 108.

Блок 302 пересылки маршрутизирует/передает инкапсулированный пакет 3808 на хост 108, и хост 108 получает инкапсулированный пакет 3808 от блока 302 пересылки.

Компонент блока 312(H) пересылки на хосте 108 обнаруживает, что инкапсулированный пакет 3808 является туннелированным пакетом, который нужно декапсулировать.

Идентификатор 3814 потока извлекается из инкапсулированного пакета 3808 и используется для поиска соответствующей упорядоченной четверки 3804 TCP/IP, которая привязана к элементу 3810(1) отображения инкапсуляции таблицы 3810 отображения инкапсуляции. Упорядоченная четверка 3804 TCP/IP используется блоком 312(H) туннелирования для восстановления заголовка упорядоченной четверки 3804 TCP/IP как изначально принятого во входящем пакете 3802 на блоке 302 пересылки.

В частности, IP-адрес F блока 302 пересылки заменяется IP-адресом источника, а IP-адрес H хоста 108 заменяется IP-адресом назначения. Кроме того, идентификатор 3814 потока заменяется TCP-портом источника и TCP-портом назначения. Декапсулированный пакет указывается через сетевой стек хоста 108 целевому приложению 316.

В целом, часть заголовка пакета, включая часть пары источник/пункт назначения, для данного пакета, который не обязательно используется для передачи данного пакета, можно использовать для переноса идентификатора 3814 потока. Благодаря предварительному предоставлению, по меньшей мере, части пары источник/пункт назначения на хост 108 идентификатор 3814 потока можно использовать для туннелирования (например, инкапсулировать и/или декапсулировать) пакетов без привлечения служебной нагрузки инкапсуляции на каждом пакете. Кроме того, пакеты, которые имеют полный размер в отношении данного протокола, можно туннелировать без отбрасывания.

На фиг.39 показана логическая блок-схема 3900 способа туннелирования пакетов между первым устройством и вторым устройством. Например, первое устройство и второе устройство могут соответствовать устройству-отправителю 3400 и устройству-адресату 3500, соответственно, инфраструктуры 106 выравнивания нагрузки и группы хостов 108, соответственно. Тем не менее, туннелирование можно использовать в реализациях, не связанных с выравниванием нагрузки.

Логическая блок-схема 3900 содержит двенадцать блоков 3902-3924. Хотя действия логической блок-схемы 3900 могут осуществляться в других средах и посредством различных программных схем, фиг.1-3, 32, 34, 35 и 38 используются, в частности, для иллюстрации определенных аспектов и примеров способа.

На блоке 3902 устройство-отправитель отправляет на устройство-адресат отображение идентификатора потока в упорядоченную четверку TCP/IP. Например, устройство-отправитель 3400 может послать отображение 3812 инкапсуляции, которое привязывает идентификатор 3814 потока к упорядоченной четверке 3804 TCP/IP. На блоке 3914 устройство-адресат принимает отображение идентификатора потока в упорядоченную четверку TCP/IP от устройства-отправителя. Например, устройство-адресат 3500 получает от устройства-отправителя 3400 отображение 3812 инкапсуляции, которое привязывает идентификатор 3814 потока к упорядоченной четверке 3804 TCP/IP.

Альтернативно, устройство-адресат 3500 может получить отображение 3812 инкапсуляции от другого устройства. Пунктирные стрелки 3926 и 3928 указывают, что действия блоков 3904-3912 и блоков 3916-3924 могут происходить через некоторое время после действий блоков 3902 и 3914, соответственно.

На блоке 3904 устройство-отправитель получает от клиента входящий пакет.

Например, устройство-отправитель 3400 может получить от клиента 102 входящий пакет 3802, имеющий заголовок с упорядоченной четверкой 3804 TCP/IP. На блоке 3906 осуществляется поиск идентификатора потока для соединения, соответствующего пакету клиента с использованием упорядоченной четверки TCP/IP входящего пакета. Например, можно искать идентификатор потока для соединения с клиентом 102 с использованием упорядоченной четверки 3804 TCP/IP, которая отображается в него, в элементе 3806(1) отображения инкапсуляции таблицы 3806 отображения инкапсуляции.

На блоке 3908 происходит замена IP источника и IP назначения входящего пакета IP-адресом отправителя устройства-отправителя и целевым IP-адресом устройства-адресата, соответственно. Например, устройство-отправитель 3400 может заменить части IP-адреса части упорядоченной четверки 3804 TCP/IP заголовка входящего пакета 3802 IP-адресами устройства-отправителя 3400 и устройства-адресата 3500.

На блоке 3910 порт источника и порт назначения входящего пакета заменяются идентификатором потока. Например, устройство-отправитель 3400 может заменить TCP-порты источника и пункта назначения части упорядоченной четверки 3804 TCP/IP заголовка входящего пакета 3802 идентификатором 3814 потока. На блоке 3912 устройство-отправитель передает инкапсулированный пакет на устройство-адресат. Например, устройство-отправитель 3400 может отправить на устройство-адресат 3500 инкапсулированный пакет 3808.

На блоке 3916 устройство-адресат получает инкапсулированный пакет от устройства-отправителя. Например, устройство-адресат 3500 может получить от устройства-отправителя 3400 инкапсулированный пакет 3808. На блоке 3918 осуществляется поиск упорядоченной четверки TCP/IP для соединения, соответствующего пакету, полученному от клиента, с использованием идентификатора потока. Например, устройство-адресат 3500 может обратиться к таблице 3810 отображения инкапсуляции в элементе 3810(1) отображения инкапсуляции, который отображает идентификатор 3814 потока в упорядоченную четверку 3804 TCP/IP.

На блоке 3920 IP-адрес отправителя и IP-адрес получателя заменяются IP-адресом источника и IP-адресом назначения, соответственно, с использованием найденной упорядоченной четверки TCP/IP. Например, устройство-адресат 3500 может заменить IP-адреса устройства-отправителя 3400 и устройства-адресата 3500 в инкапсулированном пакете 3808 IP-адресом источника и IP-адресом назначения из упорядоченной четверки 3804 TCP/IP, полученной из таблицы 3810 отображения инкапсуляции.

На блоке 3922 идентификатор потока заменяется портом источника и портом назначения входящего пакета с использованием найденной упорядоченной четверки TCP/IP. Например, устройство-адресат 3500 может заменить идентификатор 3814 потока в инкапсулированном пакете 3808 TCP-портом источника и TCP-портом назначения из упорядоченной четверки 3804 TCP/IP. На блоке 3924 пакет клиента указывается приложению на устройстве-адресате. Например, декапсулированная версия инкапсулированного пакета 3808 или входящий пакет 3802 указывается приложению 316 устройства-адресата 3500.

Действия, аспекты, признаки, компоненты и т.д., указанные на фиг.1-39, представлены в виде схем, состоящих из ряда блоков. Однако порядок, взаимосвязи, схема размещения и т.д., применяемые для описания и представления фиг.1-39, не

следует рассматривать в смысле ограничения, и для любого количества блоков допустимы любые комбинирование, изменение расположения, добавление, исключение и т.д. для реализации одного или нескольких систем, способов, устройств, протоколов, носителей, API, аппаратов, конфигураций и т.д. для выравнивания сетевой нагрузки. Кроме того, хотя приведенное здесь описание включает в себя ссылки на конкретные реализации (и иллюстративную операционную среду, представленную на фиг.40), показанные и/или описанные реализации можно реализовать посредством любого подходящего аппаратного обеспечения, программного обеспечения, программно-аппаратного обеспечения или их комбинаций и с использованием любых пригодных организаций сети, транспортных/коммуникационных протоколов, программных интерфейсов приложений (API), архитектур клиент-сервер и т.п.

Операционная среда компьютера или другого устройства

На фиг.40 иллюстративно показана операционная среда 4000 компьютера (или устройства общего назначения), способная (полностью или частично) реализовать, по меньшей мере, одну систему, устройство, аппарат, компонент, конфигурацию, протокол, подход, способ, процедуру, носитель, API, некоторую их комбинацию и т.д. для описанного здесь выравнивания сетевой нагрузки. Операционную среду 4000 можно использовать в описанных ниже компьютерных и сетевых архитектурах или в условиях автономной работы.

Операционная среда 4000 является лишь одним примером среды и не призвана как-либо ограничивать объем использования или функциональные возможности применяемых архитектур устройства (включая компьютер, сетевой узел, развлекательное устройство, мобильный прибор, общее электронное устройство и т.д.). Кроме того, операционную среду 4000 (или ее устройства) не следует рассматривать как имеющую какую-либо зависимость или требование по отношению к любому или комбинации компонентов, показанных на фиг.40.

Кроме того, выравнивание сетевой нагрузки может быть реализовано посредством многочисленных других сред или конфигураций устройства общего назначения или устройства специального назначения (включая вычислительную систему). Примеры общеизвестных устройств, систем, сред и/или конфигураций, которые могут быть пригодны для использования, включают в себя, но не исключительно, персональные компьютеры, компьютеры-серверы, тонкие клиенты, толстые клиенты, карманные персональные компьютеры (КПК) или мобильные телефоны, часы, карманные или портативные устройства, многопроцессорные системы, системы на основе микропроцессора, телевизионные приставки, программируемую бытовую электронику, видеоигровые аппараты, игровые пульты, портативные или карманные игровые устройства, сетевые ПК, мини-компьютеры, универсальные компьютеры, сетевые узлы, распределенные или мультипроцессорные вычислительные среды, которые включают в себя любые из вышеперечисленных систем или устройств, некоторые их комбинации и т.п.

Примеры осуществления выравнивания сетевой нагрузки можно описать в общем контексте команд, выполняемых процессором. В целом, команды, выполняемые процессором, включают в себя процедуры, программы, протоколы, объекты, интерфейсы, компоненты, структуры данных и т.п., которые выполняют конкретные задания или реализуют определенные абстрактные типы данных. Выравнивание сетевой нагрузки, описанное здесь в некоторых реализациях, может также осуществляться на практике в средах распределенной обработки, где задания

выполняются удаленно-связанными обрабатывающими устройствами, соединенными посредством линии и/или сети связи. Особенно в распределенной вычислительной среде команды, выполняемые процессором, могут размещаться на раздельных носителях данных, выполняться разными процессорами и/или распространяться по средам передачи данных.

Операционная среда 4000 включает в себя вычислительное устройство общего назначения в виде компьютера 4002, который может содержать любое (например, электронное) устройство с возможностями вычисления/обработки. Компоненты компьютера 4002 могут включать в себя, но не исключительно, один или несколько процессоров 4004, системную память 4006 и системную шину 4008, которая подключает различные компоненты системы, в том числе процессор 4004, к системной памяти 4006.

Процессоры 4004 не ограничиваются материалами, из которых они сформированы, или применяемыми в них механизмами обработки. Например, процессоры 4004 могут состоять из полупроводников и/или транзисторов (например, электронные интегральные схемы (ИС)). В таком контексте, команды, выполняемые процессором, могут быть электронно-выполняемыми командами. Альтернативно, механизмы процессоров 4004, а, стало быть, и компьютера 4002, могут включать в себя, но без ограничения, квантовое вычисление, оптическое вычисление, механическое вычисление (например, с использованием нанотехнологии) и т.д.

Системная шина 4008 представляет один или несколько из многочисленных типов шинных структур, включая шину памяти или контроллер памяти, двухточечное соединение, коммутирующую ткань, периферийную шину, ускоренный графический порт и шину процессора или локальную шину, использующую разнообразные шинные архитектуры. Например, такие архитектуры могут включать в себя шину архитектуры промышленного стандарта (ISA), шину микроканальной архитектуры (MCA), шину расширенного стандарта ISA (EISA), локальную шину Ассоциации по стандартам в области видеоэлектроники (VESA) и шину подключений периферийных компонентов (PCI), также именуемую шиной расширения, некоторые их комбинации и т.д.

Компьютер 4002 обычно содержит разнообразные среды, доступные процессору. Такие среды могут представлять собой любые имеющиеся среды, к которым компьютер 4002 или другое (например, электронное) устройство может осуществлять доступ, и включают в себя энергозависимые и энергонезависимые носители, сменные и стационарные носители, носители данных и среды передачи данных.

Системная память 4006 содержит носители, доступные процессору, в виде энергозависимой памяти, например, оперативной памяти (ОЗУ) 4010, и/или энергонезависимой памяти, например, постоянной памяти (ПЗУ) 4012. Базовая система ввода/вывода (BIOS) 4014, содержащая основные процедуры, которые помогают переносить информацию между элементами компьютера 4002, например, при запуске, хранится в ПЗУ 4012. ОЗУ 4010 обычно содержит данные и/или программные модули/команды, которые непосредственно доступны процессору 4004 и/или в данный момент обрабатываются им.

Компьютер 4002 может также включать в себя другие сменные/стационарные и/или энергозависимые/энергонезависимые носители данных. В порядке примера, на фиг.40 показаны жесткий диск или массив дисковых приводов 4016 для считывания с (обычно) стационарного энергонезависимого магнитного носителя (отдельно не показан) и записи на него, привод 4018 магнитного диска для считывания с (обычно)

сменного энергонезависимого магнитного диска 4020 (например, «флоппи-диска») и записи на него и привод 4022 оптического диска для считывания с (обычно) сменного энергонезависимого оптического диска 4024, например, CD, DVD, или другого оптического носителя и записи на него. Жесткий диск 4016, привод 4018 магнитного диска и привод 4022 оптического диска подключен к системной шине 4008 посредством одного или нескольких интерфейсов 4026 носителей данных. Альтернативно, жесткий диск 4016, привод 4018 магнитного диска и привод 4022 оптического диска могут быть подключены к системной шине 4008 посредством одного или нескольких других отдельных или комбинированных интерфейсов (не показаны).

Приводы и соответствующие носители, доступные процессору, обеспечивают энергонезависимое хранение команд, выполняемых процессором, например, структур данных, программных модулей и других данных для компьютера 4002. Хотя в иллюстративном компьютере 4020 показаны жесткий диск 4016, сменный магнитный диск 4020 и сменный оптический диск 4024, очевидно, что для хранения команд, доступных устройству, можно также использовать другие типы носителей, доступных процессору, например, магнитные кассеты или другие магнитные устройства хранения данных, флэш-память, компакт-диски (CD), цифровые универсальные диски (DVD) или другие оптические носители данных, ОЗУ, ПЗУ, электрически стираемые программируемые постоянные запоминающие устройства (ЭСППЗУ) и т.п. Такие носители также могут включать в себя так называемые специализированные или «зашитые» ИС. Другими словами, для реализации носителей данных иллюстративной операционной среды 4000 можно использовать любые носители, доступные процессору.

На жестком диске 4016, магнитном диске 4020, оптическом диске 4024, ПЗУ 4012 и/или ОЗУ 4010 может храниться любое количество программных модулей (или других блоков или наборов команд/кодов), включая, например, операционную систему 4028, одну или несколько прикладных программ 4030, другие программные модули 4032 и программные данные 4034.

Пользователь может вводить команды и/или информацию в компьютер 4002 через устройства ввода, например, клавиатуру 4036 и указательное устройство 4038 (например, «мышь»). Другие устройства 4040 ввода (конкретно не показаны) могут включать в себя микрофон, джойстик, игровую панель, спутниковую антенну, последовательный порт, сканер и/или др. Эти и другие устройства ввода подключены к процессору 4004 через интерфейсы 4042 ввода/вывода, которые подключены к системной шине 4008. Однако устройства ввода и/или устройства вывода могут, альтернативно, быть подключены посредством другого интерфейса и шинных структур, например, параллельного порта, игрового порта, универсальной последовательной шины (USB), инфракрасного порта, интерфейса IEEE 1394 ("FireWire"), беспроводного интерфейса IEEE 802.11, беспроводного интерфейса Bluetooth® и т.п.

Монитор/экран 4044 визуального наблюдения или устройство отображения другого типа может также быть подключено к системной шине 4008 через интерфейс, например, видеоадаптер 4046. Видеоадаптер 4046 (или другой компонент) может представлять собой или включать в себя графическую карту для обработки графики, требующей больших вычислительных ресурсов, и для выполнения требований, предъявляемых дисплеем. Обычно графическая карта содержит графический процессор (ГП), видеопамять (ВОЗУ) и т.д. для облегчения быстрого отображения

графики и выполнения графических операций. Помимо монитора 4044, другие периферийные устройства вывода могут включать в себя такие компоненты, как громкоговорители (не показаны) и принтер 4048, которые могут быть подключены к компьютеру 4002 через интерфейсы 4042 ввода/вывода.

Компьютер 4002 может работать в сетевой среде с использованием логических соединений с одним или несколькими удаленными компьютерами, например, удаленным вычислительным устройством 4050. Например, удаленное вычислительное устройство 4050 может представлять собой персональный компьютер, портативный компьютер (например, лэптоп, планшетный компьютер, КПК, мобильную станцию и др.), ручной или карманный компьютер, часы, игровое устройство, сервер, маршрутизатор, сетевой компьютер, равноправное устройство, другой сетевой узел или устройство другого типа из вышеперечисленных и т.п. Однако удаленное вычислительное устройство 4050 показано в виде портативного компьютера, который может содержать многие или все элементы и особенности, описанные здесь применительно к компьютеру 4002.

Логические соединения между компьютером 4002 и удаленным компьютером 4050 представлены в виде локальной сети (ЛС) 4052 и общей глобальной сети (ГС) 4054. Такие сетевые среды обычно используются в офисных, производственных компьютерных сетях, интрасетях, в Интернете, стационарных и мобильных телефонных сетях, специальных и инфраструктурных беспроводных сетях, других беспроводных сетях, игровых сетях, в некоторых их комбинациях и т.д. Такие сети и коммуникационные соединения являются примерами сред передачи данных.

При реализации в сетевой среде ЛС, компьютер 4002 подключен к ЛС 4052 через сетевой интерфейс или адаптер 4056. При реализации в сетевой среде ГС компьютер 4002 обычно содержит модем 4058 или другие средства установления соединений по ГС 4054. Модем 4058, который может быть внутренним или внешним по отношению к компьютеру 4002, может быть подключен к системной шине 4008 через интерфейсы 4042 ввода/вывода или любые другие соответствующие механизмы. Очевидно, что показанные сетевые соединения являются иллюстративными и что для установления линии(й) связи между компьютерами 4002 и 4050 можно использовать другие средства.

Кроме того, можно применять другое оборудование, специально разработанное для серверов. Например, для выгрузки вычислений SSL можно использовать карты ускорения SSL. Кроме того, особенно в операционной среде выравнивания сетевой нагрузки можно установить и использовать на серверных устройствах оборудование выгрузки TCP и/или классификаторы пакетов на сетевых интерфейсах или адаптерах 4056 (например, на картах сетевого интерфейса).

В сетевой среде, проиллюстрированной посредством операционной среды 4000, программные модули или другие команды, описанные применительно к компьютеру 4002, или их часть могут полностью или частично храниться на удаленном запоминающем устройстве. Например, удаленные прикладные программы 4060 размещаются в компоненте памяти удаленного компьютера 4050, но могут использоваться или могут быть иным образом доступны через компьютер 4002. Также, в целях иллюстрации, прикладные программы 4030 и другие команды, выполняемые процессором, например, операционная система 4028, показаны здесь в качестве дискретных блоков, хотя понятно, что такие программы, компоненты и другие команды размещаются в разные моменты времени в разных компонентах хранения вычислительного устройства 4002 (и/или удаленного вычислительного

устройства 4050) и выполняются процессором(ами) 4004 компьютера 4002 (и/или удаленного вычислительного устройства 4050).

Хотя системы, носители, устройства, способы, процедуры, аппараты, методики, схемы, подходы, конфигурации и другие примеры осуществления изобретения были описаны на языке, специфическом для структурных, логических, алгоритмических и функциональных особенностей и/или диаграмм, следует понимать, что изобретение, заявленное в прилагаемой формуле изобретения, не должно ограничиваться конкретными описанными выше особенностями или диаграммами. Напротив, конкретные особенности и диаграммы, раскрытые в иллюстративной форме, поясняют заявленное изобретение, объем которого определяется нижеследующей формулой изобретения.

Формула изобретения

1. Система выравнивания сетевой нагрузки, содержащая

средство (1208) для приема от множества хостов (108) информации (1206) о работоспособности и нагрузке приложения, причем информация о работоспособности и нагрузке приложения включает в себя указание о том, когда приложение, выполняемое указанными хостами, переходит в состояние неработоспособности, и

средство (106) для принятия решений о выравнивании нагрузки в соответствии с принятой информацией о работоспособности и нагрузки приложения.

2. Система по п.1, в которой средство для приема информации содержит

средство для приема информации о работоспособности и нагрузке приложений от множества хостов посредством по меньшей мере одного посредника.

3. Система по п.1, в которой информация о работоспособности и нагрузке приложения включает в себя по меньшей мере одну директиву выравнивания нагрузки, при этом средство для приема информации содержит

средство для приема по меньшей мере одной директивы выравнивания нагрузки от множества хостов посредством по меньшей мере одного посредника, который вызывает один или несколько программных интерфейсов приложения (API) для продвижения, по меньшей мере, одной директивы выравнивания нагрузки.

4. Система по п.1, которая содержит, по меньшей мере, одно единичное устройство или множество устройств.

5. Система выравнивания сетевой нагрузки, содержащая инфраструктуру (1202) работоспособности и нагрузки, содержащую таблицу (1204) работоспособности и нагрузки, имеющую множество записей, связанных с множеством приложений, которые выполняются на множестве хостов (108), при этом инфраструктура работоспособности и нагрузки приспособлена определять информацию (1206) о работоспособности и нагрузке приложения от множества хостов (108), при этом информация о работоспособности приложения включает в себя указание о том, когда приложение, выполняемое указанными хостами, находится в состоянии неработоспособности, и

инфраструктуру (106) выравнивания нагрузки, приспособленную использовать информацию о работоспособности и нагрузке приложения от множества хостов (108) при выделении запросов на множество приложений (316), при этом инфраструктура (106) выравнивания нагрузки содержит объединенный кэш работоспособности и нагрузки, который хранит информацию о работоспособности и нагрузке множества приложений, выполняемых на множестве хостов (108).

6. Система по п.5, в которой инфраструктура работоспособности и нагрузки

содержит таблицу работоспособности и нагрузки, в которой хранится, по меньшей мере, часть информации о работоспособности и нагрузке, зависящей от приложения.

7. Система по п.5, в которой инфраструктура работоспособности и нагрузки содержит таблицу работоспособности и нагрузки, в которой хранится, по меньшей мере, часть информации работоспособности и нагрузки, зависящей от приложения, причем таблица работоспособности и нагрузки содержит множество элементов, причем каждый элемент из множества элементов связан с конкретным приложением из множества приложений.

8. Система по п.5, в которой инфраструктура работоспособности и нагрузки содержит таблицу работоспособности и нагрузки, в которой хранится, по меньшей мере, часть информации о работоспособности и нагрузке, зависящей от приложения, причем таблица работоспособности и нагрузки содержит множество элементов, причем каждый элемент из множества элементов содержит идентификатор приложения для конкретного приложения, с которым связан элемент, информацию, характеризующую, по меньшей мере, один статус конкретного приложения, и, по меньшей мере, одну директиву выравнивания нагрузки в отношении конкретного приложения.

9. Система по п.5, в которой инфраструктура выравнивания нагрузки содержит объединенный кэш работоспособности и нагрузки, в котором хранится информация о работоспособности и нагрузке приложения.

10. Система по п.5, в которой инфраструктура выравнивания нагрузки содержит объединенный кэш работоспособности и нагрузки, в котором хранится информация о работоспособности и нагрузке множества приложений, выполняющихся на множестве хостов.

11. Система по п.5, в которой информация о работоспособности и нагрузке приложения содержит информацию о работоспособности и нагрузке, зависящую от конечных точек приложения.

12. Система по п.5, которая также содержит устройство-посредник, которое содержит, по меньшей мере, часть инфраструктуры работоспособности и нагрузки, причем, по меньшей мере, часть инфраструктуры работоспособности и нагрузки приспособлена определять информацию о работоспособности и нагрузке приложения, осуществляя действия внешнего слежения.

13. Система по п.5, в которой

инфраструктура работоспособности и нагрузки содержит множество таблиц работоспособности и нагрузки, в которых хранится информация о работоспособности и нагрузке приложения, и

инфраструктура выравнивания нагрузки содержит множество объединенных кэшей работоспособности и нагрузки, в которых хранится информация о работоспособности и нагрузке приложения.

14. Система по п.13, которая также содержит множество хостов, по которым распределена инфраструктура работоспособности и нагрузки, причем каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки.

15. Система по п.13, которая также содержит множество хостов, по которым распределена инфраструктура работоспособности и нагрузки, причем каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки,

причем каждый объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки содержит информацию о работоспособности и нагрузке приложения, которая хранится в каждой таблице работоспособности и нагрузки из множества таблиц работоспособности и нагрузки.

16. Система по п.13, которая также содержит множество хостов, по которым распределена инфраструктура работоспособности и нагрузки, причем каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки,

причем множество приложений выполняется на множестве хостов.

17. Система по п.13, которая также содержит множество хостов, размещенных на множестве устройств, причем инфраструктура работоспособности и нагрузки распределена по множеству хостов, каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, состоящих из, по меньшей мере, одного устройства и соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки.

18. Система по п.13, которая также содержит множество хостов, размещенных на множестве устройств, причем инфраструктура работоспособности и нагрузки распределена по множеству хостов, каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, состоящих из, по меньшей мере, одного устройства и соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки,

причем инфраструктура работоспособности и нагрузки содержит удаленную часть инфраструктуры выравнивания нагрузки.

19. Система по п.13, которая также содержит множество хостов, размещенных на множестве устройств, причем инфраструктура работоспособности и нагрузки распределена по множеству хостов, каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, состоящих из, по меньшей мере,

одного устройства и соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки,

причем, по меньшей мере, одно устройство является одним из множества устройств.

20. Система по п.13, которая также содержит множество хостов, размещенных на множестве устройств, причем инфраструктура работоспособности и нагрузки распределена по множеству хостов, каждый хост из множества хостов имеет таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки, и

множество модулей выравнивания нагрузки, состоящих из, по меньшей мере, одного устройства и соответствующих, по меньшей мере, части инфраструктуры выравнивания нагрузки, причем каждый модуль выравнивания нагрузки из множества модулей выравнивания нагрузки имеет объединенный кэш работоспособности и нагрузки из множества объединенных кэшей работоспособности и нагрузки,

причем, по меньшей мере, одно устройство является одним из множества устройств, и

инфраструктура выравнивания нагрузки также приспособлена распространять информацию работоспособности и нагрузки, зависящую от приложения, от множества устройств на, по меньшей мере, одно устройство.

21. Система по п.5, в которой инфраструктура работоспособности и нагрузки и инфраструктура выравнивания нагрузки способны использовать протокол обмена сообщениями для связи между собой в отношении информации о работоспособности и нагрузке, зависящей от приложения.

22. Система по п.21, в которой протокол обмена сообщениями содержит одно или несколько типов сообщения, а именно, тип «контрольное сообщение», тип сообщения «прощание», тип сообщения «изменение строки», тип сообщения «получить мгновенный снимок таблицы», тип сообщения «отправить мгновенный снимок таблицы», тип сообщения «постулировать состояние таблицы» и тип сообщения «постулировать ошибку».

23. Система по п.21, в которой протокол обмена сообщениями включает возможность связи с использованием группового членства.

24. Система по п.5, в которой инфраструктура выравнивания нагрузки после сбоя способна восстанавливать информацию работоспособности и нагрузки, зависящую от приложения, через инфраструктуру работоспособности и нагрузки с использованием протокола обмена сообщениями для связи между собой.

25. Система по п.5, в которой инфраструктура выравнивания нагрузки также приспособлена выделять запросы на множество приложений с использованием одной или нескольких схем распределения.

26. Система по п.25, в которой одна или несколько схем распределения содержит, по меньшей мере, одну из маркерной схемы распределения и процентной схемы распределения.

27. Система по п.25, в которой одна или несколько схем распределения предусматривают использование механизма истечения таймера.

28. Система по п.5, которая также содержит, по меньшей мере, одно устройство, которое хостирует одно или несколько приложений, при этом инфраструктура работоспособности и нагрузки содержит таблицу работоспособности и нагрузки, содержащую множество элементов, причем каждый элемент из множества элементов связан с приложением из одного или нескольких приложений, каждый элемент из

множества элементов содержит

идентификатор приложения для конкретного приложения из одного или нескольких приложений и

информацию, характеризующую, по меньшей мере, один статус конкретного приложения и,

по меньшей мере, одну директиву выравнивания нагрузки в отношении конкретного приложения.

29. Система по п.28, в которой идентификатор приложения однозначно идентифицирует конкретное приложение из одного или нескольких приложений.

30. Система по п.28, в которой идентификатор приложения содержит, по меньшей мере, один из виртуального адреса Интернет-протокола (IP) и порта, физического IP-адреса и порта, протокола, относящегося к конкретному приложению, и информации, зависящей от протокола.

31. Система по п.28, в которой идентификатор приложения содержит, по меньшей, мере, один глобально уникальный идентификатор (GUID).

32. Система по п.28, в которой информация, характеризующая, по меньшей мере, один статус конкретного приложения, содержит, по меньшей мере, одно из работоспособности приложения, нагрузки приложения и емкости приложения.

33. Система по п.32, в которой работоспособность приложения указывает, является ли статус конкретного приложения «работоспособно», «неработоспособно» или «неизвестно», нагрузка приложения указывает, насколько занято конкретное приложение, и емкость приложения указывает максимальную емкость конкретного приложения.

34. Система по п.33, в которой максимальная емкость конкретного приложения выражается (i) относительно суммарной емкости приложений того же типа, что и конкретное приложение, выполняющееся в, по меньшей мере, одном устройстве, и/или (ii) как безразмерное и ограниченное число.

35. Система по 28, в которой, по меньшей мере, одна директива выравнивания нагрузки может быть предоставлена множеством модулей выравнивания нагрузки инфраструктуры выравнивания нагрузки для обеспечения руководства для выравнивания сетевой нагрузки в отношении конкретного приложения и в отношении других приложений, относящихся к тому же типу приложения.

36. Система по п.28, в которой, по меньшей мере, одна директива выравнивания нагрузки содержит, по меньшей мере, одно из «активное», «истощающееся» и «неактивное».

37. Система по п.28, в которой, по меньшей мере, одна директива выравнивания нагрузки содержит директиву целевого состояния выравнивания нагрузки и директиву текущего состояния выравнивания нагрузки.

38. Система по п.37, в которой (i) директива целевого состояния выравнивания нагрузки указывает состояние выравнивания нагрузки, в котором инфраструктура выравнивания нагрузки должна работать, которое предусматривает инфраструктура работоспособности и нагрузки, при этом инфраструктура выравнивания нагрузки включает в себя множество модулей выравнивания нагрузки, и (ii) директива текущего состояния выравнивания нагрузки указывает состояние выравнивания нагрузки, в котором инфраструктура выравнивания нагрузки должна работать, которое предполагает инфраструктура работоспособности и нагрузки.

39. Система по п.28, которая также содержит множество устройств, при этом инфраструктура работоспособности и нагрузки содержит множество таблиц

работоспособности и нагрузки, и каждое соответствующее устройство из множества устройств включает в себя соответствующую таблицу работоспособности и нагрузки из множества таблиц работоспособности и нагрузки.

40. Система по п.5, которая реализует протокол обмена сообщениями между, по меньшей мере, одним хостом инфраструктуры работоспособности и нагрузки и одним или более модулями выравнивания нагрузки инфраструктуры выравнивания нагрузки, причем протокол обмена сообщениями используется для передачи информации о работоспособности и нагрузке между, по меньшей мере, одним хостом, и одним или более модулями выравнивания нагрузки.

41. Система по п.40, в которой протокол обмена сообщениями содержит «контрольное сообщение», указывающее одному или нескольким модулям выравнивания нагрузки, что, по меньшей мере, один хост функционирует.

42. Система по п.41, в которой формат «контрольного сообщения» содержит идентификатор для, по меньшей мере, одного хоста, данные контроля ошибок для информации о работоспособности и/или нагрузке и имя системы доменных имен (DNS).

43. Система по п.41, в которой формат «контрольного сообщения» допускает включение пары номер куска/идентификатор (ИД) поколения.

44. Система по п.40, в которой протокол обмена сообщениями включает в себя сообщение «прощание», указывающее одному или нескольким модулям выравнивания нагрузки, что, по меньшей мере, один хост планируется отключить.

45. Система по п.44, в которой формат сообщения «прощание» содержит идентификатор для, по меньшей мере, одного хоста.

46. Система по п.41, в которой протокол обмена сообщениями включает в себя сообщение «изменение строки», указывающее одному или нескольким модулям выравнивания нагрузки, что информация о работоспособности и нагрузке для приложения, по меньшей мере, одного хоста изменилась.

47. Система по п.46, в которой формат сообщения «изменение строки» содержит идентификатор для, по меньшей мере, одного хоста, идентификатор для приложения, операцию для отражения изменения и данные для операции.

48. Система по п.40, в которой протокол обмена сообщениями включает в себя сообщение «получить мгновенный снимок таблицы», отправляемое с одного или нескольких модулей выравнивания нагрузки на, по меньшей мере, один хост, сообщение «получить мгновенный снимок таблицы» запрашивает мгновенный снимок текущей информации о работоспособности и/или нагрузке, по меньшей мере, одного хоста.

49. Система по п.48, в которой формат сообщения «получить мгновенный снимок таблицы» содержит идентификацию запрашивающего модуля выравнивания нагрузки из одного или нескольких модулей выравнивания нагрузки.

50. Система по п.40, в которой протокол обмена сообщениями включает в себя сообщение «отправить мгновенный снимок таблицы», отправляемое с, по меньшей мере, одного хоста на запрашивающий модуль выравнивания нагрузки из одного или нескольких модулей выравнивания нагрузки, сообщение «отправить мгновенный снимок таблицы» обеспечивает мгновенный снимок текущей информации о работоспособности и/или нагрузке, по меньшей мере, одного хоста.

51. Система по п.50, в которой формат сообщения «отправить мгновенный снимок таблицы» содержит мгновенный снимок текущей информации о работоспособности и/или нагрузке, по меньшей мере, одного хоста.

52. Система по п.40, в которой протокол обмена сообщениями включает в себя сообщение «постулировать состояние таблицы», отправляемое с, по меньшей мере, одного хоста на один или несколько модулей выравнивания нагрузки, причем сообщение «постулировать состояние таблицы» включает в себя директиву состояния выравнивания нагрузки, указывающую директиву текущего состояния выравнивания нагрузки, в отношении которой, по меньшей мере, один хост ожидает, что она существует на одном или нескольких модулях выравнивания нагрузки.

53. Система по п.40, в которой формат сообщения «постулировать состояние таблицы» содержит идентификатор для, по меньшей мере, одного хоста и директиву текущего состояния выравнивания нагрузки.

54. Система по п.41, в которой протокол обмена сообщениями включает в себя сообщение «постулировать ошибку», отправляемое с модуля выравнивания нагрузки из одного или нескольких модулей выравнивания нагрузки на, по меньшей мере, один хост, который ранее отправил сообщение «постулировать состояние таблицы», сообщение «постулировать ошибку» указывает, что фактическая директива состояния выравнивания нагрузки модуля выравнивания нагрузки отличается от постулированной директивы состояния выравнивания нагрузки, включенной в сообщение «постулировать состояние таблицы».

55. Носитель, доступный процессору, хранящий команды, которые при исполнении компьютером (4002) побуждают его выполнять способ, содержащий этапы, на которых

принимают (1406) информацию о работоспособности и нагрузке приложения от множества хостов (108), причем информация о работоспособности и нагрузке включает в себя указание о том, когда приложение, выполняемое указанными хостами, находится в состоянии неработоспособности, и

осуществляют (1414) принятие решений о выравнивании нагрузки в соответствии с принятой информацией о работоспособности и нагрузке приложения.

56. Носитель п.55, в котором способ также содержит этап, на котором принимают запрос от клиента на новое соединение, и

при этом этап, на котором принимают решения, содержит этап, на котором выбирают пункт назначения в соответствии с принятой информацией о работоспособности и нагрузке приложения.

57. Носитель по п.55, в котором этап приема содержит по меньшей мере один этап

приема информации статуса хоста непосредственно от одного или нескольких хостов из множества хостов или

приема информации статуса хоста опосредованно от одного или нескольких хостов из множества хостов.

58. Способ выравнивания сетевой нагрузки, содержащий этапы, на которых:

определяют (1402) информацию о работоспособности и нагрузке для каждого из приложений от множества хостов (108), причем информация о работоспособности и нагрузке включает в себя указание о том, когда приложение, выполняемое указанными хостами, находится в состоянии неработоспособности, и

выбирают (1414) как часть решения о выравнивании нагрузки приложение из множества приложений в соответствии с принятой информацией о работоспособности и нагрузке.

59. Способ по п.58, в котором на этапе определения определяют, когда приложения запускают и останавливают.

60. Способ по п.58, в котором на этапе определения определяют, когда приложение

из множества приложений работоспособно и когда приложение неработоспособно.

61. Способ по п.58, в котором на этапе определения определяют нагрузку данного приложения определенного типа приложения относительно нагрузки одного или более приложений этого определенного типа приложений.

62. Способ по п.58, содержащий также этап, на котором принимают внешний входной сигнал относительно определения информации о работоспособности и нагрузке приложения, при этом на этапе определения определяют информацию о работоспособности и нагрузке приложения в соответствии с внешним входным сигналом.

63. Способ по п.58, содержащий также этап, на котором распространяют информацию о работоспособности и нагрузке от по меньшей мере одного хоста одному или более модулям выравнивания нагрузки.

64. Способ по п.58, содержащий также этап, на котором распространяют информацию о работоспособности и нагрузке от по меньшей мере одного хоста одному или более модулям выравнивания нагрузки, используя групповое членство.

65. Способ по п.64, в котором на этапе распространения направляют «контрольное сообщение» с, по меньшей мере, одного хоста на лидирующий хост, причем сообщение «пульс» содержит указание пересылки, в результате чего лидирующий хост получает предписание переслать сообщение «пульс» на один или несколько модулей выравнивания нагрузки даже в отсутствие изменения группового членства.

66. Способ по п.58, содержащий также этап, на котором распространяют информацию о работоспособности и нагрузке из, по меньшей мере, одной таблицы работоспособности и нагрузки на один или несколько объединенных кэшей работоспособности и нагрузки.

67. Способ по п.58, содержащий также этап, на котором принимают информацию работоспособности и нагрузке от множества хостов и

кэшируют информацию о работоспособности и нагрузке.

68. Способ по п.58, содержащий также этапы, на которых принимают информацию о работоспособности и нагрузке от множества хостов,

кэшируют информацию о работоспособности и нагрузке,

принимают пакет, запрашивающий инициирование соединения, и

сверяют кэшированную информацию о работоспособности и нагрузке для инициирования соединения,

при этом на этапе выбора выбирают приложение из множества приложений в соответствии со сверкой.

69. Способ по п.68, в котором инициирование соединения относится к конкретному типу сообщения.

70. Способ по п.58, в котором на этапе выбора выбирают конечную точку приложения из множества конечных точек приложения в соответствии с информацией о работоспособности и нагрузке.

71. Способ по п.58, в котором на этапе выбора выбирают в соответствии с информацией о работоспособности и нагрузке конечную точку приложения из множества конечных точек приложения, которые распределены по множеству хостов.

72. Способ по п.58, в котором на этапе выбора выбирают, в соответствии с информацией о работоспособности и нагрузке распределение конечных точек приложения из множества конечных точек приложения в отношении относительных доступных емкостей в множестве конечных точек приложения.

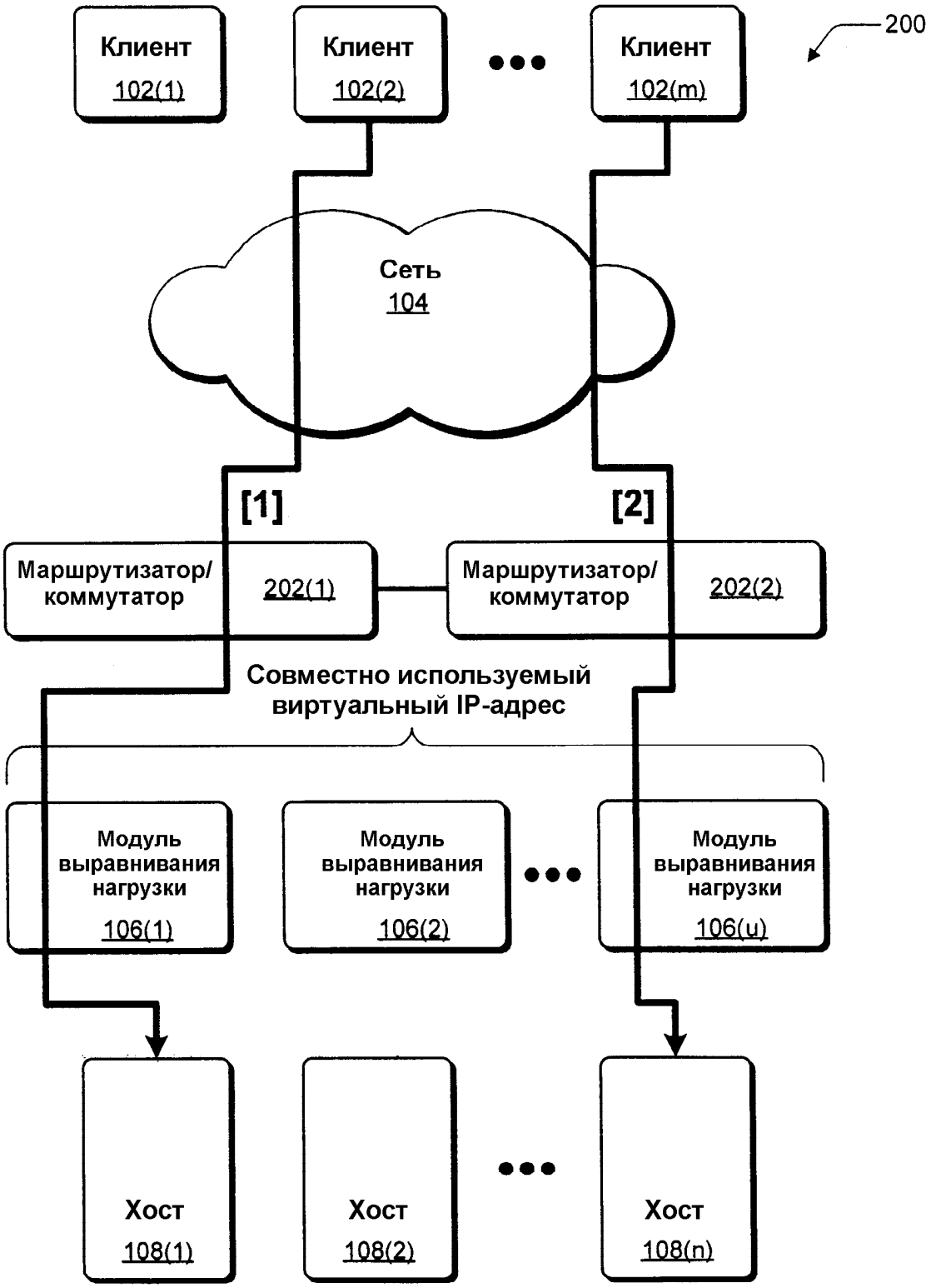
73. Способ по п.58, в котором на этапе выбора также выбирают распределение

конечных точек распределения с использованием маркерной схемы распределения и/или процентной схемы распределения.

74. Способ по п.72, в котором на множество конечных точек приложения соответствует приложениям одного типа приложения.

75. Способ по п.58, в котором на этапе выбора выбирают приложение из множества приложений в соответствии с информацией о работоспособности и нагрузке для выравнивания сетевой нагрузки, вызванной входящими пакетами.

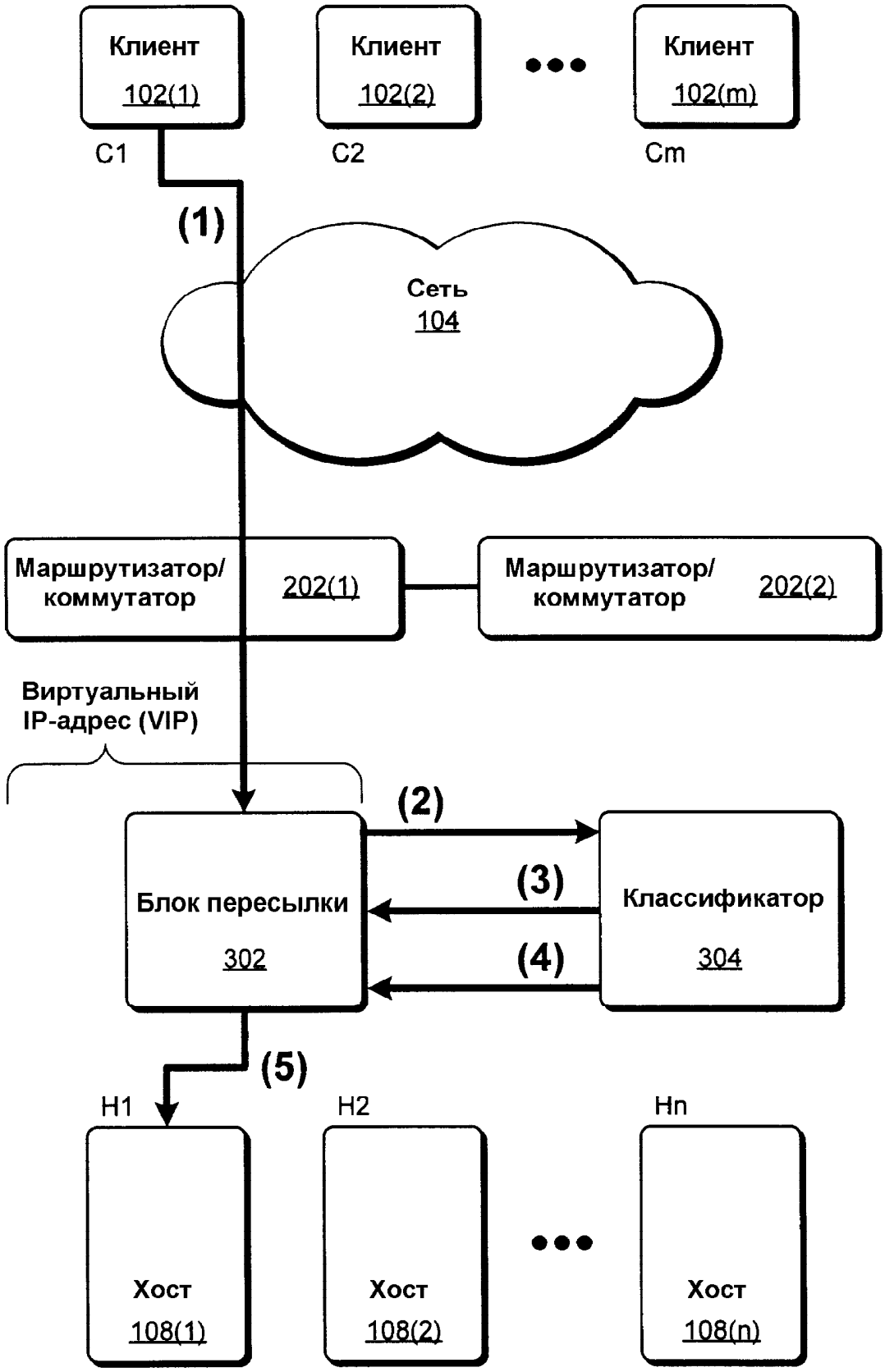
76. Способ по п.58, в котором на этапе выбора выбирают приложение из множества приложений в соответствии с информацией о работоспособности и нагрузке для выравнивания сетевой нагрузки, вызванной входящими запросами соединения.


200
Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
Сеть
104
[1]
[2]
Маршрутизатор/
коммутатор
202(1)
Маршрутизатор/
коммутатор
202(2)
Совместно используемый
виртуальный IP-адрес
Модуль
выравнивания
нагрузки
106(1)
Модуль
выравнивания
нагрузки
106(2)
Модуль
выравнивания
нагрузки
106(u)
Хост
108(1)
Хост
108(2)
Хост
108(n)

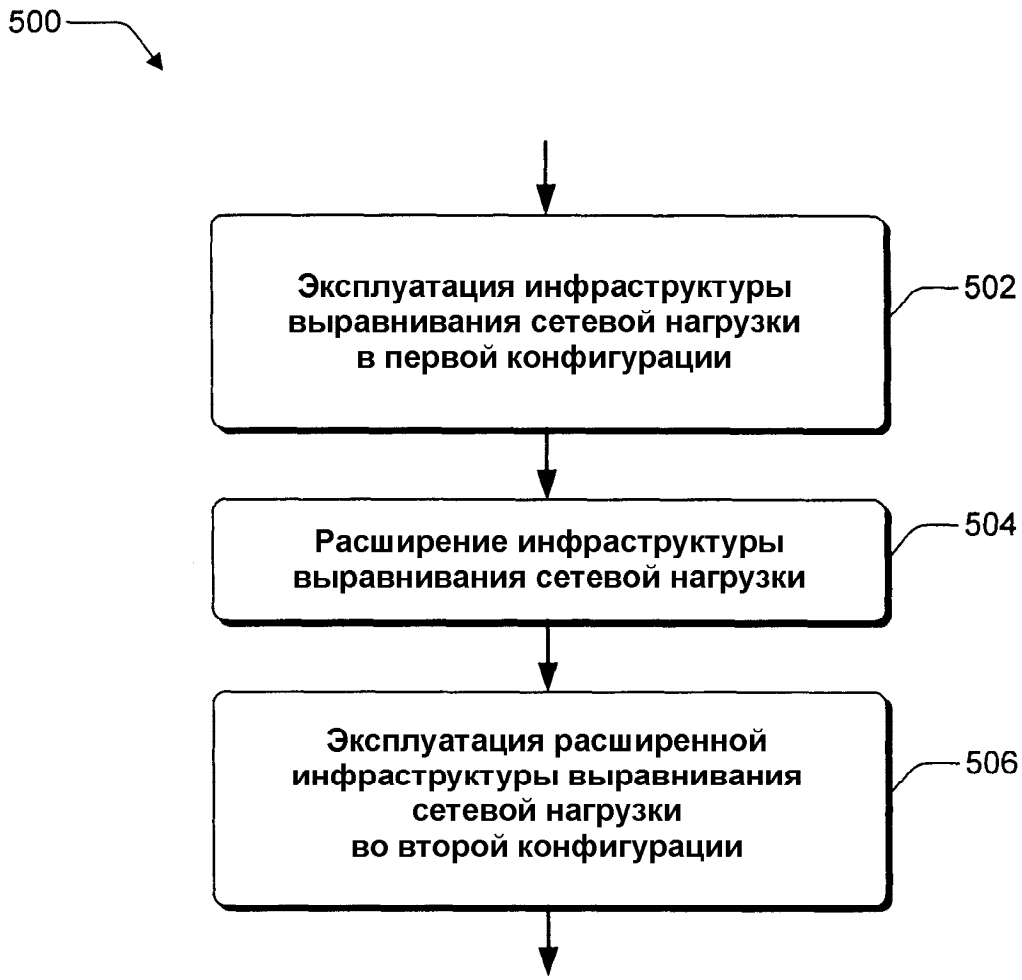
Фиг. 2


Блок пересылки 302
Классификатор 304
Маршрутизатор запросов 306
Блок отслеживания сеансов 308
Блок переноса соединений 310
Блок туннелирования 312
Обработчик работоспособности и нагрузки 314
Модуль выравнивания нагрузки 106
Приложение(я) 316
Хост 108

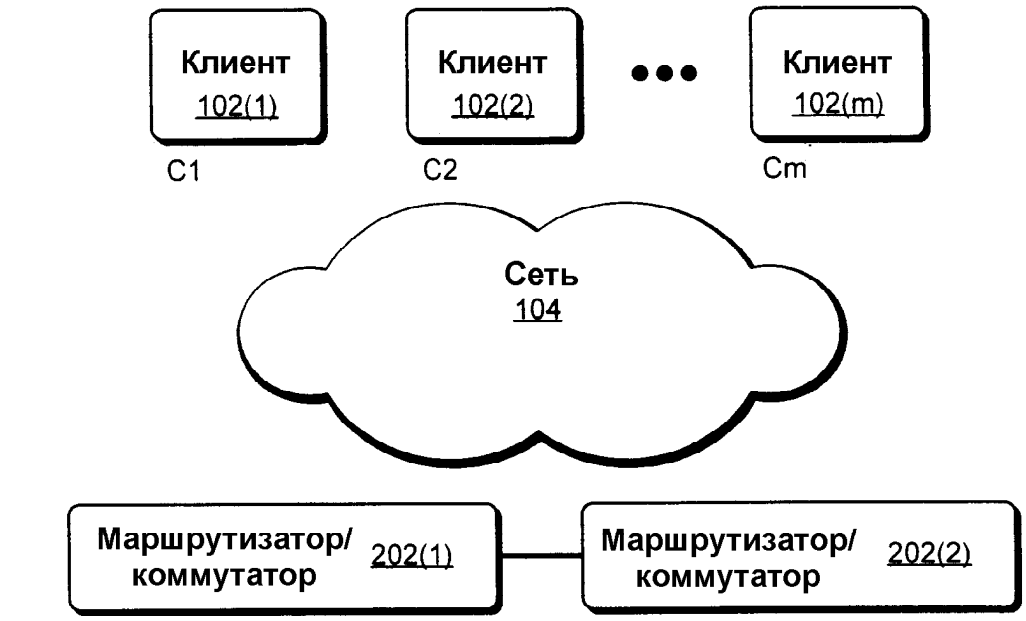
Фиг. 3

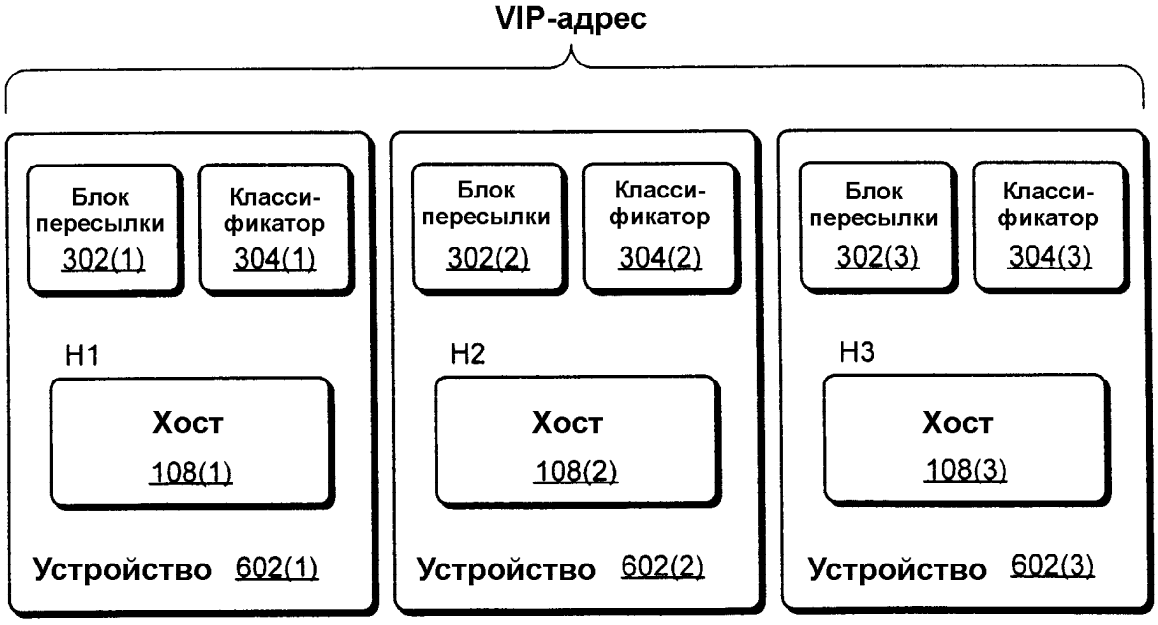

Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
C1
C2
Cm
(1)
Сеть
104
Маршрутизатор/
коммутатор
202(1)
Маршрутизатор/
коммутатор
202(2)
Виртуальный
IP-адрес (VIP)
(2)
(3)
(4)
Блок пересылки
302
Классификатор
304
(5)
H1
H2
Hn
Хост
108(1)
Хост
108(2)
Хост
108(n)

Фиг. 4

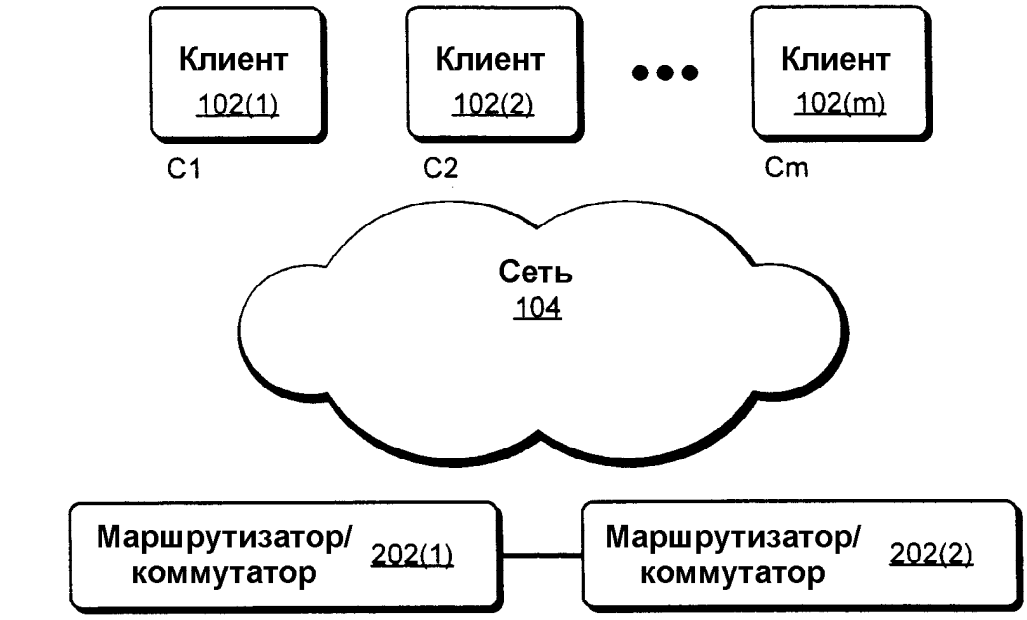

500
Эксплуатация инфраструктуры выравнивания сетевой нагрузки в первой конфигурации
502
Расширение инфраструктуры выравнивания сетевой нагрузки
504
Эксплуатация расширенной инфраструктуры выравнивания сетевой нагрузки во второй конфигурации
506

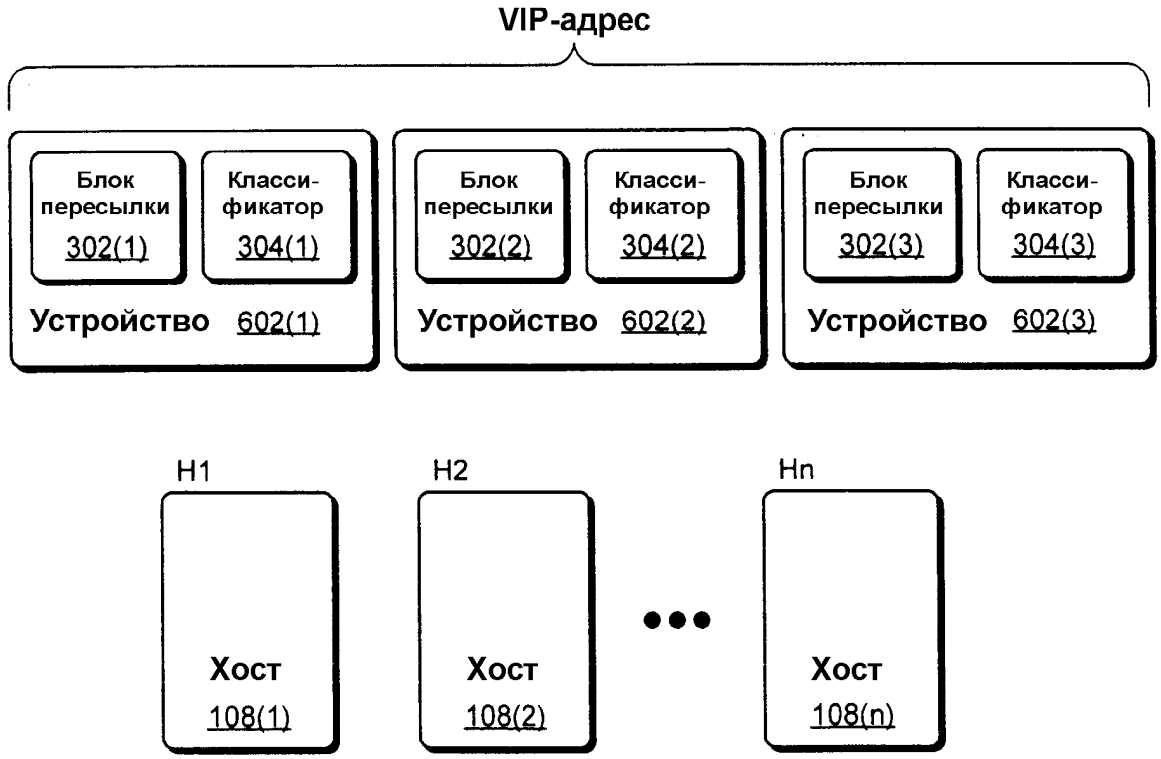
Фиг. 5


Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
C1
C2
Cm
Сеть
104
Маршрутизатор/
коммутатор
202(1)
Маршрутизатор/
коммутатор
202(2)

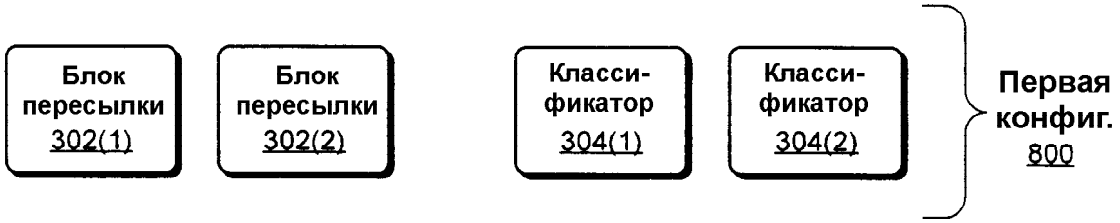

VIP-адрес
Блок
пересылки
302(1)
Класси-
фикатор
304(1)
H1
Хост
108(1)
Устройство 602(1)
Блок
пересылки
302(2)
Класси-
фикатор
304(2)
H2
Хост
108(2)
Устройство 602(2)
Блок
пересылки
302(3)
Класси-
фикатор
304(3)
H3
Хост
108(3)
Устройство 602(3)

Фиг. 6

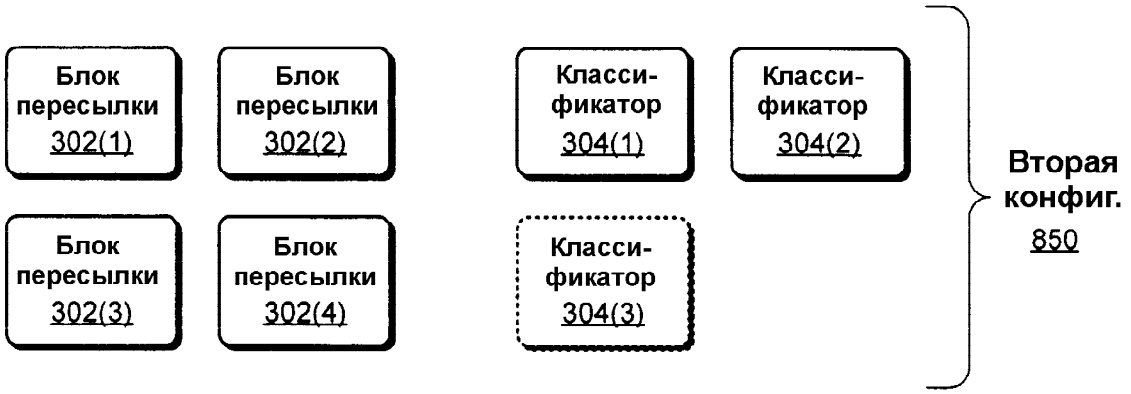

Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
C1
C2
Cm
Сеть
104
Маршрутизатор/
коммутатор
202(1)
Маршрутизатор/
коммутатор
202(2)


VIP-адрес
Блок
пересылки
302(1)
Класси-
фикатор
304(1)
Устройство 602(1)
Блок
пересылки
302(2)
Класси-
фикатор
304(2)
Устройство 602(2)
Блок
пересылки
302(3)
Класси-
фикатор
304(3)
Устройство 602(3)
H1
H2
Hn
Хост
108(1)
Хост
108(2)
Хост
108(n)

Фиг. 7

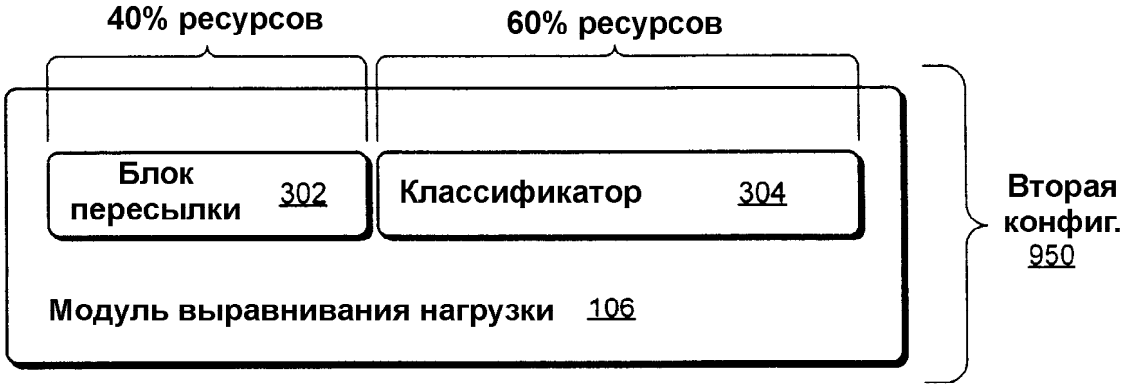

Блок
пересылки
302(1)
Блок
пересылки
302(2)
Класси-
фикатор
304(1)
Класси-
фикатор
304(2)
Первая
конфиг.
800

Фиг. 8A

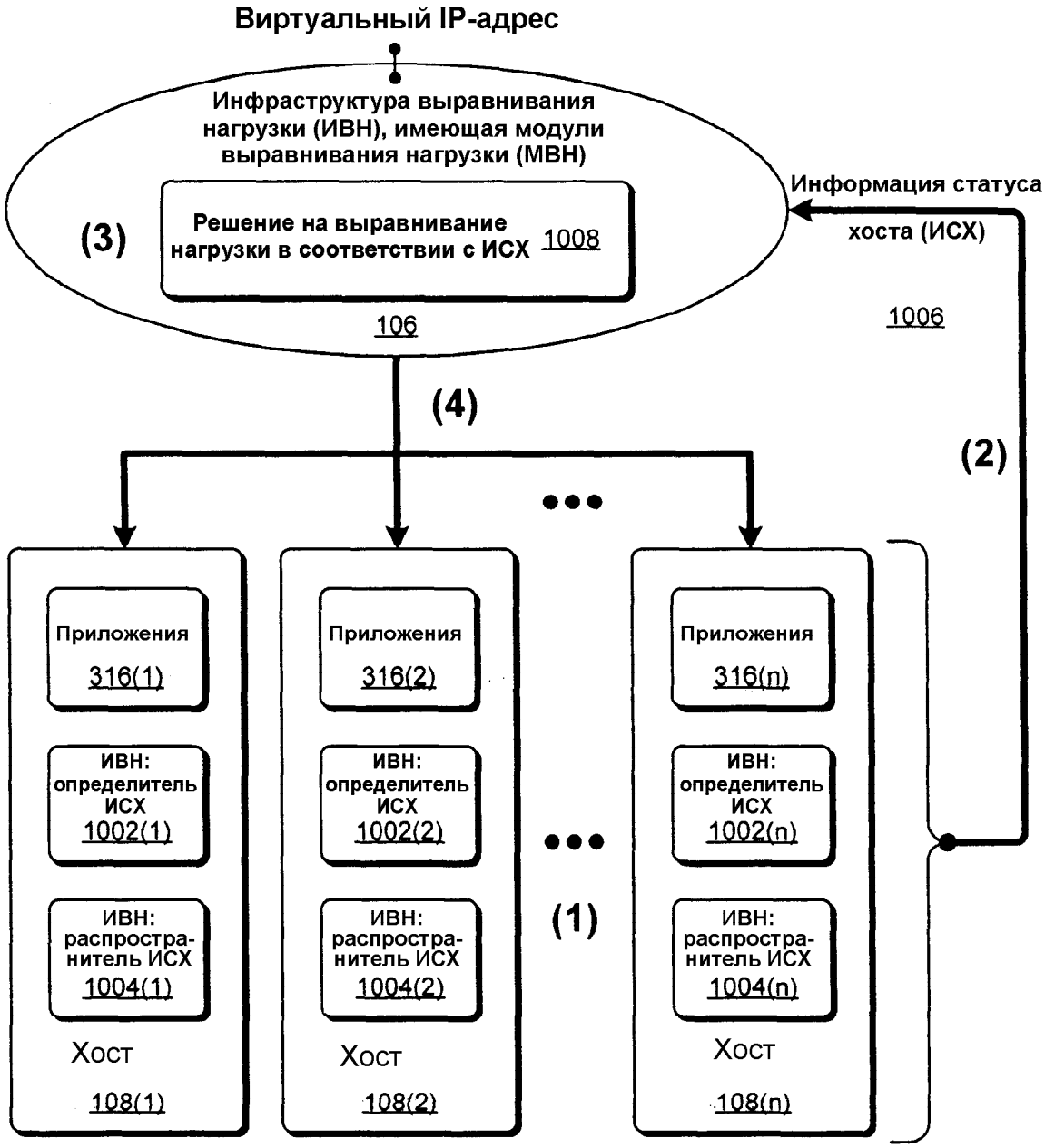


Фиг. 8B

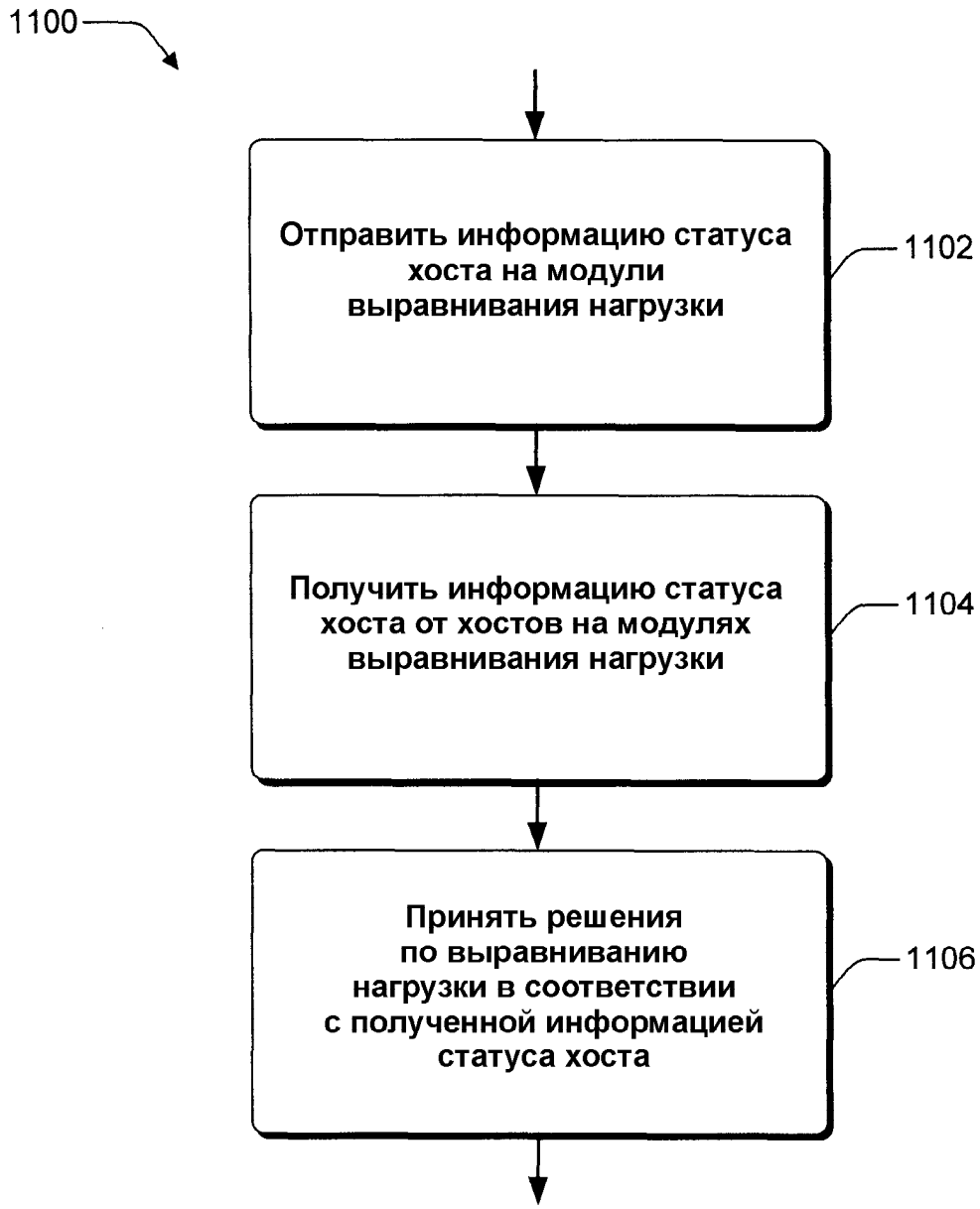
Фиг. 9A



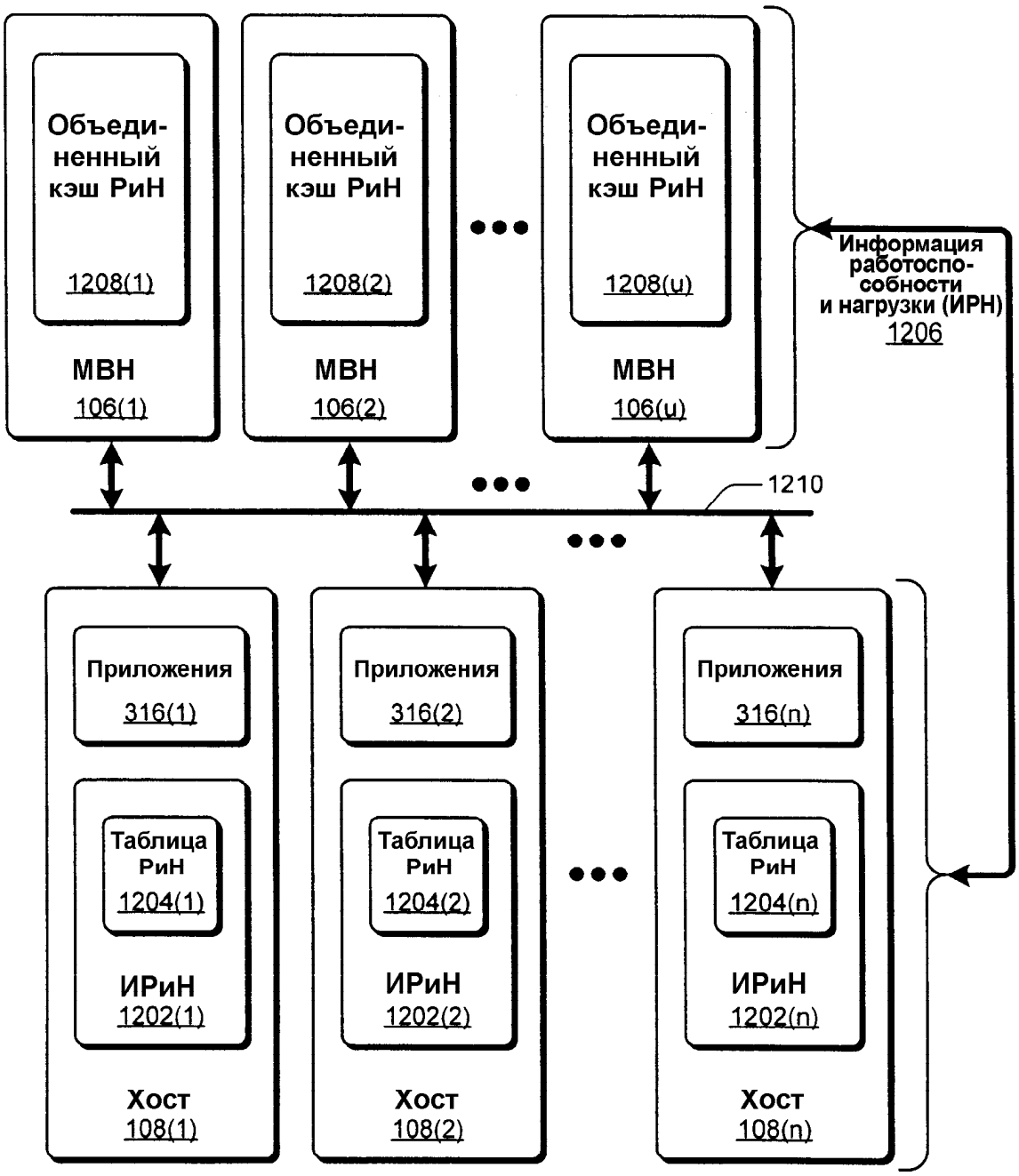
Фиг. 9B


Виртуальный IP-адрес
Инфраструктура выравнивания нагрузки (ИВН), имеющая модули выравнивания нагрузки (МВН)
(3)
Решение на выравнивание нагрузки в соответствии с ИСХ 1008
106
Информация статуса хоста (ИСХ)
1006
(4)
(2)
Приложения 316(1)
ИВН: определитель ИСХ 1002(1)
ИВН: распространитель ИСХ 1004(1)
Хост 108(1)
Приложения 316(2)
ИВН: определитель ИСХ 1002(2)
ИВН: распространитель ИСХ 1004(2)
Хост 108(2)
(1)
Приложения 316(n)
ИВН: определитель ИСХ 1002(n)
ИВН: распространитель ИСХ 1004(n)
Хост 108(n)

Фиг. 10


1100
Отправить информацию статуса хоста на модули выравнивания нагрузки
1102
Получить информацию статуса хоста от хостов на модулях выравнивания нагрузки
1104
Принять решения по выравниванию нагрузки в соответствии с полученной информацией статуса хоста
1106

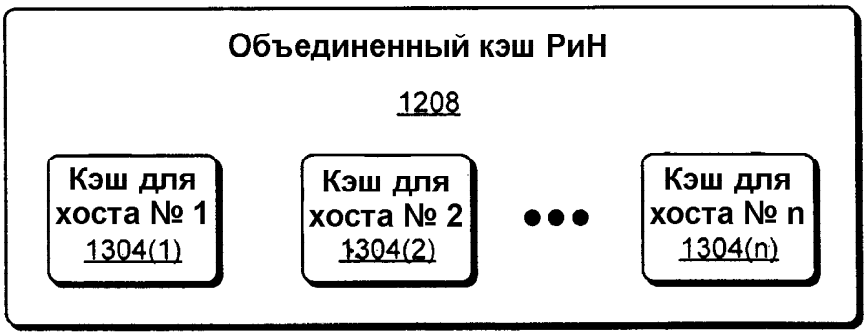
Фиг. 11


Объединенный кэш РиН
1208(1)
МВН
106(1)
Объединенный кэш РиН
1208(2)
МВН
106(2)
Объединенный кэш РиН
1208(u)
МВН
106(u)
Информация работоспособности и нагрузки (ИРН)
1206
1210
Приложения
316(1)
Таблица РиН
1204(1)
ИРиН
1202(1)
Хост
108(1)
Приложения
316(2)
Таблица РиН
1204(2)
ИРиН
1202(2)
Хост
108(2)
Приложения
316(n)
Таблица РиН
1204(n)
ИРиН
1202(n)
Хост
108(n)

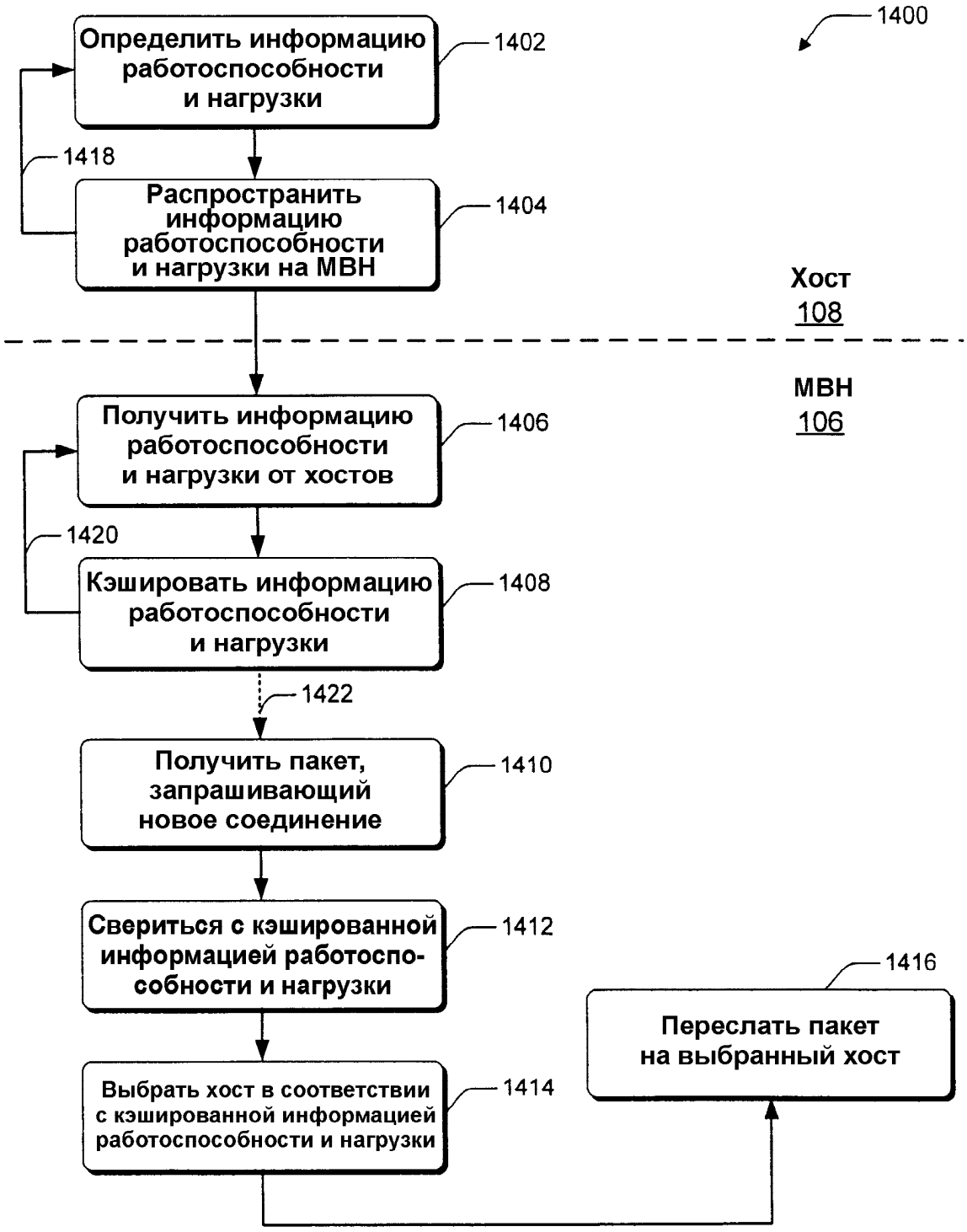
Фиг. 12

Таблица РиН 1204		
Идентификатор (ИД) приложения 1302(A)	Характеристика статуса приложения 1302(B)	Директива выравнивателя нагрузки 1302(C)
Виртуальный IP-адрес и порт	Работоспособность приложения	Целевое состояние выравнивания нагрузки
Физический IP-адрес и порт	Нагрузка приложения	
Протокол		Текущее состояние выравнивания нагрузки
Информация, зависящая от протокола	Емкость приложения	

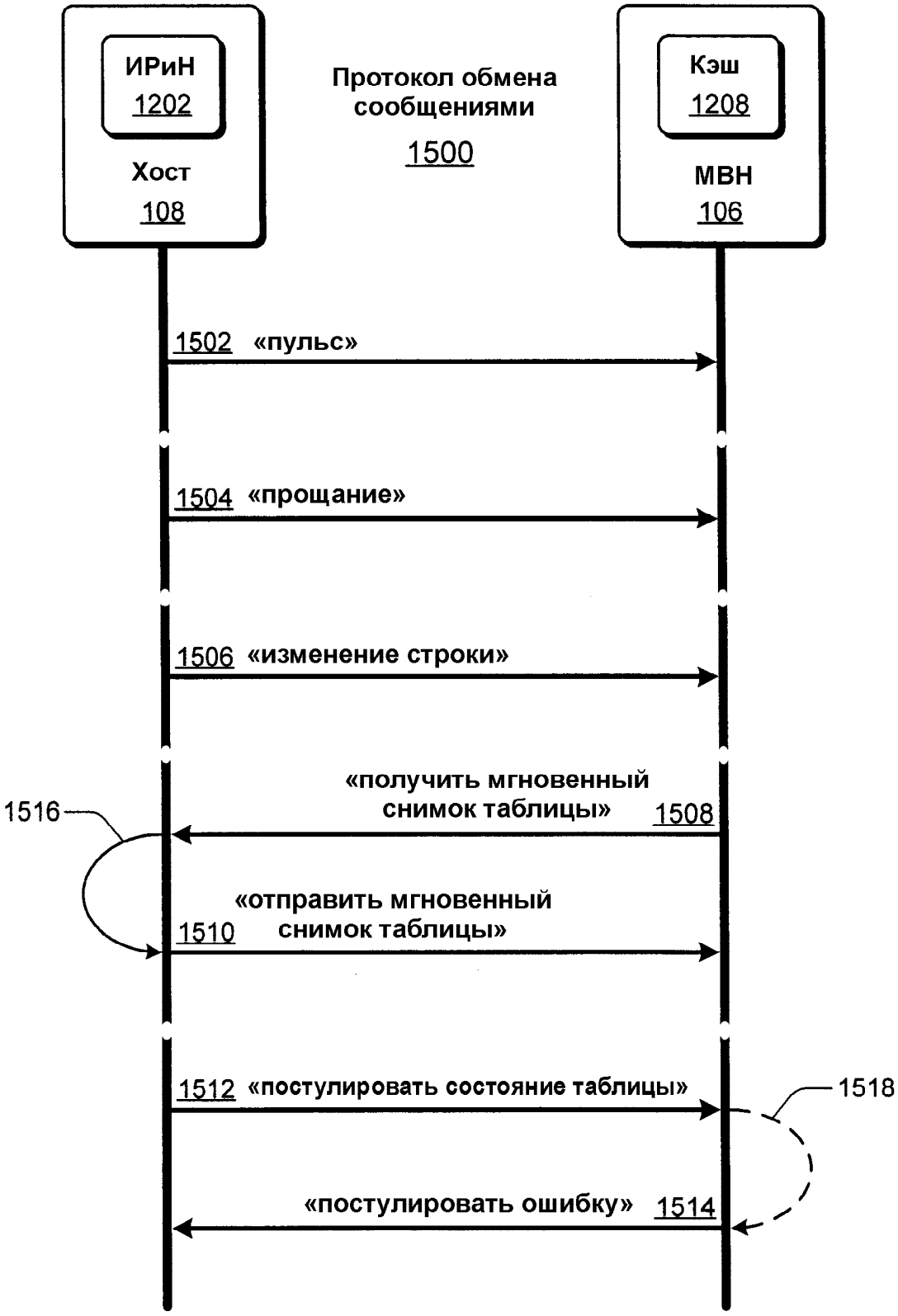
Фиг. 13A



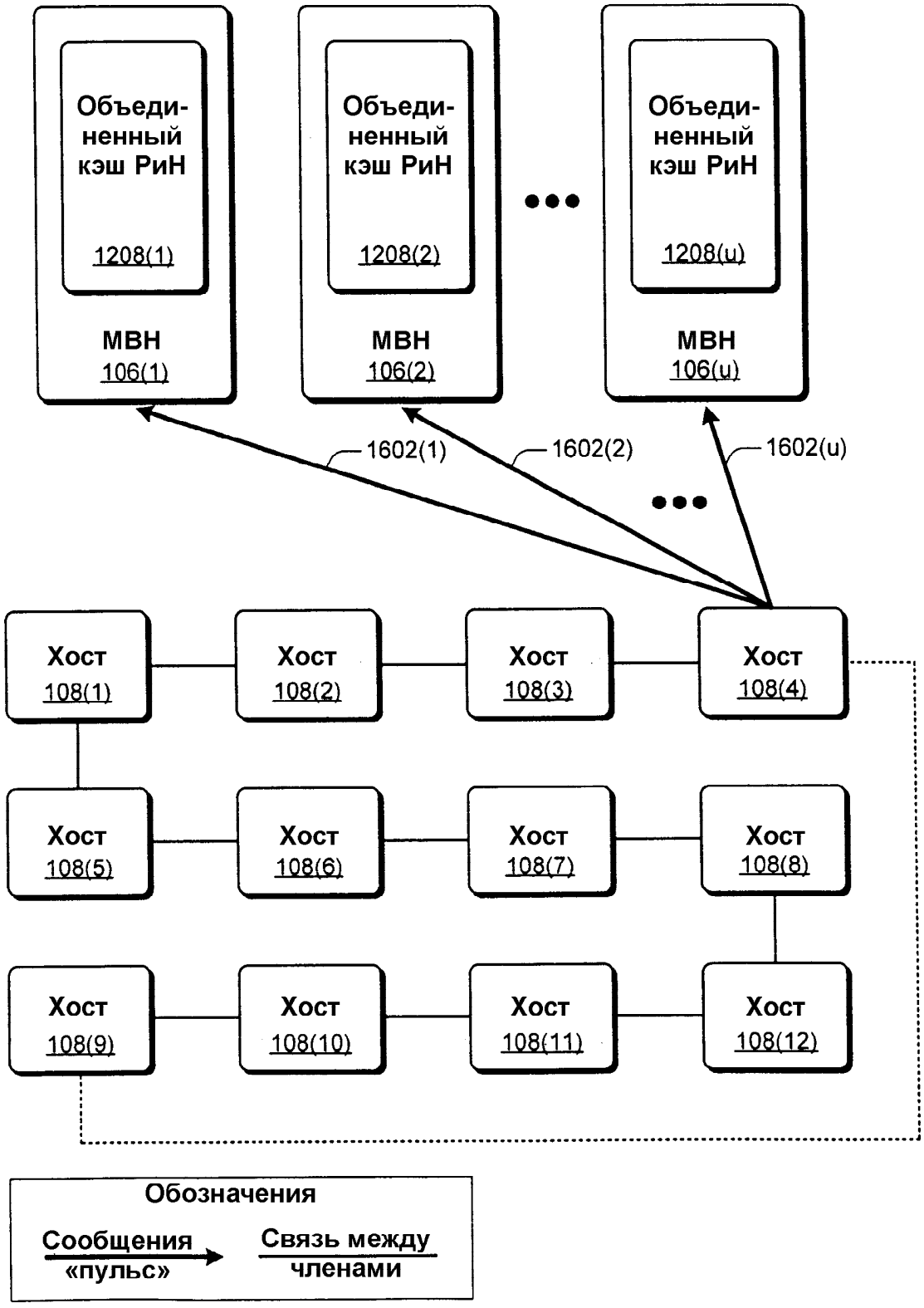
Фиг. 13B


1400
Определить информацию работоспособности и нагрузки
1402
1418
Распространить информацию работоспособности и нагрузки на МВН
1404
Хост
108
МВН
106
Получить информацию работоспособности и нагрузки от хостов
1406
1420
Кэшировать информацию работоспособности и нагрузки
1408
1422
Получить пакет, запрашивающий новое соединение
1410
Свериться с кэшированной информацией работоспособности и нагрузки
1412
1416
Переслать пакет на выбранный хост
Выбрать хост в соответствии с кэшированной информацией работоспособности и нагрузки
1414

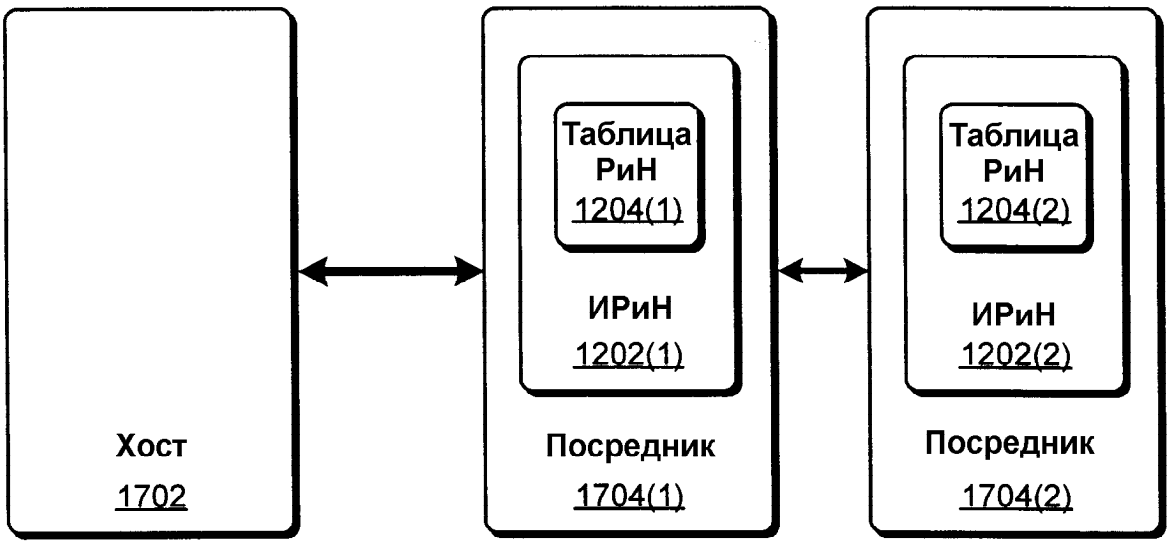
Фиг. 14


ИРиН
1202
Хост
108
Протокол обмена сообщениями
1500
Кэш
1208
МВН
106
1502 «пульс»
1504 «прощание»
1506 «изменение строки»
«получить мгновенный снимок таблицы» 1508
1516
«отправить мгновенный снимок таблицы»
1510
1512 «постулировать состояние таблицы»
1518
«постулировать ошибку» 1514

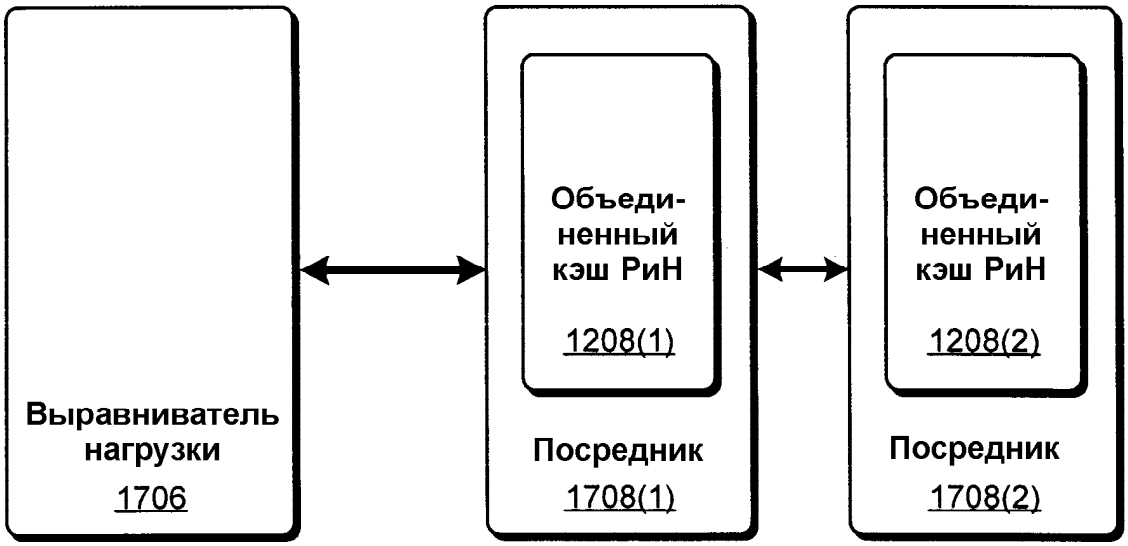
Фиг. 15


Объединенный кэш РиН
1208(1)
МВН
106(1)
Объединенный кэш РиН
1208(2)
МВН
106(2)
Объединенный кэш РиН
1208(u)
МВН
106(u)
1602(1)
1602(2)
1602(u)
Хост 108(1)
Хост 108(2)
Хост 108(3)
Хост 108(4)
Хост 108(5)
Хост 108(6)
Хост 108(7)
Хост 108(8)
Хост 108(9)
Хост 108(10)
Хост 108(11)
Хост 108(12)
Обозначения
Сообщения «пульс»
Связь между членами

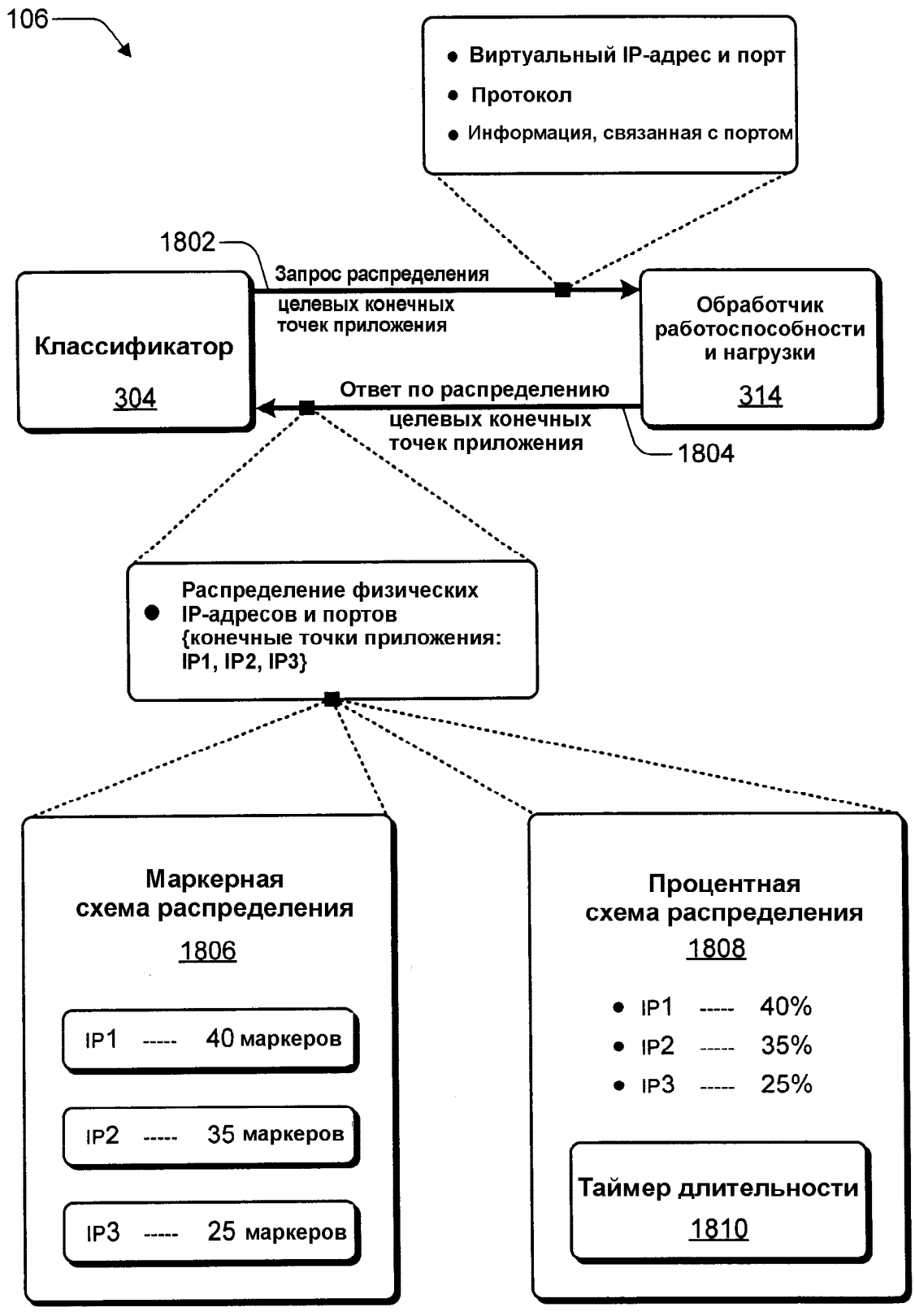
Фиг. 16



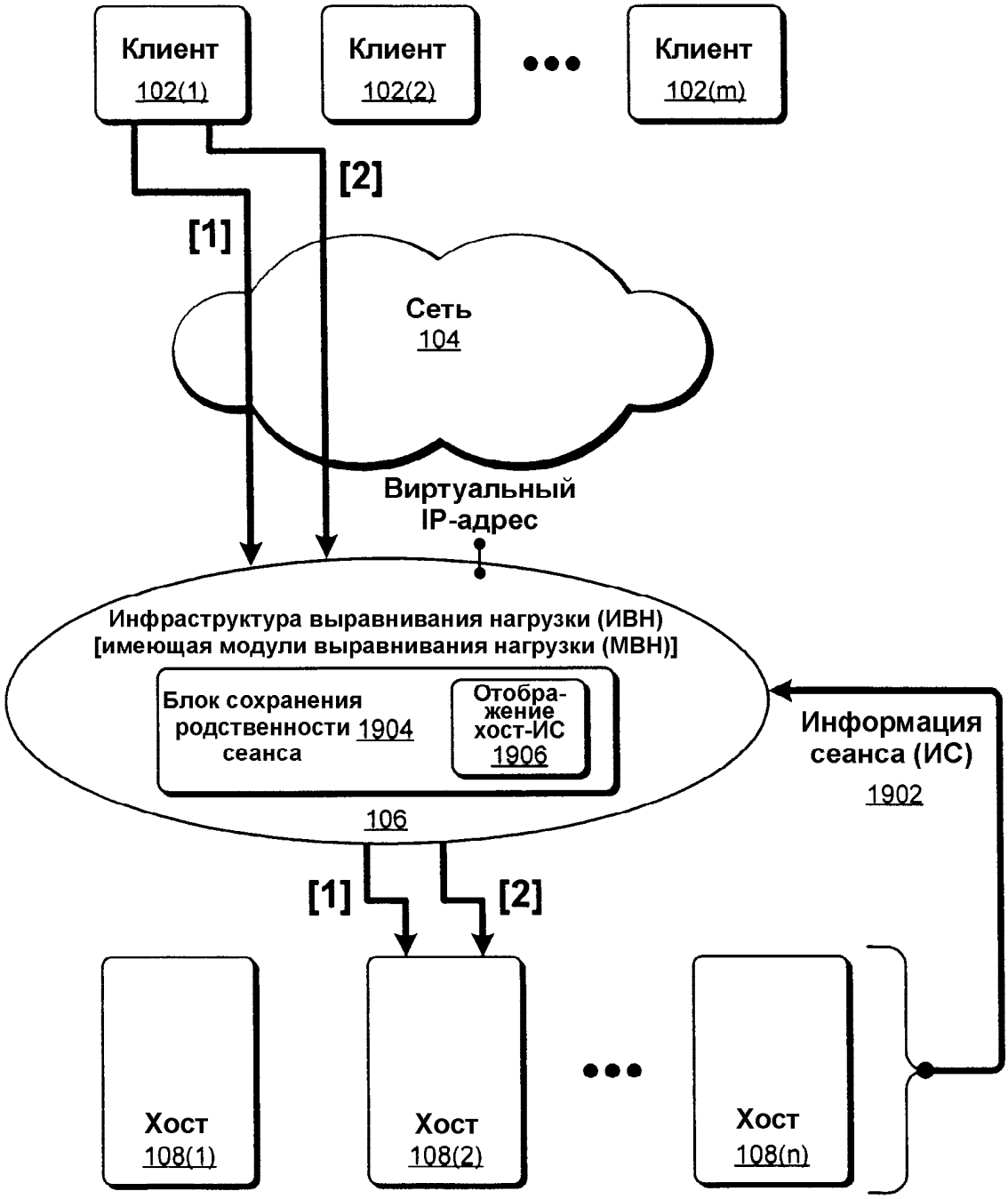
Фиг. 17A



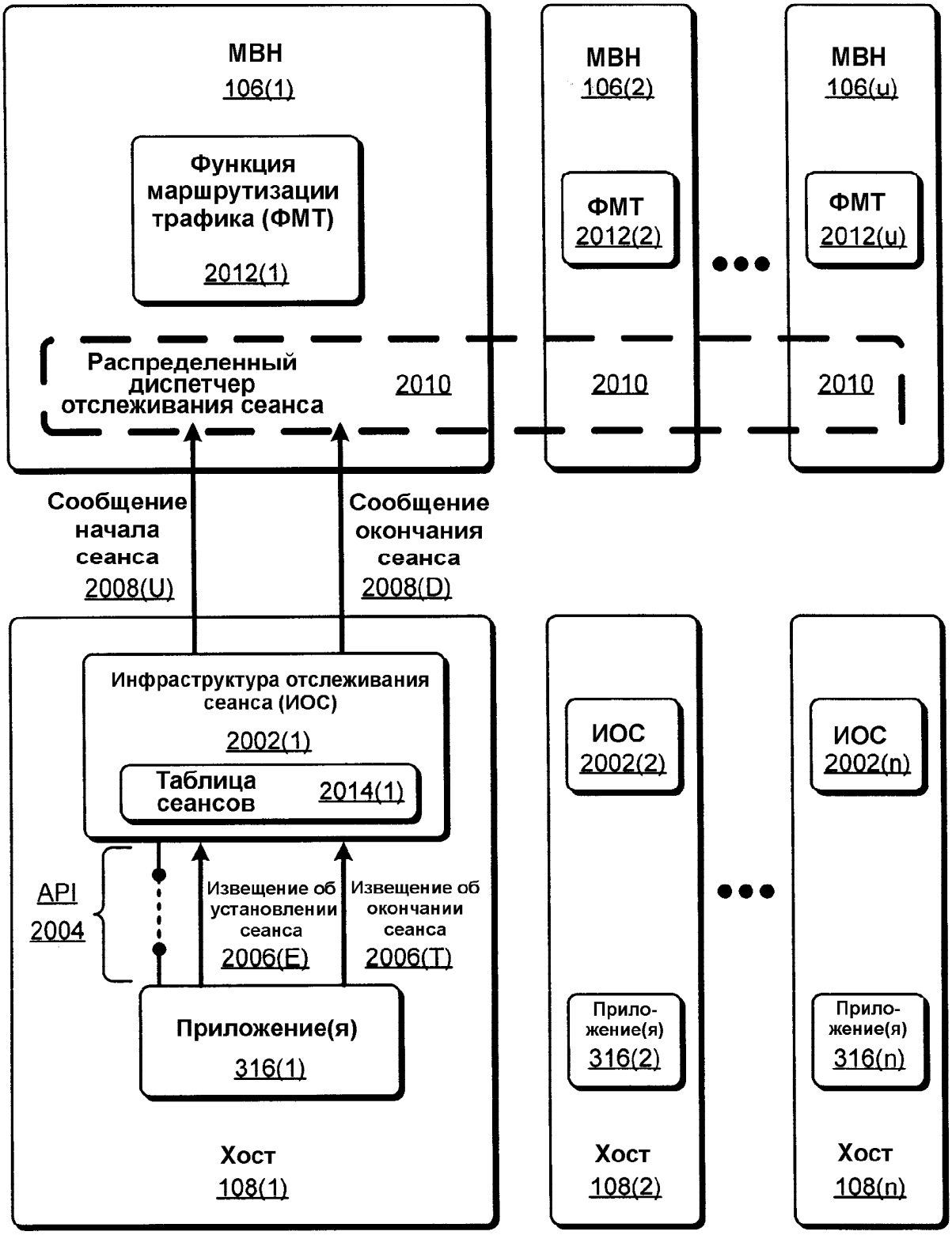
Фиг. 17B


106
• Виртуальный IP-адрес и порт
• Протокол
• Информация, связанная с портом
1802
Запрос распределения целевых конечных точек приложения
Классификатор
304
Обработчик работоспособности и нагрузки
314
Ответ по распределению целевых конечных точек приложения
1804
• Распределение физических IP-адресов и портов {конечные точки приложения: IP1, IP2, IP3}
Маркерная схема распределения
1806
IP1 ----- 40 маркеров
IP2 ----- 35 маркеров
IP3 ----- 25 маркеров
Процентная схема распределения
1808
• IP1 ----- 40%
• IP2 ----- 35%
• IP3 ----- 25%
Таймер длительности
1810

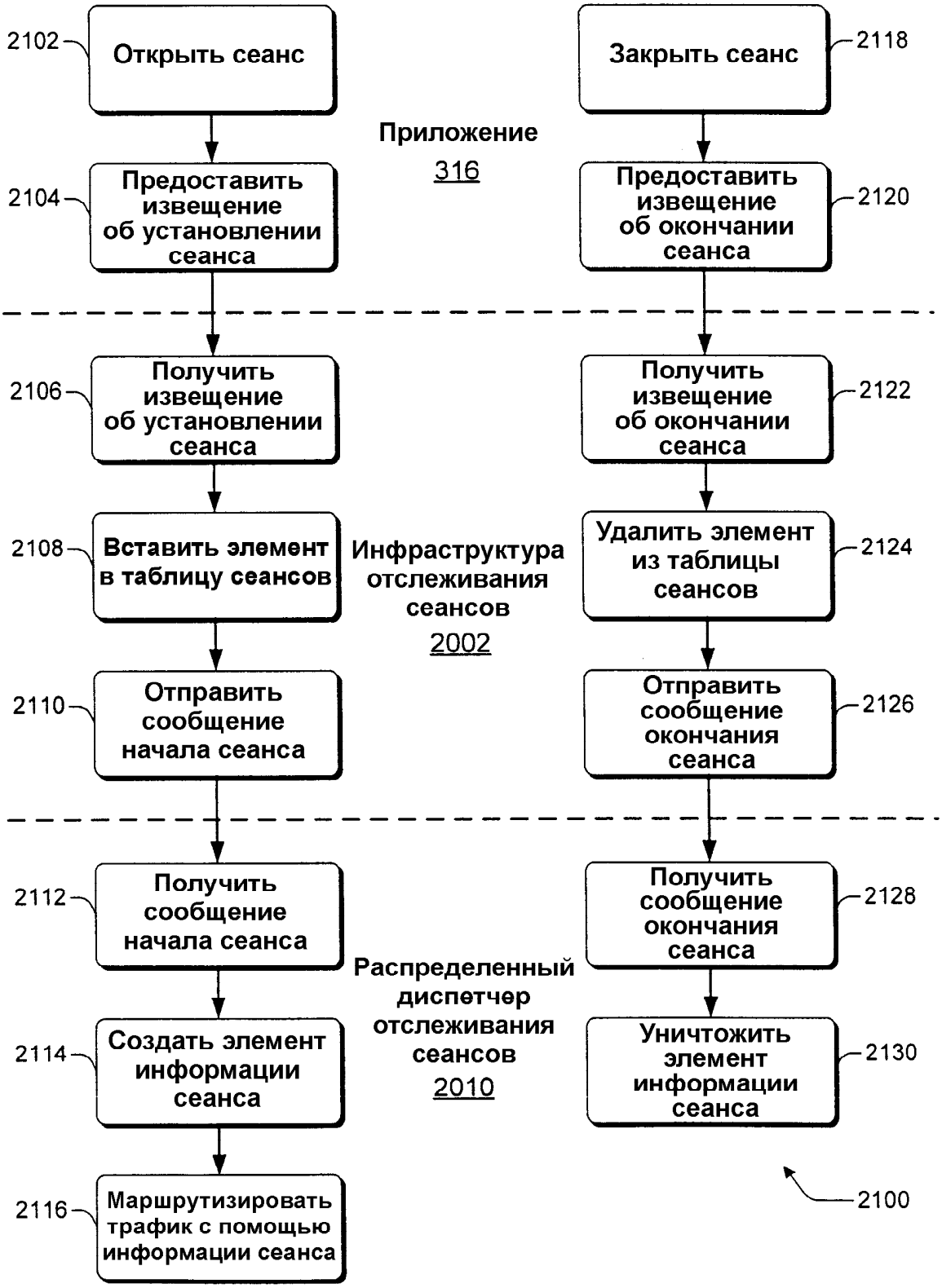
Фиг. 18


Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
[1]
[2]
Сеть
104
Виртуальный
IP-адрес
Инфраструктура выравнивания нагрузки (ИВН)
[имеющая модули выравнивания нагрузки (МВН)]
Блок сохранения родственности сеанса 1904
Отобра-жение хост-ИС 1906
106
Информация сеанса (ИС)
1902
[1]
[2]
Хост
108(1)
Хост
108(2)
Хост
108(n)

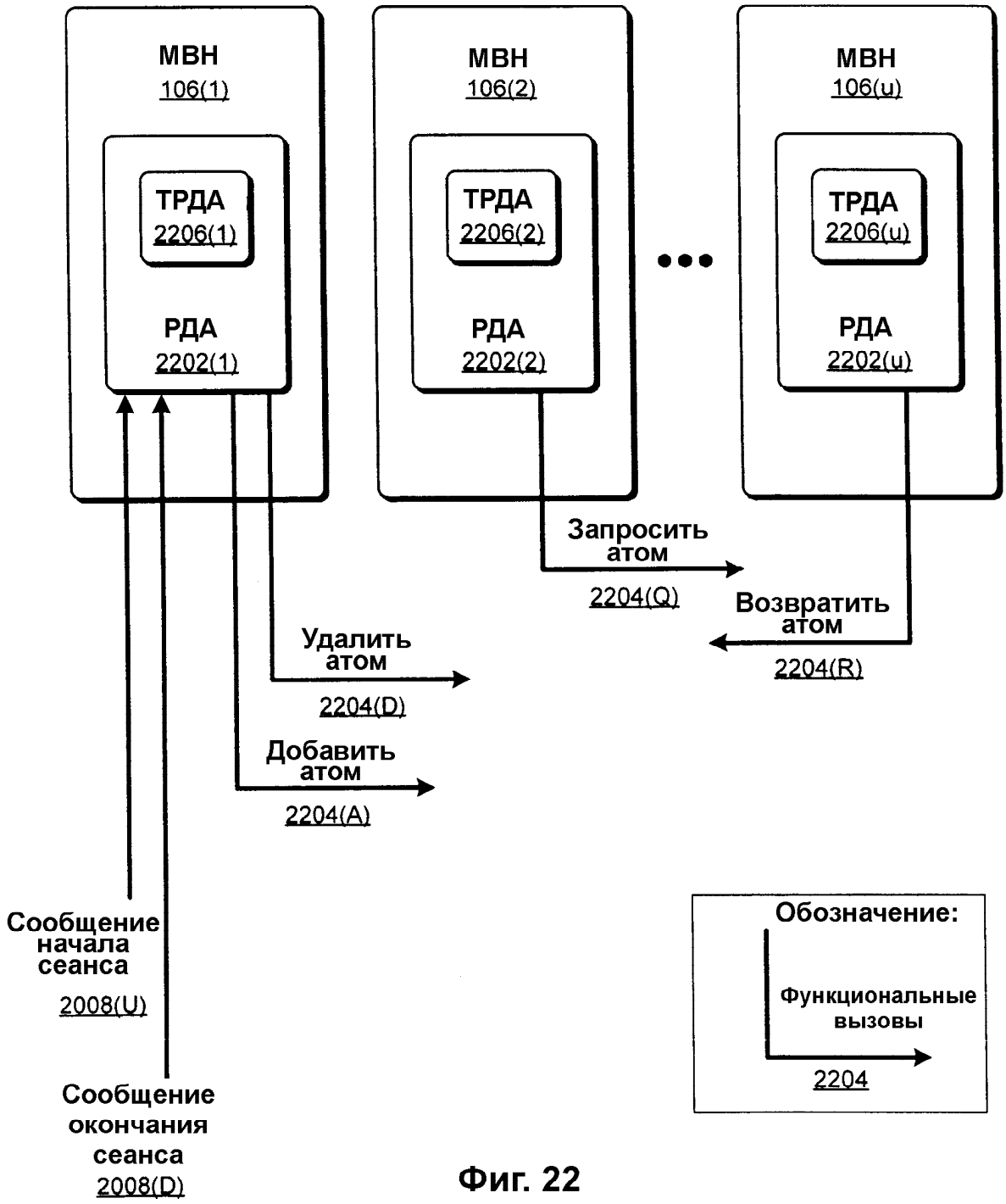
Фиг. 19


МВН
106(1)
Функция маршрутизации трафика (ФМТ)
2012(1)
Распределенный диспетчер отслеживания сеанса
2010
МВН
106(2)
ФМТ
2012(2)
2010
МВН
106(u)
ФМТ
2012(u)
2010
Сообщение начала сеанса
2008(U)
Сообщение окончания сеанса
2008(D)
Инфраструктура отслеживания сеанса (ИОС)
2002(1)
Таблица сеансов
2014(1)
API
2004
Извещение об установлении сеанса
2006(E)
Извещение об окончании сеанса
2006(T)
Приложение(я)
316(1)
Хост
108(1)
ИОС
2002(2)
Прило-жение(я)
316(2)
Хост
108(2)
ИОС
2002(n)
Прило-жение(я)
316(n)
Хост
108(n)

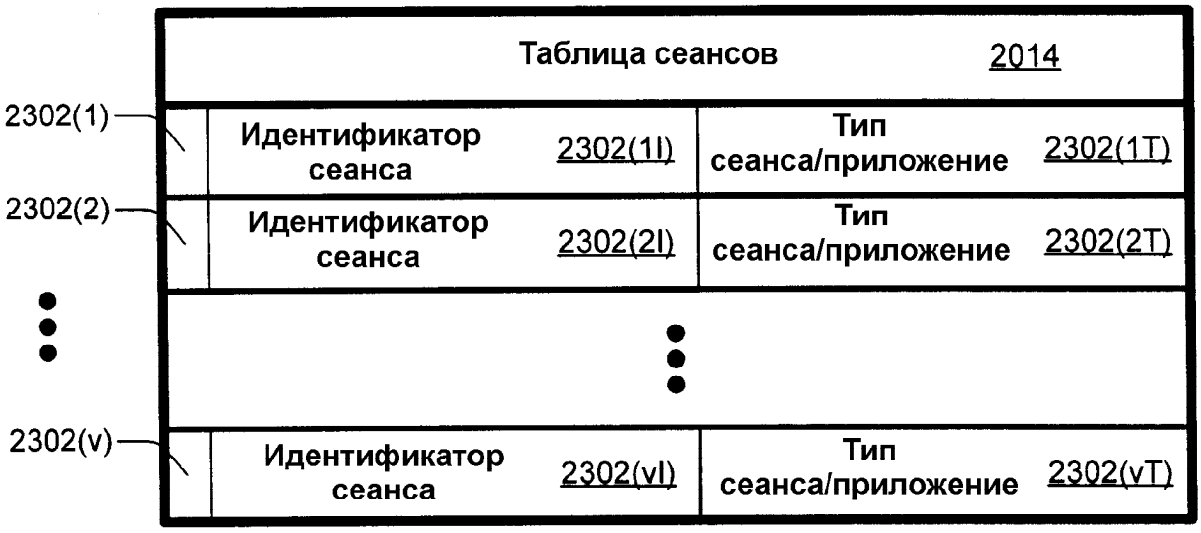
Фиг. 20


2102
Открыть сеанс
2104
Предоставить извещение об установлении сеанса
2106
Получить извещение об установлении сеанса
2108
Вставить элемент в таблицу сеансов
2110
Отправить сообщение начала сеанса
2112
Получить сообщение начала сеанса
2114
Создать элемент информации сеанса
2116
Маршрутизировать трафик с помощью информации сеанса
Приложение 316
Инфраструктура отслеживания сеансов 2002
Распределенный диспетчер отслеживания сеансов 2010
2118
Закрыть сеанс
2120
Предоставить извещение об окончании сеанса
2122
Получить извещение об окончании сеанса
2124
Удалить элемент из таблицы сеансов
2126
Отправить сообщение окончания сеанса
2128
Получить сообщение окончания сеанса
2130
Уничтожить элемент информации сеанса
2100

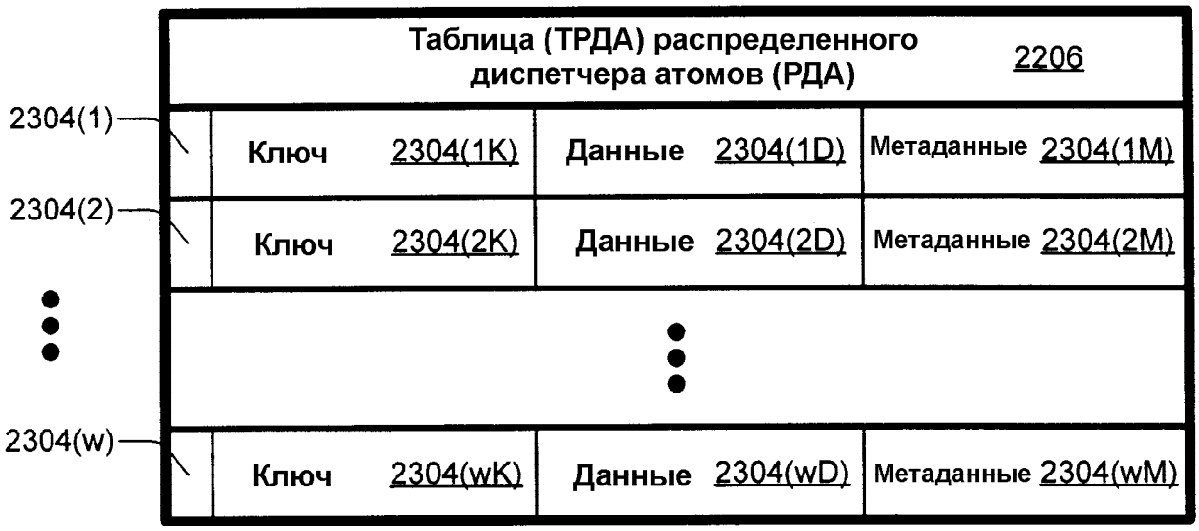
Фиг. 21


МВН
106(1)
ТРДА
2206(1)
РДА
2202(1)
МВН
106(2)
ТРДА
2206(2)
РДА
2202(2)
МВН
106(u)
ТРДА
2206(u)
РДА
2202(u)
Запросить атом
2204(Q)
Возвратить атом
2204(R)
Удалить атом
2204(D)
Добавить атом
2204(A)
Сообщение начала сеанса
2008(U)
Сообщение окончания сеанса
2008(D)
Обозначение:
Функциональные вызовы
2204

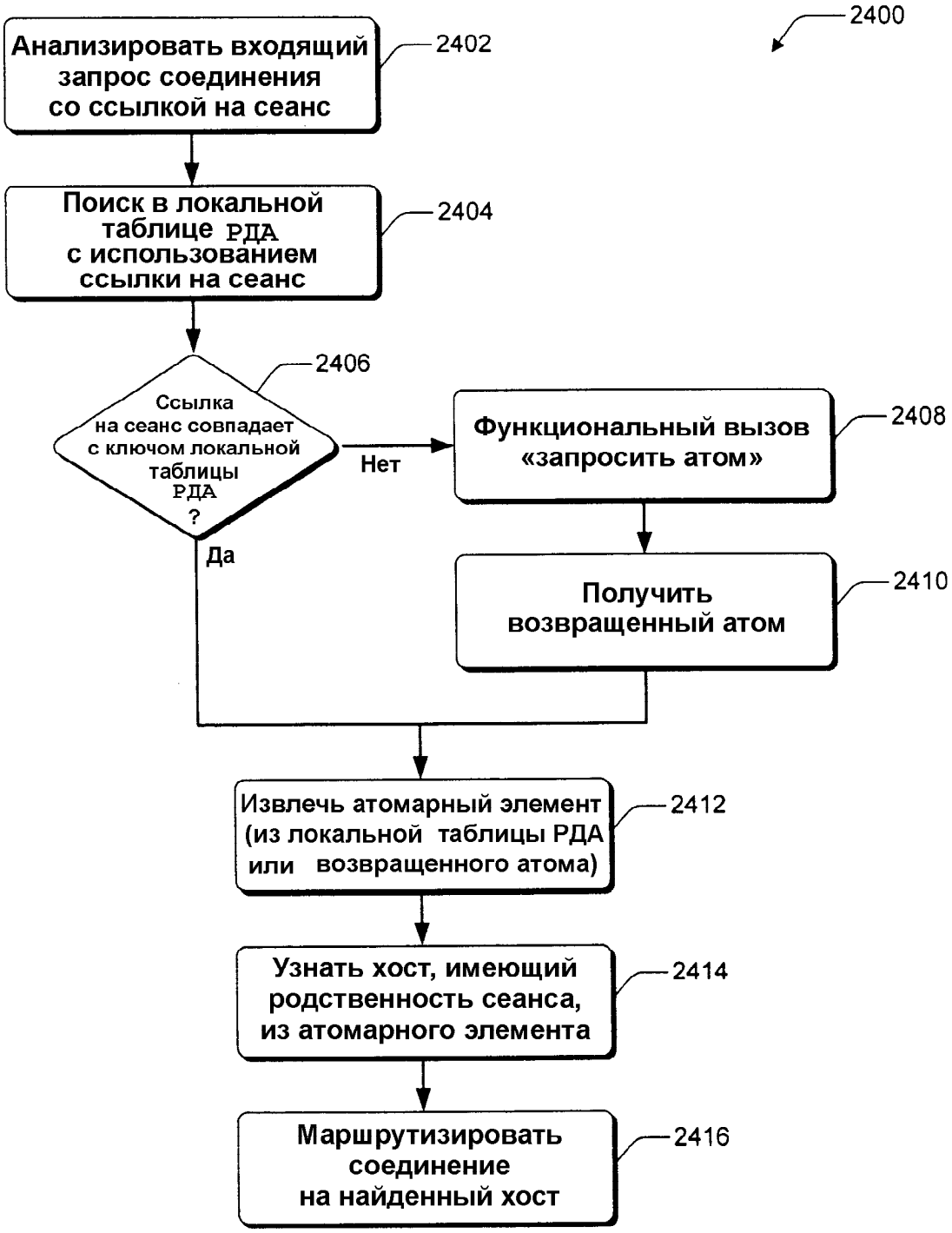
Фиг. 22



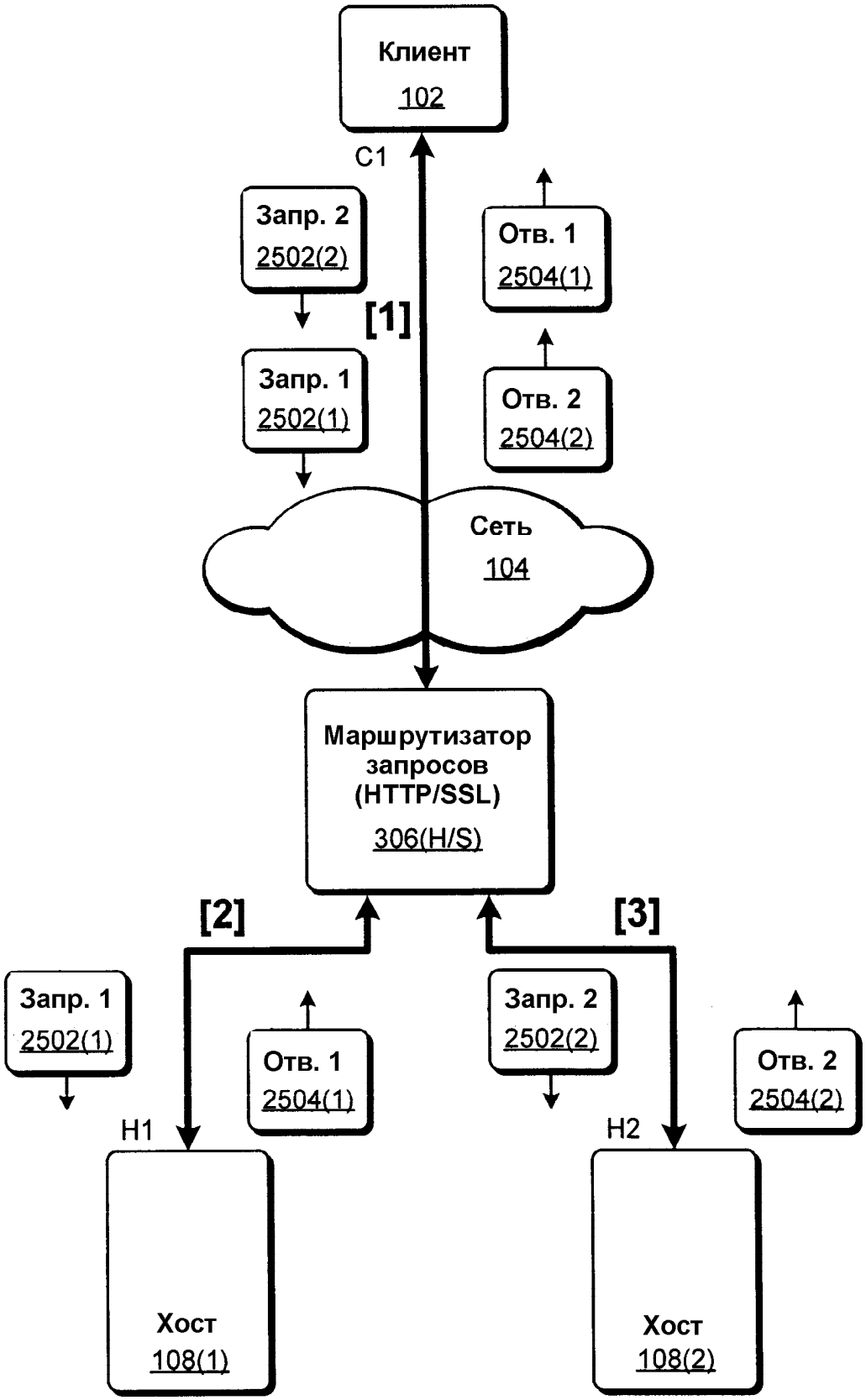
Фиг. 23A



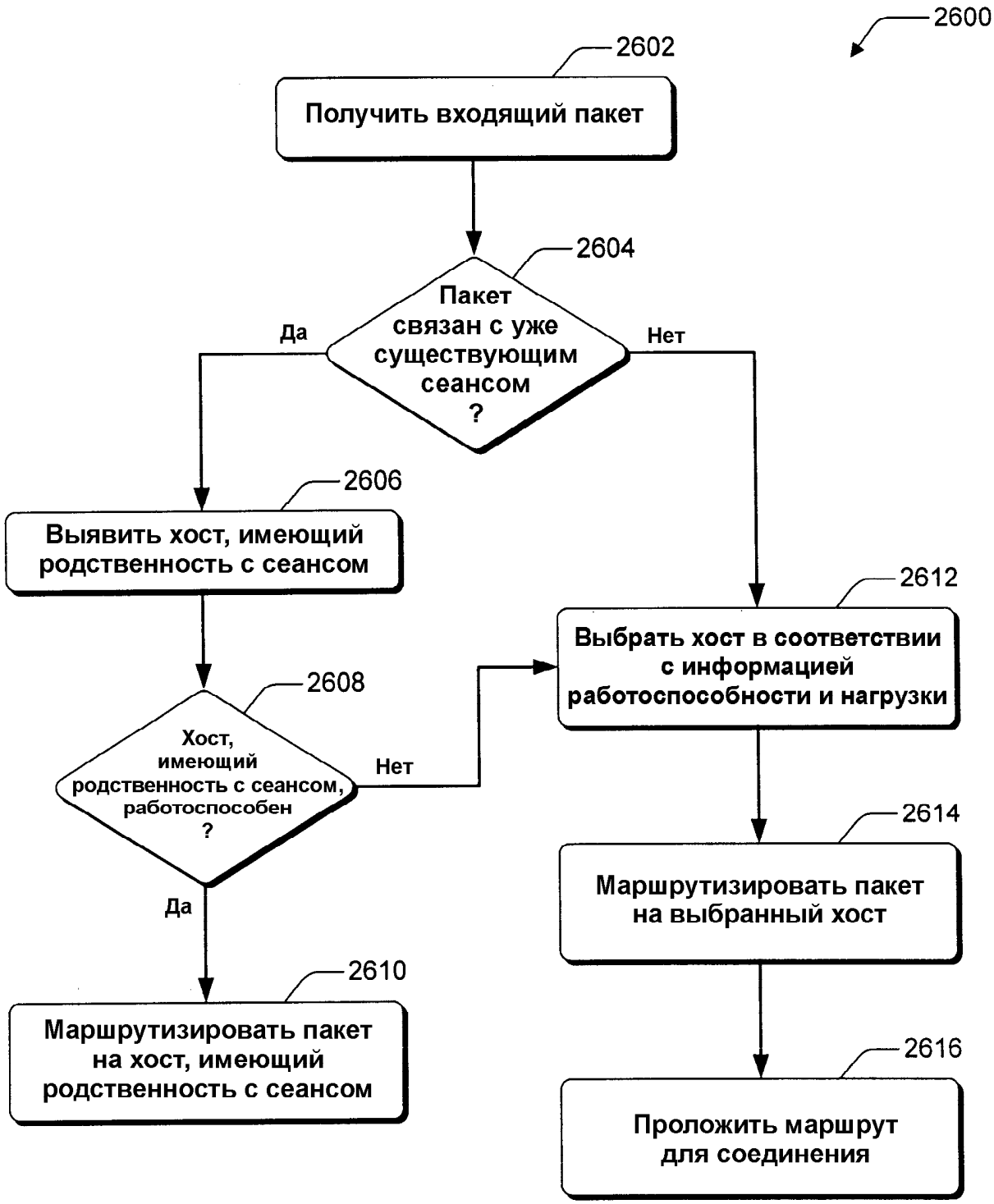
Фиг. 23B


2400
Анализировать входящий запрос соединения со ссылкой на сеанс
2402
Поиск в локальной таблице РДА с использованием ссылки на сеанс
2404
Ссылка на сеанс совпадает с ключом локальной таблицы РДА ?
2406
Нет
Да
Функциональный вызов «запросить атом»
2408
Получить возвращенный атом
2410
Извлечь атомарный элемент (из локальной таблицы РДА или возвращенного атома)
2412
Узнать хост, имеющий родственность сеанса, из атомарного элемента
2414
Маршрутизировать соединение на найденный хост
2416

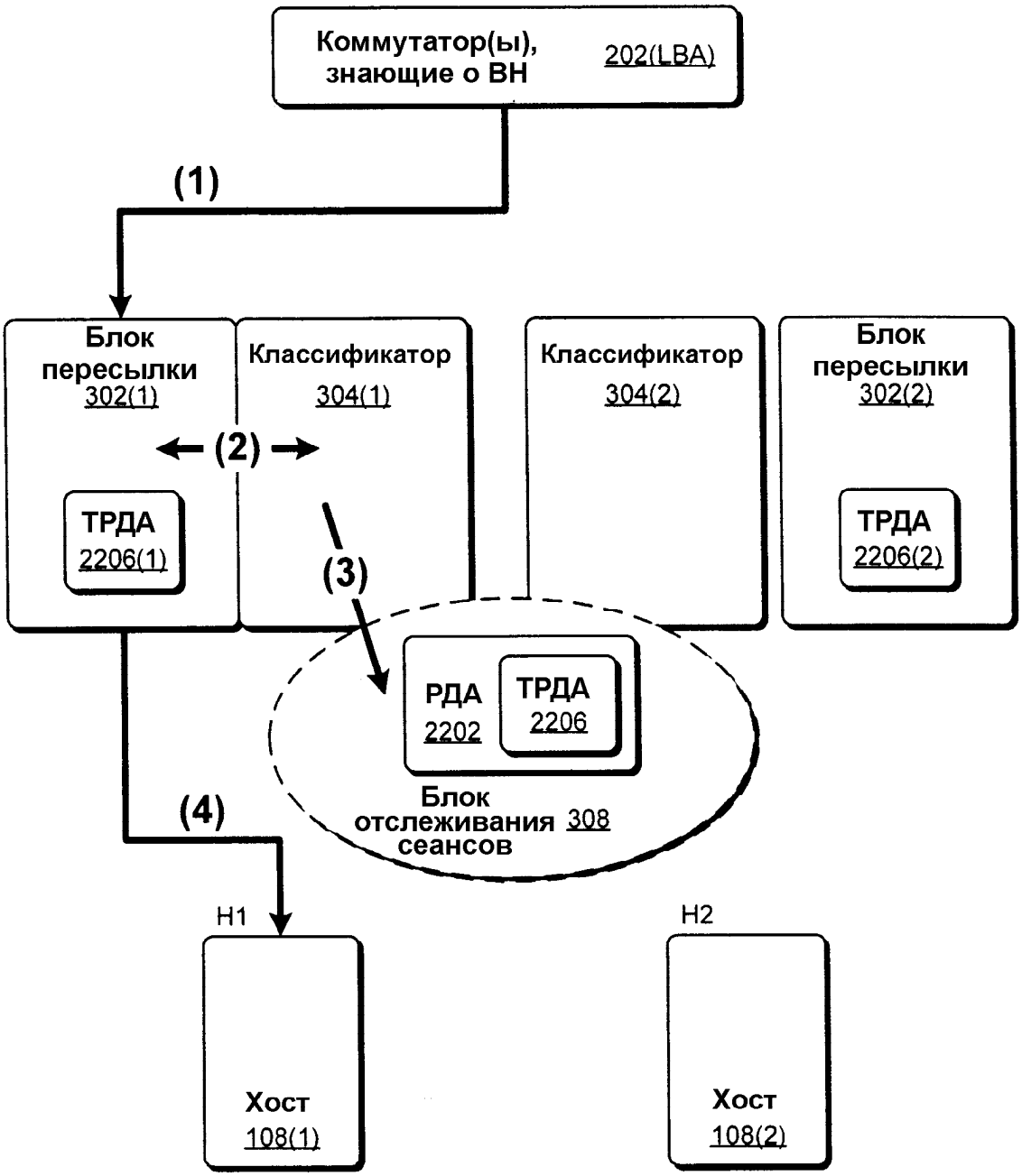
Фиг. 24


Клиент
102
C1
Запр. 2
2502(2)
[1]
Отв. 1
2504(1)
Запр. 1
2502(1)
Отв. 2
2504(2)
Сеть
104
Маршрутизатор запросов (HTTP/SSL)
306(H/S)
[2]
[3]
Запр. 1
2502(1)
Отв. 1
2504(1)
Запр. 2
2502(2)
Отв. 2
2504(2)
H1
H2
Хост
108(1)
Хост
108(2)

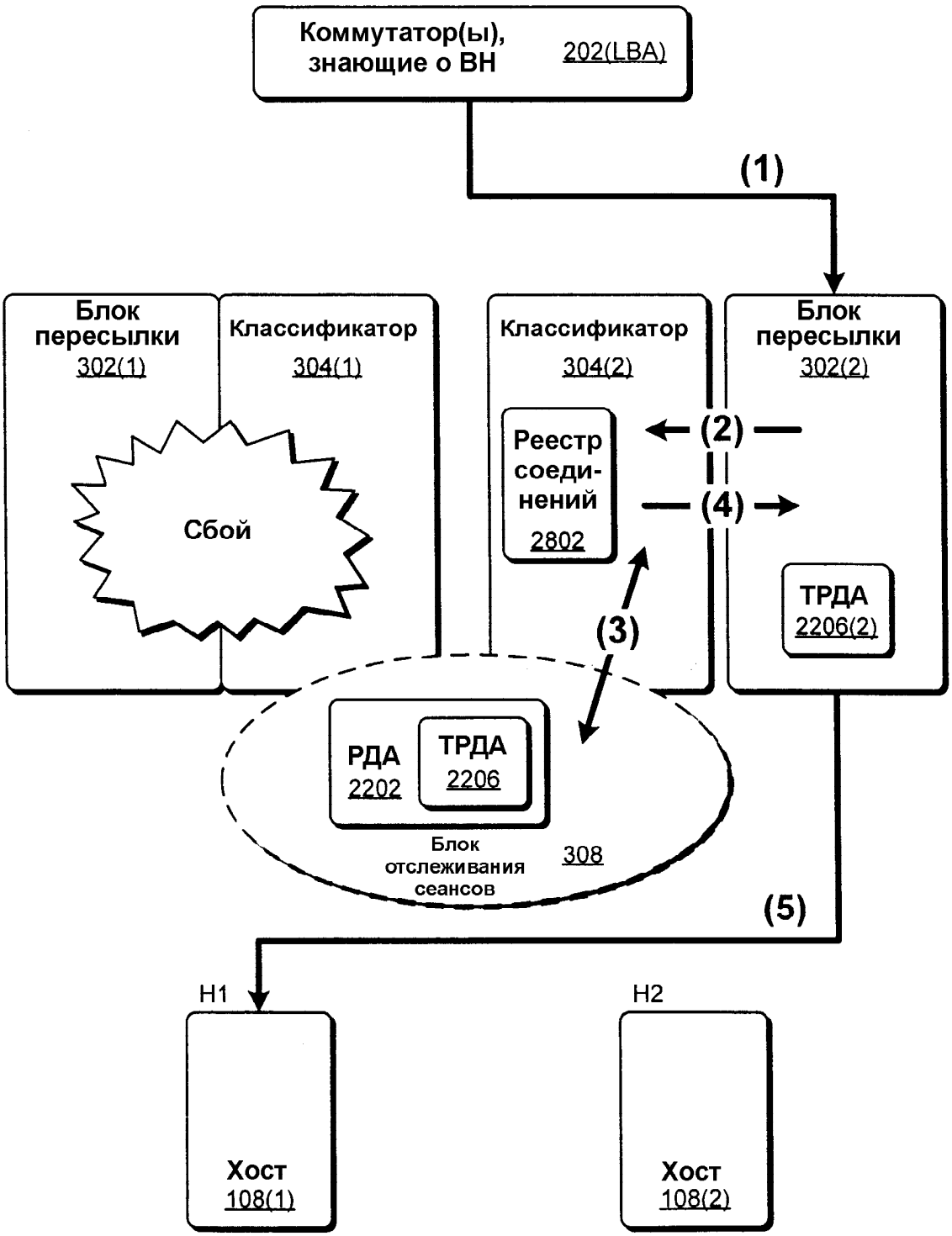
Фиг. 25


2600
2602
Получить входящий пакет
2604
Пакет связан с уже существующим сеансом ?
Да
Нет
2606
Выявить хост, имеющий родственность с сеансом
2608
Хост, имеющий родственность с сеансом, работоспособен ?
Нет
Да
2610
Маршрутизировать пакет на хост, имеющий родственность с сеансом
2612
Выбрать хост в соответствии с информацией работоспособности и нагрузки
2614
Маршрутизировать пакет на выбранный хост
2616
Проложить маршрут для соединения

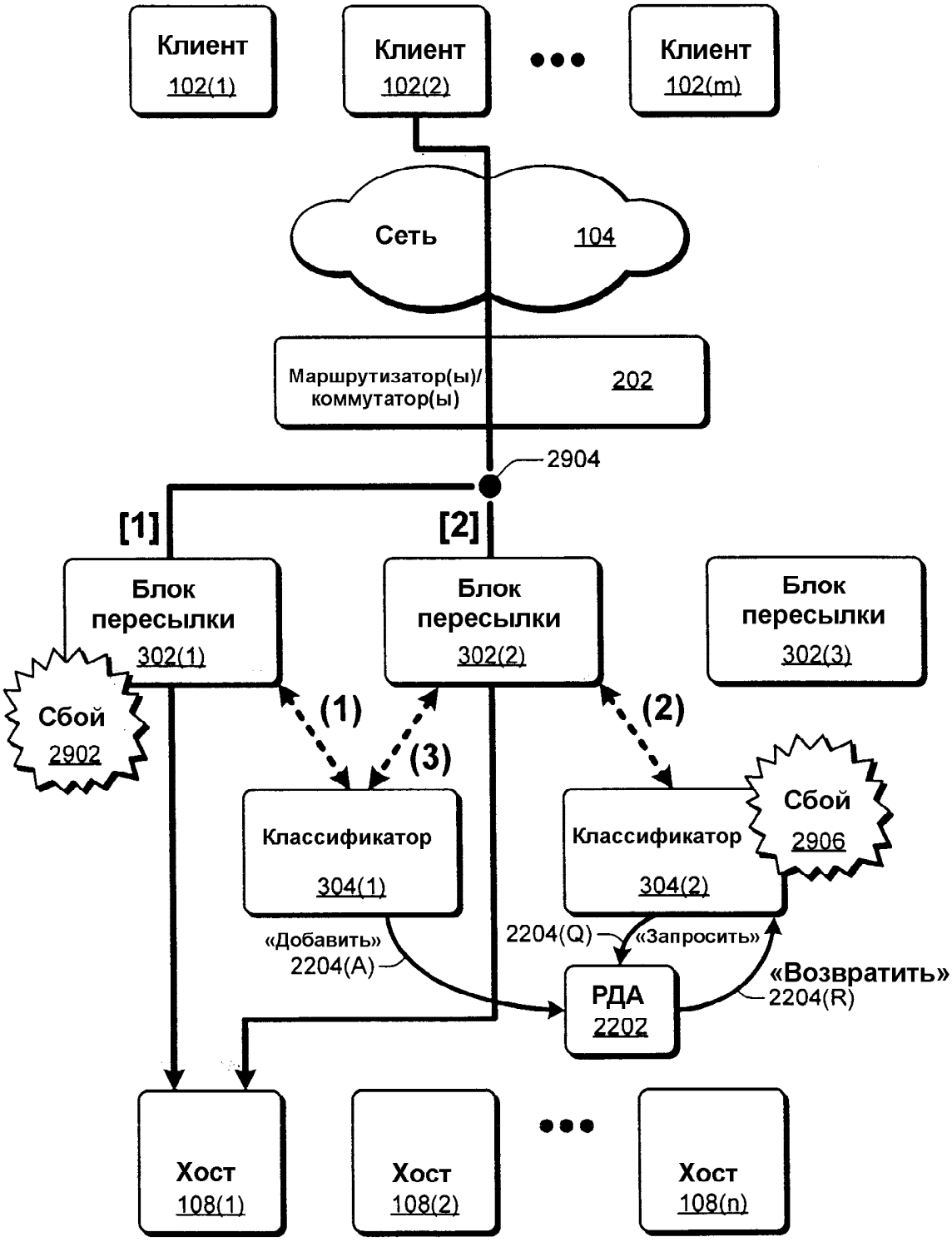
Фиг. 26


Коммутатор(ы), знающие о ВН
202(LBA)
(1)
Блок пересылки 302(1)
Классификатор 304(1)
Классификатор 304(2)
Блок пересылки 302(2)
(2)
ТРДА 2206(1)
(3)
ТРДА 2206(2)
РДА 2202
ТРДА 2206
Блок отслеживания 308 сеансов
(4)
H1
H2
Хост 108(1)
Хост 108(2)

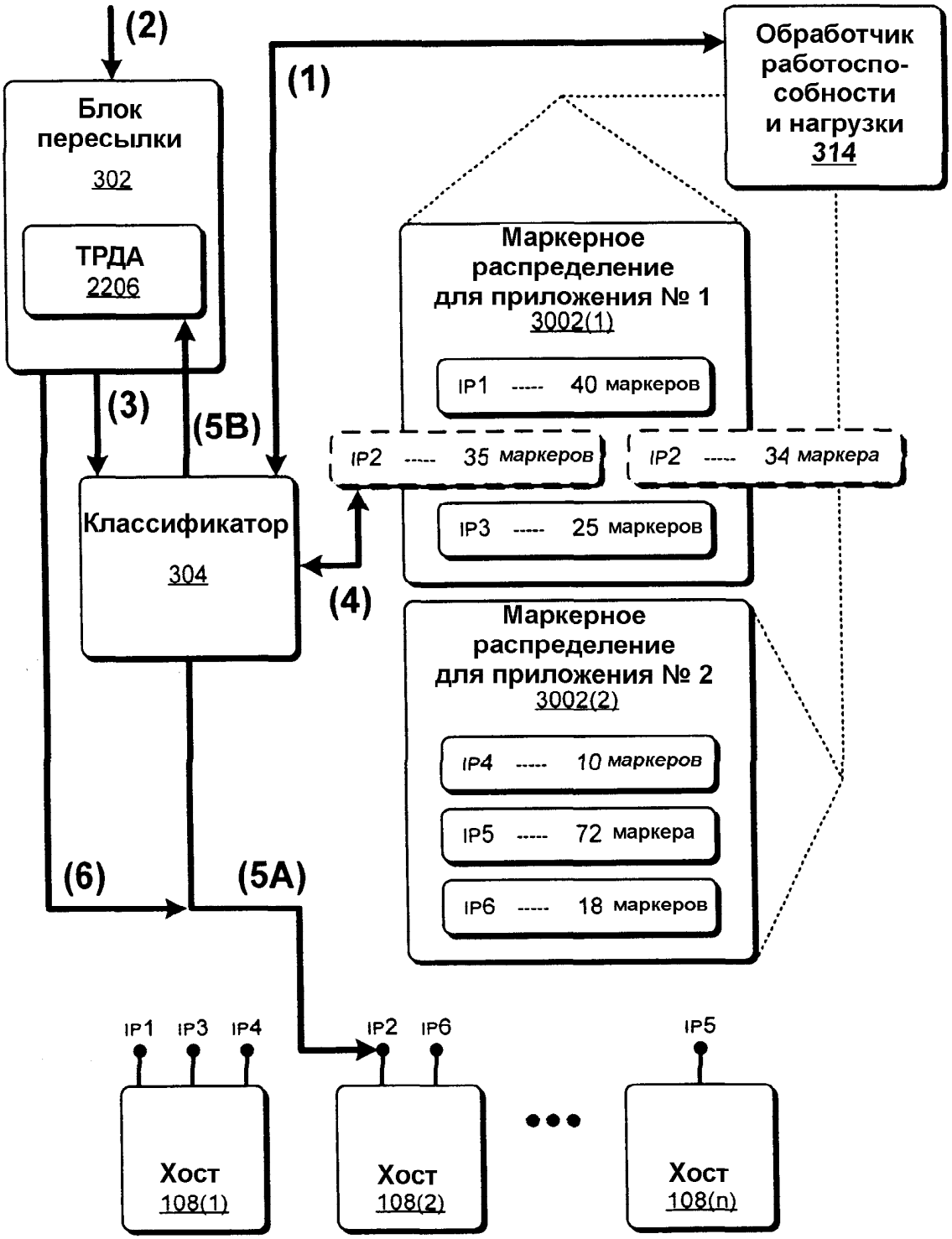
Фиг. 27


Коммутатор(ы), знающие о ВН
202(LBA)
(1)
Блок пересылки
302(1)
Классификатор
304(1)
Сбой
Классификатор
304(2)
Блок пересылки
302(2)
Реестр соединений
2802
(2)
(4)
ТРДА
2206(2)
(3)
РДА
2202
ТРДА
2206
Блок отслеживания сеансов
308
(5)
H1
H2
Хост
108(1)
Хост
108(2)

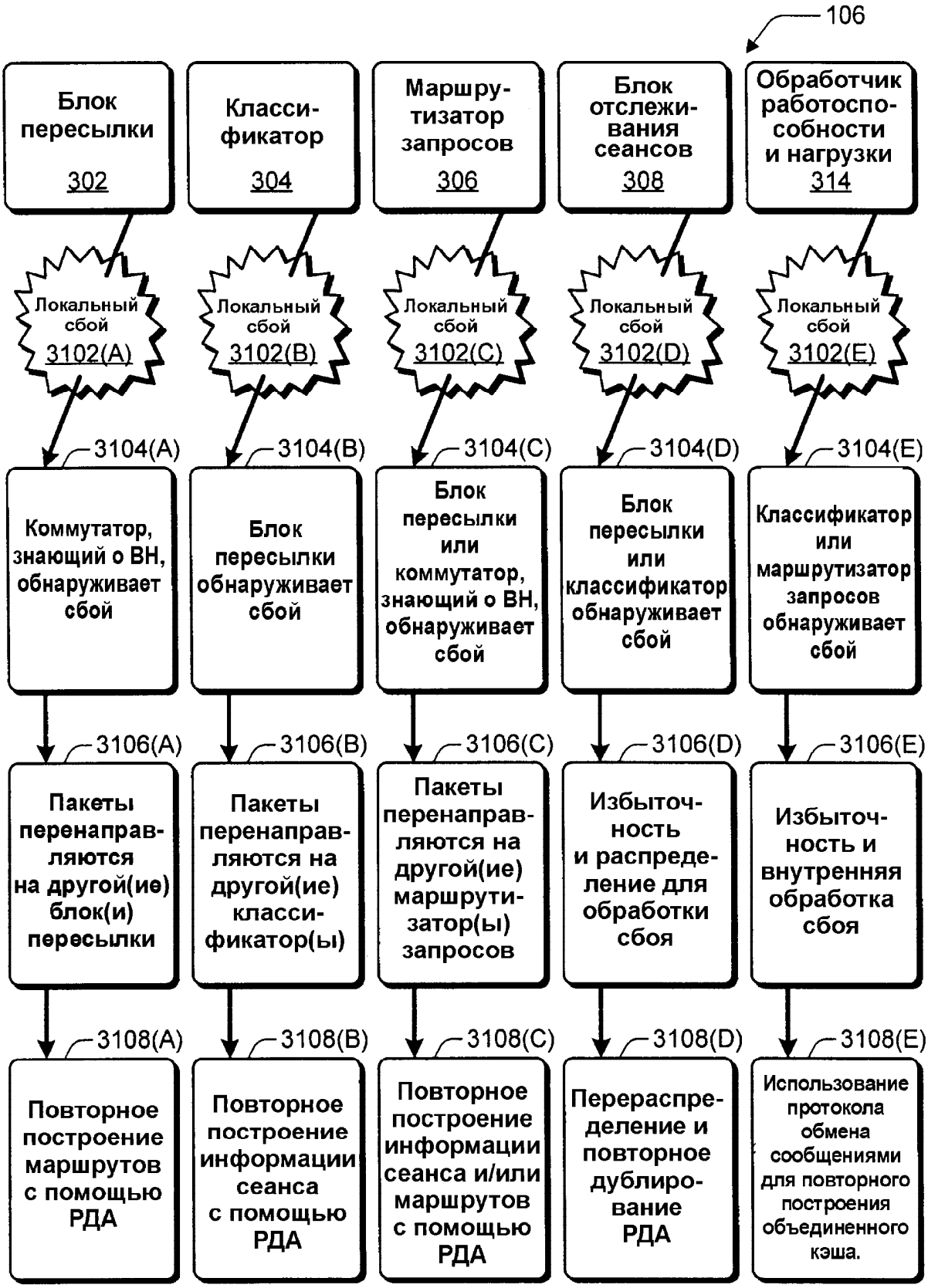
Фиг. 28


Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
Сеть
104
Маршрутизатор(ы)/
коммутатор(ы)
202
2904
[1]
[2]
Блок
пересылки
302(1)
Блок
пересылки
302(2)
Блок
пересылки
302(3)
Сбой
2902
(1)
(2)
(3)
Классификатор
304(1)
Классификатор
304(2)
Сбой
2906
«Добавить»
2204(A)
2204(Q)
«Запросить»
«Возвратить»
2204(R)
РДА
2202
Хост
108(1)
Хост
108(2)
Хост
108(n)

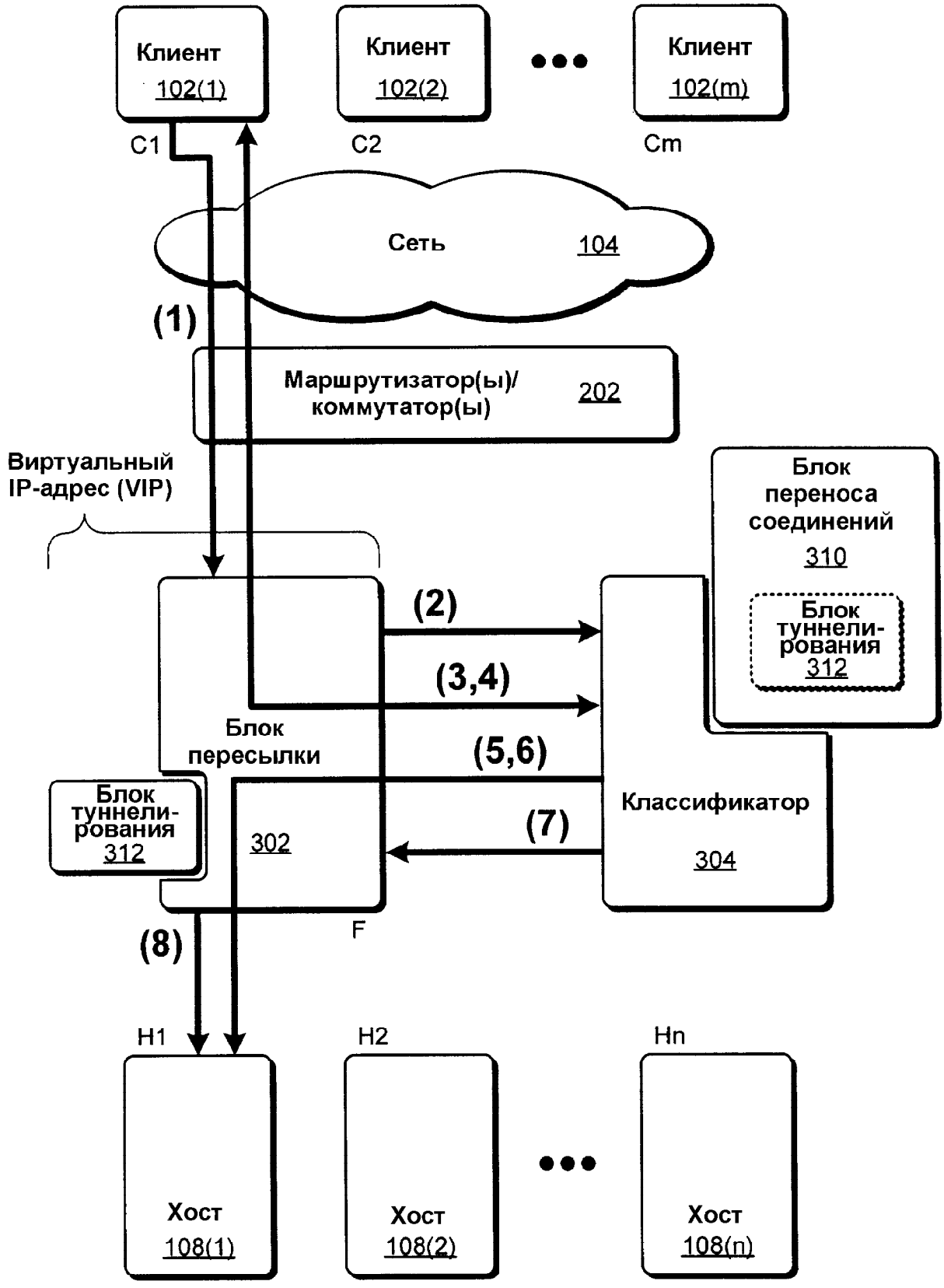
Фиг. 29


(2)
(1)
Обработчик работоспо-собности и нагрузки 314
Блок пересылки 302
ТРДА 2206
Маркерное распределение для приложения № 1 3002(1)
IP1 ----- 40 маркеров
IP2 ----- 35 маркеров
IP2 ----- 34 маркера
IP3 ----- 25 маркеров
(3)
(5B)
Классификатор 304
(4)
Маркерное распределение для приложения № 2 3002(2)
IP4 ----- 10 маркеров
IP5 ----- 72 маркера
IP6 ----- 18 маркеров
(6)
(5A)
IP1 IP3 IP4
IP2 IP6
IP5
Хост 108(1)
Хост 108(2)
Хост 108(n)

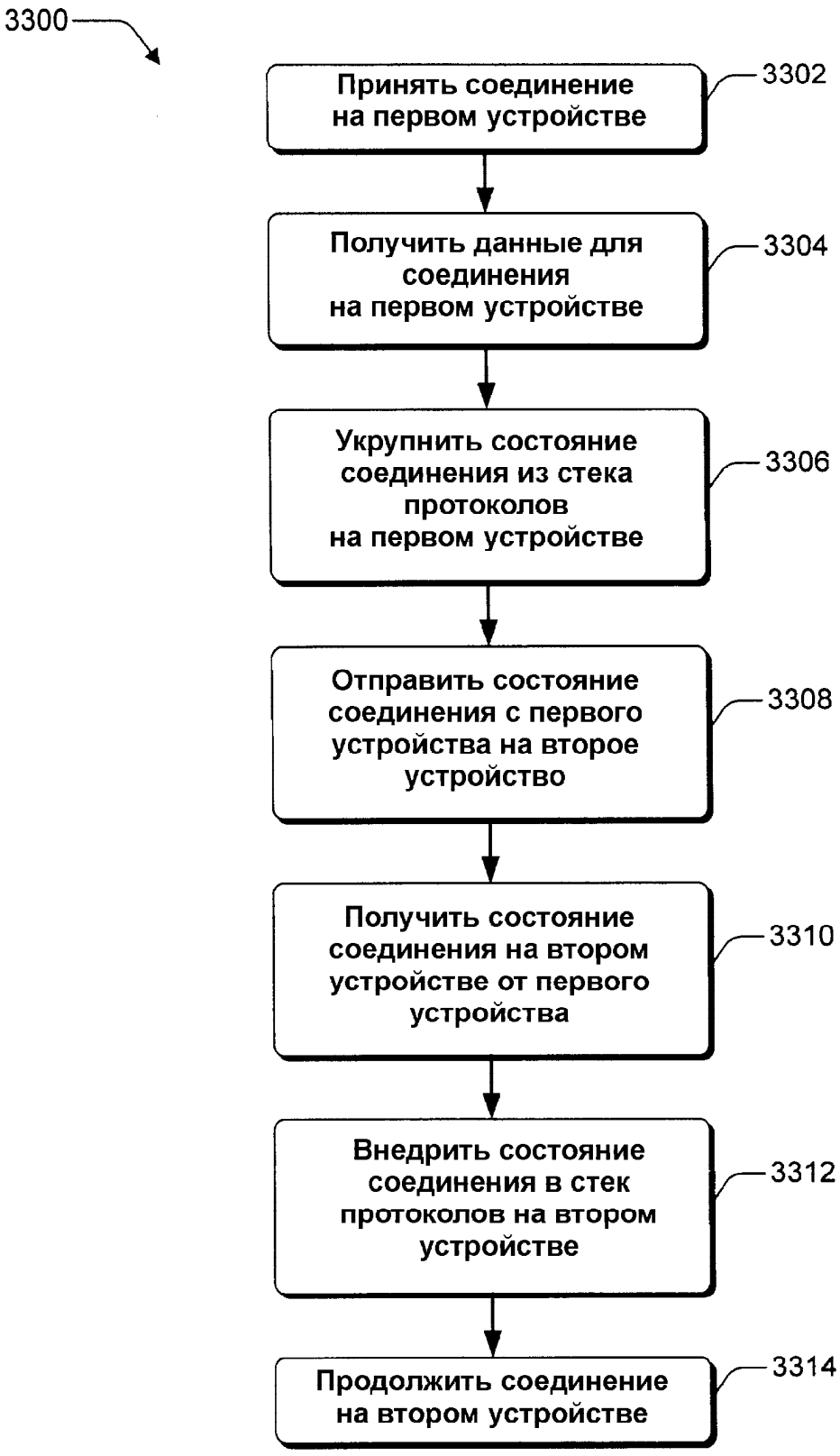
Фиг. 30


106
Блок пересылки 302
Классификатор 304
Маршрутизатор запросов 306
Блок отслеживания сеансов 308
Обработчик работоспособности и нагрузки 314
Локальный сбой 3102(A)
Локальный сбой 3102(B)
Локальный сбой 3102(C)
Локальный сбой 3102(D)
Локальный сбой 3102(E)
3104(A)
3104(B)
3104(C)
3104(D)
3104(E)
Коммутатор, знающий о ВН, обнаруживает сбой
Блок пересылки обнаруживает сбой
Блок пересылки или коммутатор, знающий о ВН, обнаруживает сбой
Блок пересылки или классификатор обнаруживает сбой
Классификатор или маршрутизатор запросов обнаруживает сбой
3106(A)
3106(B)
3106(C)
3106(D)
3106(E)
Пакеты перенаправляются на другой(ие) блок(и) пересылки
Пакеты перенаправляются на другой(ие) классификатор(ы)
Пакеты перенаправляются на другой(ие) маршрутизатор(ы) запросов
Избыточность и распределение для обработки сбоя
Избыточность и внутренняя обработка сбоя
3108(A)
3108(B)
3108(C)
3108(D)
3108(E)
Повторное построение маршрутов с помощью РДА
Повторное построение информации сеанса с помощью РДА
Повторное построение информации сеанса и/или маршрутов с помощью РДА
Перераспределение и повторное дублирование РДА
Использование протокола обмена сообщениями для повторного построения объединенного кэша.

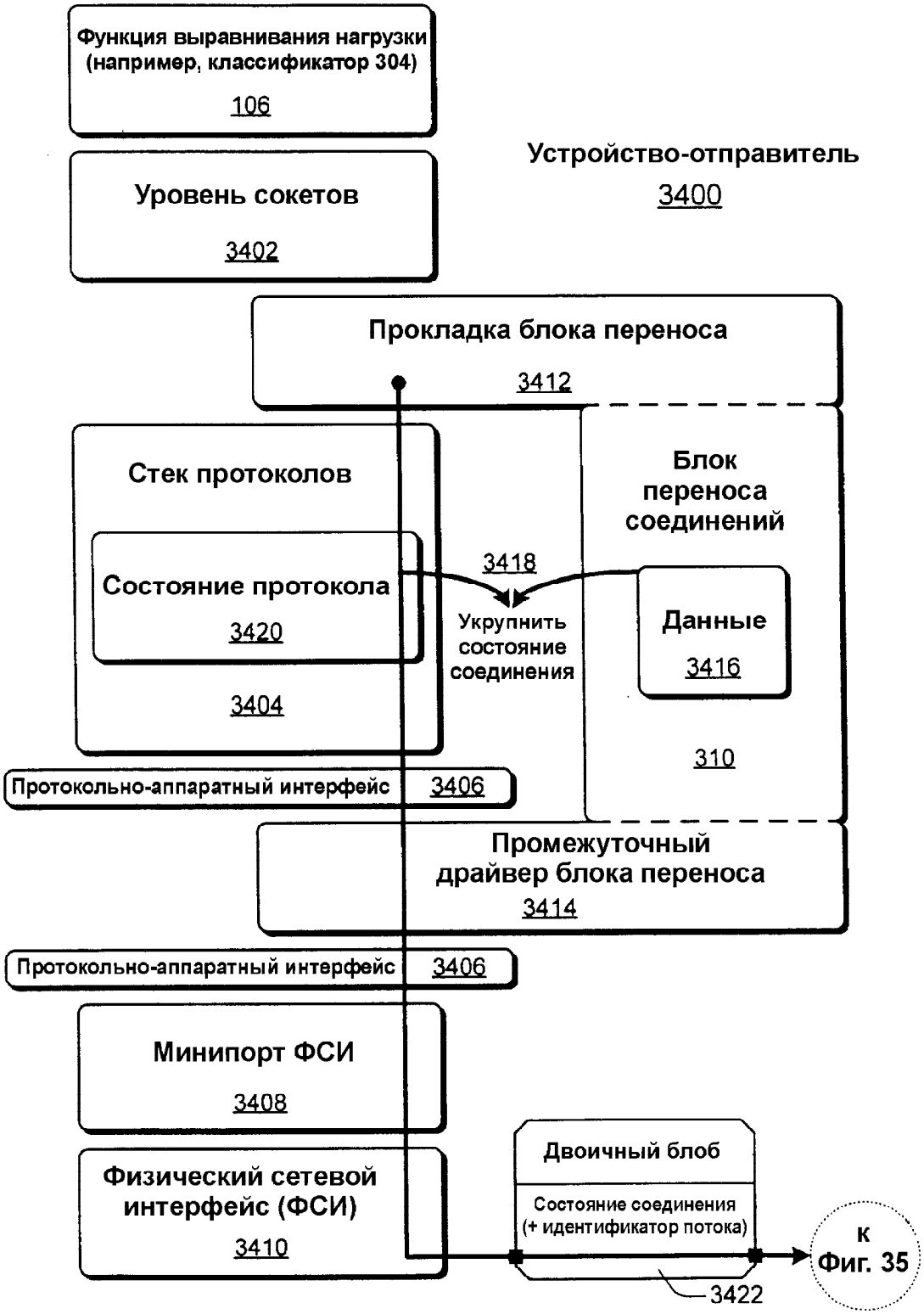
Фиг. 31


Клиент
102(1)
Клиент
102(2)
Клиент
102(m)
C1
C2
Cm
Сеть
104
(1)
Маршрутизатор(ы)/
коммутатор(ы)
202
Виртуальный
IP-адрес (VIP)
Блок
переноса
соединений
310
Блок
туннели-
рования
312
(2)
(3,4)
Блок
пересылки
(5,6)
Блок
туннели-
рования
312
302
Классификатор
304
(7)
F
(8)
H1
H2
Hn
Хост
108(1)
Хост
108(2)
Хост
108(n)

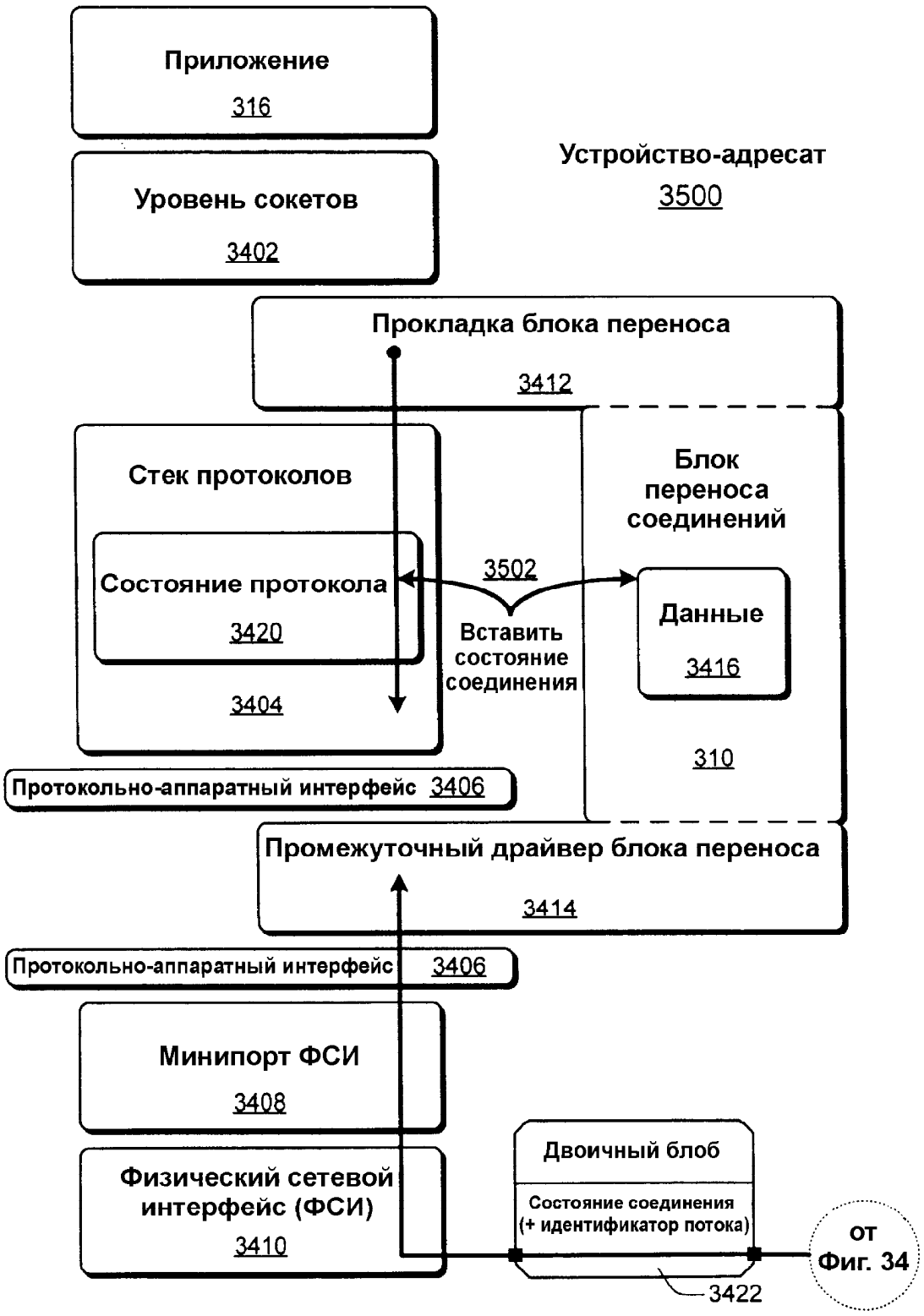
Фиг. 32


3300
Принять соединение на первом устройстве
3302
Получить данные для соединения на первом устройстве
3304
Укрупнить состояние соединения из стека протоколов на первом устройстве
3306
Отправить состояние соединения с первого устройства на второе устройство
3308
Получить состояние соединения на втором устройстве от первого устройства
3310
Внедрить состояние соединения в стек протоколов на втором устройстве
3312
Продолжить соединение на втором устройстве
3314

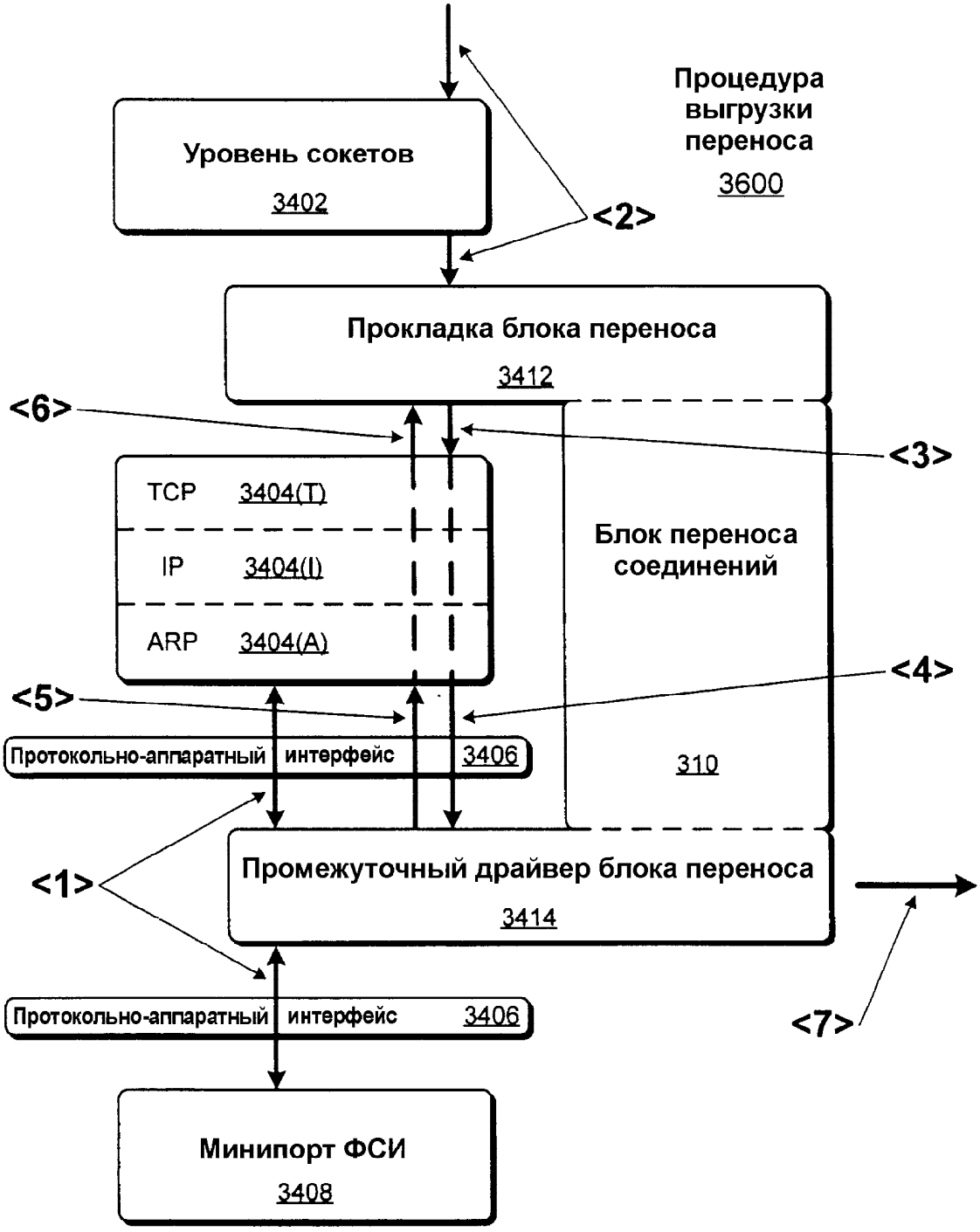
Фиг. 33


Функция выравнивания нагрузки (например, классификатор 304)
106
Уровень сокетов
3402
Устройство-отправитель
3400
Прокладка блока переноса
3412
Стек протоколов
Состояние протокола
3420
3404
3418
Укрупнить состояние соединения
Блок переноса соединений
Данные
3416
310
Протокольно-аппаратный интерфейс
3406
Промежуточный драйвер блока переноса
3414
Протокольно-аппаратный интерфейс
3406
Минипорт ФСИ
3408
Физический сетевой интерфейс (ФСИ)
3410
Двоичный блоб
Состояние соединения (+ идентификатор потока)
3422
к Фиг. 35

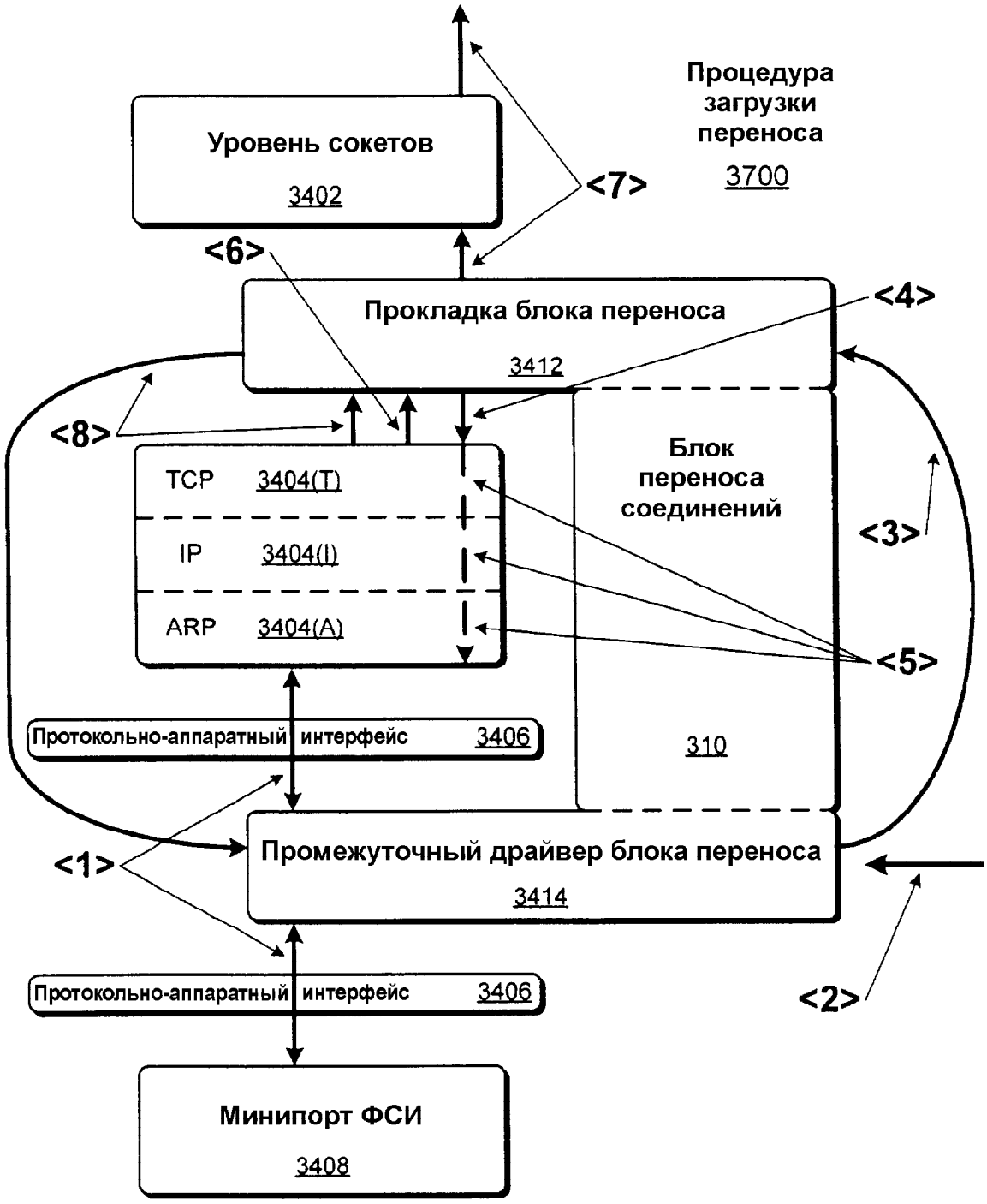
Фиг. 34


Приложение
316
Уровень сокетов
3402
Устройство-адресат
3500
Прокладка блока переноса
3412
Стек протоколов
Состояние протокола
3420
3404
3502
Вставить состояние соединения
Блок переноса соединений
Данные
3416
310
Протокольно-аппаратный интерфейс 3406
Промежуточный драйвер блока переноса
3414
Протокольно-аппаратный интерфейс 3406
Минипорт ФСИ
3408
Физический сетевой интерфейс (ФСИ)
3410
Двоичный блоб
Состояние соединения
(+ идентификатор потока)
3422
от Фиг. 34

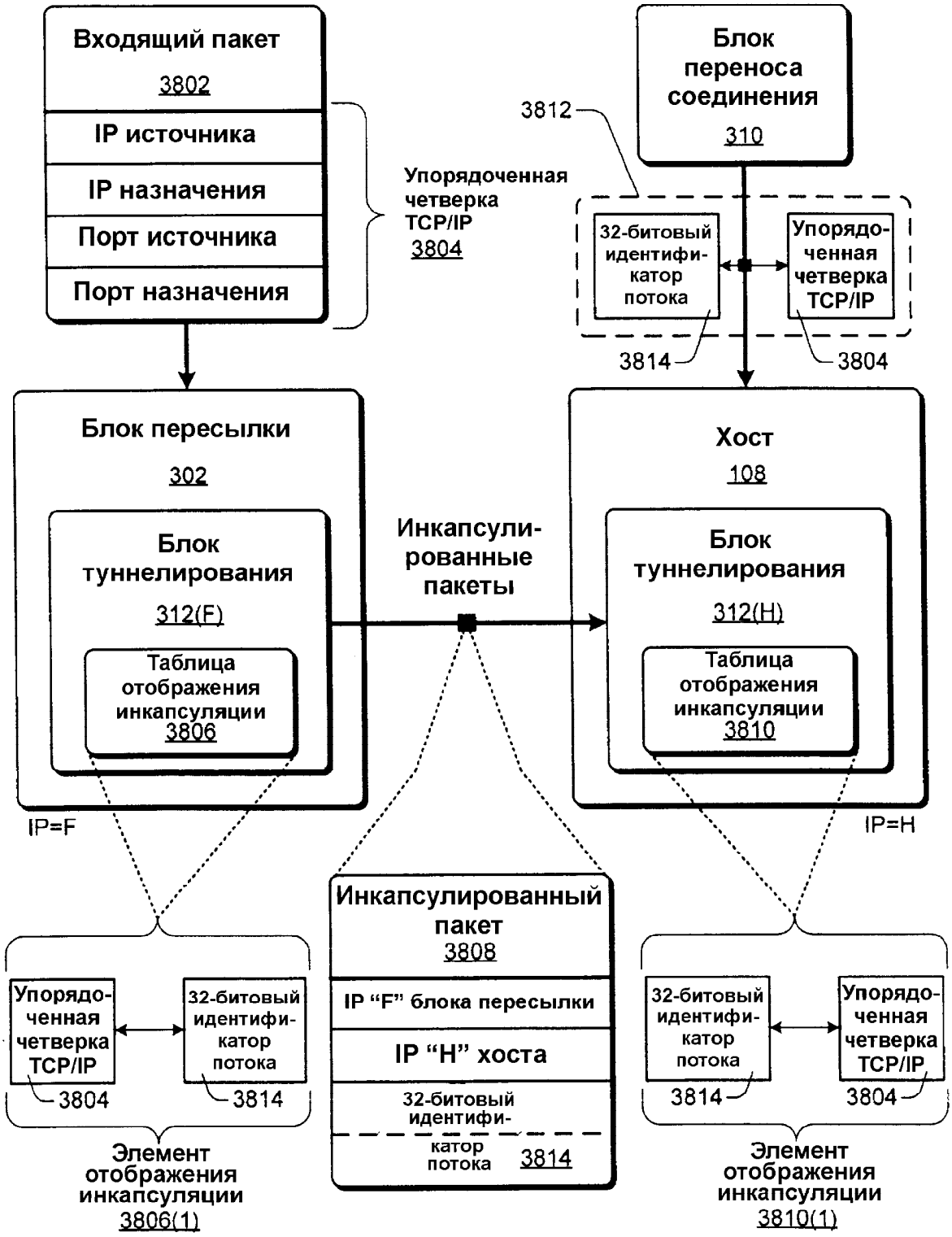
Фиг. 35


Процедура выгрузки переноса
3600
Уровень сокетов
3402
<2>
Прокладка блока переноса
3412
<6>
<3>
TCP 3404(T)
IP 3404(I)
ARP 3404(A)
Блок переноса соединений
<4>
<5>
Протокольно-аппаратный интерфейс 3406
310
<1>
Промежуточный драйвер блока переноса
3414
<7>
Протокольно-аппаратный интерфейс 3406
Минипорт ФСИ
3408

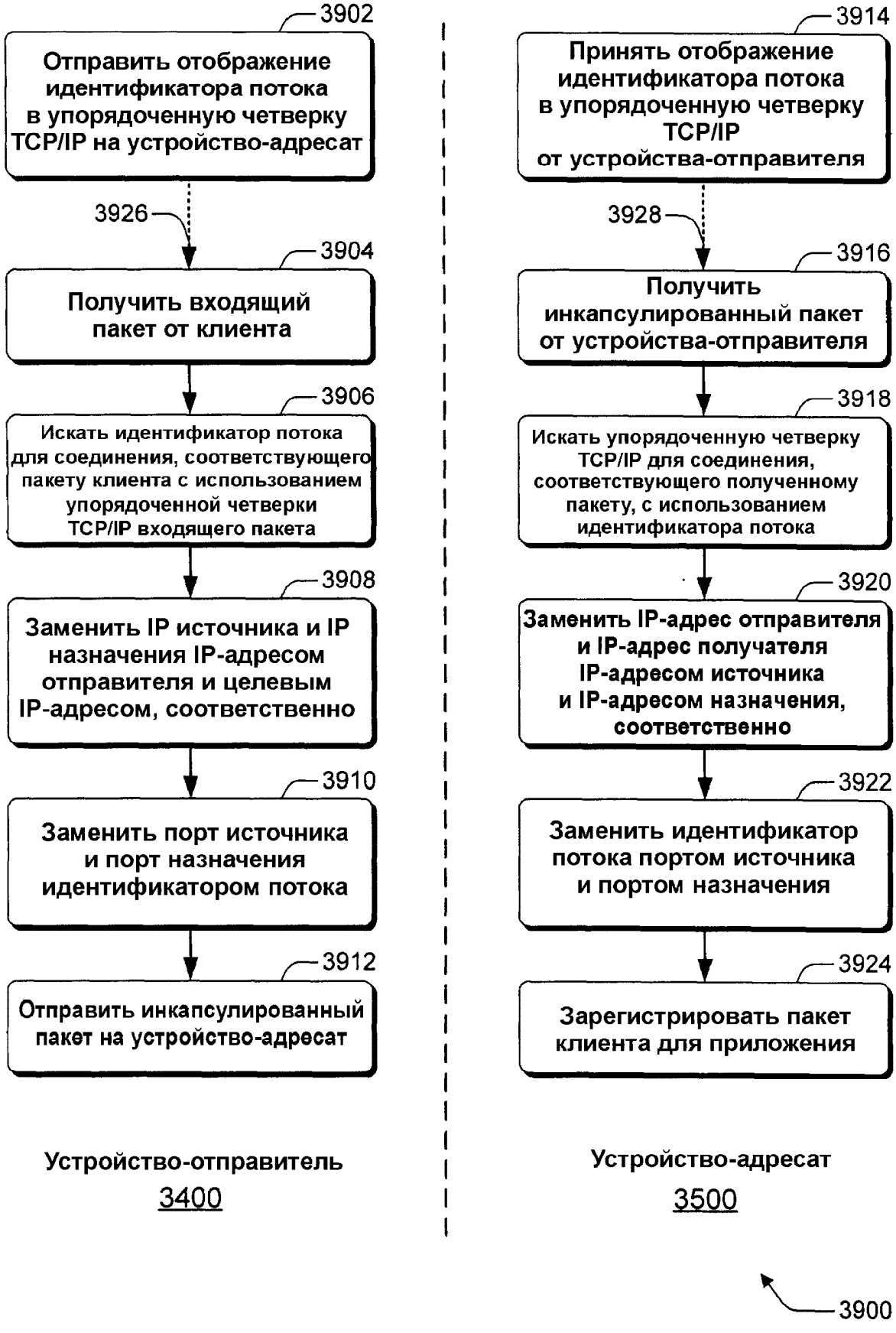
Фиг. 36


Процедура загрузки переноса
3700
Уровень сокетов
3402
<7>
<6>
Прокладка блока переноса
3412
<4>
<8>
TCP 3404(T)
IP 3404(I)
ARP 3404(A)
Блок переноса соединений
<3>
<5>
Протокольно-аппаратный интерфейс 3406
310
<1>
Промежуточный драйвер блока переноса
3414
<2>
Протокольно-аппаратный интерфейс 3406
Минипорт ФСИ
3408

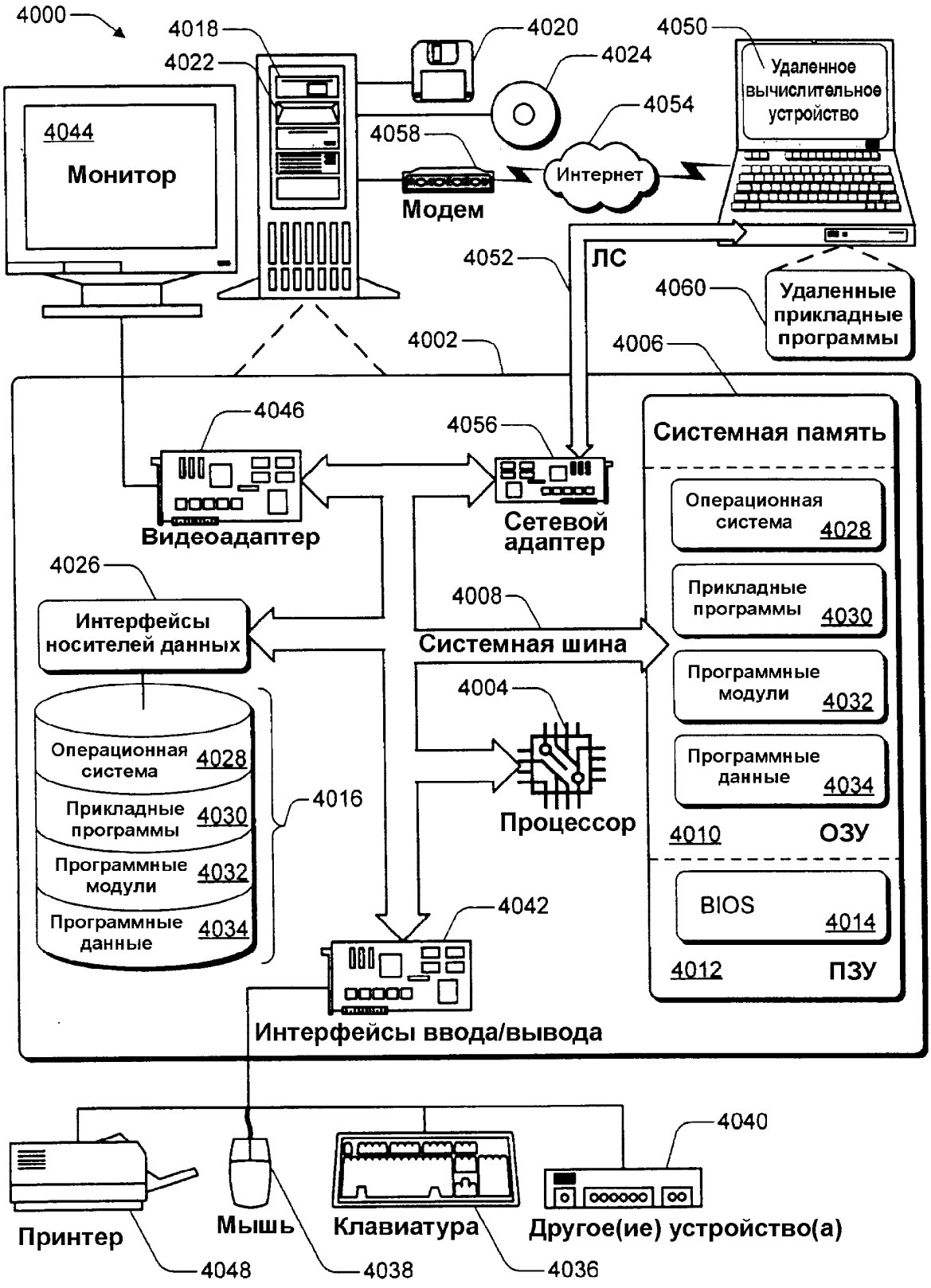
Фиг. 37


Входящий пакет
3802
IP источника
IP назначения
Порт источника
Порт назначения
Упорядоченная четверка TCP/IP
3804
Блок переноса соединения
310
3812
32-битовый идентификатор потока
Упорядоченная четверка TCP/IP
3814
3804
Блок пересылки
302
Блок туннелирования
312(F)
Таблица отображения инкапсуляции
3806
IP=F
Инкапсулированные пакеты
Хост
108
Блок туннелирования
312(H)
Таблица отображения инкапсуляции
3810
IP=H
Инкапсулированный пакет
3808
IP "F" блока пересылки
IP "H" хоста
32-битовый идентификатор потока
3814
Упорядоченная четверка TCP/IP
32-битовый идентификатор потока
3804
3814
Элемент отображения инкапсуляции
3806(1)
32-битовый идентификатор потока
Упорядоченная четверка TCP/IP
3814
3804
Элемент отображения инкапсуляции
3810(1)

Фиг. 38


3902
Отправить отображение идентификатора потока в упорядоченную четверку TCP/IP на устройство-адресат
3926
3904
Получить входящий пакет от клиента
3906
Искать идентификатор потока для соединения, соответствующего пакету клиента с использованием упорядоченной четверки TCP/IP входящего пакета
3908
Заменить IP источника и IP назначения IP-адресом отправителя и целевым IP-адресом, соответственно
3910
Заменить порт источника и порт назначения идентификатором потока
3912
Отправить инкапсулированный пакет на устройство-адресат
Устройство-отправитель
3400
3914
Принять отображение идентификатора потока в упорядоченную четверку TCP/IP от устройства-отправителя
3928
3916
Получить инкапсулированный пакет от устройства-отправителя
3918
Искать упорядоченную четверку TCP/IP для соединения, соответствующего полученному пакету, с использованием идентификатора потока
3920
Заменить IP-адрес отправителя и IP-адрес получателя IP-адресом источника и IP-адресом назначения, соответственно
3922
Заменить идентификатор потока портом источника и портом назначения
3924
Зарегистрировать пакет клиента для приложения
Устройство-адресат
3500
3900

Фиг. 39


4000
4018
4022
4020
4024
4050
Удаленное вычислительное устройство
4044
Монитор
4058
4054
Интернет
Модем
4052
ЛС
4060
Удаленные прикладные программы
4002
4006
4046
4056
Системная память
Операционная система 4028
Видеоадаптер
Сетевой адаптер
4026
Прикладные программы 4030
Интерфейсы носителей данных
4008
Системная шина
Программные модули 4032
4004
Операционная система 4028
Прикладные программы 4030
Программные модули 4032
Программные данные 4034
4016
Процессор
Программные данные 4034
4010
ОЗУ
BIOS 4014
4042
4012
ПЗУ
Интерфейсы ввода/вывода
4040
Принтер
Мышь
Клавиатура
Другое(ие) устройство(а)
4048
4038
4036

Фиг. 40