РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 420 897** (13) **C2**

(51) МПК
H04L 12/28 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2008109226/09, 11.09.2006**

(24) Дата начала отсчета срока действия патента:
11.09.2006

Приоритет(ы):
(30) Конвенционный приоритет:
12.09.2005 US 60/716,122
16.12.2005 US 11/275,185

(43) Дата публикации заявки: **10.10.2009** Бюл. № 28

(45) Опубликовано: **10.06.2011** Бюл. № **16**

(56) Список документов, цитированных в отчете о поиске: **US 6732189 B1, 04.05.2004. US 2004/0008717 A1, 15.01.2004. RU 2004102029 A, 10.07.2005. US 5923854 A, 13.07.1999. US 6430622 B1, 06.08.2002. RU 2047899 C1, 10.11.1995.**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **11.03.2008**

(86) Заявка PCT:
US 2006/035497 (11.09.2006)

(87) Публикация заявки PCT:
WO 2007/033179 (22.03.2007)

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):
МАССА Майкл Т. (US),
ДИОН Дэвид А. (US),
ОПАВСКИ Рудольф (US)

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

RU 2420897 C2

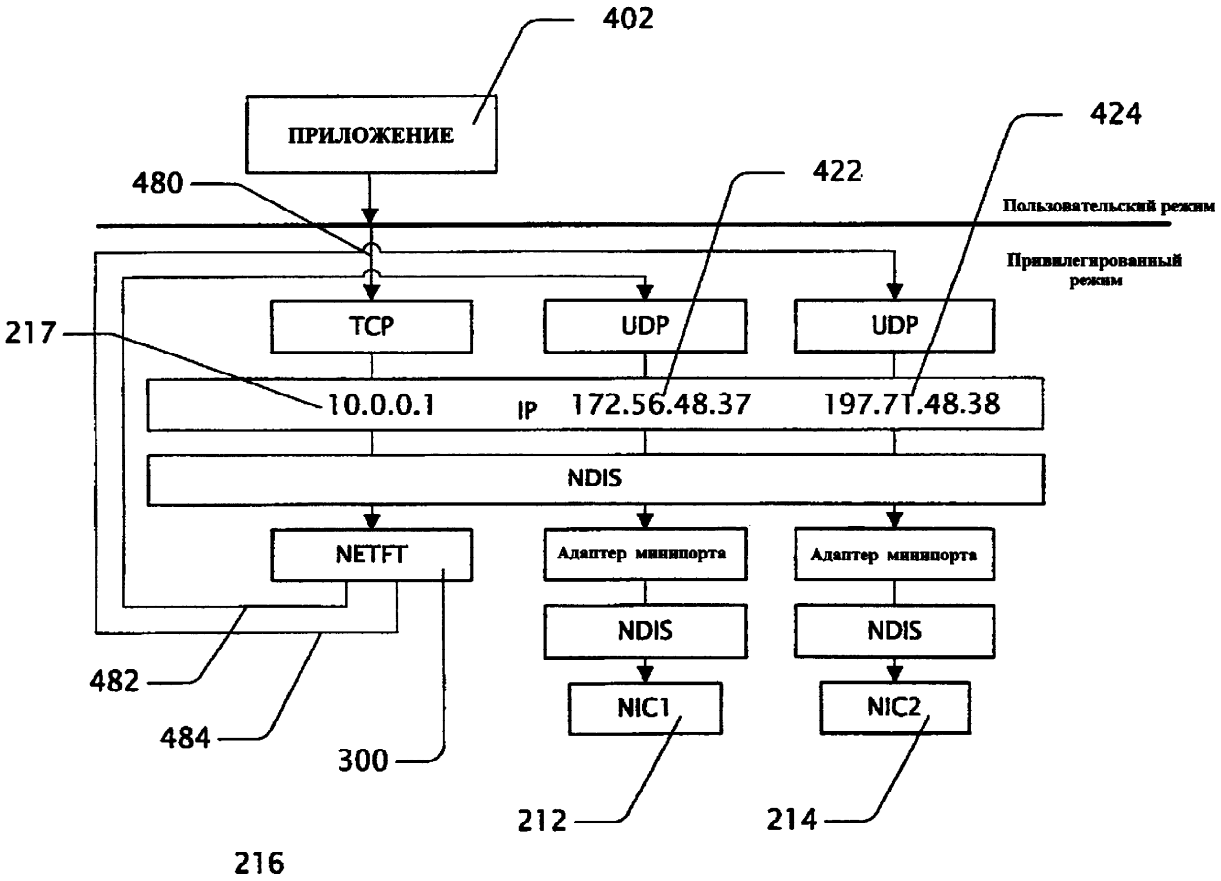
RU 2420897 C2

(54) ОТКАЗОУСТОЙЧИВАЯ СВЯЗЬ В МАРШРУТИЗОВАННЫХ СЕТЯХ

(57) Реферат:

Изобретение относится к области сетей передачи информации, а именно к компьютерной сети передачи информации, где многие узлы могут осуществлять обмен информацией друг с другом по сети. Технический результат заключается в обеспечении отказоустойчивости связи вычислительной сети посредством уникальных архитектур сетевого стека, требующих минимального рассмотрения прикладным программным обеспечением, действующим на связанных в сеть узлах. Для этого способ обеспечения отказоустойчивой сетевой связи между множеством узлов для приложения включает в себя этапы обеспечения множества исходных маршрутов передачи данных по множеству сетей, связанных между множеством узлов, и приема пакета данных на передающем узле от приложения. Причем передающий узел

является одним из множества узлов, пакет данных адресуется приложением на адрес на одном из множества узлов. Также способ содержит этап выбора первого выбранного маршрута для пакета данных из числа множества исходных маршрутов передачи, причем первый выбранный маршрут передачи является предпочтительным маршрутом. 3 н. и 14 з.п. ф-лы, 2 табл., 11 ил.



ФИГ. 4

RU 2420897 C2

RU 2420897 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(19) **RU** (11) **2 420 897** (13) **C2**

(51) Int. Cl.
H04L 12/28 (2006.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2008109226/09, 11.09.2006**

(24) Effective date for property rights:
11.09.2006

Priority:
(30) Priority:
12.09.2005 US 60/716,122
16.12.2005 US 11/275,185

(43) Application published: **10.10.2009** Bull. 28

(45) Date of publication: **10.06.2011 Bull. 16**

(85) Commencement of national phase: **11.03.2008**

(86) PCT application:
US 2006/035497 (11.09.2006)

(87) PCT publication:
WO 2007/033179 (22.03.2007)

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):
MASSA Majkl T. (US),
DION Dehvid A. (US),
OPAVSKI Rudol'f (US)

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

RU 2420897 C2

(54) FAULT-TOLERANT COMMUNICATION IN ROUTED NETWORKS

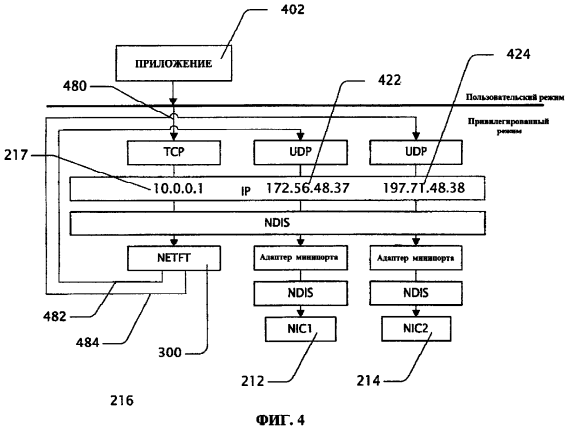
(57) Abstract:

FIELD: information technology.

SUBSTANCE: method for providing fault-tolerant network communication between a plurality of nodes for an application includes providing a plurality of initial communication pathways over a plurality of networks coupled between the plurality of nodes, and receiving a data packet on a sending node from the application. The sending node is one of the plurality of nodes, and the data packet is addressed by the application to an address on one of the plurality of nodes. The method also comprises a step for selecting a first selected pathway for the data packet from among the plurality of initial communication pathways, where the first selected pathway is a preferred pathway.

EFFECT: providing fault-tolerant communication for a computer network through unique architectures of the network stack, which require minimum consideration by an application software acting on nodes connected in the network.

17 cl, 2 tbl, 11 dwg



ФИГ. 4

RU 2420897 C2

ПРЕДШЕСТВУЮЩИЙ УРОВЕНЬ ТЕХНИКИ

В среде компьютерной сети передачи информации многие узлы могут осуществлять обмен информацией друг с другом по сети. Если сеть испытывает отказ, связь между узлами может быть нарушена.

КРАТКОЕ ОПИСАНИЕ СУЩНОСТИ ИЗОБРЕТЕНИЯ

Нижеследующее представляет упрощенное краткое изложение раскрытия, чтобы обеспечить читателю основное понимание. Это краткое изложение не является всесторонним представлением раскрытия, и оно не определяет основные/критические элементы изобретения и не устанавливает границы объема изобретения. Его единственной целью является представление в упрощенной форме некоторых раскрытых в документе концепций в качестве вводной части к более подробному описанию, которое представлено далее.

Нижеследующие примеры осуществления представляют обеспечение отказоустойчивости связи вычислительной сети посредством уникальных архитектур сетевого стека, требующих минимального рассмотрения прикладным программным обеспечением, действующим на связанных в сеть узлах.

Многие из сопутствующих признаков будут с большей легкостью оценены или таковые становятся лучше понятыми путем ссылки на нижеследующее подробное описание, рассматриваемое вместе с сопроводительными чертежами.

ОПИСАНИЕ ЧЕРТЕЖЕЙ

Настоящее описание будет лучше понято из нижеследующего подробного описания, изученного с рассмотрением сопроводительных чертежей, на которых:

Фиг.1 - блок-схема, показывающая примерную архитектуру сетевого стека.

Фиг.2 - блок-схема, показывающая организованную в сеть вычислительную среду, включающую в состав два примерных узла, соединенных через две сети.

Фиг.3 - блок-схема, показывающая примерный драйвер (NETFT) отказоустойчивой связи.

Фиг.4 - блок-схема, показывающая примерную архитектуру отказоустойчивой связи, включающую в состав NETFT и приложение.

Фиг.5 - блок-схема потока данных, показывающая организацию потока данных через среду отказоустойчивой связи, включая узел-источник и узел-адресат, соединенные через Тракт А по сети 1 и Тракт В по сети 2.

Фиг.6 - блок-схема потока данных, показывающая организацию потока данных через среду отказоустойчивой связи, показанную на фиг.5 с дополнением нескольких возможных отказов связи.

Фиг.7 - показ блок-схемы и другой пример драйвера, NETFT, отказоустойчивой связи.

Фиг.8 - блок-схема, показывающая примерную архитектуру отказоустойчивой связи, включая NETFT и приложение.

Фиг.9 - блок-схема потока данных, показывающая организацию потока данных через среду отказоустойчивой связи, включая узел-источник и узел-адресат, соединенные через Тракт А по сети 1 и Тракт В по сети 2.

Фиг.10 - блок-схема потока данных, показывающая организацию потока данных через среду отказоустойчивой связи, показанную на фиг.9, с дополнением нескольких возможных отказов связи.

Фиг.11 - блок-схема, показывающая примерную вычислительную среду, в которой может быть осуществлена описанная выше технология.

Используются одинаковые ссылочные числовые позиции, чтобы обозначать

одинаковые части на сопроводительных чертежах.

ПОДРОБНОЕ ОПИСАНИЕ

Подробное описание, представленное ниже вместе с сопроводительными чертежами, предназначено в качестве описания имеющихся примеров осуществления и не подразумевает представления только форм, в которых имеющийся пример может быть создан или использован. Описание излагает функции примеров осуществления и последовательность этапов для создания и действия примеров осуществления. Однако одинаковые или эквивалентные функции и последовательности могут выполняться согласно различным примерам осуществления.

Хотя имеющиеся примеры осуществления описаны и проиллюстрированы в документе в виде реализуемых в вычислительной системе, описанная система представлена в качестве примера, а не ограничения. Как должны понимать специалисты в данной области техники, имеющиеся примеры осуществления являются подходящими для применения во множестве различных типов вычислительных и сетевых систем.

На фиг.1 показана блок-схема примерной архитектуры 100 сетевого стека. Сетевой стек ("стек") обычно связывается с программными приложениями через интерфейсы сетевого стека и/или другие интерфейсы, чтобы предоставить приложениям функциональность сети связи. Обычно считают, что приложение должно находиться на "вершине" (или связываться с вершиной) стека. Обычно считают, что сеть должна находиться в "основании" (или связываться с основанием («дном»)) стека. Различные элементы сетевого стека могут упоминаться как находящиеся на или около вершины или основания стека, или выше или ниже в стеке относительно друг друга. Например, на фиг.1, драйвер 130 протокола находится в стеке выше протокола сетевой интерфейсной платы NIC 180, которая показывается в основании стека на этом конкретном чертеже. Различные описания сетевого стека могут включать или могут не включать некоторые элементы стека, или могут группировать, определять порядок или именовать элементы различным образом в зависимости от назначения или цели описания, как понятно специалистам в данной области техники.

Как используется в документе, термин "драйвер" относится к управляющей программе, или подобному, что дает возможность узлу работать с конкретным устройством, таким как принтер, сетевая интерфейсная плата, или другая подсистема компьютера, или работать с одной или несколькими программами, такими как сетевые стеки, драйверы протокола, и/или другое программное обеспечение или микропрограммное обеспечение или подобное. Например, драйвер протокола обычно работает с сетевым стеком.

Приложение может передавать пакет данных на стек, предназначенный для приложения, работающего на другом узле. В этом случае говорят, что данные проходят "вниз" стека и посылаются по сети. Говорят, что принятые узлом данные проходят "наверх" стека, пока не достигнут предназначенного приложения. Такие сетевые системы являются известными специалистам в данной области техники.

В одном примере стек основывается на спецификации (NDIS) интерфейса сетевых драйверов, который определяет интерфейс (API) прикладного программирования для сетевых интерфейсных плат (адаптеров) (NIC), например, NIC 180, и абстрагирует сетевые аппаратные средства от сетевых драйверов. NDIS также определяет стандартный интерфейс между многоуровневыми сетевыми драйверами, тем самым абстрагируя драйверы нижнего уровня, управляющие аппаратными средствами, например, драйвер мини-порта, от драйверов верхнего уровня, например, драйверов

протокола. Множество соответствующих NDIS драйверов протокола могут сосуществовать на одном узле. Также, если узел включает в состав многие сетевые интерфейсные платы, возможно вследствие того, что он соединен с несколькими сетями, NDIS маршрутизует сетевой трафик на надлежащую NIC через ее связанный драйвер, как указано трафиком. Иллюстрация NDIS показана на фиг.1. Другие стандарты сетевого стека, технологии и/или архитектуры, такие как открытый интерфейс (ODI) канала данных, интерфейс (DLPI) поставщика услуг линии передачи данных, унифицированный интерфейс (UDI) драйверов, или другие технологии могут использоваться с нижеследующими примерами, а также с соответствующими модификациями, как будет понятно специалистам в данной области техники. Для удобства NDIS и терминология NDIS используется вместе с примерами осуществления по всему данному описанию, но другие стандарты, технологии и/или архитектуры могут использоваться во всех из этих примеров с соответствующими модификациями, если не указано иное.

Как показано на фиг.1, сопрягаемым с NIC 180 через NDIS 120 является драйвер 160 мини-порта. Драйвер мини-порта обычно взаимодействует с NDIS через интерфейс 162 мини-порта NDIS. Драйвер 160 мини-порта может быть связан с NIC 180 и может управлять ее операциями, включая посылку и прием данных через NIC. Драйвер 160 мини-порта обычно взаимодействует с драйверами более высокого уровня, такими как промежуточный драйвер 140 и драйвер 130 протокола. Драйвер мини-порта считают драйвером NIC. Мини-порты NIC обычно выполняют те зависимые от аппаратного обеспечения операции, которые необходимы для управления конкретной NIC с помощью общей или NIC-независимой функциональности, обеспечиваемой посредством NDIS. Узел может включать в состав несколько NIC, причем каждая NIC обычно имеет связанный драйвер NIC. Некоторые примеры в этом описании представляют использование драйверов мини-порта, но, как будет понятно специалистам в данной области техники, любой тип драйвера NIC или подобное могут использоваться в этих примерах осуществления, если не указано иное.

Драйвер 130 протокол или транспортный драйвер сопрягаются с NDIS 120 через интерфейс 134 протокола NDIS. Драйверы протокола или драйверы транспортного протокола обычно обеспечивают функциональность для создания, посылки и приема пакетов данных, которые посылаются от одного узла на другой через сетевой стек и по сети. Как известно специалистам в данной области техники, общим транспортным протоколом надежной или гарантированной доставки (пакетов данных) может быть протокол TCP/IP (протокол управления передачей/протокол IP). Протокол UDP (протокол пользовательских дейтаграмм) поверх IP может быть общим протоколом ненадежной или негарантированной доставки пакетов. Протоколы TCP, UDP и/или другие протоколы, такие как стек протоколов IPX/SPX (пакетный обмен/последовательный обмен пакетами), могут использоваться с нижеследующими примерами осуществления, если не указано иное.

На фиг.1 показаны промежуточные (IM) драйверы 140 NDIS между драйверами 130 протокола и NDIS-мини-портами 160 NIC. По отношению к драйверам протокола IM-драйверы предстают NDIS-мини-портами, тогда как по отношению к драйверам NIC они имеют вид драйверов протокола. Пакеты данных, проходящие вверх или вниз по сетевому стеку, проходят через IM-драйвер 140, который может игнорировать, проверять, фильтровать, пересылать, перенаправлять и/или модифицировать пакеты данных. Промежуточный драйвер 140 также может быть известным в качестве драйвера фильтра.

На фиг.2 показана блок-схема сетевой вычислительной среды 200, включающей в состав два примерных узла 210 и 260, связанных через две сети 202 и 282. Узлы 210 и 260 могут быть персональными компьютерами (ПК, PC), клиентскими компьютерами, серверами, ведущими компьютерами, портативными ЭВМ, портативными устройствами, устройствами бытовой электроники, или любыми из различных других типов вычислительных или предназначенных для обработки данных устройств, машин или систем. Один неограничительный пример относительно типа вычислительной системы описан подробно ниже по отношению к фиг.11. Окружности 212, 214, 262, и 264 представляют устройства NIC, связанные с их соответственными узлами. Один неограничительный пример относительно типа NIC в виде сетевого адаптера 1113 дополнительно описан ниже в отношении фиг.11.

Как используется в документе, термин узел относится к любой вычислительной системе, устройству или процессу, который является уникально адресуемым, или иначе, уникально идентифицируемым в сети (например, сети 202), и он действует, чтобы осуществлять взаимодействие с другими узлами в сети. Например и неограничительно, узлом может быть персональный компьютер, компьютер сервера, переносное или портативное устройство, планшетное устройство, многопроцессорная система, микропроцессорная система, телевизионная абонентская приставка, устройство бытовой электроники, сетевой ПК, мини-компьютер, универсальная ЭВМ, или подобное. Неограничительный пример узла 210, в виде вычислительной системы, изложен ниже по отношению к фиг.11.

Сети 202 и 282 могут быть одинаковыми сетями, могут существовать на одной и той же или на различных подсетях, могут быть логически или физически соединенными или изолированными друг от друга, могут использовать сходные или различные технологии организации сети, и т.д. В частности, сети 202 и 282 могут быть сетями с маршрутизацией, то есть сетями, включающими в состав маршрутизаторы, которые пересылают пакеты маршрутизируемого протокола. Маршрутизируемыми протоколами обычно считают протоколы связи, используемые, чтобы направлять данные от одной сети на другую. Примером маршрутизируемого протокола является TCP/IP. Посылка пакета данных маршрутизируемым образом подразумевает использование маршрутизируемого транспортного протокола для форматирования и/или посылки пакета данных. Специалисты в данной области техники будут знакомы с топологиями маршрутизируемых протоколов и сетевой маршрутизации, системами и архитектурами.

В одном примере сети 202 и 282 могут быть независимыми друг от друга, так что если имеется проблема или отказ, связанные с одной сетью, это вряд ли воздействует на рабочее состояние другой. В других примерах могут использоваться три или более сетей. В примерах, где желательны более высокие степени отказоустойчивости, может использоваться большое число сетей, наряду с совместной связностью узлов для этих сетей, включая сходное число NIC, установленных на узле.

Относящаяся к узлу 210 NIC 212 показана с примерным адресом 172.56.48.37 и связана с сетью 1 202. Также относящаяся к узлу 210 NIC 214 показана с примерным адресом 197.71.48.38 и связана с сетью 2 282. Относящаяся к узлу 260 NIC 262 показана с примерным адресом 172.56.48.38 и также связана с сетью 1 202. Также относящаяся к узлу 260 NIC 264 показана с примерным адресом 197.71.48.39 и также связана с сетью 2 282. Эти адреса на практике могут быть адресами по протоколам IPv4 или IPv6 или подобным, или любым другим типом сетевого адреса, обычно связанного с используемым протоколом.

Каждый узел может включать в себя одну или несколько NIC. Стрелки 201 и 203, показанные также на фиг.11 в виде стрелки 1114, представляют первый маршрут передачи информации или маршрут ("Тракт А") по сети 1 202 между узлами 210 и 260. Стрелки 281 и 283 представляют второй маршрут передачи информации или маршрут ("Тракт В") по сети 2 282 между узлами 210 и 260. Практически, может быть один или несколько маршрутов по одной или нескольким сетям между двумя или несколькими узлами в среде 200. Термин "маршрут", как используется в документе, определен как маршрут передачи информации, или канал связи, между узлами в сети. Такой маршрут или канал может быть динамическим в том, что точный маршрут между узлами может изменяться во времени.

Блоки 216 и 266 представляют приложение и сетевой стек, включая драйвер отказоустойчивой связи (FT), предусмотренный на каждом из узлов 210 и 260. FT-драйвер блока 216 показан с примерным адресом 10.0.0.1, и FT-драйвер блока 266 показан с примерным адресом 10.0.0.2. Эти адреса обычно считаются виртуальными адресами. Эти адреса могут быть IPv4- или IPv6-адресом или подобными, или любым другим типом сетевого адреса или адреса связи. FT-драйверы могут иметь или не иметь виртуальные адреса, как показано в различных примерах ниже.

Отказоустойчивый сетевой стек является сетевым стеком, включающим в себя FT-драйвер, например, NETFT, описанный ниже в связи с фиг.3, или подобный. FT-драйвер, например, NETFT, действующий в комбинации с сетевым стеком, обычно позволяет узлам взаимодействовать друг с другом через один или несколько трактов передачи информации, например, Тракт А и Тракт В, по одной или нескольким сетям. Если какой-либо из этих трактов связи данных терпит отказ, узлы могут продолжать осуществлять связь, получив, по меньшей мере, один рабочий тракт (маршрут) передачи. Такой отказ тракта передачи может следовать из отказа NIC или отказа какого-либо элемента тракта передачи, включая соединения, кабельную сеть или другие среды передачи данных (включая в них радиочастотные ("RF") или инфракрасные ("IR") и т.п.), маршрутизаторы, концентраторы, коммутаторы, межсетевые защиты, поставщиков ("ISP") услуг сети Интернет, сбой питания для любого узла, устройства или системы сети, или подобное.

В одном примере, отказ связи может иметь результатом событие «вновь подключенного устройства» ("PnP"). PnP-событие может указывать удаление NIC из ее узла или по отношению к изменению восприятия (считывания) среды передачи. Разъединение восприятия среды передачи, например, обычно является результатом отказа, который вызывает, что NIC теряет сигнал или несущую в сетевой среде передачи, такой как сетевой кабель, RF- или IR-канал или подобное. Разъединение восприятия среды передачи может быть вызвано разъединением из NIC сетевого кабеля или несущей или выключением питания другого конца кабеля (концентратора или коммутатора, например). Соединение восприятия среды передачи является обычно обратным (событием), таким как повторное соединение кабеля, повторное включение концентратора или коммутатора или подобное. Эти типы событий, известные также как события связности, являются обычно локальными событиями в том, что они происходят непосредственно на узле или в близости к нему. Такие локальные события связности обычно имеют результатом на узле указание события, такого как PnP-событие или подобного.

В другом примере осуществления отказ связи может быть выявлен путем использования контрольных пакетов, посылаемых между узлами. Отказ такого контрольного пакета может указывать отказ тракта передачи между узлами.

Контрольные пакеты обычно помечаются, так что FT-драйвер может выявлять их после приема и удалять их для потока пакетов, передаваемых вверх по сетевому стеку. В одном примере контрольные пакеты могут быть осуществлены с использованием протокола (RCP) управления маршрутом путем формирования RCP-пакетов. Такие контрольные пакеты могут использоваться, чтобы проверять действительность рабочего состояния сквозной передачи для маршрута. То есть, согласно посылке контрольного пакета от узла 210 по Тракту A на узел 260 и приему узлом 210 от узла 260 ответа на посланный контрольный пакет, обычно считают, что Тракт A является рабочим для сквозной передачи. Если контрольный пакет терпит неудачу (нет ответа на контрольный пакет, принятого в ответ на посланный контрольный пакет), такой отказ может указывать, что Тракт A не является рабочим для сквозной передачи, возможно вследствие отказа некоторого элемента сети 1 202, такого как маршрутизатор, коммутатор, соединение, или подобное, или вследствие отказавшего непосредственно целевого узла. В частности узел 210 может иметь рабочую NIC 212 и действительное восприятие среды передачи, указывая, что он надлежаще соединен с сетью, но может, тем не менее, выявить отказ контрольного пакета вследствие некоторого сетевого или системного отказа вниз по линии связи.

На фиг.3 показана блок-схема примерного драйвера, NETFT, 300 отказоустойчивой связи. NETFT 300 может быть осуществлен в виде драйвера (фиг.1, 160) мини-порта NDIS для использования с сетевым стеком NDIS и для обеспечения сетевой связи между узлами, устойчивой к отказам маршрута передачи. То есть, связь между двумя или несколькими узлами может продолжаться, если каждый использует NETFT, несмотря на отказ какого-либо компонента в маршруте пока, по меньшей мере, один маршрут передачи остается рабочим.

В одном примере исполнение FT-драйвера в виде драйвера мини-порта NDIS обеспечивает, по меньшей мере, два преимущества. Во-первых, поскольку такой FT-драйвер обычно находится в стеке ниже любых драйверов протокола, надежность протокола обычно обеспечивается посредством любого драйвера более высокоуровневого надежного протокола, на который обычно не воздействует дополнение отказоустойчивости канального уровня, обеспеченной посредством FT-драйвера. Например, при использовании FT-драйвера в комбинации с драйвером протокола, например, драйвером TCP/IP, FT-драйвер обычно будет выявлять отказавшие маршруты передачи и направлять пакеты данных по рабочим трактам сквозной передачи, независимым от какого-либо драйвера протокола. Если какая-либо потеря пакета происходит из-за переключения маршрутов, то драйвер протокола TCP/IP, который обычно в стеке находится выше FT-драйвера, обычно выявляет такие потери и выполняет любые операции повтора передачи или повторной посылки, чтобы гарантировать, что надежный протокол достигает цели доставки пакета.

Вторым преимуществом размещения FT-драйвера в стеке ниже драйвера протокола является то, что обычно нет вводимого ухудшения возможности маршрутизации по протоколу. При таком определении конфигурации, любая операция туннелирования, которую выполняет FT-драйвер над пакетом данных, может использовать маршрутизируемый протокол, например, TCP или UDP, гарантируя таким образом, что такие данные являются маршрутизируемыми, в дополнение будучи отказоустойчивыми на канальном уровне. "Туннелированием маршрутизируемым образом" пакета данных является туннелирование пакета данных с использованием маршрутизируемого протокола.

NETFT, в качестве части сетевого стека, обычно связывается с программным приложением через NDIS или другие интерфейсы сетевого стека. Такая связность обычно дает приложениям возможность посылки и приема пакетов данных по сетям, связанным с основанием стека. В одном примере осуществления приложения склонны использовать виртуальный адрес в качестве адреса-источника для своих пакетов данных, этот виртуальный адрес известен для NETFT и отображается и сообщается на другие узлы в сети, как описано ниже. Как показано на фиг.3, NETFT включает в состав адаптер 302 мини-порта (известный также как обрабатывающий элемент), базу данных 304 маршрутизации и один или несколько адаптеров 306 монитора маршрута и туннельных адаптеров 308.

Туннельный адаптер 308 обычно представляет одну NIC на локальном узле (или, в некоторых случаях, виртуальную NIC) и поддерживает сокет, используемый для туннелирования пакетов на NETFT на целевом узле. Обычно имеется один туннельный адаптер 308, связанный с каждой NIC на локальном узле, причем каждая NIC связывается с сетью, обеспечивая маршрут передачи на другой узел. Каждая сеть может быть или не быть изолированной от любой другой сети. Туннельный адаптер 308 обычно связан с драйвером протокола туннелирования и туннеллирует пакеты данных через протокол туннелирования на его связанную NIC и от нее через интерфейсы NDIS. Одним примером протокола туннелирования является UDP. В качестве альтернативы, для туннелирования могут использоваться другие протоколы, такие как TCP, IPX, или SPX. Туннельный адаптер 308 может становиться неактивным, если соответствующие NIC или соединение среды передачи стали неактивными.

База данных 304 маршрутизации, как реализовано в NETFT, является обычно простой структурой данных, которая может находиться в системной памяти, структура включает в состав элементы, отображающие виртуальный адрес для одного или нескольких каналов на подобный NETFT на другом узле. В одном примере осуществления, отображения представляются посредством адаптеров монитора маршрутов, например, адаптера 306 монитора маршрутов, которые обычно связаны с туннельным адаптером, например, туннельным адаптером 308. Обычно база данных маршрутизации, например, база данных 304 маршрутизации, будет включать в состав один набор адаптеров маршрутов для каждого туннельного адаптера, каждый адаптер маршрута является связываемым с другим целевым узлом, достижимым по маршруту, связанному с туннельным адаптером. При использовании TCP/IP, например, база данных может отображать виртуальный адрес назначения на физический адрес конкретного удаленного узла.

База данных 304 маршрутизации также может включать в состав приоритетную информацию для каждого маршрута передачи. Такая приоритетная информация может использоваться, чтобы указывать предпочтительный или первичный маршрут на другой узел и/или может включать в состав информацию о скорости тракта передачи или другие характеристики. Предпочтительным маршрутом является вычисленный посредством NETFT маршрут, насколько возможно подлежащий использованию над другими возможными маршрутами, на основании приоритетной информации и/или состояния тракта передачи. Приоритетная информация может альтернативно указывать балансировку загрузки по алгоритму кругового обслуживания для использования множества маршрутов на целевой узел для выравнивания по нагрузке трафика по трактам передачи, или допускать некоторую другую схему назначения приоритетов маршрутов.

В Таблице 1 показана примерная таблица отображения базы данных 304 таблицы

маршрутов.

Таблица 1

Тип адреса назначения	Адрес	Приоритет
Виртуальный	10.0.0.2	--
Физический: тракт А	172.56.48.38	1
Физический: тракт В	197.71.48.39	2

Что касается таблицы 1 и фиг.2, в таблице 1 показана примерная таблица отображения, которая использовалась бы NETFT, действующим на узле 216. В таблице 1 показан виртуальный адрес 10.0.0.2 назначения, как показано для узла 266, виртуальный адрес отображен на физический адрес 172.56.48.38, связанный с Трактом А на узел 266, и физический адрес 197.71.48.39, связанный с Трактом В на узел 266. Тракт А показан с первым приоритетом и Тракт В со вторым приоритетом. Таблица 1 представлена в качестве примера и не подразумевает являться ограничительной.

При посылке данных от узла 216 на узел 266 такая таблица отображения обычно используется для туннелирования пакета, предназначенного для виртуального адреса назначения 10.0.0.2 путем пересылки пакета по протоколу туннелирования, например, UDP, на физический адрес назначения 172.56.48.38, таким образом, туннелируя пакет от узла 216 по Тракту А на узел 266. Одна такая таблица отображения может быть создана в базе данных маршрутизации (фиг.3, 304) для каждого набора маршрутов, установленных между двумя узлами. Такая таблица отображения может реализовываться в различных формах, использовать различные схемы приоритетов и/или хранить другую информацию, включая рабочее состояние маршрута. Показанные в Таблице 1 структура таблицы отображения, число маршрутов, форматы адресов, и т.д. представлены в качестве примера и не подразумевают являться ограничительными.

Виртуальный адрес локального узла, виртуальные адреса удаленных узлов и приоритет и другая информация маршрута обычно обеспечиваются на узлы посредством механизма передачи вне полосы и передаются на NETFT через его интерфейсы NDIS. Этот механизм передачи вне полосы может быть максимально простым в виде системного администратора, использующего приложение административного управления, чтобы задавать информацию, или это может быть автоматизированная система или подобное. Такие механизмы передачи вне полосы известны специалистам в данной области техники.

Как показано на фиг.3, адаптер 302 мини-порта (известный также как обрабатывающий элемент драйвера) обычно анализирует пакет данных, проходящий вниз по сетевому стеку, исследует виртуальный адрес назначения пакета и использует информацию из базы данных 304 маршрутизации, чтобы определить, через какой туннельный адаптер 308 туннелировать пакет данных. Входящие пакеты, или пакеты данных, движущиеся вверх по стеку, пересылаются вверх по стеку по направлению к их виртуальному адресу назначения, причем протокол туннелирования предварительно удаляет заголовки пакета туннелирования. В частности, туннельный адаптер 308 проверяет входящие пакеты, и пересылает контрольные пакеты на адаптер 306 монитора маршрута, и пересылает другие пакеты вверх по стеку через адаптер 302 мини-порта. Специалистам в данной области техники известны аспекты пакетов данных туннелирования, использующих протокол туннелирования, и как заголовки протокола добавляются и удаляются драйверами протокола.

Адаптер 306 монитора маршрута обычно представляет удаленный узел, доступный

по конкретному маршруту, идентифицированному посредством связанного туннельного адаптера. Адаптер 306 монитора маршрута будет обычно обеспечивать физический адрес для удаленного узла, физический адрес также соответствует конкретному маршруту на удаленный узел. Этот физический адрес обычно используется для отображений в базе данных 304 маршрутизации. Обычно имеется один адаптер монитора маршрута для каждого отдельного маршрута на удаленный узел, каждый адаптер монитора маршрута является связываемым с туннельным адаптером, представляющим маршрут. В одном примере, возвращаясь на фиг.2, узел 210 показан соединенным с узлом 260 по двум маршрутам, один через сеть 1 202 ("Тракт А") и другой через сеть 2 282 ("Тракт В"). Действующий на узле 210 NETFT может включать в состав первый адаптер монитора маршрута ("RMA-A"), обеспечивающий физический адрес 172.56.48.38 удаленного узла 260, связанный с его NIC 262. RMA-A может быть связан с первым туннельным адаптером ("TA-A") на узле 210, который может быть связан с Маршрутом А. NETFT на узле 210 может также включать в состав второй адаптер ("RMA-B") монитора маршрута, обеспечивающий второй физический адрес 197.71.48.39 удаленного узла 260, связанный с его NIC 264. RMA-B может быть связан со вторым туннельным адаптером ("TA-B") на узле 210, который может быть связан с Трактом В.

Что касается фиг.3, адаптер 306 монитора маршрута обычно осуществляет мониторинг состояния (степени исправности) маршрута на удаленный узел и указывает в базе данных 304 маршрутизации отказавший или нерабочий маршрут. Мониторинг обычно включает в себя прием любых указаний событий и/или обозначение любых отказов контрольного пакета и обновление базы данных 304 соответственно. В одном примере, событие, указывающее отказ NIC или соединения среды передачи, может иметь результатом отключение туннельного адаптера 308. В другом примере, отказ контрольного пакета может иметь результатом отключение адаптера 306 монитора маршрута, связанного с конкретным удаленным узлом, для которого контрольный пакет получил отказ.

На фиг.4 показана блок-схема примерной архитектуры 216 отказоустойчивой связи, включающей в состав NETFT 300 и приложение 402. В этом примере приложение 402 посылает пакеты данных на NETFT 300 через стек, используя виртуальный адрес 217 источника и виртуальный адрес назначения, представляющий узел-адресат. Такие исходящие пакеты данных проходят через маршрут 480 от приложения и через сетевой стек на драйвер 300. Драйвер 300 обычно определяет, какой из возможных маршрутов каждый пакет должен получить, обычно используя приоритетную информацию и информацию о рабочем состоянии маршрута, хранимую в базе данных маршрутизации, и туннелирует пакет на целевой узел по выбранному маршруту, используя соответствующий физический адрес 422 или 424 источника.

Приложение 402 может посылать пакет данных через NETFT 300 по протоколу TCP, как показано на фиг.4. В качестве альтернативы могут использоваться UDP или любой другой протокол. Также, как показано, NETFT 300 может использовать протокол UDP, чтобы туннелировать пакеты на целевой узел. В качестве альтернативы для туннелирования могут использоваться TCP или любой другой протокол. Дополнительно, альтернативные примеры могут не использовать адаптеры мини-порта или драйверы NDIS, но могут использовать другие механизмы или архитектуры, чтобы выполнять подобные функции. В заключение, различные элементы сетевого стека и т.п. могут действовать либо в пользовательском режиме или в привилегированном режиме, либо как показано или иным образом, либо на

системах с эквивалентными режимами работы или без них.

На фиг.5 показана блок-схема организации потока данных через среду 500 отказоустойчивой связи, включающей в состав узел-источник 216 и узел-адресат 266, соединенные через Тракт А по сети 1 202 и Тракт В по сети 2 282. В этой примерной среде 500 данные показываются посылаемыми от приложения, действующего на узле 216, на приложение, принимающее на узле 266 по виртуальному адресу назначения. Пакеты данных проходят вниз по действующему на узле 216 сетевому стеку с использованием протокола TCP в NETFT, как показано посредством маршрута 501. Если, как показано, Тракт А является выбранным маршрутом, NETFT отображает пакеты данных от виртуального адреса источника, используемого приложением, на Тракт А и туннелирует данные через протокол UDP, используя физический адрес назначения Тракта А для целевого узла 266, из NIC 1 узла 216, как дополнительно показано посредством трактом 501, и на сеть 1 202 через канал 201 связи. Данные затем проходят через сеть 1 202 по каналу 203 связи и на узел 266, проходя вверх по сетевому стеку, действующему на узле 266, как показано маршрутом 503. Данные затем проходят через драйвер протокола UDP, того же протокола, который использовался на передающей стороне в качестве протокола туннелирования, где заголовки протокола UDP удаляются из пакетов данных, которые затем пропускаются в NETFT, действующий на узле 266. NETFT затем пересылает пакеты данных вверх по стеку на приложение, которое является принимающим по виртуальному адресу назначения. Ответы обычно проходят в обратном порядке.

На фиг.6 показана блок-схема организации потока данных через среду 500 отказоустойчивой связи, показанной на фиг.5, с дополнением нескольких возможных отказов 610, 612, 620, 622 и 630 связи. Возможны также другие отказы связи. Отказ 610 указывает отказ NIC1, действующей на передающем узле 216. Такой отказ может происходить, если NIC1 была удалена из узла, драйвер NIC1 потерпел отказ, непосредственно NIC1 потерпела отказ, или подобное. Отказ может быть выявлен посредством NETFT через указание события, например, события PnP или подобного, и/или отказа контрольного пакета. В такой ситуации считают, что Тракт А потерпел отказ, и NETFT выберет альтернативный рабочий маршрут сквозной передачи. Рабочий маршрут сквозной передачи является обычно маршрутом, в который может успешно доставлять данные от узла-источника и приложения полностью на узел назначения и приложение.

Отказ 620 указывает отказ связанности сетевой среды передачи с NIC1 узла 216. Этот отказ может быть из-за разъединяемого от NIC1 кабеля от того, что разъединяется кабель от некоторого устройства сети 1 202, от выключаемого или отказывающего устройства на сетевой стороне, соединенного с кабелем, или подобного. Этот тип отказа также может быть выявлен посредством NETFT через указание события, например, события PnP или подобного, и/или отказа контрольного пакета и дополнительного выбранного маршрута.

Отказ 630 указывает отказ некоторого типа в сети 202, имея следствием, что пакетам данных не удается достичь узел-адресат 266. В этом случае отказа, посылающий узел 216 может тем не менее быть соединенным с сетью 202 с указанием надлежащего восприятия среды передачи, хотя Тракт А стал нарушенным далее вниз по сети. Получив такой отказ, действующий на передающем узле 216, NETFT может не выявить отказ через указание события, если локальные указания показывают связность с сетью 202 как хорошую, но может выявить отказ через отказ

контрольного пакета для Тракта А.

Отказ 622 канала 203 связи и отказ 612, действующий на приемном узле 266 NIC1, обычно являются сходными с соответствующими отказами, показанными для узла 216. Но эти отказы, не являющиеся локальными на узле 216, могут не быть выявлены через указания события, а могут быть выявлены через отказ контрольного пакета.

Любой из этих отказов и другие отказы могут быть выявлены посредством действующего на узле 216 NETFT и иметь результатом, что он выбирает альтернативный сквозной рабочий маршрут, например, Тракт В по сети 2 282. В этом примере, как показано на фиг.6, NETFT туннелирует данные вниз по альтернативному маршруту 681 и по сети 2 282 на приемный узел 266. Если состояние отказа исправлено и рабочее состояние сквозной передачи восстановлено на Тракте А, то действующий на передающем узле 216 NETFT может выявить восстановление и снова использовать Тракт А. Дополнительно, любые ответы от узла 266 обратно на узел 216 могут быть туннелированы подобным отказоустойчивым способом посредством NETFT.

На фиг.7 показаны блок-схема и другой пример драйвера отказоустойчивой связи, NETFT 700. Этот пример подобен примеру, показанному на фиг.3, но включает разновидности, как описано ниже. В этом примере программное приложение может не нуждаться в использовании виртуального адреса. Вместо этого приложение может использовать физический адрес назначения, чтобы адресовать пакеты данных на целевой узел.

Адаптер 710 протокола обычно соединяется с адаптером 702 мини-порта (известным также как обрабатывающий элемент драйвера) и с адаптером мини-порта NIC (не показано). Обычно имеется один адаптер протокола для каждой NIC, установленной на узле, каждый адаптер протокола является связываемым с NIC через его адаптер NIC. Поскольку каждый адаптер протокола связан с NIC, он также связан с маршрутом, соединенным с NIC. Адаптер протокола 710 действует, чтобы принять пакеты данных от приложения через обрабатывающий элемент 702 и передать пакеты данных на связанную NIC без потребности туннелирования.

Обрабатывающий элемент 702 обычно анализирует пакет данных, проходящий вниз по сетевому стеку, исследует физический адрес назначения пакета и использует информацию из базы данных 704 маршрутизации, чтобы определить, может ли пакет пересылаться через адаптер 710 протокола или должен быть туннелирован через туннельный адаптер 308 на целевой узел. Обычно, если маршрут, указанный посредством физического адреса назначения, является рабочим для сквозной передачи, пакет данных будет послан по этому маршруту. Иначе может быть выбран альтернативный маршрут, по которому пакет может быть туннелирован.

В этом примере база данных 704 маршрутизации поддерживает отображения физических адресов назначения и маршрутов вместе с приоритетной и другой информацией, как описано выше. Пример таблицы отображения базы данных 704 маршрутизации показан в Таблице 2.

Таблица 2

Тип адреса назначения	Адрес	Приоритет
Физический: Тракт А	172.56.48.38	1
Физический: Тракт В	197.71.48.39	2

Что касается Таблицы 2 и фиг.2, в Таблице 2 показана примерная таблица отображения, которую мог бы использовать действующий на узле 216 NETFT. В

Таблице 2 показано отображение, включающее в состав физический адрес назначения 172.56.48.38, относящийся к Тракту А на узел 266, и физический адрес назначения 197.71.48.39, относящийся к Тракту В на узел 266. Тракт А показан с первым приоритетом и Тракт В со вторым приоритетом.

При посылке данных от узла 216 на узел 266 такая таблица отображения обычно используется в пересылке (или туннелировании, если необходимо) пакета данных, посылаемого от узла 266 на физический адрес назначения 172.56.48.38. Если маршрут, связанный с исходным адресом назначения, является рабочим, пакет данных обычно пересылается на узел назначения без туннелирования. Если этот маршрут не является доступным, то пакет данных посылается от узла 266 по альтернативному маршруту по физическому адресу назначения 197.71.48.39 через туннелирование. Другие аспекты NETFT 700 в целом подобны таковым для NETFT, описанного для фиг.3.

На фиг.8 показана блок-схема, показывающая примерную архитектуру 216 отказоустойчивой связи, включая NETFT 700 и приложение 402. В этом примере приложение 402 посылает пакеты данных на NETFT 700 через стек, используя физический адрес источника и физический адрес 801 назначения, представляющий узел-адресат. Такие исходящие пакеты данных проходят по маршруту 880 от приложения и через сетевой стек на драйвер 700. Драйвер 700 обычно определяет, какой из возможных маршрутов должен получить каждый пакет, обычно используя приоритетную информацию и информацию рабочего состояния маршрута, хранимую в базе данных маршрутизации, и пересылает пакет на целевой узел по маршруту, указанному посредством исходного физического адреса назначения или, если этот маршрут не является сквозной связью, туннелирует пакет по альтернативному маршруту, как указано в этом примере посредством маршрута 882 и NIC2 892.

Приложение 402 может посылать пакет данных через NETFT 700 посредством протокола TCP, как показано на фиг.8. В качестве альтернативы могут использоваться UDP или любой другой протокол. Также, как показано, NETFT 700 может использовать протокол UDP для туннелирования пакетов на целевой узел. В качестве альтернативы, для туннелирования могут использоваться TCP или любой другой протокол. Дополнительно, другие примеры осуществления могут не использовать драйверы NDIS, но могут использовать другие механизмы или архитектуру, чтобы выполнять подобные функции. В заключение, различные элементы сетевого стека и т.п. могут действовать либо в пользовательском режиме или привилегированном режиме, либо как показано или иным образом, либо на системах с эквивалентными режимами работы или без них.

На фиг.9 показана блок-схема организации потока данных через среду 900 отказоустойчивой связи, включающую в состав узел-источник 816 и узел-адресат 966, соединенные через Тракт А по сети 1 202 и Тракт В по сети 2 282. В этой примерной среде 900 данные показаны посылаемыми от приложения, действующего на узле 216, на приложение, принимающее по физическому адресу назначения на узле 266. Пакеты данных проходят вниз по сетевому стеку, действующему на узле 216 с использованием протокола TCP, в NETFT, как показано посредством маршрута 901. Если, как показано, Тракт А является выбранным маршрутом, NETFT пересылает пакеты данных с использованием физического адреса назначения, предоставленного приложением, через NIC1 от узла 216 по Тракту А и сеть 1 202 через канал связи 201. Данные затем проходят через сеть 1 202, по каналу связи 203 и на узел 966, проходя вверх по сетевому стеку, действующему на узле 966, как показано маршрутом 903. Данные затем проходят через NETFT и драйвер протокола (драйвер протокола для

того же протокола, который использовался на передающей стороне в качестве протокола для посылки) и вверх на приложение. Ответы обычно проходят в обратном порядке.

На фиг.10 показана блок-схема организации потока данных через среду 900 отказоустойчивой связи, показанную на фиг.9, с дополнением нескольких возможных отказов 1010, 1012, 1020, 1022 и 1030 связи. Другие отказы связи также возможны. Отказ 1010 указывает отказ NIC1, действующей на передающем узле 816. Такой отказ может происходить, если NIC1 должна быть удалена из узла, его драйвер NIC терпит отказ, непосредственно NIC терпит отказ, или подобное. Отказ может быть выявлен посредством NETFT через указание события, например, события PnP или подобного и/или отказа контрольного пакета. В такой ситуации считают, что Тракт А терпит отказ, и NETFT выберет альтернативный рабочий маршрут сквозной передачи.

Отказ 1020 указывает отказ связанности сетевой среды передачи с NIC1 узла 816. Этот отказ может быть из-за разъединяемого от NIC1 кабеля, от разъединенного кабеля от некоторого устройства сети 1 202, от выключаемого или отказывающего устройства на сетевой стороне, соединенного с кабелем, или подобного. Этот тип отказа также может быть выявлен посредством NETFT через указание события, например, события PnP или подобного, и/или отказа контрольного пакета и выбранного дополнительного маршрута.

Отказ 1030 указывает отказ некоторого типа в сети 202, который имеет следствием, что пакеты данных не в состоянии достичь узел-адресат 966. В этом случае отказа передающий узел 816 тем не менее может быть соединен с сетью 202 с указанием надлежащего восприятия среды передачи, хотя Тракт А стал нарушенным далее вниз по сети. Получив такой отказ, действующий на передающем узле 816 NETFT может не выявить отказ через указание события, например, события PnP или подобного, если локальные указания показывают связность с сетью 202 как хорошую, но может выявить отказ через отказ контрольного пакета для Тракта А.

Отказ 1022 канала 203 связи и отказ 1012 NIC1, действующей на приемном узле 966, является подобным соответствующим отказам, показанным для узла 816. Но эти отказы, не являющиеся локальными для узла 816, могут не быть выявлены через указание события, а могут быть выявлены через отказ контрольного пакета.

Любой из этих отказов и другие отказы могут быть выявлены посредством действующего на узле 816 NETFT и иметь результатом выбор альтернативного рабочего маршрута сквозной передачи, например, Тракта В по сети 2 282. В этом примере, как показано на фиг.10, NETFT туннелирует данные вниз по альтернативному маршруту 1081 и по сети 2 282 на приемный узел 966. Если состояние отказа исправлено и восстановлено рабочее состояние сквозной передачи на Тракте А, то действующий на передающем узле 816 NETFT может выявить восстановление и снова использовать Тракт А. Дополнительно, любые ответы от узла 966 обратно на узел 816 могут пересылаться или туннелироваться в зависимости от рабочего состояния Тракт А и Тракта В подобным отказоустойчивым образом посредством его NETFT.

На фиг.11 показана блок-схема примерной вычислительной среды 1100, в которой описанная выше технология может быть осуществлена. Подходящая вычислительная среда может быть осуществлена с помощью многих универсальных или специализированных систем. Примеры известных систем могут включать в себя, но не ограничиваться указанным, персональные компьютеры (ПК, PC), переносные или портативные устройства, микропроцессорные системы, многопроцессорные системы,

серверы, рабочие станции, устройства бытовой электроники, телевизионные абонентские приставки и т.п.

Вычислительная среда 1100 обычно включает в себя универсальную вычислительную систему в виде вычислительного устройства 1101, соединенного с различными периферийными устройствами 1102, 1103, 1104 и т.п. Система 1100 может соединяться с различными устройствами 1103 ввода, включая клавиатуры и указательные устройства, такие как мышь или шаровой манипулятор, через один или несколько интерфейсов 1112 ввода/вывода (I/O). Компоненты вычислительного устройства 1101 могут включать в себя один или несколько процессоров (включая блоки центрального процессора (ЦПУ), графические процессоры данных (GPU), микропроцессоры (uP), и т.п.) 1107, системную память 1109 и системную шину 1108, которая обычно соединяет различные компоненты. Процессор 1107 обычно обрабатывает или исполняет различные машиноисполняемые команды, чтобы управлять действием вычислительного устройства 1101 и взаимодействовать с другими электронными и/или вычислительными устройствами, системами или средой (не показано) через различные соединения связи, такого как сетевое соединение 1114 или подобное. Системная шина 1108 представляет несколько типов шинных структур, включая в них шину памяти или контроллер памяти, периферийную шину, последовательную шину, ускоренный графический порт, процессорную или локальную шину, использующую любую из множества шинных архитектур, и т.п.

Системная память 1109 может включать в себя компьютерные носители в виде энергозависимого запоминающего устройства, например, оперативное запоминающее устройство (ОЗУ, RAM), и/или энергонезависимого запоминающего устройства, такого как постоянное запоминающее устройство (ПЗУ, ROM) или флэш-память. Базовая система 833 ввода-вывода (BIOS) может храниться в энергонезависимом ЗУ или подобном. Системная память 1109 обычно хранит данные, машиноисполняемые команды и/или программные модули, содержащие машиноисполняемые команды, которые непосредственно доступны для одного или нескольких процессоров 1107 и/или обрабатываются ими в данный момент времени.

Запоминающие устройства 1104 и 1110 большой емкости могут быть соединены с вычислительным устройством 1101 или включены в состав вычислительного устройства 1101 через соединение с системной шиной. Такие ЗУ 1104 и 1110 большой емкости могут включать в себя накопитель на магнитном диске, который осуществляет считывание со съемного, энергонезависимого магнитного диска (например, с "гибкого диска") 1105 или запись на него, и/или накопитель на оптическом диске, который осуществляет считывание с энергонезависимого оптического диска или запись на такой диск, например, компакт-диск (CD ROM), цифровой диск универсального назначения (DVD, ROM) 1106. В качестве альтернативы, ЗУ большой емкости, например, накопитель 1110 на жестком диске, может включать в себя несъемный носитель данных. Другие ЗУ большой емкости могут включать в себя платы памяти, карты памяти, накопители на магнитной ленте, и т.п.

Несколько компьютерных программ, файлов, структур данных и т.п. могут храниться на накопителе на жестком диске 1110, других устройствах 1104, 1105, 1106 хранения данных и системной памяти 1109 (обычно ограниченными доступным пространством) включая, в качестве примера, операционные системы, прикладные программы, файлы данных, структуры каталогов и машиноисполняемые команды.

Устройства вывода, например, устройство 110 отображения, могут быть соединены

с вычислительным устройством 1101 через интерфейс, например, видеоадаптер 1111. Другие типы устройств вывода могут включать в себя принтеры, аудиовыходы, тактильные устройства или другие сенсорные устройства вывода, или подобное. Устройства вывода могут давать вычислительному устройству 1101 возможность взаимодействовать с человеком-оператором или другими машинами или системами. Пользователь может осуществлять взаимодействие с вычислительной средой 1100 через несколько различных устройств 1103 ввода, таких как клавиатура, мышь, джойстик, игровая панель, порт данных и т.п. Эти и другие устройства ввода могут быть соединены с процессором 1107 через интерфейсы 1112 ввода/вывода, который может быть соединен с системной шиной 1108 и может быть соединен посредством других интерфейсами и шинных структур, таких как параллельный порт, игровой порт, универсальная последовательная шина (USB), шина FireWire, инфракрасный порт и т.п.

Вычислительное устройство 1101 может действовать в сетевой среде через соединения связи с одним или несколькими удаленными вычислительными устройствами через одну или несколько локальных сетей (ЛВС, LAN), глобальных сетей (ГВС, WAN), сетей (устройств) хранения данных (SAN), Internet, каналов радиосвязи, оптический линий связи и т.п. Вычислительное устройство 1101 может быть соединено с сетью через сетевой адаптер 1113 или подобное, или, в качестве альтернативы, через модем, цифровую абонентскую линию (DSL), цифровую сеть (ISDN) связи с комплексными услугами, линию связи сети Internet, беспроводную связь, или подобное.

Соединение 1114 связи, например, сетевое соединение, обычно предусматривает сопряжение со средой передачи информации, например, сетью. Среда передачи обычно обеспечивает машиночитаемые и машиноисполняемые команды, структуры данных, файлы, программные модули и другие данные, с использованием модулированного сигнала данных, например, сигнал передачи модулированной информации или другого механизма переноса информации. Термин "модулированный сигнал данных" обычно означает сигнал, который имеет одну или несколько своих характеристик, задаваемых или изменяемых таким образом, чтобы кодировать информацию в сигнале. В качестве примера, а не ограничения, среда передачи может включать в себя проводную среду передачи, например, проводную сеть или непосредственное соединение проводом или подобное, и беспроводные среды передачи, такие как акустические, радиочастотные, инфракрасные, или другие средства беспроводной связи.

Специалисты в данной области техники поймут, что устройства хранения данных, используемые, чтобы обеспечивать машиночитаемые и машиноисполняемые команды и данные, могут быть распределены по сети. Например, удаленный компьютер или устройство хранения данных могут хранить машиночитаемые и машиноисполняемые команды в форме программных приложений и данных. Локальный компьютер может осуществлять доступ к удаленному компьютеру или устройству хранения данных по сети и загружать по сети часть или полное программное приложение или данные, и может исполнять любые машиноисполняемые команды. В качестве альтернативы, локальный компьютер может загружать по сети части программного обеспечения или данных по необходимости, или распределенным образом обрабатывать программное обеспечение путем исполнения некоторых команд на локальном компьютере и некоторых команд на удаленных компьютерах и/или устройствах.

Специалисты в данной области техники также поймут, что путем использования

традиционных способов, все или части машиноисполняемых команд программного обеспечения могут выполняться посредством специализированной электронной схемы, такой как цифровой процессор сигналов (ЦПС, DSP), программируемая логическая матрица (ПЛМ, PLA), дискретные схемы, и подобное. Термин "электронная аппаратура" может включать в себя вычислительные устройства или устройства бытовой электроники, содержащие любое программное обеспечение, микропрограммное обеспечение или подобное, или электронные устройства или схемы, не содержащие программное обеспечение, микропрограммное обеспечение или подобное.

Термин "микропрограммное обеспечение" типично относится к исполняемым командам, программному коду или данным, поддерживаемым в электронном устройстве, например, ПЗУ. Термин "программное обеспечение" типично относится к исполняемым командам, программному коду, данным, приложениям, программам, или подобному, поддерживаемым в или на машиночитаемой среде передачи любого вида. Термин "машиночитаемая среда передачи" типично относится к системной памяти, устройствам хранения данных и связанных с ними носителями, средами передачи данных и т.п.

Формула изобретения

1. Способ обеспечения приложения отказоустойчивой сетевой связи между множеством узлов, содержащий этапы, на которых:

обеспечивают множество исходных маршрутов передачи данных через множество сетей, связанных с множеством узлов, при этом каждая из множества сетей является действующей независимо от всех других из множества сетей и при этом каждый из множества исходных маршрутов передачи отличается от всех других исходных маршрутов передач;

принимают на втором драйвере протокола множества драйверов протоколов, действующем на передающем узле, пакет данных от приложения, действующего на передающем узле, причем передающий узел является одним из множества узлов, пакет данных является адресуемым посредством приложения на адрес одного из множества узлов, отличного от передающего узла; при этом второй драйвер протокола сконфигурирован для поддержки второго маршрутизируемого транспортного протокола; и

выбирают посредством драйвера отказоустойчивых связей, действующего на передающем узле, первый выбранный маршрут из числа множества исходных маршрутов передачи данных, причем первый выбранный маршрут является предпочтительным маршрутом, который вычислен посредством драйвера отказоустойчивых связей на основании приоритетной информации, определенной, по меньшей мере, в части посредством выравнивания по загрузке сетевого трафика по множеству трактов передачи, и при этом передающий узел включает в себя множество сетевых интерфейсных плат, каждая из множества сетевых интерфейсных плат связана с отдельным одним из множества исходных маршрутов передачи данных и при этом первая сетевая интерфейсная плата (NIC) из множества сетевых интерфейсных плат управляется посредством первого NIC драйвера из множества NIC драйверов, действующих на передающем узле, первый NIC драйвер связан с первым драйвером протокола из множества драйверов протокола, первый драйвер протокола связан с драйвером отказоустойчивых связей, который связан со вторым драйвером протокола, который связан с приложением;

принимают на драйвере отказоустойчивых связей пакет данных из второго драйвера протокола и

обеспечивают посредством драйвера отказоустойчивых связей пакет данных на первый драйвер протокола, основанный на выборе, при этом первый выбранный маршрут связан с первой NIC, которая управляется посредством первого NIC драйвера, который связан с первым драйвером протокола, и при этом первый драйвер протокола сконфигурирован для поддержки первого маршрутизируемого транспортного протокола.

2. Способ по п.1, дополнительно содержащий этапы, на которых:

выявляют событие локальной связности, связанное с одним из множества исходных маршрутов передачи данных; и

указывают, является ли действующим один маршрут из множества исходных маршрутов передачи данных на основании события локальной связности.

3. Способ по п.1, дополнительно содержащий этапы, на которых:

посылают маршрутизируемый контрольный пакет по одному из множества исходных маршрутов передачи данных;

осуществляют мониторинг ответа на маршрутизируемый контрольный пакет и

указывают, является ли один маршрут из множества исходных маршрутов передачи данных сквозным рабочим на основании мониторинга ответа.

4. Способ по п.1, в котором адресом является адрес по протоколу Internet версии 4 или адрес по протоколу Internet версии 6.

5. Способ по п.1, в котором пакетом данных, принятым от приложения, является пакет по протоколу управления передачей или пакет по протоколу пользовательских дейтаграмм.

6. Способ по п.1, в котором машиноисполняемые команды для выполнения способа сохраняют на машиночитаемом носителе.

7. Способ по п.1, в котором адресом является виртуальный адрес, ассоциированный со вторым драйвером протокола на передающем узле.

8. Способ по п.7, дополнительно содержит этап маршрутизируемого туннелирования пакета данных по первому выбранному маршруту без требования, чтобы приложение было осведомлено о том, какой маршрут из множества исходных маршрутов передачи данных является выбранным маршрутом.

9. Способ по п.8, в котором пакетом данных, туннелируемым с маршрутизацией по первому выбранному маршруту, является пакет по протоколу управления передачей или пакет по протоколу пользовательских дейтаграмм.

10. Способ по п.7, дополнительно содержащий этапы, на которых:

выявляют отказ первого выбранного маршрута;

выбирают второй выбранный маршрут из числа множества исходных маршрутов передачи данных, причем второй выбранной маршрут является и сквозным рабочим и предпочтительным маршрутом;

обеспечивают посредством драйвера отказоустойчивых связей пакет данных на третий драйвер протокола множества драйверов протокола, основанный на выборе второго выбранного маршрута, при этом второй выбранный маршрут связан со второй NIC множества NIC, которая управляется посредством второго NIC драйвера множества NIC драйверов, который связан с третьим драйвером протокола, и при этом третий драйвер протокола сконфигурирован для поддержки третьего маршрутизируемого транспортного протокола; и

туннелируют с маршрутизацией пакет данных по второму выбранному маршруту

без требования, чтобы приложение было осведомлено, какой маршрут из одного или нескольких исходных маршрутов передачи данных является вторым выбранным маршрутом.

11. Способ по п.1, в котором первый выбранный маршрут выбирают на основании физического адреса назначения, включенного вместе с пакетом данных, физический адрес назначения ассоциирован со вторым драйвером протокола на передающем узле.

12. Способ обеспечения приложения отказоустойчивой сетевой связи между множеством узлов, содержащий этапы, на которых:

обеспечивают множество исходных маршрутов передачи данных через множество сетей, связанных с множеством узлов, при этом каждая из множества сетей является действующей независимо от всех других из множества сетей и при этом каждый из множества исходных маршрутов передачи отличается от всех других исходных маршрутов передачи;

принимают на первом драйвере протокола множества драйверов протоколов, действующем на передающем узле, пакет данных, приемный узел включает в себя драйвер отказоустойчивых связей и является одним из множества узлов, пакет данных предназначается для приложения, при этом приложение действует на принимающем узле, и при этом пакет данных был отослан из передающего узла множества узлов через первый маршрутизируемый транспортный протокол по первому выбранному маршруту из множества исходных маршрутов передачи, и при этом первый драйвер протокола сконфигурирован для поддержки первого маршрутизируемого транспортного протокола;

определяют, что пакет данных был туннелирован посредством передающего узла; пересылают пакет данных посредством первого драйвера протокола в ответ на определение к драйверу отказоустойчивых связей;

обеспечивают посредством драйвера отказоустойчивых связей пакет данных на второй драйвер протокола множества драйверов протокола, при этом первый выбранный маршрут связан с первой сетевой интерфейсной платой (NIC) из множества NIC, действующей на приемном узле, первая NIC управляемая посредством первой NIC драйвера множества NIC драйверов, действующих на приемном узле, первый NIC драйвер связан с первым драйвером протокола который связан с драйвером отказоустойчивых связей, который связан со вторым драйвером протокола, который связан с приложением, и при этом второй драйвер протокола сконфигурирован для поддержки второго маршрутизируемого транспортного протокола.

13. Способ по п.12, дополнительно содержащий этапы, на которых:

принимают маршрутизируемый контрольный пакет на приемном узле по одному маршруту из множества исходных маршрутов передачи данных и

отвечают ответом на маршрутизируемый контрольный пакет, ответ, указывающий действующее состояние сквозной передачи для одного маршрута из множества исходных маршрутов передачи данных.

14. Способ по п.12, в котором пакетом данных является пакет по протоколу управления передачей или пакет по протоколу пользовательских дейтаграмм.

15. Способ по п.12, в котором машиноисполняемые команды, предназначенные для выполнения способа, хранятся на машиночитаемом носителе.

16. Система для обеспечения приложения отказоустойчивой сетевой связи между множеством узлов, содержащая:

первый узел из множества узлов;

первый драйвер отказоустойчивых связей, связанный с первым сетевым стеком и действующий на первом узле; и

первый драйвер отказоустойчивых связей, связанный со вторым драйвером отказоустойчивых связей, действующим на втором узле из множества узлов,

первый и второй драйверы отказоустойчивых связей связаны через множество исходных маршрутов передачи данных по множеству сетей, связанных с множеством узлов, при этом каждая из множества сетей является операционно независимой от других из множества сетей, и при этом каждый из множества исходных маршрутов передачи отличается от всех других исходных маршрутов передачи, и при этом передающий узел включает в себя множество сетевых интерфейсных плат, каждая из множества сетевых интерфейсных плат связана с отличным одним из множества исходных маршрутов передачи, и при этом первая сетевая интерфейсная плата (NIC) из множества сетевых интерфейсных плат управляется посредством первого NIC драйвера из множества NIC драйверов, действующих на первом узле, первый NIC драйвер связан с первым драйвером протокола из множества драйверов протокола, действующих на первом узле, первый драйвер протокола связан с первым драйвером отказоустойчивых связей, который связан со вторым драйвером протокола из множества драйверов протокола, второй драйвер протокола связан с приложением, при этом каждый из первого драйвера протокола и второго драйвера протокола сконфигурирован для поддержки маршрутизируемого транспортного протокола.

17. Система по п.16, в которой первый отказоустойчивый драйвер содержит:

адаптер мини-порта, связанный с приложением через первый сетевой стек;

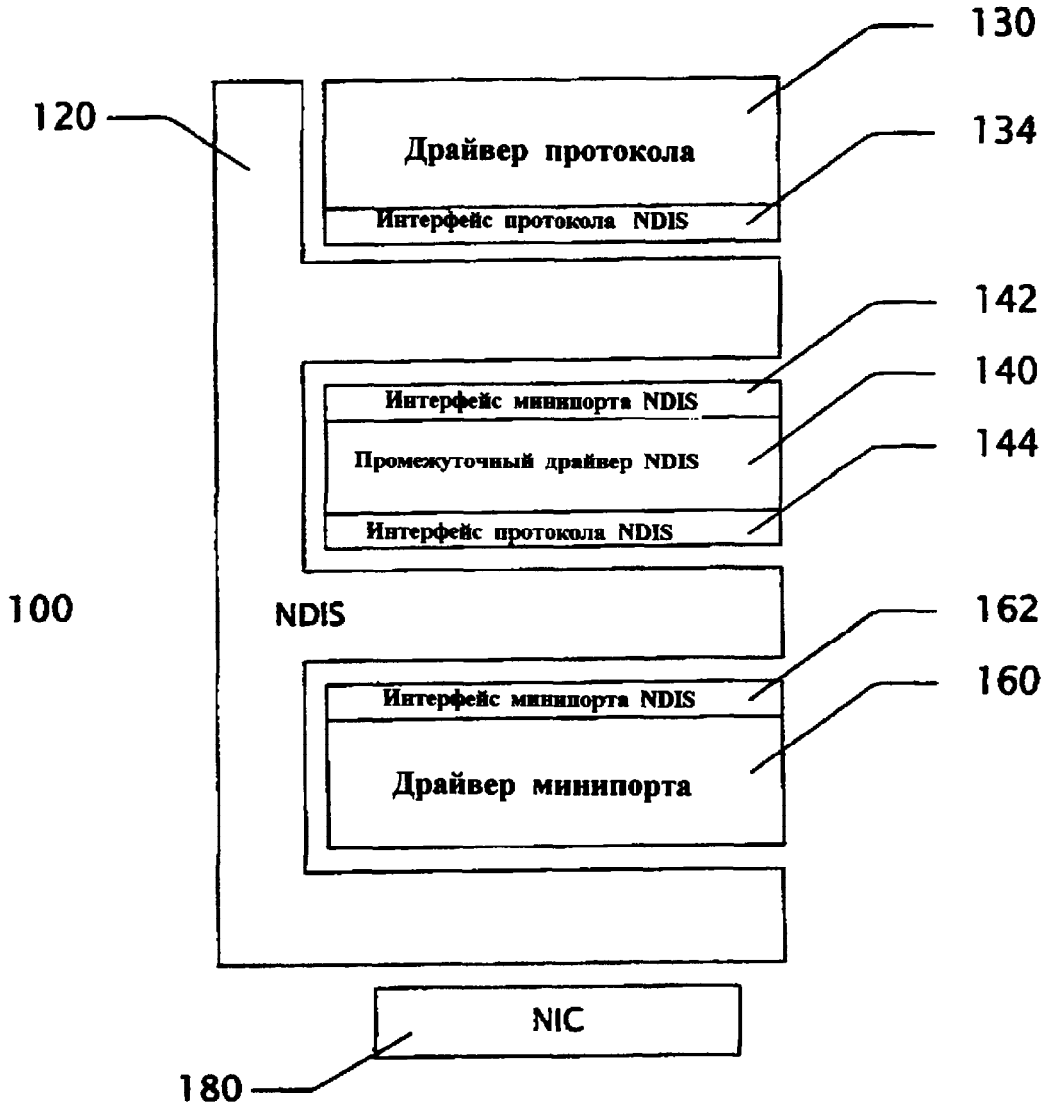
базу данных маршрутизации, связанную с адаптером мини-порта, включающую в себя:

элемент, представляющий маршрут с первого узла на второй узел и включающий в себя физический адрес второго узла, маршрут является одним маршрутом из множества исходных маршрутов передачи данных; и

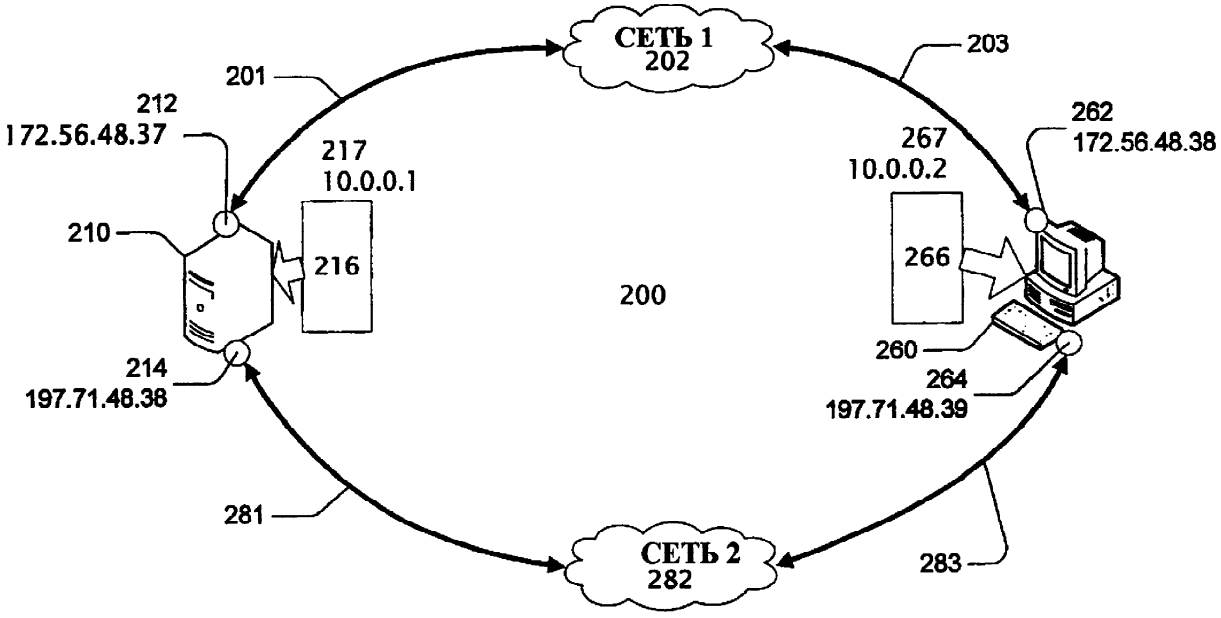
указание действующего состояния сквозной передачи для маршрута с первого узла на второй узел;

адаптер протокола, связанный с адаптером мини-порта и связанный с одной сетью из множества сетей через первую сетевую интерфейсную плату (NIC), действующую на первом узле, одну из множества сетей, обеспечивающих маршрут от первого узла ко второму узлу; и

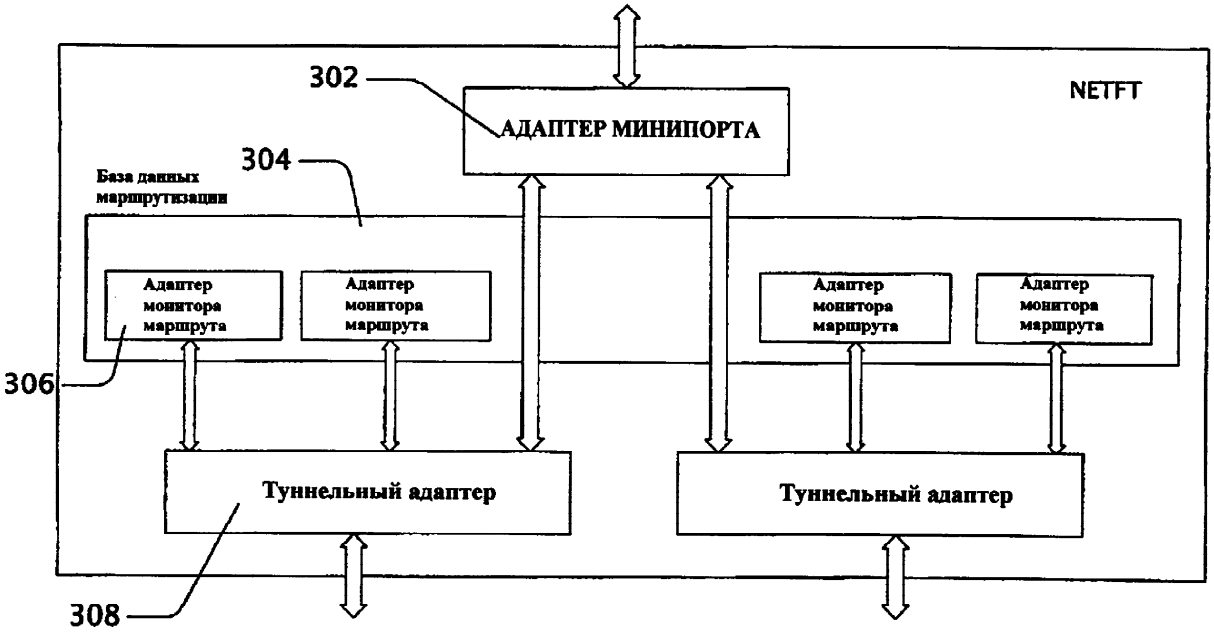
туннельный адаптер, связанный с адаптером мини-порта и связанный с одним вторым из множества сетей через третий драйвер протокола, связанный со второй NIC множества NIC, при этом третий драйвер протокола сконфигурирован для поддержки другого маршрутизируемого транспортного протокола.


130
120
Драйвер протокола
134
Интерфейс протокола NDIS
142
140
Интерфейс минипорта NDIS
Промежуточный драйвер NDIS
144
Интерфейс протокола NDIS
100
NDIS
162
160
Интерфейс минипорта NDIS
Драйвер минипорта
NIC
180

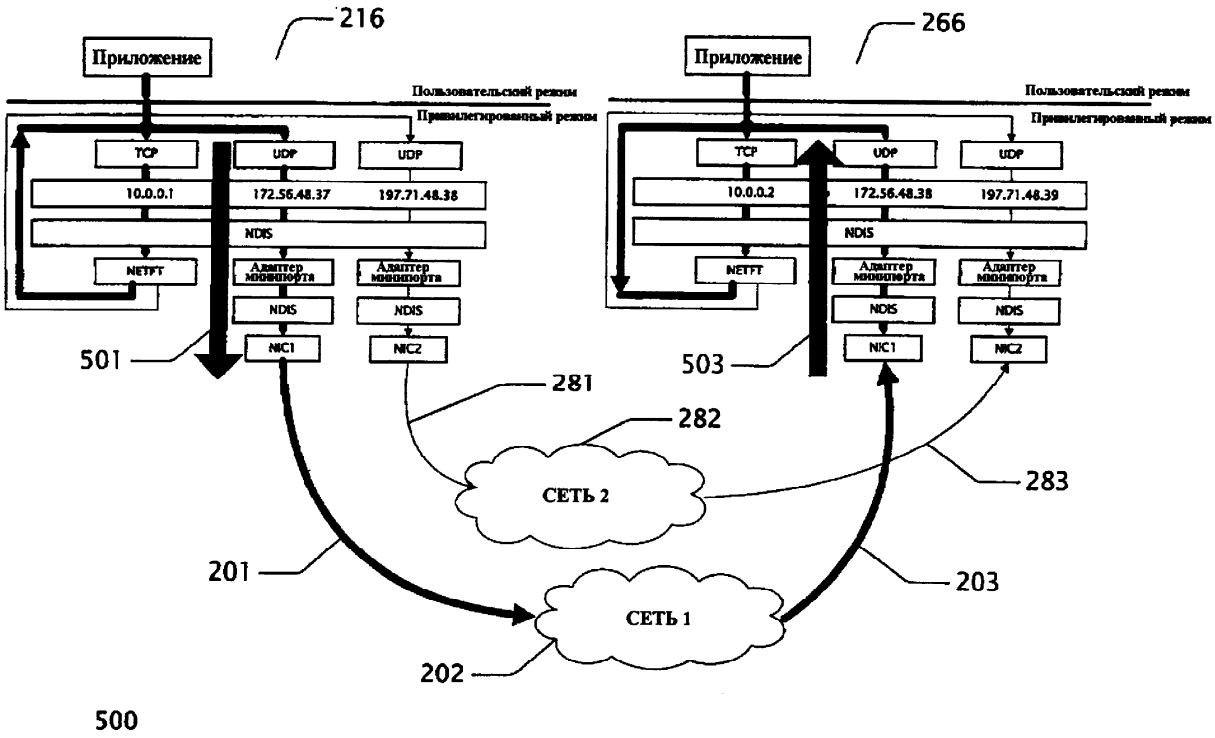
ФИГ. 1


СЕТЬ 1
202
201
203
212
172.56.48.37
217
10.0.0.1
267
10.0.0.2
262
172.56.48.38
210
216
266
200
214
197.71.48.38
260
264
197.71.48.39
281
283
СЕТЬ 2
282

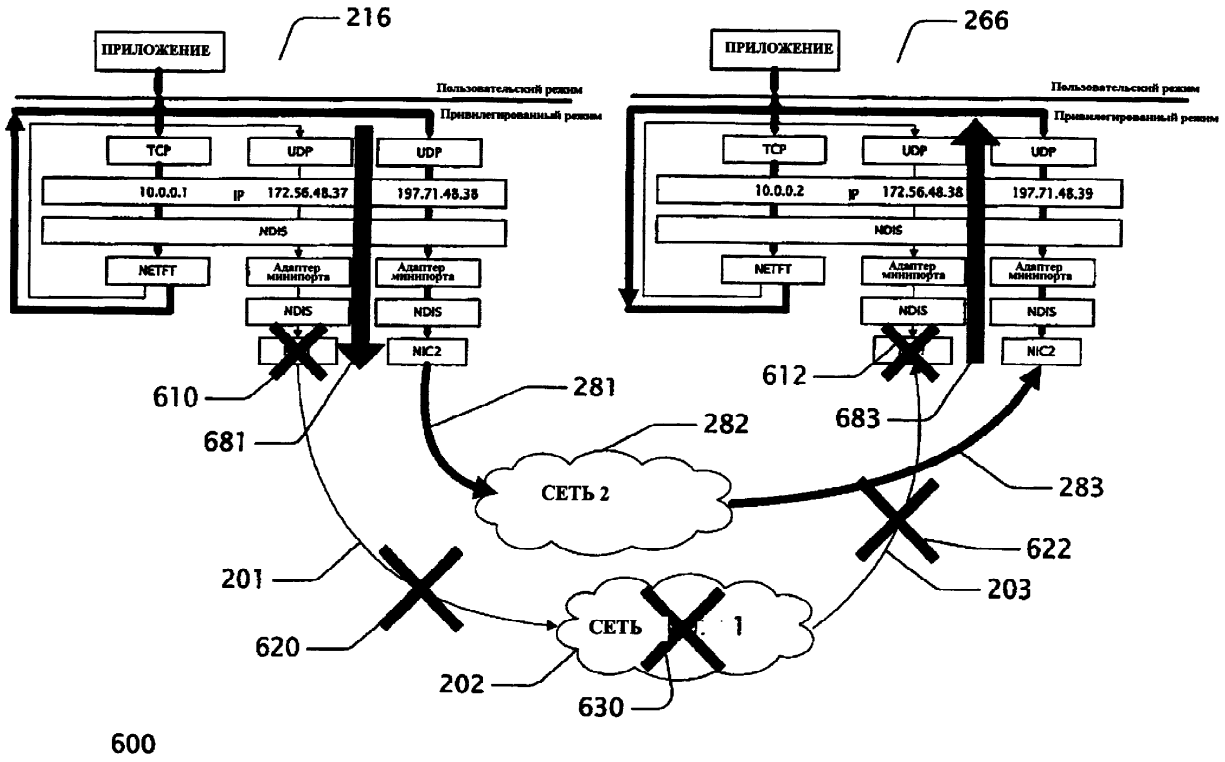
ФИГ. 2


302
NETFT
АДАПТЕР МИНИПОРТА
304
База данных маршрутизации
Адаптер монитора маршрута
Адаптер монитора маршрута
Адаптер монитора маршрута
Адаптер монитора маршрута
306
Туннельный адаптер
Туннельный адаптер
308
300

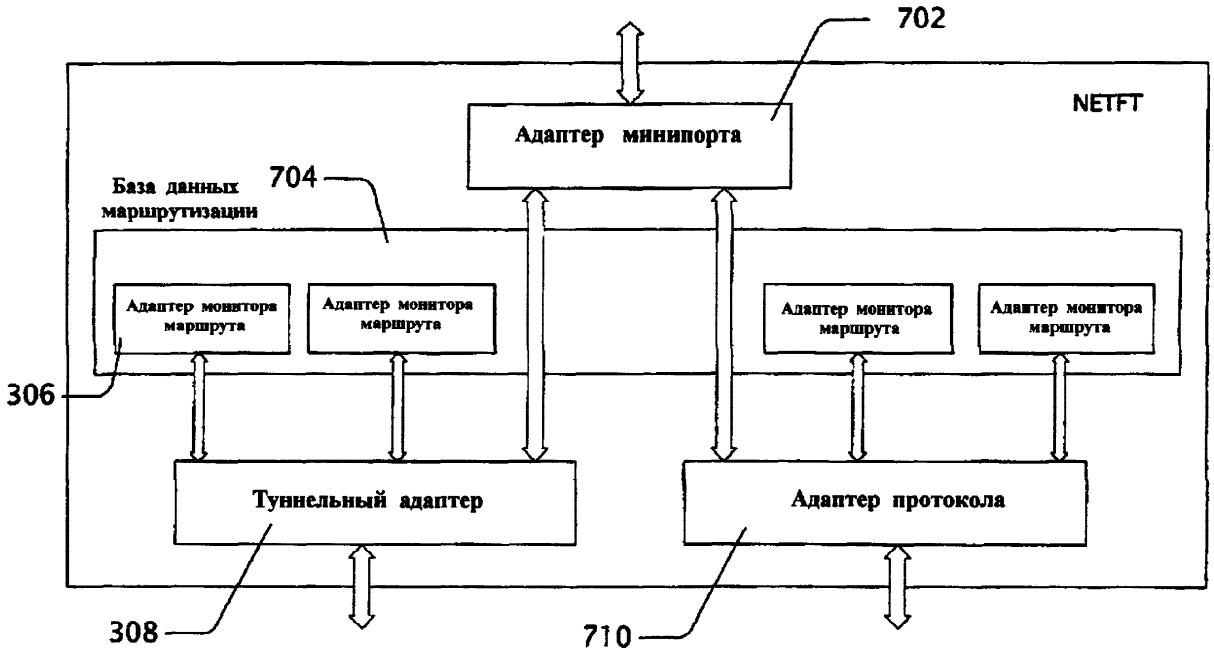
ФИГ. 3


216
266
Приложение
Приложение
Пользовательский режим
Привилегированный режим
TCP
UDP
UDP
10.0.0.1
172.56.48.37
197.71.48.38
10.0.0.2
172.56.48.38
197.71.48.39
NDIS
NETFT
Адаптер минипорта
NIC1
NIC2
501
503
281
282
283
СЕТЬ 2
201
203
СЕТЬ 1
202
500

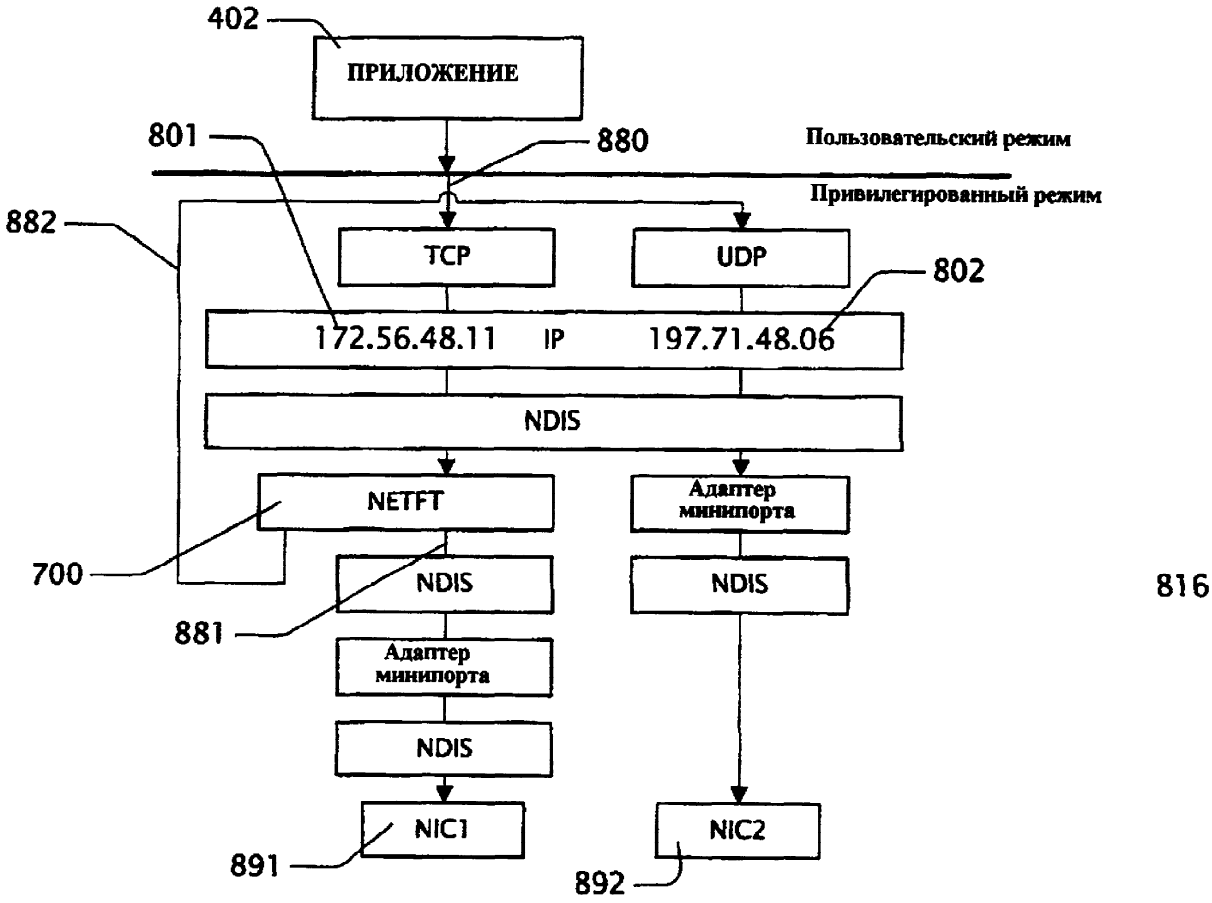
ФИГ. 5


ПРИЛОЖЕНИЕ
216
Пользовательский режим
Привилегированный режим
TCP
UDP
UDP
10.0.0.1
IP
172.56.48.37
197.71.48.38
NDIS
NETFT
Адаптер минипорта
Адаптер минипорта
NDIS
NDIS
NIC2
ПРИЛОЖЕНИЕ
266
Пользовательский режим
Привилегированный режим
TCP
UDP
UDP
10.0.0.2
IP
172.56.48.38
197.71.48.39
NDIS
NETFT
Адаптер минипорта
Адаптер минипорта
NDIS
NDIS
NIC2
610
681
281
282
612
683
283
СЕТЬ 2
622
203
201
620
СЕТЬ 1
202
630
600

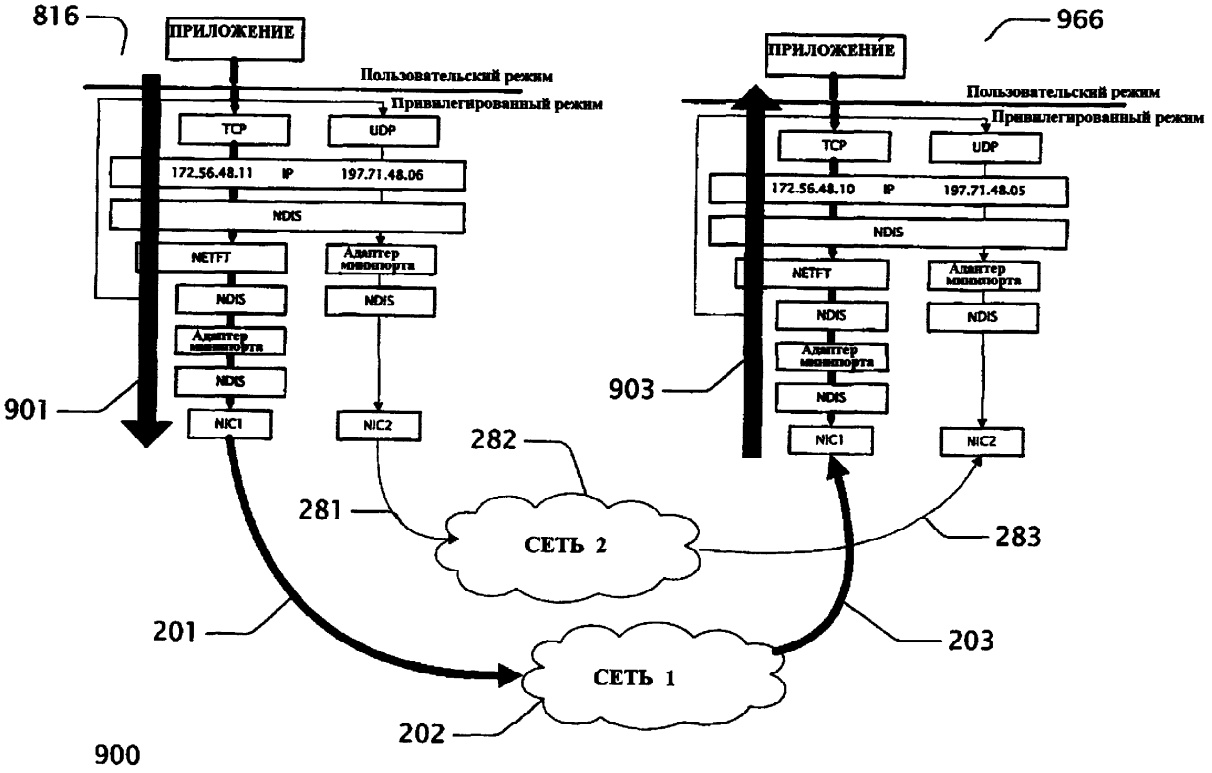
ФИГ. 6


702
NETFT
Адаптер минипорта
База данных маршрутизации
704
Адаптер монитора маршрута
Адаптер монитора маршрута
Адаптер монитора маршрута
Адаптер монитора маршрута
306
Туннельный адаптер
Адаптер протокола
308
710
700

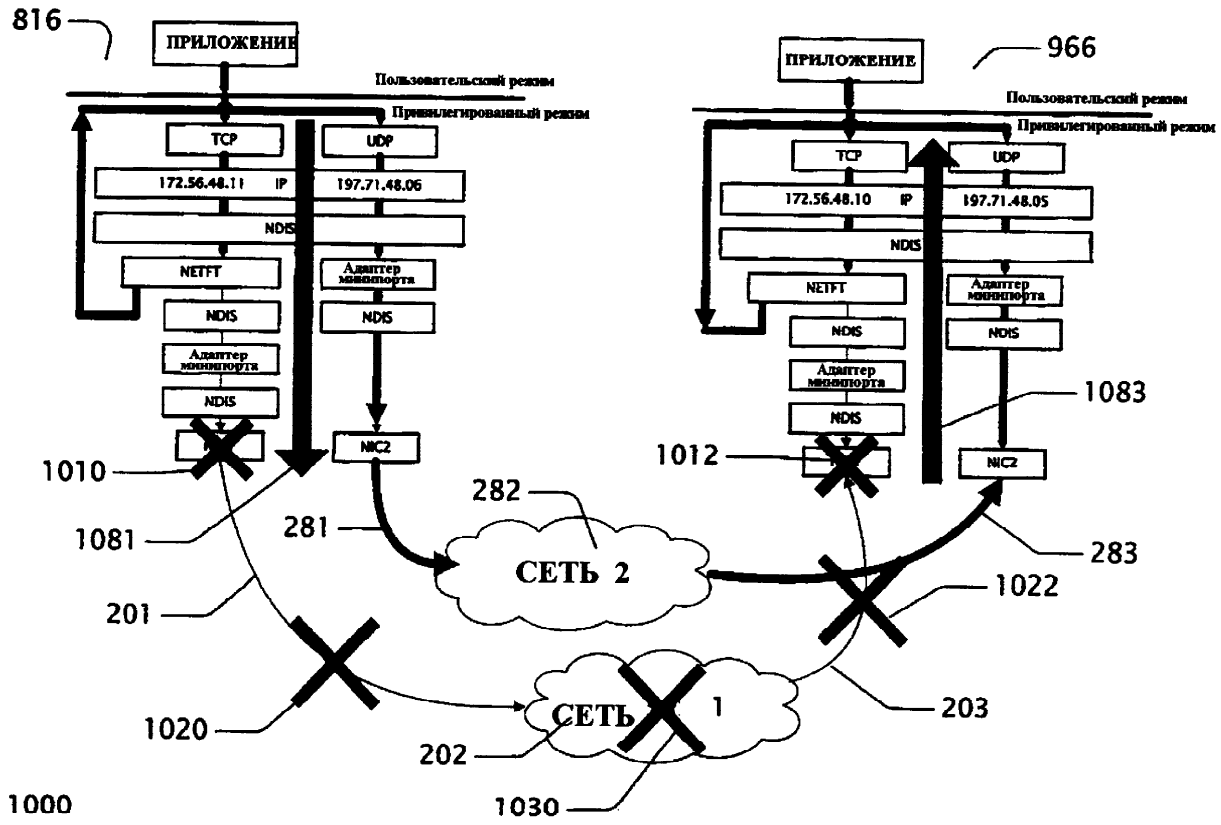
ФИГ. 7


402
ПРИЛОЖЕНИЕ
801
880
Пользовательский режим
Привилегированный режим
882
TCP
UDP
802
172.56.48.11
IP
197.71.48.06
NDIS
NETFT
Адаптер минипорта
700
NDIS
NDIS
816
881
Адаптер минипорта
NDIS
NIC1
NIC2
891
892

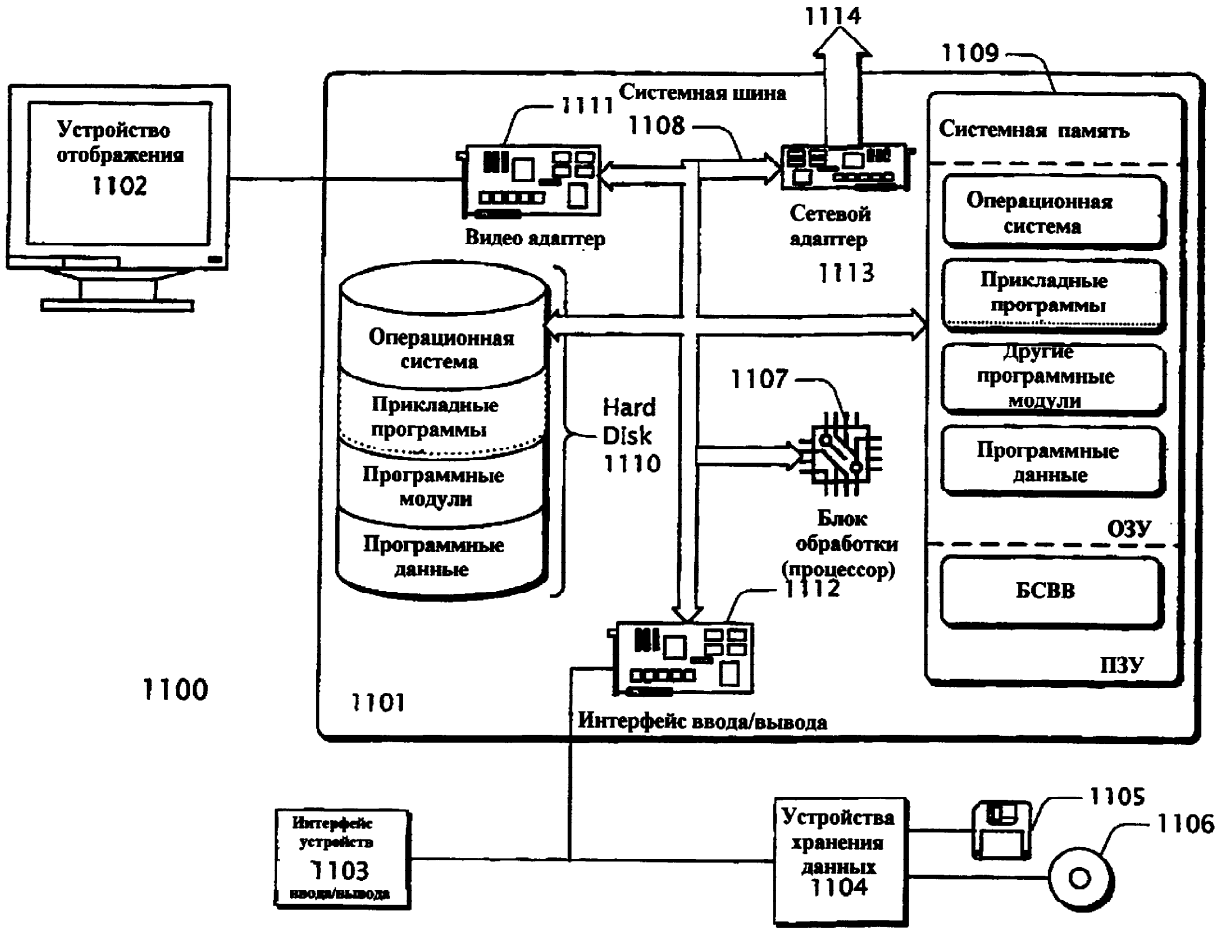
ФИГ. 8


816
ПРИЛОЖЕНИЕ
Пользовательский режим
Привилегированный режим
TCP
UDP
172.56.48.11
IP
197.71.48.06
NDIS
NETFT
Адаптер минипорта
NDIS
NDIS
Адаптер минипорта
NDIS
NIC1
NIC2
901
966
ПРИЛОЖЕНИЕ
Пользовательский режим
Привилегированный режим
TCP
UDP
172.56.48.10
IP
197.71.48.05
NDIS
NETFT
Адаптер минипорта
NDIS
NDIS
Адаптер минипорта
NDIS
NIC1
NIC2
903
282
281
СЕТЬ 2
283
201
203
СЕТЬ 1
202
900

ФИГ. 9


816
ПРИЛОЖЕНИЕ
Пользовательский режим
Привилегированный режим
TCP
UDP
172.56.48.11
IP
197.71.48.06
NDIS
NETFT
Адаптер мини-порта
NDIS
Адаптер мини-порта
NDIS
NIC2
1010
1081
201
1020
281
282
СЕТЬ 2
СЕТЬ 1
202
1030
1000
966
ПРИЛОЖЕНИЕ
Пользовательский режим
Привилегированный режим
TCP
UDP
172.56.48.10
IP
197.71.48.05
NDIS
NETFT
Адаптер мини-порта
NDIS
Адаптер мини-порта
NDIS
NIC2
1012
1083
283
1022
203

ФИГ. 10


1114
1109
Устройство отображения 1102
1111
Системная шина
1108
Системная память
Видео адаптер
Сетевой адаптер 1113
Операционная система
Прикладные программы
Другие программные модули
Программные данные
ОЗУ
БСВВ
ПЗУ
Операционная система
Прикладные программы
Программные модули
Программные данные
Hard Disk 1110
1107
Блок обработки (процессор) 1112
1100
1101
Интерфейс ввода/вывода
Интерфейс устройств 1103 ввода/вывода
Устройства хранения данных 1104
1105
1106

ФИГ. 11