



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,  
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2007143547/09, 25.04.2006

(24) Дата начала отсчета срока действия патента:  
25.04.2006

Приоритет(ы):

(30) Конвенционный приоритет:  
26.04.2005 US 11/115,656

(43) Дата публикации заявки: 10.06.2009 Бюл. № 16

(45) Опубликовано: 27.05.2011 Бюл. № 15

(56) Список документов, цитированных в отчете о поиске: US 2005044162 A1, 24.02.2005. RU 2185657 C2, 10.04.2002. «iSCSI Management API», Storage Networking Industry Association, 2004, [найдено 20.10.2010], найдено из Интернета: <URL: <http://www.t11.org/ftp/t11/admin/snia/04-732v0.pdf>>. D.SHEINWALD, J.SATRAN, P.THALER, V.CAVANNA, M.WAKELEY «iSCSI CRC/Checksum Considerations», (см. прод.)

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 26.11.2007

(86) Заявка РСТ:  
US 2006/015847 (25.04.2006)(87) Публикация заявки РСТ:  
WO 2006/116521 (02.11.2006)

Адрес для переписки:

129090, Москва, ул.Б.Спасская, 25, стр.3,  
ООО "Юридическая фирма Городиский и  
Партнеры", пат.пов. А.В.Мицу, рег.№ 364

(72) Автор(ы):

ДЖОГЛЕКАР Абхиджит (US),  
БЕРРИ Франк (US)

(73) Патентообладатель(и):

ИНТЕЛ КОРПОРЕЙШН (US)

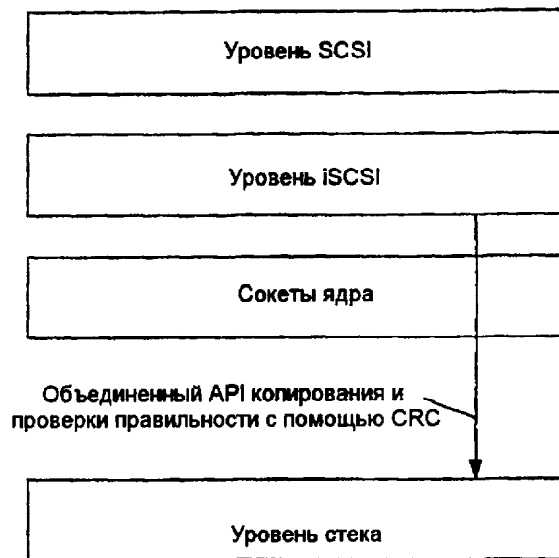
## (54) ТЕХНОЛОГИИ ДЛЯ ОБЕСПЕЧЕНИЯ ПРОВЕРКИ ПРАВИЛЬНОСТИ И ПЕРЕДАЧИ ИНФОРМАЦИИ

(57) Реферат:

Изобретение относится к области сетей передачи данных. Технический результат заключается в расширении функциональных возможностей. Сущность изобретения заключается в том, что проверки правильности

и передачи информации выдают объединенный запрос интерфейса прикладного программирования (API), включающий запрос копирования данных и запрос проверки правильности с помощью циклически избыточного кода (CRC), из первого уровня

обработки протокола в промежуточный уровень; передают объединенный запрос API из промежуточного уровня во второй уровень обработки протокола; и выдают по меньшей мере одну инструкцию логике из второго уровня обработки протокола на основе логики, доступной второму уровню обработки протокола, чтобы выполнить запрос API. 4 н. и 30 з.п. ф-лы, 8 ил.



Фиг. 1С

(56) (продолжение):

07.05.2001, [найдено 20.10.2010], найдено из Интернета: <URL: <http://tools.ietf.org/pdf/draft-sheinwald-iscsi-crc-00.pdf>>.

RU 242007 007 C2

RU 242007 007 C2



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY,  
PATENTS AND TRADEMARKS

**(12) ABSTRACT OF INVENTION**

(21)(22) Application: **2007143547/09, 25.04.2006**

(24) Effective date for property rights:  
**25.04.2006**

Priority:

(30) Priority:  
**26.04.2005 US 11/115,656**

(43) Application published: **10.06.2009 Bull. 16**

(45) Date of publication: **27.05.2011 Bull. 15**

(85) Commencement of national phase: **26.11.2007**

(86) PCT application:  
**US 2006/015847 (25.04.2006)**

(87) PCT publication:  
**WO 2006/116521 (02.11.2006)**

Mail address:

**129090, Moskva, ul.B.Spaskaja, 25, str.3, OOO  
"Juridicheskaja firma Gorodisskij i Partnery",  
pat.pov. A.V.Mitsu, reg.№ 364**

(72) Inventor(s):

**DZhOGLEKAR Abkhidzhit (US),  
BERRI Frank (US)**

(73) Proprietor(s):

**INTEL KORPOREJShN (US)**

**(54) TECHNOLOGIES TO ENSURE VERIFICATION OF CORRECTNESS AND INFORMATION TRANSFER**

(57) Abstract:

FIELD: information technologies.

SUBSTANCE: inspections of correctness and information transfer generate a combined request of an applied programming interface (API), including a request of data copying, and a request to verify correctness with the help of a cyclically redundancy code (CRC), from the first level of the protocol processing into an intermediate level; a combined

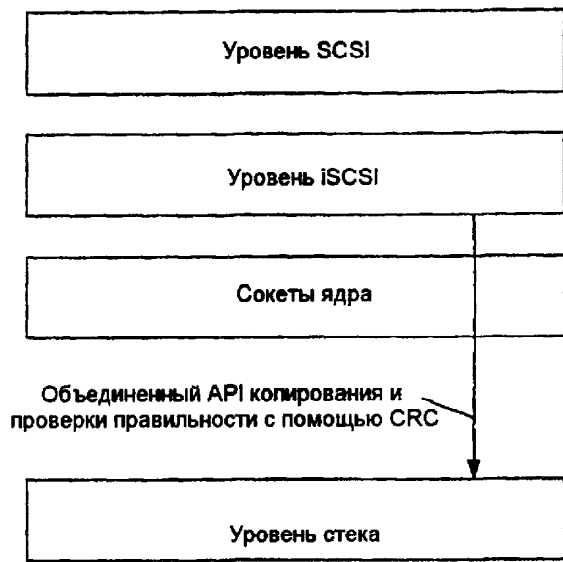
API request is sent from the intermediate level into the second level of the protocol processing; and at least one instruction is issued to a logic from the second level of the protocol processing on the basis of a logic available for the second level of the protocol processing, in order to execute an API request.

EFFECT: expansion of functional capabilities.

34 cl, 8 dwg

RU 2 420 007 C2

RU 2 420 007 C2



Фиг. 1С

Область техники, к которой относится изобретение

Предмет изучения, раскрытый в данном документе, относится к технологиям проверки правильности и передачи информации.

Уровень техники

5 Некоторые компьютеры, которые принимают трафик из сети, используют протокол Интернет-интерфейса малых вычислительных систем (iSCSI). Некоторые  
10 осуществления iSCSI (к примеру, реализованные в Linux® или Microsoft Windows®) требуют передачи данных из сетевого буфера в буфер назначения iSCSI и затем  
15 выдают запрос, чтобы проверить правильность с помощью циклически избыточного кода (CRC) по данным в буферах назначения. В некоторых случаях уровень стека, такого как стек обработки TCP/IP, запрашивает передачу данных из сетевого буфера в буфер назначения и отдельно, слой iSCSI выполняет проверку правильности с помощью CRC по переданным данным (или запрашивает проверку правильности с помощью CRC по переданным данным). Уровень сокетов ядра может использоваться, чтобы отделить уровень iSCSI от уровня стека и коммуникативно связать уровень iSCSI с уровнем стека.

Краткое описание чертежей

20 Фиг.1А изображает примерную вычислительную систему, приспособленную использовать варианты осуществления настоящего изобретения.

Фиг.1В изображает примерное осуществление содержимого памяти хост-компьютера, которая может быть использована в варианте осуществления настоящего изобретения.

25 Фиг.1С изображает примерное функционирование варианта осуществления настоящего изобретения, которое может обеспечить операции передачи данных и проверку правильности с помощью CRC в одном запросе API.

30 Фиг.2-5 изображают примерные операции вариантов осуществления настоящего изобретения, которые могут предоставить операции копирования пакета и проверки правильности с помощью CRC.

Фиг.6 изображает подходящую блок-схему процесса 600 в соответствии с вариантом осуществления настоящего изобретения.

35 Отметим, что использование одинаковых номеров ссылок в разных чертежах указывает одинаковые или похожие элементы.

Подробное описание изобретения

40 Ссылка по всей данной спецификации на "один вариант осуществления" или "вариант осуществления" означает, что отдельный признак, структура или характеристика, описанная в связи с вариантом осуществления, включена, по меньшей мере, в один вариант осуществления настоящего изобретения. Таким образом, появление фразы "в одном варианте осуществления" или "вариант осуществления" в различных местах в этой спецификации необязательно всегда обращается к одному и тому же варианту осуществления. Более того, отдельные признаки, структуры или  
45 характеристики могут быть объединены в один или более вариантов осуществления.

50 Фиг.1А изображает в вычислительной системе 100 подходящую систему, в которой могут использоваться некоторые варианты осуществления настоящего изобретения. Вычислительная система 100 может включать в себя хост-систему 102, шину 116 и сетевой интерфейс 118.

Хост-система 102 может включать в себя набор 105 микросхем, процессор 110, хост-память 112 и хранилище 114. Набор 105 микросхем может обеспечивать связь между процессором 110, хост-памятью 112, хранилищем 114, шиной 116, так же как и

графическим адаптером, который может использоваться для передачи графики и информации для отображения на устройстве отображения (оба не изображены). Например, набор 105 микросхем может включать в себя адаптер хранилища (не изображен), приспособленный обеспечивать связь с хранилищем 114 в соответствии с любым из следующих протоколов: интерфейс малых вычислительных систем (SCSI), стандарт волоконно-оптического канала (FC) и/или последовательная улучшенная технология присоединения (S-ATA).

В некоторых вариантах осуществления набор 105 микросхем включает в себя механизм прямого доступа к памяти (DMA), чтобы выполнить прямые обращения к памяти для информации из и в хост-память и/или главное хранилище хост-системы.

Процессор 110 может быть осуществлен как процессор со сложным набором команд (CISC) или процессор с сокращенным набором команд (RISC), многоядерный или любой другой микропроцессор или центральный процессор.

Хост-память 112 может быть осуществлена как энергозависимое запоминающее устройство, такое как, но не ограниченное этим, оперативное запоминающее устройство (RAM), динамическое оперативное запоминающее устройство (DRAM) или статическое ОЗУ (SRAM).

Хранилище 114 может быть осуществлено как энергонезависимое устройство хранения, такое как, но не ограниченное этим, накопитель на магнитных дисках, накопитель на оптических дисках, ленточный накопитель, внутреннее устройство хранения, присоединенное устройство хранения и/или доступное в сети устройство хранения.

Шина 116 может обеспечивать связь, по меньшей мере, между хост-системой 102 и сетевым интерфейсом 118, так же как и другими периферийными устройствами.

Шина 116 может поддерживать последовательную или параллельную связь. Шина 116 может поддерживать связь узел-узел или узел-много узлов. Шина 116 может быть совместима с шиной соединения периферийных компонентов (PCI), описанной, например, в спецификации локальной шины соединения периферийных устройств (PCI), версия 2.2, 18 декабря 1998 года, доступной от специальной группы по интересам PCI, Портланд, штат Орегон, США (так же, как и ее версии); шина PCI Express, описанная в базовой спецификации PCI Express специальной группы по интересам PCI, версия 1.0a (так же, как и ее версии); PCI-x, описанная в спецификации PCI-X версия 1.0a, 24 июля 2000 года, доступной от вышеупомянутой специальной группы по интересам, Портланд, штат Орегон, США (так же, как и ее версии); и/или универсальная последовательная шина (USB) (и родственные стандарты), так же как и другие стандарты соединения.

Сетевой интерфейс 118 может быть выполнен с возможностью обеспечить соединение между хост-системой 102 и сетью 120 в соответствии с протоколами, поддерживаемыми сетью 120. Сетевой интерфейс 118 может осуществлять соединение с хост-системой 102 с использованием шины 116. В одном варианте осуществления сетевой интерфейс 118 может быть интегрирован в набор 105 микросхем.

Сеть 120 может быть любой сетью, такой как Интернет, интранет, локальная вычислительная сеть (LAN), сеть устройств хранения данных (SAN), глобальная вычислительная сеть (WAN) или беспроводная сеть. Сеть 120 может обмениваться трафиком с сетевым интерфейсом 118 с помощью стандарта Ethernet (описанным в IEEE 802.3 и связанных стандартах) или любого стандарта связи.

Фиг.1В изображает примерное осуществление содержимого памяти хост-компьютера, которое может использоваться в варианте осуществления настоящего

изобретения. Например, хост-память может хранить операционную систему (OS) 150, стек 152, драйвер 154 устройства, буфер 156 источника, буфер 158 назначения и приложения 160.

5 Некоторые варианты осуществления OS 150 могут включать в себя, по меньшей мере, стек Интернет-интерфейса малых вычислительных систем (iSCSI), способный выполнять обработку и кодирование в соответствии со стандартами iSCSI, уровень сокетов ядра, чтобы обеспечить связь, по меньшей мере, прикладных интерфейсов программирования (API) между стеком iSCSI и уровнем 152 стека. Например, iSCSI 10 описывается, по меньшей мере, в проекте описания 3720 рабочей группы Интернет по IP-хранилищам, озаглавленном "Internet Small Computer Systems Interface (iSCSI)" (апрель 2004 года). Например, подходящие варианты осуществления OS 150 включают в себя, но не ограничиваются этим, операционные системы, совместимые с Linux® или Microsoft Windows®.

15 Стек 152 может обрабатывать пакеты, по меньшей мере, чтобы определить совместимость с TCP/IP в соответствии со стандартами TCP/IP. Например, протокол TCP/IP описывается, по меньшей мере, в публикации, озаглавленной "Transmission Control Protocol: DARPA Internet Program Protocol Specification", подготовленной для Агентства перспективных исследований 20 министерства обороны США (RFC 793, опубликованное в сентябре 1981 года).

В некоторых вариантах осуществления настоящего изобретения уровень iSCSI может предоставлять один запрос интерфейса прикладного программирования (API) стеку 152, чтобы запросить и операцию копирования данных, и операцию проверки 25 правильности с помощью CRC. Один запрос API может определять копирование и контроль CRC множественных данных. Например, данные могут включать в себя пакеты, принятые из сети, и/или другие типы информации. В некоторых вариантах осуществления стек 152 может наблюдать, какая логика (к примеру, программное 30 обеспечение или аппаратные средства и/или комбинации программного обеспечения и аппаратных средств) доступна, чтобы выполнить команды для операций копирования данных и проверки правильности с помощью CRC. Термин "логика" может включать в себя, в качестве примера, программное обеспечение или аппаратные средства и/или комбинации программного обеспечения и аппаратных средств. В некоторых 35 вариантах осуществления стек 152 может выдавать запросы операций копирования данных и проверки правильности с помощью CRC на основе доступных вычислительных ресурсов в ответ на один запрос API как для операций копирования данных, так и проверки правильности с помощью CRC. В некоторых вариантах 40 осуществления механизм DMA доступен для использования, по меньшей мере, в передачах данных, тогда как в некоторых вариантах осуществления CPU используется, чтобы копировать данные. Например, в некоторых вариантах осуществления механизм DMA может выполнять операции проверки правильности с помощью CRC, тогда как в некоторых вариантах осуществления инструкции, выполняемые посредством CPU, используются, чтобы выполнить операции проверки 45 правильности с помощью CRC. В некоторых вариантах осуществления механизм DMA может буферизовать данные и выполнить проверку правильности с помощью CRC по буферизованным данным. API может разрешить уровню iSCSI быть независимым от используемой операционной системы и платформы и еще принимать наиболее 50 эффективное выполнение копирования и контроль CRC от платформы, которая доступна.

В некоторых вариантах осуществления уровень сокетов ядра OS 150 может, по

меньшей мере, модифицироваться, чтобы разрешить передачу частей запроса API, которые запрашивают объединенные операции копирования данных и проверки правильности с помощью CRC.

5 В некоторых вариантах осуществления стек 152 может быть встроен в OS 150. В таких вариантах осуществления стек 152 может модифицироваться, чтобы выполнять API, запрашивающий операции копирования данных и проверки правильности с помощью CRC. В таких вариантах осуществления стек 152 может быть модифицирован, чтобы разрешить стеку наблюдать, какая логика доступна, чтобы  
10 выполнить операции копирования данных и проверки правильности с помощью CRC.

Соответственно, несмотря на необязательные признаки любого варианта осуществления может быть обеспечено ускорение операций, определенных в протоколе iSCSI, при сохранении общих накладываемых ограничений операционной системы между iSCSI и уровнем стека (т.е. уровнем сокетов ядра). Сохранение границ  
15 интерфейса может разрешить стеку гибко администрировать операции копирования данных и проверки правильности с помощью CRC ниже границы интерфейса на основе доступной логики.

Драйвер 154 устройства может быть драйвером устройства для сетевого  
20 интерфейса 118. Драйвер 154 устройства может, по меньшей мере, координировать передачу данных между хост-системой и сетевым интерфейсом.

Буфер 156 источника может хранить данные, которые должны быть переданы в буфер 158 назначения. Например, буфер 156 источника может хранить данные, принятые из сети. Например, в iSCSI буфер 158 назначения может реализовывать  
25 буфер файловой системы.

Приложения 160 могут быть одной или более машиноисполняемыми программами, которые могут использовать данные, по меньшей мере, принятые из или переданные в сеть. Приложение может включать в себя, но не ограничиваться этим, например, веб-  
30 браузер, фильтр ввода/вывода, приложение обслуживания электронной почты, приложение обслуживания файлов или приложение базы данных.

Фиг.1С изображает примерную работу варианта осуществления настоящего изобретения для того, чтобы обеспечить операции копирования данных и проверки правильности с помощью CRC в одном запросе интерфейса прикладного  
35 программирования (API) из уровня iSCSI в уровень стека. В некоторых вариантах осуществления уровень, отличный от уровня iSCSI, может выдавать запрос API. В некоторых вариантах осуществления уровень, отличный от уровня стека, может принимать запрос API. Например, вызов API может иметь следующий формат:

40 `getBufferListWithCRC (copy list, CRC list)`, где:

список копирования (`copy list`) представляет количество данных, которые должны быть скопированы, и сборный список разброса точек назначения, куда данные должны быть скопированы; а

45 список CRC (`CRC list`) представляет длину данных, по которым должен быть выполнен контроль CRC, и сборный список разброса, где расположены данные.

В некоторых вариантах осуществления адрес источника не указывается в списке копирования и CRC, так как данные источника хранятся в последовательном порядке, однако адрес источника может быть определен в других вариантах осуществления. В  
50 некоторых вариантах осуществления содержимое списка копирования и списка CRC может не иметь одинаковой размерности, так что только некоторые элементы в списке могут быть скопированы, а по другим может быть выполнена только проверка правильности с помощью CRC. В некоторых вариантах осуществления проверка

правильности с помощью CRC может быть выполнена по части скопированных данных.

В некоторых вариантах осуществления данные могут включать в себя маркеры (описанные, по меньшей мере, посредством iSCSI) или другие области, и операции проверки правильности с помощью CRC не должны выполняться по маркерам или другим областям. Например, маркеры могут использоваться, чтобы определить границы кадра, и могут использоваться в случае, когда пакеты теряются или разрушены, чтобы доставить данные для использования. API может определить местоположения маркеров посредством смещения к первому маркеру, размера каждого маркера и интервала между маркерами. Например, в одном варианте осуществления, если маркеры присутствуют в данных, которые должны быть скопированы, то контроль CRC не выполняется по маркерам. В некоторых вариантах осуществления, если маркеры не присутствуют в данных, которые должны быть скопированы, то список CRC может быть таким же, что и список копирования.

Фиг.2-5 изображают примерные операции вариантов осуществления настоящего изобретения, которые могут предоставить операции копирования пакета и проверки правильности с помощью CRC. Конечно, на фиг.2-5 могут быть скопированы и проверены посредством CRC другие типы информации. Фиг.2 изображает примерную работу варианта осуществления настоящего изобретения, которая предоставляет операции копирования данных и проверки правильности с помощью CRC. В этом примере вычислительные ресурсы, доступные уровню стека, не включают в себя доступность использования механизма DMA. Соответственно, уровень стека может отвечать на запрос API копирования и проверки правильности с помощью CRC, выдавая инструкцию CPU, чтобы выполнить предварительную выборку в кэш-пространстве памяти буфера назначения, в которое должны быть записаны данные, и инструкцию логике проверки правильности с помощью CRC, чтобы выполнить вычисление CRC по данным, которые должны быть переданы в пространство памяти буфера назначения. Предварительная выборка содержимого буфера назначения параллельно с проверкой правильности с помощью CRC обеспечивает преимущество в скорости, но не является обязательным признаком какого-либо варианта осуществления, отменяя расходуемую время передачу содержимого из буфера назначения в кэш после проверки правильности с помощью CRC.

На этапе 201 уровень 250 iSCSI выдает объединенный API копирования и контроля CRC стеку 252 в соответствии с вариантом осуществления настоящего изобретения. Объединенный запрос API копирования и контроля CRC может определить копирование и контроль CRC для множественных пакетов. Например, объединенный API копирования и контроля CRC может быть осуществлен с использованием API `getBufferListWithCRC`. Например, объединенный запрос API копирования и контроля CRC может определить, что производится копирование и проверка правильности с помощью CRC данных в буфере источника.

На этапе 202, частично в ответ на объединенный запрос API копирования и контроля CRC, стек 252 может запросить операцию проверки правильности с помощью CRC у операционной логики проверки правильности с помощью CRC. Операция проверки правильности с помощью CRC выполняется по данным, сохраненным в буфере источника, которые должны быть переданы в буфер 256 назначения на этапе 204. Например, логика операции проверки правильности с помощью CRC может быть интегрирована в стек 252 и доступна посредством выдачи вызова функции. В других осуществлениях логика операции проверки правильности с

помощью CRC может быть доступна как отдельный элемент.

На этапе 203-A-1, частично в ответ на запрос операции проверки правильности с помощью CRC из этапа 202, логика операции проверки правильности с помощью CRC может выдать запрос на доступ к данным из буфера 254 источника, чтобы по ним  
5 выполнить проверку правильности с помощью CRC. Например, данные могут быть указаны объединенным запросом API копирования и CRC.

На этапе 203-A-2, частично в ответ на этап 203-A-1, запрошенные данные передаются из буфера 254 источника в кэш 258 (показанные как "кэшированная копия  
10 буфера источника").

На этапе 203-A-3 операция проверки правильности с помощью CRC выполняется по данным, сохраненным в кэше 258 (показанным как кэшированная копия буфера источника).

На этапе 203-B-1 стек 252 выдает запрос, чтобы предварительно выбрать данные из  
15 буфера 256 назначения для сохранения в кэше 258. Предварительно выбранные данные могут быть частью буфера 256 назначения, которая должна быть перезаписана данными, которые должны быть переданы из буфера 254 источника в буфер 256 назначения, частично в ответ на объединенный запрос API копирования и  
20 контроля CRC. На этапе 203-B-2, частично в ответ на запрос, чтобы предварительно выбрать данные на этапе 203-B-1, кэш 258 может найти запрошенные данные для предварительной выборки из буфера назначения (найденные данные показаны как "кэшированная копия буфера назначения").

В одном варианте осуществления этапы 203-A-1 и 203-B-1 могут перекрываться по  
25 времени или происходить приблизительно в одно и то же время. В одном варианте осуществления элемент 203-B-2 может перекрываться по времени или происходит, по существу, в то же время, что и этап 203-A-2 и/или 203-A-3.

На этапе 204 данные кэшированной копии буфера источника в кэше 258 передаются  
30 из кэшированной копии буфера источника в кэшированную копию буфера назначения. Например, кэшированная копия буфера источника передается, чтобы перезаписать кэшированную копию буфера назначения. На этапе 205, более позднем по времени, например, всякий раз, когда буфер назначения должен быть очищен от кэша 258, кэшированная копия буфера назначения копируется в буфер 256 назначения,  
35 чтобы выполнить операцию копирования данных, запрошенную в объединенном запросе API копирования и контроля CRC из этапа 201.

На этапе 203-A-4 указание успеха/неудачи копирования и контроля CRC предоставляется уровню iSCSI, частично в ответ на выполнение операций  
40 копирования (т.е. этапа 204) и проверки правильности с помощью CRC (т.е. этапа 203-A-3). Например, указание успеха/неудачи копирования и контроля CRC может быть предоставлено уровню iSCSI в ответ на завершение действий, запрошенных посредством запроса API на этапе 201. Например, указание успеха/неудачи копирования и контроля CRC уровню iSCSI может указывать успех, когда успешное  
45 копирование и проверка правильности с помощью CRC произошли для всех пакетов, идентифицированных в API, выданном на этапе 201, тогда как неудача может быть указана, когда происходит неудача для любого запрошенного копирования и контроля CRC. Указание неудачи может дополнительно указывать, какая операция (к  
50 примеру, копирование или проверка правильности с помощью CRC) не удалась и для какого пакета.

Фиг.3 изображает примерную работу варианта осуществления настоящего изобретения, которая обеспечивает операции копирования данных и проверки

правильности с помощью CRC. В этом примере вычислительные ресурсы, доступные уровню стека, включают в себя доступность использования механизма DMA.

Соответственно, уровень стека может отвечать на запрос копирования и проверки правильности с помощью CRC, выдавая инструкцию механизму DMA выполнить передачу данных из буфера 254 источника в буфер 256 назначения и инструкцию, чтобы выполнить вычисление CRC по переданным данным в отдельной логике проверки правильности с помощью CRC. Могут иметь место, но они не являются обязательными признаками какого-либо варианта осуществления, преимущества по времени при запросе механизма DMA для загрузки данных, по которым запланирована проверка правильности с помощью CRC в последующем в течение, по меньшей мере, части выполнения проверки правильности с помощью CRC.

На этапе 301 уровень 250 iSCSI выдает объединенный запрос API копирования и контроля CRC стеку 252, чтобы запросить копирование и проверку правильности с помощью CRC пакетов 1 и 2. В этом примере пакет 2 запланирован быть скопированным и проверенным по CRC после копирования и проверки правильности с помощью CRC пакета 1. Например, объединенный API копирования и контроля CRC может быть осуществлен с использованием `getBufferListWithCRC` API. Фиг.3 не изображает передачу пакета 1 из буфера 254 источника в буфер 256 назначения с использованием механизма 260 DMA. Передача пакета 1 может иметь место в ответ на запрос API, выданный на этапе 301.

На этапе 302 стек 252 выдает запрос передачи пакета механизму 260 DMA, чтобы передать пакет 2 из буфера 254 источника в буфер 256 назначения. Частично в ответ на запрос передачи пакета 2, на этапе 303-1, механизм 260 DMA находит пакет 2 в буфере 254 источника, и на этапе 303-2 механизм 260 DMA сохраняет пакет 2 в буфере 256 назначения. Например, пакет 2 может быть пакетом, по которому передача данных и проверка правильности с помощью CRC происходит после проверки правильности с помощью CRC пакета 1.

На этапе 304-1 стек 252 запрашивает операцию проверки правильности с помощью CRC по пакету 1, сохраненному в кэше 258. Например, операция проверки правильности с помощью CRC может быть доступна стеку 252. На этапе 304-2 пакет 1, сохраненный в буфере 256 назначения, передается в кэш 258, частично в ответ на начало операции проверки правильности с помощью CRC. Например, на этапе 304-2 CPU, который выполняет операцию проверки правильности с помощью CRC, может запросить передачу пакета 1 в кэш 258. На этапе 304-3 может иметь место проверка правильности с помощью CRC пакета 1. Проверка правильности с помощью CRC пакета 2 не изображена. На этапе 304-4 стек 252 может предоставить указание успеха/неудачи копирования и контроля CRC уровню iSCSI в ответ на завершение операций копирования и проверки правильности с помощью CRC для пакетов 1 и 2. Например, указание успеха/неудачи копирования и контроля CRC может быть предоставлено уровню iSCSI в ответ на завершение действий, запрошенных запросом API на этапе 301. Например, указание успеха/неудачи копирования и контроля CRC уровню iSCSI может указывать успех, когда успешное копирование и проверка правильности с помощью CRC происходят для пакетов 1 и 2, тогда как неудача может быть указана, когда неудача происходит для любого из запрошенных копирования или контроля CRC. Указание неудачи может дополнительно указывать, какая операция (к примеру, копирование или проверка правильности с помощью CRC) не удалась и для какого пакета.

Например, запрос API, выданный на этапе 301, может быть расширен, чтобы

запросить копирование и проверку правильности с помощью CRC пакета 3 в дополнение к копированию и проверке CRC пакетов 1 и 2. В таком случае пакет 3 может быть передан из буфера 254 источника в буфер 256 назначения механизмом 260 DMA, по меньшей мере, перекрываясь по времени с проверкой правильности с помощью CRC пакета 2.

Фиг.4 изображает примерную работу варианта осуществления настоящего изобретения, чтобы обеспечить операции копирования данных и проверки правильности с помощью CRC. В этом примере вычислительные ресурсы, доступные уровню стека, включают в себя механизм DMA с возможностью передачи данных, так же как и выполнения операций проверки правильности с помощью CRC (показанных как DMA с возможностью 262 проверки правильности с помощью CRC). Соответственно, уровень стека может отвечать на запрос копирования и проверки правильности с помощью CRC, выдавая инструкцию механизму DMA выполнить передачу данных из буфера источника в буфер назначения и инструкцию механизму DMA, чтобы выполнить проверку правильности с помощью CRC по данным, которые должны быть переданы.

На этапе 401 уровень 250 iSCSI выдает объединенный запрос API копирования и CRC стеку 252, чтобы запросить копирование и контроль CRC, в соответствии с вариантом осуществления настоящего изобретения. Объединенный запрос API копирования и CRC может определить копирование и CRC для множественных пакетов. Например, объединенный API копирования и контроля CRC может быть осуществлен с использованием API `getBufferListWithCRC`. Например, объединенный запрос API копирования и контроля CRC может определять, что должна производиться передача и проверка правильности с помощью CRC пакета 1. Пакет 1 может быть сохранен в буфере 254 источника.

На этапе 402 стек 252 может запросить операцию копирования у механизма 262 DMA. Например, этап 402 может включать в себя CPU, который выполняет стек 252, выдающий запрос передачи данных механизму 262 DMA. На этапе 403 стек 252 может посредством механизма 262 DMA запросить операцию проверки правильности с помощью CRC по запрошенным данным, которые должны быть переданы на этапе 402.

На этапе 404-1, частично в ответ на запрос на этапе 402, механизм 262 DMA может запросить передачу пакета 1 из буфера 254 источника, а на этапе 404-2 механизм 262 DMA может запросить сохранение пакета 1 в буфере 256 назначения.

На этапе 405-1, частично в ответ на запрос на этапе 403, механизм 262 DMA может запросить пакет 1 из буфера 254 источника. На этапе 405-2, частично в ответ на запрос на этапе 403, механизм 262 DMA может выполнить операцию проверки правильности с помощью CRC по пакету 1. На этапе 405-3 механизм 262 DMA может указать успех/неудачу копирования и контроля CRC стеку 252 в ответ на завершение операций копирования и контроля CRC. На этапе 405-4 стек 252 может указать успех/неудачу копирования и контроля CRC уровню 250 iSCSI в ответ на указание успеха/неудачи копирования и контроля CRC, принятое стеком 252. Например, указание успеха/неудачи копирования и контроля CRC может быть предоставлено уровню iSCSI в ответ на завершение действий, запрошенных в запросе API на этапе 401. Указание успеха/неудачи копирования и контроля CRC уровню iSCSI может указывать успех, когда успешное копирование и проверка правильности с помощью CRC происходят для всех пакетов, идентифицированных в запросе API, выданном на этапе 401, тогда как неудача может быть указана, когда неудача

происходит для любого запрошенного копирования или контроля CRC. Указание неудачи может дополнительно указывать, какая операция (к примеру, копирование или контроль CRC) не удалась и для какого пакета(ов).

5 Фиг.5 изображает примерную работу варианта осуществления настоящего изобретения, чтобы обеспечить операции копирования и проверки правильности с помощью CRC. В этом примере вычислительные ресурсы, доступные уровню стека, включают в себя механизм DMA с возможностью передачи данных, так же как и выполнения операций проверки правильности с помощью CRC по буферизованным 10 данным (показан как DMA с возможностью 264 проверки правильности с помощью CRC). Соответственно, уровень стека может отвечать на запрос копирования и проверки правильности с помощью CRC, выдавая инструкцию механизму DMA выполнить передачу данных из буфера источника в буфер назначения и инструкцию механизму DMA, чтобы выполнить проверку правильности с 15 помощью CRC по данным, которые должны быть переданы.

Этапы 401-403 описываются относительно фиг.4, кроме, по меньшей мере, того, что этапы 402 и 403 выдаются механизму 264 DMA.

На этапе 504-1, частично в ответ на запрос на этапе 402, механизм 264 DMA 20 запрашивает передачу пакета 1 из буфера 254 источника. На этапе 504-2 механизм 264 DMA сохраняет пакет 1 в буфер, доступный посредством механизма 264 DMA. На этапе 504-3 механизм 264 DMA может передать пакет 1 в буфер 256 назначения для сохранения. Этапы 504-1 и 504-3 могут осуществить запрошенную операцию копирования.

25 На этапе 505-1, частично в ответ на запрос на этапе 403, механизм 264 DMA может выполнить проверку правильности с помощью CRC по буферизованному пакету 1. Этап 505-1 может осуществить операцию контроля CRC. На этапе 505-2 механизм 264 DMA может указать успех/неудачу копирования и контроля CRC стеку 252 в ответ на 30 завершение операций копирования и контроля CRC. На этапе 505-3 стек 252 может указать успех/неудачу копирования и контроля CRC уровню 250 iSCSI в ответ на указание успеха/неудачи копирования и контроля CRC, принятое стеком 252. Например, указание успеха/неудачи копирования и контроля CRC может быть 35 предоставлено уровню iSCSI в ответ на завершение действий, запрошенных в запросе API на этапе 401. Указание успеха/неудачи копирования и контроля CRC уровню iSCSI может указывать успех, когда успешное копирование и проверка правильности с помощью CRC происходят для всех пакетов, идентифицированных в запросе API, выданном на этапе 401, тогда как неудача может быть указана, когда 40 неудача происходит для любого запрошенного копирования или контроля CRC. Указание неудачи может дополнительно указывать, какая операция (к примеру, копирование или контроль CRC) не удалась и для какого пакета(ов).

Фиг.6 изображает подходящую блок-схему процесса 600 в соответствии с вариантом осуществления настоящего изобретения.

45 В блоке 602 процесс 600 может выдать единый запрос API для копирования данных и проверки правильности с помощью CRC. Например, уровень iSCSI может выдать единый запрос API для передачи данных и проверки правильности с помощью CRC. Объединенный запрос API копирования и контроля CRC может определить 50 копирование и контроль CRC для множественных данных. В блоке 603 процесс 600 может определить, какой вычислительный ресурс доступен для копирования данных и для проверки правильности с помощью CRC.

В блоке 604 процесс 600 может выдать одну или более команд, чтобы выполнить

копирование данных и проверку правильности с помощью CRC частично на основе доступной вычислительной среды. Например, уровень стека TCP может наблюдать, какое устройство, программное обеспечение или логика доступна, чтобы выполнить команды для операций передачи данных и проверки правильности с помощью CRC. В некоторых вариантах осуществления механизм DMA доступен для использования, по меньшей мере, для передач данных, тогда как в некоторых вариантах осуществления CPU используется, чтобы передать данные. Например, в некоторых вариантах осуществления механизм DMA может выполнять операции проверки правильности с помощью CRC, тогда как в некоторых вариантах осуществления инструкции, выполняемые посредством CPU, используются, чтобы выполнить операции проверки правильности с помощью CRC. В некотором варианте осуществления механизм DMA может буферизовать данные и выполнить проверку правильности с помощью CRC по буферизованным данным.

Варианты осуществления настоящего изобретения могут быть осуществлены как любое или комбинация из следующего: микросхемы или интегрированные схемы, соединенные с помощью материнской платы, логические схемы с жесткими соединениями, программное обеспечение, сохраненное запоминающим устройством и выполняемое микропроцессором, программно-аппаратные средства, специализированная интегральная микросхема (ASIC) и/или программируемая пользователем матричная БИС (FPGA).

Чертежи и следующее описание дают примеры настоящего изобретения. Хотя изображено несопоставимое число функциональных элементов, специалисты в области техники оценят, что один или более из таких элементов могут быть хорошо объединены в один функциональный элемент. Альтернативно, определенные элементы могут быть разделены на множественные функциональные элементы. Элементы из одного варианта осуществления могут быть добавлены в другой вариант осуществления. Например, порядок процессов, описанных в данном документе, может быть изменен и не ограничивается образом, описанным в данном документе. Рамки настоящего изобретения, однако, не ограничены посредством этих конкретных примеров. Возможны многочисленные вариации, несмотря на то, даны ли они явно в спецификации или нет, такие как различия в структуре, размерах и использовании материала. Рамки изобретения, по меньшей мере, широко даны следующей формулой изобретения.

#### Формула изобретения

1. Способ проверки правильности и передачи информации, содержащий этапы, на которых:

выдают объединенный запрос интерфейса прикладного программирования (API), причем этот объединенный запрос API имеет запрос копирования данных и запрос проверки правильности с помощью циклически избыточного кода (CRC), из первого уровня обработки протокола в промежуточный уровень;

- передают объединенный запрос API из промежуточного уровня во второй уровень обработки протокола; и

- выдают по меньшей мере одну инструкцию логике из второго уровня обработки протокола на основе логики, доступной второму уровню обработки протокола, чтобы выполнить запрос API.

2. Способ по п.1, в котором первый уровень обработки протокола содержит уровень, совместимый с Интернет-интерфейсом малых вычислительных систем.

3. Способ по п.1, в котором промежуточный уровень содержит уровень сокетов ядра.

4. Способ по п.1, в котором второй уровень обработки протокола содержит уровень, совместимый с протоколом управления передачей.

5. Способ по п.1, в котором логика, доступная второму уровню обработки протокола, содержит по меньшей мере одно из: центрального процессора, кэша и механизма прямого доступа к памяти, способного выполнить запрос копирования данных или выполнить запрос проверки правильности с помощью CRC или выполнить объединенный запрос API.

6. Способ по п.1, в котором объединенный запрос API содержит инструкцию передать копированные данные и выполнить проверку правильности с помощью CRC в отношении по меньшей мере части копированных данных.

7. Способ по п.6, в котором объединенный запрос API содержит инструкцию передать копированные данные и выполнить проверку правильности с помощью CRC другой части копированных данных.

8. Способ по п.1, в котором объединенный запрос API задает копирование данных и проверку правильности с помощью CRC для множества данных.

9. Способ по п.1, в котором объединенный запрос API разрешает первому уровню обработки протокола быть независимым от используемых операционных систем и логики платформы.

10. Способ по п.1, в котором логика, доступная второму уровню обработки протокола, содержит по меньшей мере один вычислительный ресурс, доступный второму уровню обработки протокола, при этом способ дополнительно содержит этапы, на которых:

определяют по меньшей мере один вычислительный ресурс, доступный второму уровню обработки протокола, чтобы выполнять запрос копирования данных и выполнять запрос проверки правильности с помощью CRC.

11. Машиночитаемый носитель, содержащий инструкции, сохраненные на нем, которые, когда выполняются машиной, заставляют машину:

выдавать объединенный запрос интерфейса прикладного программирования (API), причем этот объединенный запрос API имеет запрос копирования данных и запрос проверки правильности с помощью

циклически избыточного кода (CRC) из первого уровня обработки протокола в промежуточный уровень;

- передавать объединенный запрос API из промежуточного уровня во второй уровень обработки протокола; и

- выдавать по меньшей мере одну инструкцию логике из второго уровня обработки протокола на основе логики, доступной второму уровню обработки протокола, чтобы выполнить объединенный запрос API.

12. Машиночитаемый носитель по п.11, в котором первый уровень обработки протокола содержит уровень, совместимый с Интернет-интерфейсом малых вычислительных систем.

13. Машиночитаемый носитель по п.11, в котором промежуточный уровень содержит уровень сокетов ядра.

14. Машиночитаемый носитель по п.11, в котором второй уровень обработки протокола содержит уровень, совместимый с протоколом управления передачей.

15. Машиночитаемый носитель по п.11, в котором логика, доступная второму уровню обработки протокола, содержит по меньшей мере одно из: центрального

процессора, кэша и механизма прямого доступа к памяти, способного выполнить запрос копирования данных или выполнить запрос проверки правильности с помощью CRC или выполнить объединенный запрос API.

5 16. Машиночитаемый носитель по п.11, в котором объединенный запрос API содержит инструкцию, чтобы передать копированные данные и выполнить проверку правильности с помощью CRC в отношении по меньшей мере части копированных данных.

10 17. Машиночитаемый носитель по п.16, в котором объединенный запрос API содержит инструкцию, чтобы передать копированные данные и выполнить проверку правильности с помощью CRC в отношении другой части копированных данных.

18. Машиночитаемый носитель по п.11, в котором объединенный запрос API задает копирование данных и проверку правильности с помощью CRC для множества данных.

15 19. Машиночитаемый носитель по п.11, в котором объединенный запрос API разрешает первому уровню обработки протокола быть независимым от используемых операционных систем и логики платформы.

20 20. Машиночитаемый носитель по п.11, в котором логика, доступная второму уровню обработки протокола, содержит по меньшей мере один вычислительный ресурс, доступный второму уровню обработки протокола, и в котором инструкции, хранящиеся в нем, которые при исполнении машиной, заставляют машину дополнительно

25 определять по меньшей мере один вычислительный ресурс, доступный второму уровню обработки протокола, чтобы выполнять запрос копирования данных и выполнять запрос проверки правильности с помощью CRC.

21. Устройство проверки правильности и передачи информации, которое содержит в себе:

30 - логику первого уровня обработки протокола, чтобы выдавать объединенный запрос интерфейса прикладного программирования (API), при этом этот объединенный запрос API имеет запрос копирования данных и запрос проверки правильности с помощью циклически избыточного кода (CRC), логике промежуточного уровня;

35 - логику промежуточного уровня, чтобы передавать объединенный запрос API логике второго уровня обработки протокола; и

40 - логику второго уровня обработки протокола, чтобы выдавать по меньшей мере одну инструкцию логике из логики второго уровня обработки протокола на основе логики, доступной логике второго уровня обработки протокола, чтобы выполнить объединенный запрос API.

22. Устройство по п.21, в котором логика первого уровня обработки протокола содержит логику, совместимую с Интернет-интерфейсом малых вычислительных систем.

45 23. Устройство по п.21, в котором логика промежуточного уровня содержит логику уровня сокетов ядра.

24. Устройство по п.21, в котором логика второго уровня обработки протокола содержит логику, совместимую с протоколом управления передачей.

50 25. Устройство по п.21, в котором логика, доступная логике второго уровня обработки протокола, содержит по меньшей мере одно из: центрального процессора, кэша и механизма прямого доступа к памяти, способного выполнить запрос копирования данных или выполнить запрос проверки, правильности с помощью CRC

данных или выполнить объединенный запрос API.

26. Устройство по п.21, в котором объединенный запрос API содержит инструкцию, чтобы передать данные и выполнить проверку правильности с помощью CRC в отношении по меньшей мере части копированных данных.

27. Устройство по п.26, в котором объединенный запрос API содержит инструкцию, чтобы передать копированные данные и выполнить проверку правильности с помощью CRC в отношении другой части копированных данных.

28. Устройство по п.21, в котором комбинированный запрос API задает копирование данных и проверку правильности с помощью CRC для множества данных.

29. Устройство по п.21, в котором логика, доступная логике второго уровня обработки протокола, содержит по меньшей мере один вычислительный ресурс, доступный логике второго уровня обработки протокола, и в котором логика второго уровня обработки протокола дополнительно

определяет по меньшей мере один вычислительный ресурс, доступный второму уровню обработки протокола, чтобы выполнять запрос копирования данных и выполнять запрос проверки правильности с помощью CRC.

30. Система проверки правильности и передачи информации, содержащая:

- хост-систему, содержащую процессор и запоминающее устройство; причем хост-система содержит:

- логику первого уровня обработки протокола, чтобы выдавать объединенный запрос интерфейса прикладного программирования (API), причем этот объединенный запрос API имеет запрос копирования данных и запрос проверки правильности с помощью циклически избыточного кода (CRC), логике промежуточного уровня;

- логику промежуточного уровня, чтобы передавать объединенный запрос API логике второго уровня обработки протокола; и

- логику второго уровня обработки протокола, чтобы выдавать по меньшей мере одну инструкцию логике, доступной логике второго уровня обработки протокола, чтобы выполнить объединенный запрос API.

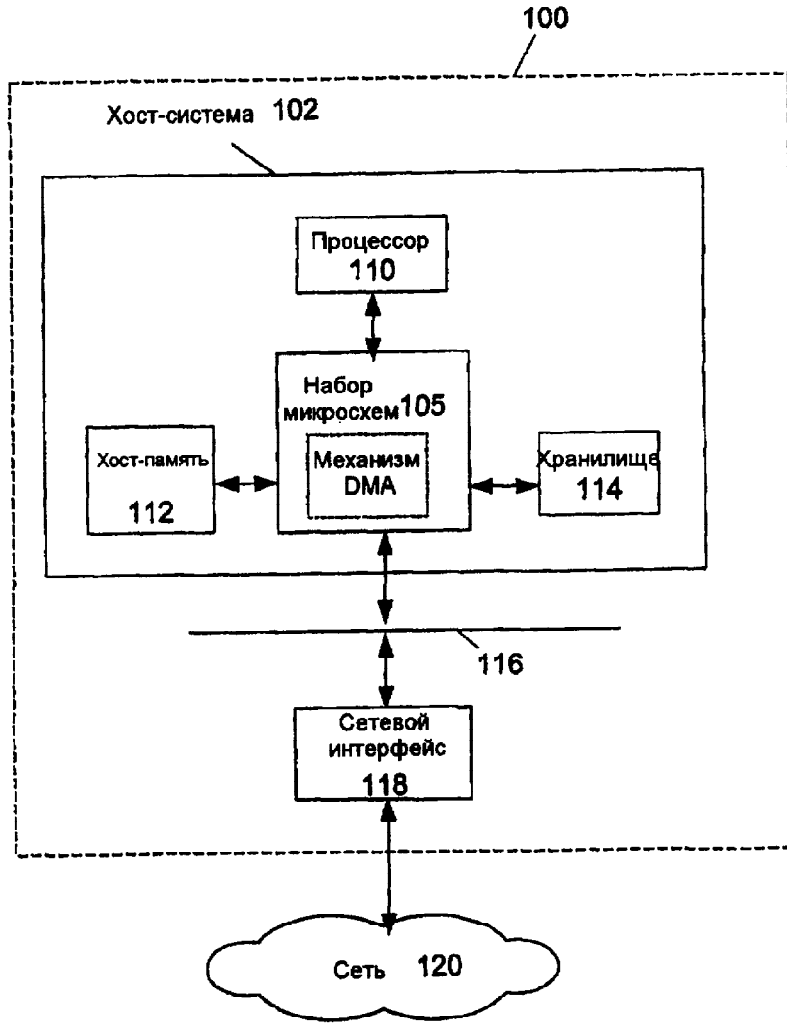
31. Система по п.30, в которой объединенный запрос API задает копирование данных и проверку правильности с помощью CRC для множества данных.

32. Система по п.30, в которой объединенный запрос API содержит инструкцию, чтобы передать копированные данные и выполнить проверку правильности с помощью CRC в отношении по меньшей мере части копированных данных.

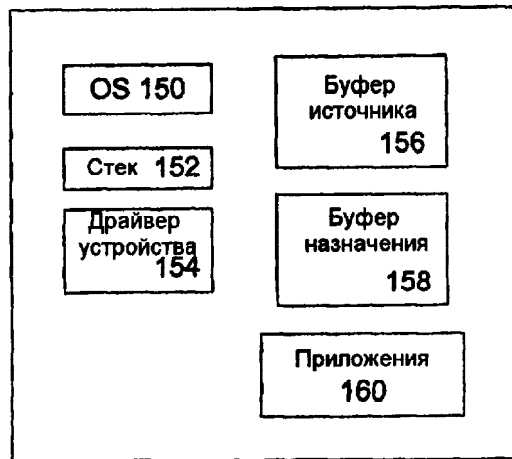
33. Система по п.31, в которой объединенный запрос API содержит инструкцию, чтобы передать копированные данные и выполнить проверку правильности с помощью CRC в отношении другой части копированных данных.

34. Система по п.30, в которой логика, доступная логике второго уровня обработки протокола, содержит по меньшей мере один вычислительный ресурс, доступный логике второго уровня обработки протокола, и в котором логика второго уровня обработки протокола дополнительно

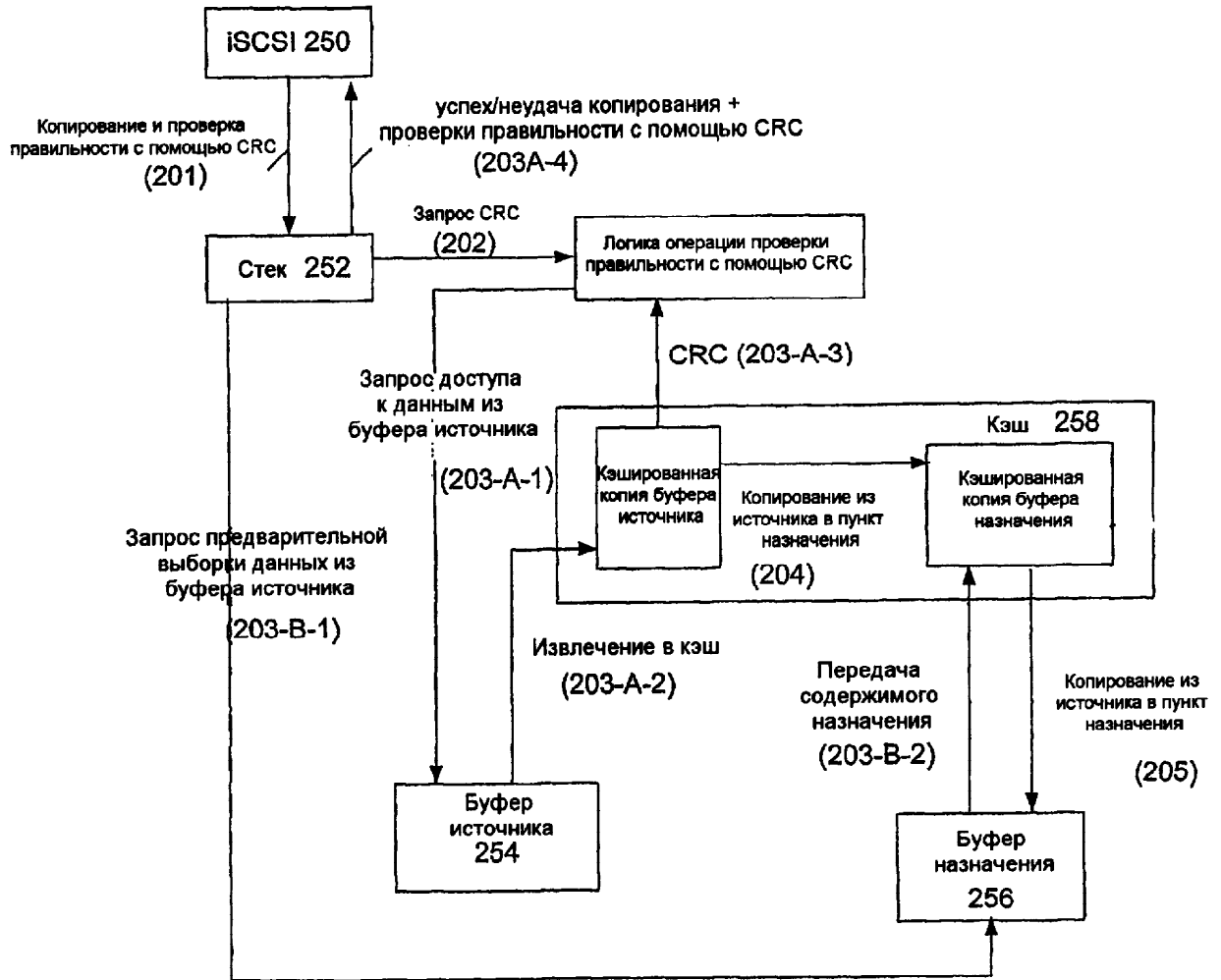
определяет по меньшей мере один вычислительный ресурс, доступный второму уровню обработки протокола, чтобы выполнять запрос копирования данных и выполнять запрос проверки правильности с помощью CRC.



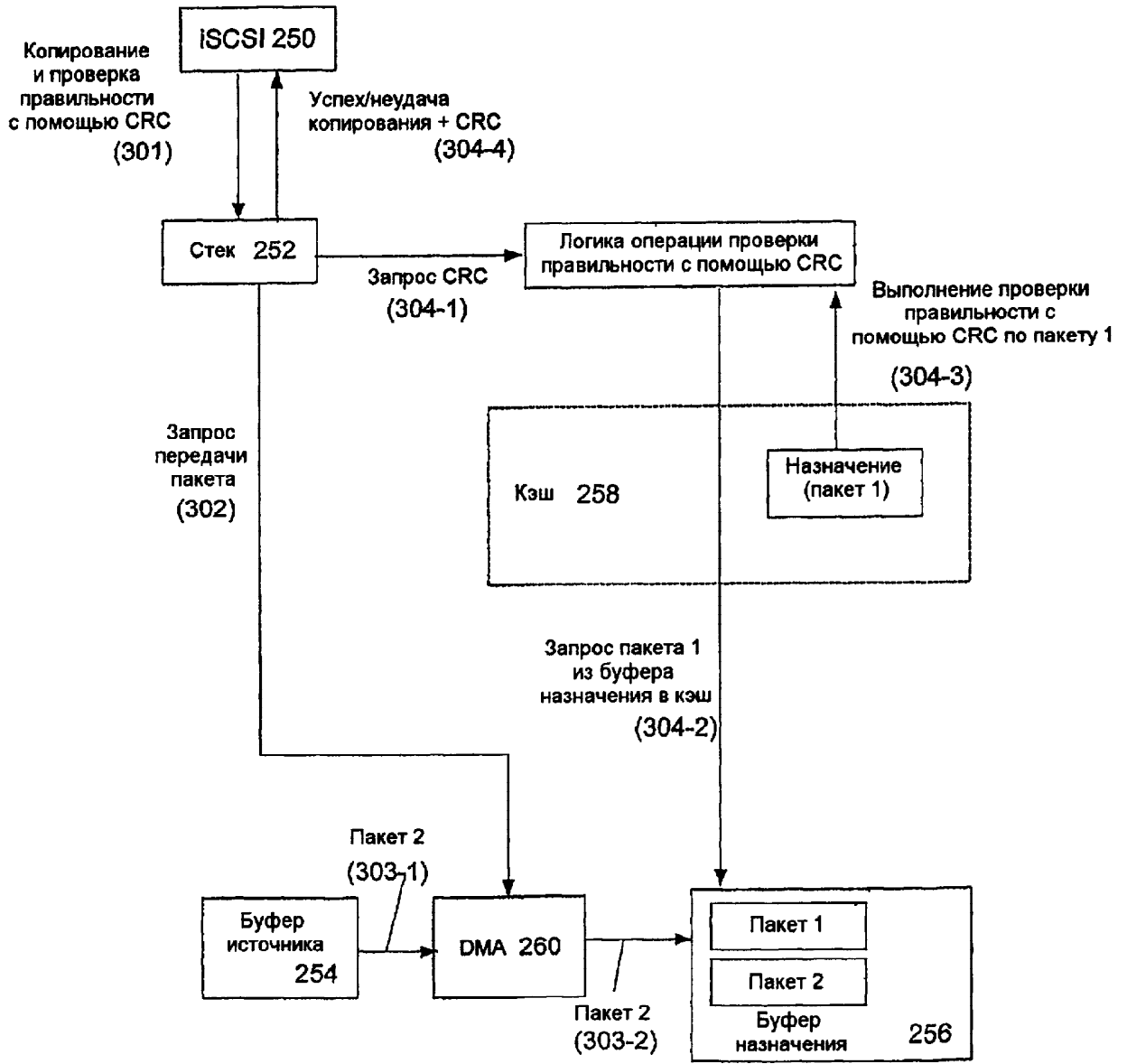
Фиг. 1А



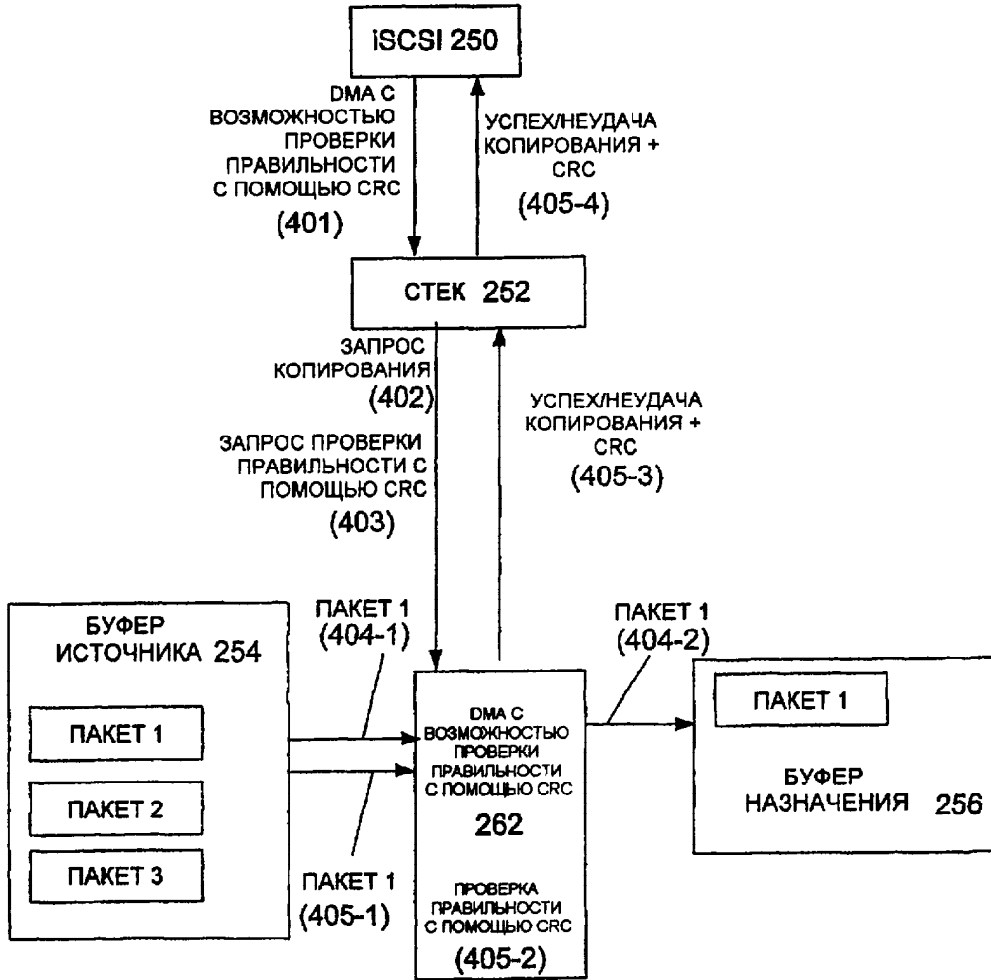
Фиг. 1В



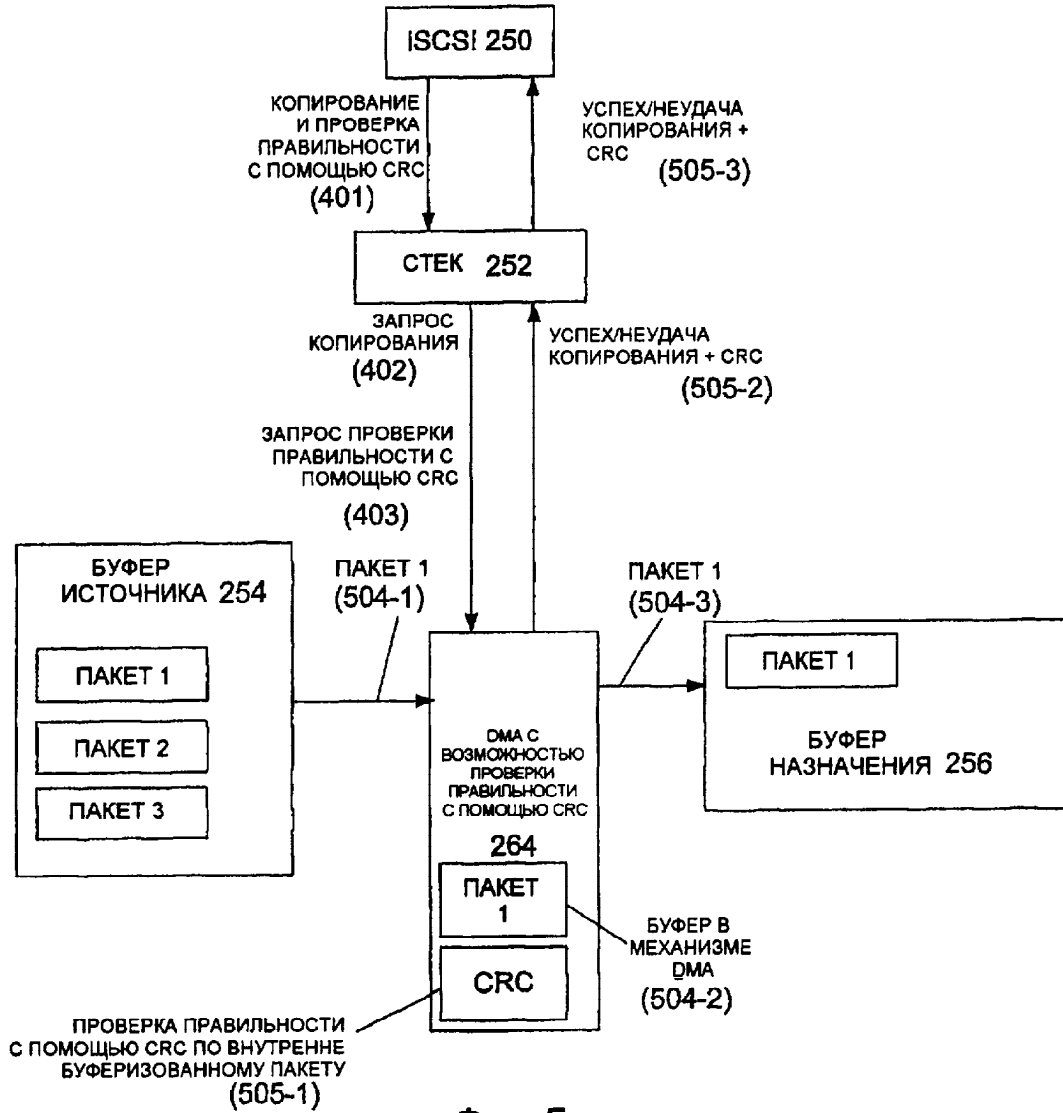
Фиг. 2



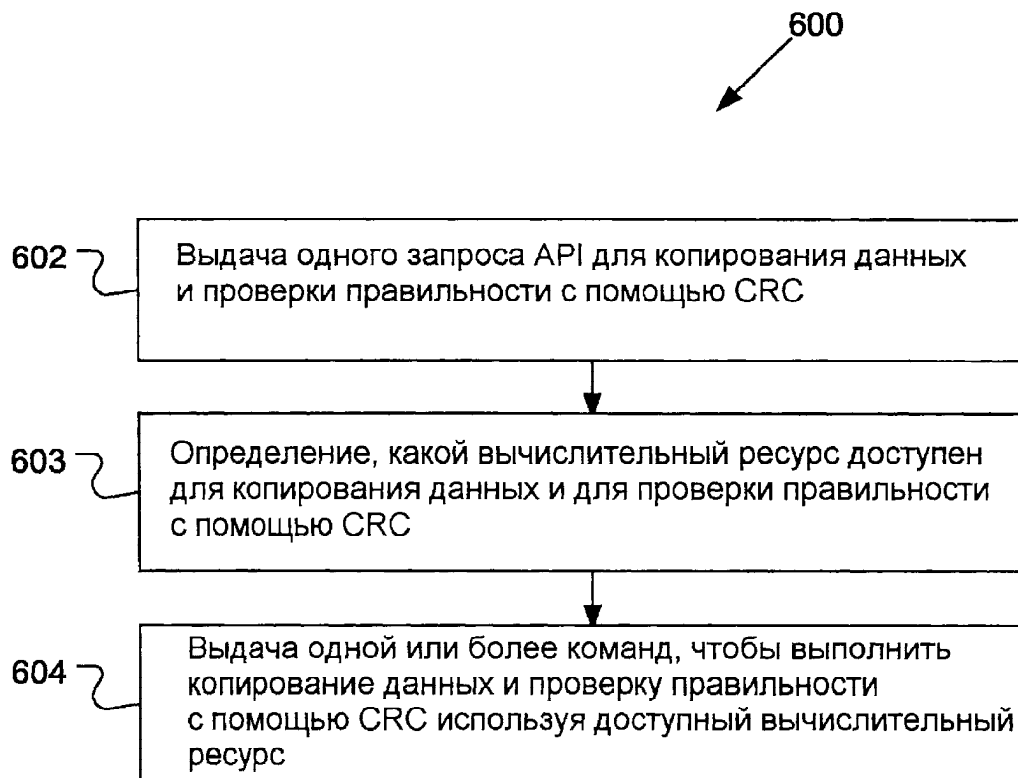
Фиг. 3



Фиг.4



Фиг.5



Фиг. 6