

Network Working Group
Request for Comments: 3549
Category: Informational

J. Salim
Znyx Networks
H. Khosravi
Intel
A. Kleen
Suse
A. Kuznetsov
INR/Swsoft
July 2003

Netlink как протокол для служб IP

Linux Netlink as an IP Services Protocol

Статус документа

Этот документ содержит информацию, предназначенную для сообщества Internet, и не задает каких-либо стандартов Internet. Документ может распространяться без ограничений.

Авторские права

Copyright (C) The Internet Society (2003). All Rights Reserved.

Тезисы

Данный документ описывает интерфейс Netlink ОС Linux, который используется операционной системой для обмена сообщениями как между процессами ядра, так и между ядром и пользовательскими процессами. Основное внимание в документе уделяется описанию функциональности Netlink как протокола, связывающего компоненты FEC¹ и CPC², которые определяют работу сервиса IP. Прочие варианты использования Netlink, включая обмен сообщениями внутри ядра и между процессами IPC³, а также настройка конфигурации служб, не относящихся к IP (несетевые службы или сетевые службы других протоколов), в данном документе не рассматриваются.

Документ предназначен для создания информационного контекста на начальном этапе работы группы ForCES⁴ IETF.

¹ Forwarding Engine Component – машина пересылки.

² Control Plane Component

³ Inter-process communication – обмен информацией между процессами.

⁴ Forwarding & Control Element Separation – разделение функций пересылки и управления. Страница рабочей группы доступна по адресу <http://www.ietf.org/html.charters/forces-charter.html>. Прим. перев.

Оглавление

1. Введение.....	3
1.1. Определения.....	3
1.1.1. Компоненты CPCS.....	3
1.1.2. Компоненты FEC.....	3
1.1.2.1. Модель машины пересылки IP в Linux.....	3
1.1.3. Службы IP.....	4
2. Архитектура Netlink.....	4
2.1. Логическая модель Netlink.....	5
2.2. Формат сообщений.....	5
2.3. Модель протокола.....	5
2.3.1. Адресация служб.....	5
2.3.2. Заголовок сообщений Netlink.....	6
2.3.2.1. Механизмы создания протоколов.....	7
2.3.2.2. Сообщение ACK в Netlink.....	7
2.3.3. Шаблоны FE системных служб.....	7
2.3.3.1. Сервисный модуль сетевого интерфейса.....	7
2.3.3.2. Модуль службы адресов IP.....	8
3. Определенные в данный момент IP-службы Netlink.....	9
3.1. Служба NETLINK_ROUTE.....	9
3.1.1. Модуль службы маршрутизации.....	9
3.1.2. Модуль учета соседей.....	10
3.1.3. Служба контроля трафика.....	11
3.2. Служба NETLINK_FIREWALL.....	12
3.3. Служба NETLINK_ARPD.....	13
4. Литература.....	14
4.1. Нормативные документы.....	14
4.2. Дополнительная литература.....	14
5. Вопросы безопасности.....	14
6. Благодарности.....	14
Приложение 1: Пример иерархии служб.....	14
Приложение 2: Пример протокола для IP-службы Foo.....	15
Приложение 2а: Взаимодействие с другими службами IP.....	15
Приложение 3: Примеры.....	15

1. Введение

Концепция разделения служб IP на управление и пересылку впервые была реализована в начале 1990-х годов в сокетях маршрутизации BSD 4.4 [9]. В то время наибольшую важность представляло простое решение вопроса пересылки пакетов IP (v4) и управление таблицами пересылки IPv4 в CPC (с помощью консольного интерфейса или демона динамической маршрутизации).

Мир IP-сетей с тех давних пор существенно изменился. Linux Netlink с точки зрения обеспечения сервиса и управления кроме поддержки сокетов маршрутизации обеспечивает ряд дополнительных функций. Начиная с ядра Linux 2.1, сокет Netlink обеспечивает абстракцию служб IP для нескольких типов сервиса кроме классической пересылки IPv4 в соответствии с .

Мотивом создания этого документа послужило отнюдь не желание описать весь набор служб, для которых можно использовать Netlink. Фактически многие типы сервиса (групповая маршрутизация, туннелирование, маршрутизация на основе правил и т. д) просто не рассматриваются в данном документе. Не предназначен документ и для использования в качестве учебника по Netlink. Идея документа заключается в общем описании Netlink и более подробном рассмотрении обязательных компонент в контексте работы группы ForCES - IPv4 и QoS. Документ также служит предварительным описанием множества механизмов, изучение которых представляет интерес в рамках ForCES. Рассматривается подмножество функций, доступных в ядре версии 2.4.6, которая была последней во время подготовки данного документа. Кроме того, документ рассматривает лишь функции, связанные с IPv4.

Документ начинается с концептуальных определений, после чего приводится рассмотрение Netlink в свете этих определений.

1.1. Определения

CPC⁵ представляет собой среду исполнения, которая может иметь несколько субкомпонент, которые будут обозначаться как CPC⁶. Все CPC, обеспечивающие контроль для разных служб IP, будут выполняться посредством машины пересылки FE⁷. Такие отношения между компонентами означают возможность наличия нескольких CPC для одной физической CP, если они контролируют несколько служб IP. По сути, связь между CP и FE является абстракцией сервиса.

1.1.1. Компоненты CPC

Компоненты управляющего плана CPC включают сигнальные протоколы от динамических протоколов маршрутизации (например, OSPF [5]) до протоколов распространения тегов (например, CR-LDP [7]). Классические протоколы и операции управления также входят в эту категорию. Среди них такие механизмы, как SNMP [6], COPS [4] и фирменные средства настройки конфигурации CLI/GUI. Задача управляющего плана состоит в обеспечении среды исполнения для перечисленных действий с целью настройки конфигурации и управления второй компонентой элемента сети (NE⁸) - машиной пересылки FE. Результат настройки конфигурации определяет способ трактовки пакетов, проходящих через FE.

1.1.2. Компоненты FEC⁹

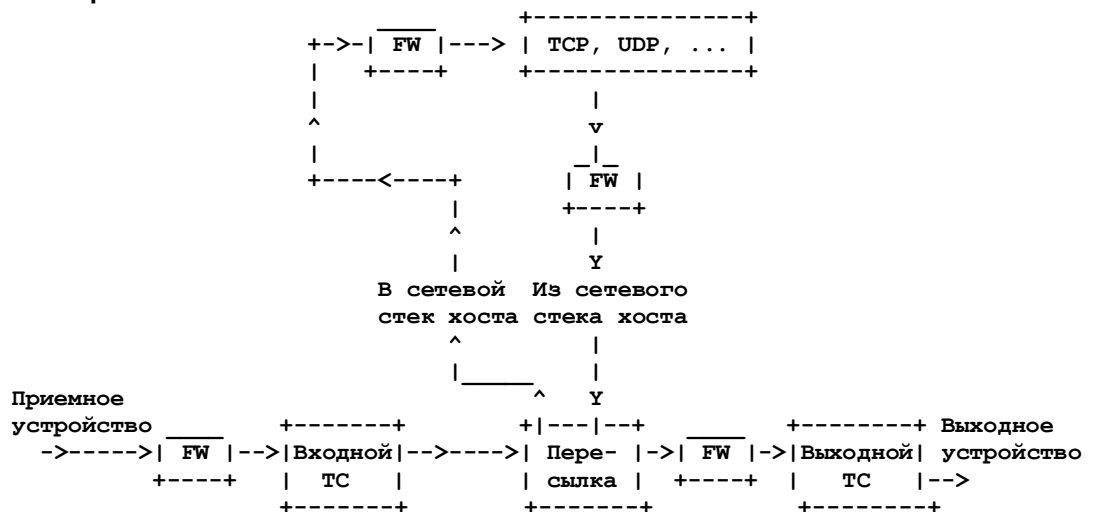
Машина пересылки FE представляет собой объект NE, который первым получает сетевые пакеты (из сети в NE).

Связанная с сервисом компонента FE просматривает пакет с целью обеспечения для него обработки, определенной компонентами CPC для данного типа сервиса IP. Различные службы будут использовать различные компоненты FEC. Сервисные модули могут объединяться в цепочки для поддержки более сложных типов сервиса (в рамках описанной ниже модели Linux FE).

Будучи созданной для поддержки конкретной службы, сервисная компонента FE будет по-прежнему соответствовать принципам модели пересылки.

1.1.2.1. Модель машины пересылки IP в Linux

На рисунке справа показана модель Linux FE для отдельного устройства. Единственной обязательной частью этой модели является модуль пересылки (Пересылка), соответствующий RFC 1812. Различные модули сетевого экранирования (FW), управления входящим и исходящим трафиком (TC¹⁰) не являются обязательными и могут даже использоваться для обхода модуля RFC 1812. Эти модули показаны в виде простых блоков на пути передачи данных и, фактически, могут представлять собой каскады из множества субмодулей. Дополнительную информацию о таких модулях вы найдете на сайтах [10] и [11].



Пакеты, прибывающее на входное устройство, сначала проходят через модуль межсетевого экранирования (FW), который может отбрасывать (drop) и изменять (mangle) пакеты или выполнять с ними иные операции. После прохождения модуля FW входящие пакеты в зависимости от принятой политики, могут попадать во входной модуль контроля трафика TC, который выполняет операции по измерению и регулированию потоков входящего трафика. Пакеты могут отбрасываться входным модулем TC в зависимо-

⁵ Control Plane – плоскость управления

⁶ Control Plane Components – компоненты управляющего плана.

⁷ Forwarding Engine – машина пересылки.

⁸ Network Element

⁹ Forwarding Engine Components – компоненты машины пересылки.

¹⁰ Traffic Control – контроль трафика.

сти от результатов измерения уровня трафика и принятой политики. После этого пакет передается единственному обязательному модулю, который обеспечивает пересылку в соответствии с требованиями RFC 1812. Пакет может быть отброшен, если он не соответствует требованиям RFC 1812, 1122, а также дополняющих их документов. Этот модуль является точкой выбора пути, из которой пакет, направленный принявшему его сетевому элементу NE, может быть передан сетевому стеку хоста.

Пакеты, которые не адресованы данному NE, могут проходить через submodule маршрутизации на базе правил (часть модуля пересылки), если такая маршрутизация поддерживается. После этого пакет передается следующему модулю сетевого экранирования, который может отбросить или изменить пакет в зависимости от настроек своих submodule и выбранной политики. После прохождения модуля экранирования пакет попадает в выходной фильтр контроля трафика (TC).

Выходной TC может отбрасывать пакеты с учетом политики, состояния очередей, уровня насыщения и правил управления скоростью исходящего потока. На этом этапе используются выходные очереди и задержки или отбрасывание пакета могут происходить как до его включения в очередь, так и после этого. Судьба пакета определяется выбранными для модуля алгоритмами и политикой.

1.1.3. Службы IP

Служба IP представляет собой процессы обработки пакета IP внутри NE. Эти процессы определяются комбинацией CPC и FEC.

Занимаемое службой время начинается с момента прихода пакета в NE и заканчивается в момент, когда пакет покидает NE. Существенно, что поведение служб IP в этом контексте определяет конкретным хостом. Компоненты CP, запущенные на NE, определяют сквозной для всего пути контроль служб с помощью управляющих приложений и сигнальных протоколов. Такие распределенные компоненты CPC унифицируют сквозное представление служб IP. Как было отмечено выше такие компоненты CP определяют поведение FE (и, следовательно, NE) по отношению к описываемому пакету.

Простым примером службы IP может служить классическая пересылка¹¹ IPv4. В этом случае управляющие компоненты (протоколы маршрутизации OSPF, RIP и т. п.) и фирменные средства настройки конфигурации CLI/GUI изменяют таблицы пересылки FE для того, чтобы обеспечить простой сервис по пересылке пакетов на следующий интервал (next hop). Обычно NE, обеспечивающие такой сервис, называют маршрутизаторами.

На приведенном справа рисунке показан простой пример реализации FE->CP для обеспечения классической пересылки IPv4 с некоторыми дополнительными функциями QoS для управления выходными очередями.

Демон ospfd управляет работой протокола OSPF, а COPS PEP¹² представляет собой дополнительную компоненту CPC. Компоненту IPv4 FE включает модуль пересылки IPv4, а также модуль выходного планировщика QoS. В качестве дополнительной службы может быть добавлен сервис пересылки на основе правил между модулем пересылки IPv4 и модулем планировщика QoS. Простейший классический вариант будет включать только модуль пересылки IPv4.

Опыт использования сетей говорит о важности добавления в маршрутизаторы новых типов сервиса, удовлетворяющих современным требованиям. Для решения этих задач были созданы и стандартизованы новые службы, которые могут выходить за пределы содержимого заголовков сетевого уровня. Однако, для обеспечивающих пересылку пакетов устройств NE по-прежнему используется термин "маршрутизатор". Новые службы (которые могут выходить за классические пределы заголовков L3) включают межсетевое экранирование, QoS с использованием Diffserv и RSVP, NAT, маршрутизацию на базе правил и т. п. Для таких служб создаются новые протоколы и средства управления.

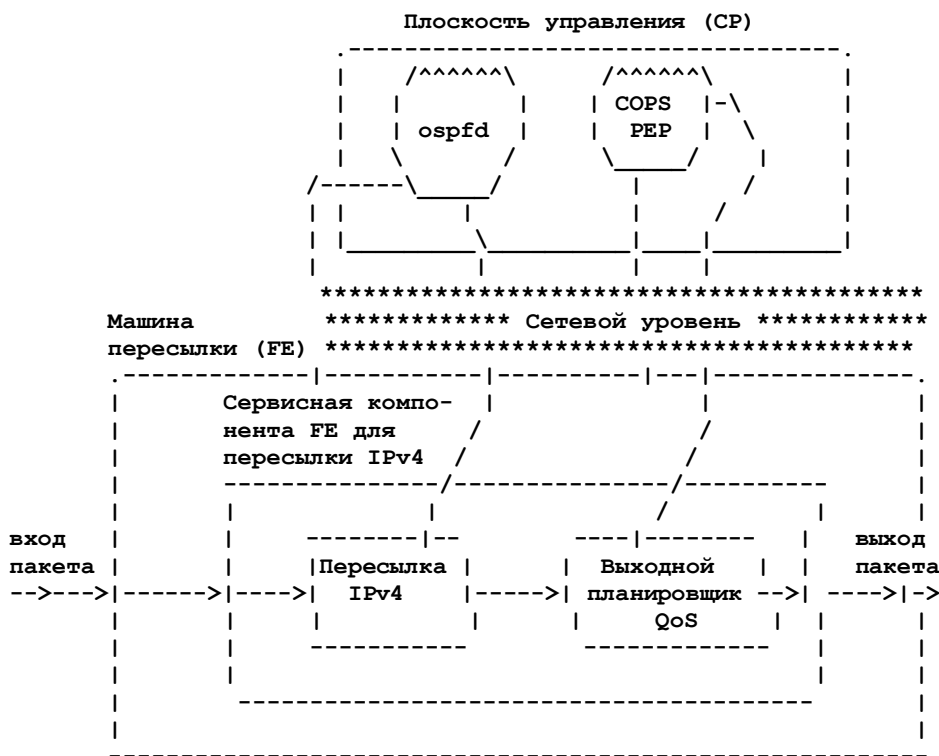
Одним из экстремистских определений сервиса IP является "все, за что сервис-провайдеры могут взять деньги".

2. Архитектура Netlink

Управление компонентами IP-сервиса определяется с использованием шаблонов.

Компоненты FEC и CPC участвуют в предоставлении услуг IP-сервиса путем обмена данными с использованием таких шаблонов. FEC может непрерывно получать обновления от компоненты CPC, указывающие как предоставлять услуги (например, для пересылки пакетов IPv4, добавления, удаления или изменения маршрутов).

Взаимодействие между FEC и CPC в контексте Netlink определяется протоколом. Netlink предоставляет механизмы для CPC(находится в пользовательском пространстве) и FEC (находится в ядре), позволяющие им получить свои собственные определения для протокола. Это связано с тем, что пользовательское пространство и ядро находятся на разных уровнях безопасности. Следовательно, для обмена информацией между компонентами требуется протокол. Такой протокол обычно обеспечивается неким привилегированным сервисом, который имеет возможность копирования данных между различными уровнями безопасности. Будем назы-



¹¹ Forwarding

¹² Policy Enforcement Point – точка реализации политики.

вать такую службу сервисом Netlink. Этот сервис может также инкапсулироваться в протоколы транспортного уровня, если CPC и FEC выполняются на разных узлах. Компоненты FEC и CPC, используя механизмы Netlink, могут выбрать надежный протокол для обмена данными. По умолчанию, однако, Netlink не обеспечивает гарантированного обмена данными.

Отметим, что FEC и CPC могут располагаться на одном уровне защиты памяти и использовать системный вызов connect() для создания прямого пути и обмена информацией через этот путь. В данном документе этот механизм рассматриваться не будет – отметим лишь возможность его реализации. В данном документе предполагается, что FEC является частью ядра, а CPC размещается в пользовательском пространстве. Это не означает однако, что приведенная в документе информация относится лишь к случаю размещения этих компонент в разных областях защиты и не привязывает компоненты к одному узлу.

Отметим, что Netlink позволяет обоим компонентам участвовать в предоставлении сервиса IP.

2.1. Логическая модель Netlink

На приведенном рисунке показана простая диаграмма логических связей между компонентами FEC и CPC. В качестве примера использована FEC пересылки IPv4 (служба NETLINK_ROUTE, описанная ниже).

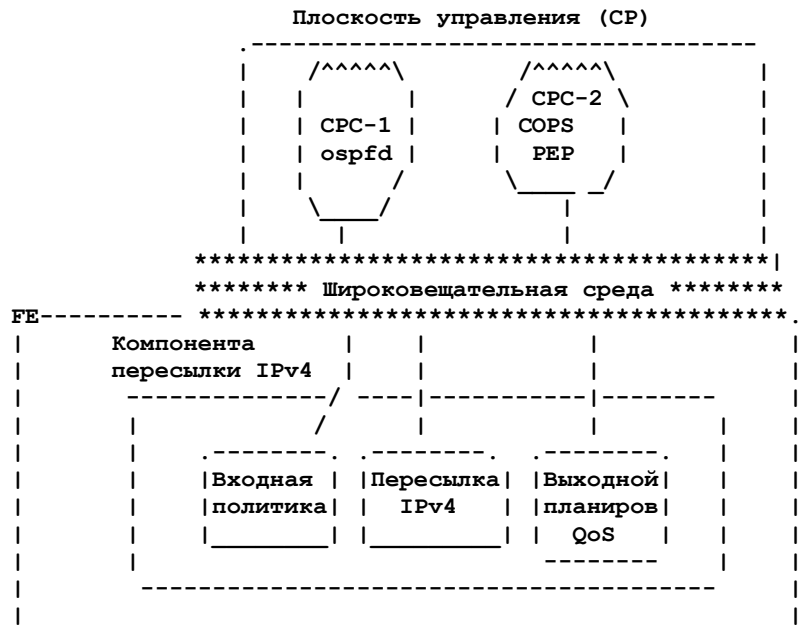
Netlink логически моделирует FEC и CPC в форме узлов, связанных между собой через ширококвещательную среду.

Свойства среды обусловлены сервисом. В приведенном примере показана ширококвещательная среда, принадлежащая к расширенному сервису пересылки IPv4.

Узлы (CPC и FEC в рассматриваемом примере) подключены к среде передачи и регистрируются для получения сообщений определенных типов. CPC может подключаться к множеству сред, если это способствует более эффективному управлению сервисом. Все узлы (CPC и FEC) принимают пакеты из ширококвещательной среды. Пакеты могут отбрасываться средой передачи, если они имеют некорректный формат или содержат ошибки. Отброшенные пакеты не поступают ни на один из узлов. Сервис Netlink может передавать отправителю сигналы об ошибках при обнаружении некорректных пакетов Netlink.

Передаваемые в среду пакеты могут быть ширококвещательными, групповыми или адресованными конкретному узлу. Узлы FEC и CPC регистрируют свою заинтересованность в сообщениях определенного типа для их обработки или простого мониторинга.

В приложениях 1 и 2 приведено более детальное рассмотрение этого взаимодействия.



2.2. Формат сообщений

В сообщениях Netlink существует три уровня - заголовок сообщения Netlink, шаблон IP-сервиса и связанные с IP-сервисом данные.

Сообщения Netlink используются для обмена данными между FEC и CPC, параметризации FEC, асинхронной передачи сведений о событиях FEC компонентам CPC и сбора/просмотра статистики (обычно с помощью CPC).

Заголовок сообщения Netlink используется для всех типов сервиса, тогда как шаблоны (IP Service Template) связаны с конкретными типами сервиса. Каждая служба IP передает данные параметризации (от CPC к FEC) или отклики (от FEC к CPC). Эти данные передаются в формате TLV¹³ и являются уникальными для сервиса.

Отдельные компоненты сообщений Netlink подробно рассматриваются ниже.

2.3. Модель протокола

В этом разделе описано как Netlink обеспечивает механизм ориентированного на службы взаимодействия между FEC и CPC.

2.3.1. Адресация служб

Для получения доступа сначала нужно соединиться с сервисом на FE. Соединение организуется путем системного вызова socket() для домена PF_NETLINK. Каждая компонента FEC идентифицируется номером протокола. В результате вызова могут создаваться сокеты типа SOCK_RAW или SOCK_DGRAM, хотя Netlink не различает сокеты этих типов. Соединение с сокетом обеспечивает основу для адресации FE<->CP.

После этого организуется подключение к сервису (в любой момент в течение срока существования соединения) путем ввода обусловленной сервисом команды (от CPC к FEC, в основном для настройки конфигурации), команды сбора статистики или подписки/отказа на уведомление о связанных с сервисом событиях. Закрытие сокета прерывает транзакцию.

¹³ Type-Length-Value – тип-размер-значение

Флаг	Значение	Флаг	Значение
IFF_UP	Интерфейс активизирован администратором	IFF_NOTRAILERS	Следует избегать использования трейлеров.
IFF_BROADCAST	Установлен корректный широковещательный адрес.	IFF_ALLMULTI	Принимать все пакеты с групповыми адресами.
IFF_DEBUG	Флаг режима отладки для интерфейса.	IFF_MASTER	Ведущий интерфейс для транка с распределением нагрузки.
IFF_LOOPBACK	Петлевой интерфейс (loopback).	IFF_SLAVE	Ведомый интерфейс для транка с распределением нагрузки.
IFF_POINTOPOINT	Интерфейс типа "точка-точка".	IFF_MULTICAST	Поддержка групповой адресации.
IFF_RUNNING	Интерфейс находится в работающем состоянии.	IFF_PORTSEL	Интерфейс может выбирать тип среды с помощью ifmap.
IFF_NOARP	Для интерфейса не требуется протокол ARP.	IFF_AUTOMEDIA	Активизирован автоматический выбор типа среды.
IFF_PROMISC	Интерфейс работает в режиме захвата ¹⁷ .	IFF_DYNAMIC	Интерфейс создан в динамическом режиме.

Change Mask - 32 бита

Зарезервированное поле, которое должно иметь значение 0xFFFFFFFF.

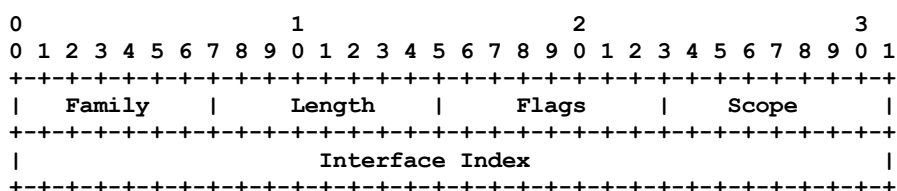
Применимые к данному сервису атрибуты перечислены в таблице.

Атрибут	Описание	Атрибут	Описание
IFLA_UNSPEC	Не определен.	IFLA_MTU	Значение MTU для устройства
IFLA_ADDRESS	Аппаратный адрес интерфейса на уровне L2.	IFLA_LINK	Значение ifindex для канала, к которому подключено устройство.
IFLA_BROADCAST	Аппаратный широковещательный адрес интерфейса на уровне L2.	IFLA_QDISC	Строка ASCII, указывающая имя дисциплины управления выходными очередями ¹⁸ .
IFLA_IFNAME	Имя устройства (строка ASCII).	IFLA_STATS	Статистика для интерфейса.

К данному типу сервиса относятся сообщения Netlink **RTM_NEWLINK**, **RTM_DELLINK** и **RTM_GETLINK**.

2.3.3.2. Модуль службы адресов IP

Эта служба обеспечивает возможность добавления и удаления адресов, а также получения сведений об IP-адресах, связанных с данным интерфейсом. Шаблон сообщения службы предоставления адресов¹⁹ показан на рисунке справа.



Family - 8 битов

Идентификатор семейства адресов: **AF_INET** для IPv4 и **AF_INET6** для IPv6.

Length - 8 битов

Размер маски адреса.

Flags - 8 битов

Флаг	Описание
IFA_F_SECONDARY	Вторичный адрес (псевдоним интерфейса)
IFA_F_PERMANENT	Постоянный адрес, установленный пользователем. Отсутствие этого флага говорит о динамическом выделении адреса (например, с помощью системы автоматической настройки конфигурации)
IFA_F_DEPRECATED	Недействующий (deprecated) адрес IP.
IFA_F_TENTATIVE	Предполагаемый (tentative) адрес IP. Процедура обнаружения дубликатов адресов находится в стадии разработки.

Scope - 8 битов

Область корректности адреса.

SCOPE_UNIVERSE	Адрес глобального действия.
SCOPE_SITE	Адрес корректен в пределах данного сайта (только для IPv6).
SCOPE_LINK	Адрес имеет смысл только для данного устройства.
SCOPE_HOST	Адрес имеет смысл только для данного хоста.

Атрибуты сервиса перечислены в таблице.

¹⁷ promiscuous mode.

¹⁸ egress root queuing discipline.

¹⁹ address provisioning service.

Атрибут	Описание	Атрибут	Описание
IFA_UNSPEC	Не определен.	IFA_BROADCAST	Широковещательный адрес для протокола RAW.
IFA_ADDRESS	Адрес интерфейса для протокола RAW.	IFA_ANYCAST	Anycast-адрес для протокола RAW.
IFA_LOCAL	Локальный адрес для протокола RAW.	IFA_CACHEINFO	Кэшированная информация об адресе.
IFA_LABEL	Имя интерфейса (строка ASCII).		

К данному типу сервиса относятся сообщения Netlink `RTM_NEWADDR`, `RTM_DELADDR` и `RTM_GETADDR`.

3. Определенные в данный момент IP-службы Netlink

Хотя, как было отмечено выше, существует множество других служб IP, использующих Netlink, в данном документе рассматривается лишь небольшая часть этих служб, интегрированных в ядро версии 2.4.6. К таким службам относятся `NETLINK_ROUTE`, `NETLINK_FIREWALL` и `NETLINK_ARPD`²⁰.

3.1. Служба NETLINK_ROUTE

Эта служба позволяет СРС изменять таблицу маршрутизации IPv4 в машине пересылки FE. Кроме того, данный сервис может применяться СРС для получения данных об обновлении маршрутов и сбора статистики.

3.1.1. Модуль службы маршрутизации

Эта служба обеспечивает возможность создания и удаления маршрутов, а также получения информации о сетевых маршрутах. Формат шаблона сообщения показан на рисунке справа.

Family - 8 битов

Идентификатор семейства адресов: `AF_INET` для IPv4 и `AF_INET6` для IPv6.

Src length - 8 битов

Размер префикса IP-адреса отправителя.

Dest length - 8 битов

Размер префикса IP-адреса получателя.

TOS - 8 битов

Восьмибитовое поле TOS (следует отказаться от него для освобождения места под DSCP).

Table ID - 8 битов

Идентификатор таблицы. Поддерживается до 255 таблиц маршрутизации.

<code>RT_TABLE_UNSPEC</code>	Неуказанная таблица.	<code>RT_TABLE_MAIN</code>	Основная таблица.
<code>RT_TABLE_DEFAULT</code>	Используемая по умолчанию таблица.	<code>RT_TABLE_LOCAL</code>	Локальная таблица.

Пользователь может выделять дополнительные значения в диапазоне²¹ от `RT_TABLE_UNSPEC` (0) до `RT_TABLE_DEFAULT` (253).

Protocol - 8 битов

Указывает кто добавил маршрут в таблицу.

Протокол	Источник маршрута	Протокол	Источник маршрута
<code>RTPROT_UNSPEC</code>	Неизвестен.	<code>RTPROT_BOOT</code>	При загрузке системы.
<code>RTPROT_REDIRECT</code>	Из сообщения ICMP redirect.	<code>RTPROT_STATIC</code>	Администратор.
<code>RTPROT_KERNEL</code>	Ядро.		

Значения, превышающие `RTPROT_STATIC` (4)²², не интерпретируются ядром и включены только с информационными целями. Эти значения могут использоваться, чтобы пометить источник маршрутной информации или различать разные демоны маршрутизации. Идентификаторы уже присвоенные демонам маршрутизации вы можете найти в файле `<linux/rtnetlink.h>`.

Scope - 8 битов

Область видимости маршрута (корректная дистанция до получателя).

<code>RT_SCOPE_UNIVERSE</code>	Глобальный маршрут.
<code>RT_SCOPE_SITE</code>	Внутренний маршрут локальной автономной системы.
<code>RT_SCOPE_LINK</code>	Маршрут на данном канале (соединении).
<code>RT_SCOPE_HOST</code>	Маршрут на локальном хосте.
<code>RT_SCOPE_NOWHERE</code>	Получателя не существует.

²⁰ На момент перевода документа (ядро 2.6.10) был определен целый ряд дополнительных служб, информацию о которых вы найдете в файле `<linux/netlink.h>`. *Прим. перев.*

²¹ Не включая граничные значения 0 и 253. *Прим. перев.*

²² В файле `<linux/rtnetlink.h>` указано, что значение `RTPROT_STATIC` (4) также не интерпретируется ядром. *Прим. перев.*

Значения в диапазоне от **RT_SCOPE_UNIVERSE** (0) до **RT_SCOPE_SITE** (200), не включая граничные, могут использоваться для пользовательских идентификаторов.

Type - 8 битов

Тип маршрута.

<i>Type</i>	<i>Получатель</i>
RTN_UNSPEC	Неизвестный маршрут.
RTN_UNICAST	Шлюз или прямой маршрут.
RTN_LOCAL	Маршрут к локальному интерфейсу.
RTN_BROADCAST	Локальный широковещательный маршрут (передается как broadcast).
RTN_ANYCAST	Локальный anycast-маршрут (передается как unicast)
RTN_MULTICAST	Локальный групповой (multicast) маршрут.
RTN_BLACKHOLE	Маршрут для отбрасывания пакетов без уведомления (черная дыра).
RTN_UNREACHABLE	Недостижимый получатель. Пакеты отбрасываются с передачей отправителю сообщения ICMP о недоступности адресата.
RTN_PROHIBIT	Запрещенный маршрут. Пакеты отбрасываются с передачей отправителю сообщения ICMP о запрете доступа к адресату.
RTN_THROW	При использовании маршрутизации на базе правил указывает на продолжение просмотра маршрутов в другой таблице. При обычной маршрутизации пакеты отбрасываются с передачей отправителю сообщения ICMP о недоступности адресата.
RTN_NAT	Правило трансляции сетевых адресов.
RTN_XRESOLVE	Указывает на внешний преобразователь (resolver). В настоящее время еще не реализовано.

Flags - 32 бита

Дополнительная информация о маршруте.

RTM_F_NOTIFY	При изменении маршрута пользователю передается уведомление.
RTM_F_CLONED	Маршрут клонирован из другого маршрута.
RTM_F_EQUALIZE	Маршрут допускает случайный выбор следующего интервала (next hop) в случае наличия нескольких путей (в настоящее время не реализовано).

Имеющие отношение к данному сервису атрибуты перечислены в таблице.

<i>Атрибут</i>	<i>Описание</i>
RTA_UNSPEC	Игнорируется.
RTA_DST	Протокольный адрес источника маршрута.
RTA_SRC	Протокольный адрес конечной точки маршрута.
RTA_IF	Индекс входного интерфейса.
RTA_OIF	Индекс выходного интерфейса.
RTA_GATEWAY	Протокольный адрес шлюза для маршрута.
RTA_PRIORITY	Приоритет маршрута.
RTA_PREFSRC	Предпочтительный адрес отправителя при наличии нескольких адресов.
RTA_METRICS	Присвоенная маршруту метрика (например, RTT, начальный размер окна TCP и т. п.).
RTA_MULTIPATH	Атрибуты следующего интервала для маршрута с множеством путей (Multipath route).
RTA_PROTOINFO	Атрибут маршрутизации, основанный на политике межсетевого экрана.
RTA_FLOW	Область маршрута (Route realm).
RTA_CACHEINFO	Кэшированная информация о маршруте.

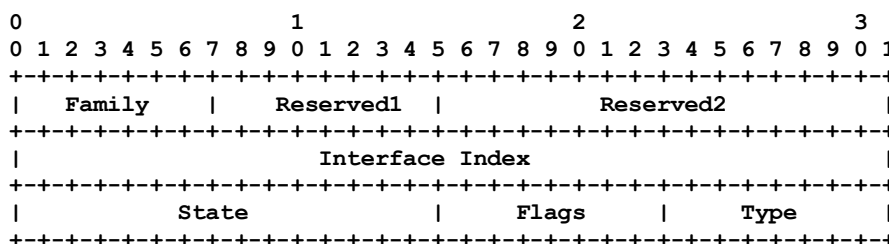
Для этого типа сервиса поддерживаются дополнительные сообщения Netlink **RTM_NEWROUTE**, **RTM_DELROUTE** и **RTM_GETROUTE**.

3.1.2. Модуль учета соседей

Этот сервис обеспечивает возможность добавления и удаления записей о соседях (например, ARP, IPv4 neighbor solicitation и т. п.), а также получения информации о существующих записях таблицы соседей. Шаблон сообщений этой службы показан на рисунке справа.

Family - 8 битов

Идентификатор семейства адресов:
AF_INET для IPv4 и **AF_INET6** для IPv6.



Interface Index - 32 бита

Уникальный индекс интерфейса.

State - 16 битов

Битовая маска, которая может включать перечисленные в таблице биты состояния.

NUD_INCOMPLETE	Продолжаются попытки преобразования адреса.
NUD_REACHABLE	Подтверждено наличием рабочей записи в кэше.
NUD_STALE	Просроченная запись из кэша.
NUD_DELAY	Сосед больше не достижим. Трафик передан, ожидается подтверждение.
NUD_PROBE	В настоящее время осуществляется запрос на обновление записи в кэше.
NUD_FAILED	Некорректная запись в кэше.
NUD_NOARP	Устройство, которое не выполняет обнаружения соседей (ARP).
NUD_PERMANENT	Статическая запись.

Flags - 8 битов

NTF_PROXY	Запись проху ARP	NTF_ROUTER	Маршрутизатор IPv6
------------------	------------------	-------------------	--------------------

Применимые к этому сервису атрибуты перечислены в таблице.

Атрибут	Описание
NDA_UNSPEC	Неизвестный тип.
NDA_DST	Адрес сетевого уровня для кэша соседей.
NDA_LLADDR	Адрес канального уровня для кэша соседей.
NDA_CACHEINFO	Статистика кэширования.

Для этого типа сервиса поддерживаются дополнительные сообщения Netlink **RTM_NEWNEIGH**, **RTM_DELNEIGH** и **RTM_GETNEIGH**.

3.1.3. Служба контроля трафика

Этот сервис обеспечивает возможность генерации, запроса и прослушивания событий, связанных с контролем трафика. Эта служба включает дисциплины очередей (планировщики и алгоритмы обслуживания очередей – например, планировщики на основе уровней приоритета или алгоритм RED) и классификаторы трафика. Система управления трафиком в Linux²³ обеспечивает высокий уровень гибкости и поддерживает иерархическое каскадирование различных блоков для совместного использования ресурсов каналов передачи трафика.

На приведенном справа рисунке показана пример схемы выходного блока TC. В этом документе приводится весьма краткое рассмотрение этого вопроса; дополнительную информацию можно найти на сайте [11]. Пакет сначала проходит через фильтр, используемый для идентификации класса трафика, к которому может быть отнесен данный пакет. Термин “класс” относится к дисциплинам очередей и связан с конкретной очередью. Очередь может использоваться простой алгоритм (например, FIFO) или более сложные механизмы типа RED или token bucket. Дисциплину очереди, наиболее удаленную от родительской дисциплины, обычно называют планировщиком. В показанной здесь иерархии планировщик может включать различные алгоритмы планирования, что делает системы управления трафиком на выходе²⁴ в ОС Linux очень гибкими.



Шаблон сообщения для этого типа сервиса показан ниже. Этот шаблон используется для дисциплин входных и выходных очередей (относительно модели управления трафиком на выходе, описанной в разделе для модели FE на стр. 3). Каждая специфическая компонента модели имеет уникальные атрибуты, описывающие ее наилучшим способом. Атрибуты общего назначения рассматриваются ниже.

Family - 8 битов

Идентификатор семейства адресов: **AF_INET** для IPv4 и **AF_INET6** для IPv6.

Interface Index - 32 бита

Уникальный индекс интерфейса.

Qdisc handle - 32 бита

²³ Traffic Control Service

²⁴ Egress TC

Уникальный идентификатор экземпляра дисциплины очередей. Обычно эти идентификаторы рассматриваются как двух-компонентные (старшая:младшая) по 16 битов в каждой части. Старшая часть номера будет также старшей частью в номере родителя данного экземпляра.

Parent Qdisc - 32 бита

Используется для иерархической структуризации дисциплин очередей. Если это значение совпадает с идентификатором и TC_H_ROOT, данный экземпляр qdisc является называется корневым²⁵ (старшим).

TCM Info - 32 бита

Для этого поля FE обычно устанавливает значение 1 за исключением тех случаев, когда экземпляр Qdisc уже используется (в этом случае в поле помещается значение счетчика использования данного экземпляра). При передаче со стороны CPC в направлении FEC это поле обычно имеет значение 0 за исключением тех случаев, когда оно используется в контексте фильтрации. В таких случаях 32-битовое поле делится на 16-битовые поля приоритета и протокола. Протоколы определены в исходных кодах ядра (файл <include/linux/if_ether.h>). Наиболее широко используемым протоколом является ETH_P_IP (протокол IP).

Значение приоритета используется для разрешения конфликтов при пересечении фильтрующих выражений.

Базовые атрибуты этого типа сервиса перечислены в таблице.

<i>Атрибут</i>	<i>Описание</i>
TCA_KIND	Каноническое имя компоненты FE.
TCA_STATS	Базовая статистика использования FEC.
TCA_RATE	Оценка скорости для FEC (расчет на основе текущего состояния).
TCA_XSTATS	Специфическая статистика FEC.
TCA_OPTIONS	Вложенные атрибуты, связанные с FEC.

В приложении 3 дается пример конфигурации компоненты FE для дисциплины FIFO.

Для этого типа сервиса поддерживаются дополнительные сообщения Netlink **RTM_NEWQDISC**, **RTM_DELQDISC**, **RTM_GETQDISC**, **RTM_NEWTCCLASS**, **RTM_DELTCLASS**, **RTM_GETTCLASS**, **RTM_NEWTFILTER**, **RTM_DELTFILTER** и **RTM_GETTFILTER**.

3.2. Служба NETLINK_FIREWALL

Эта служба позволяет CPC принимать пакеты через сервисные модули межсетевое экрана IPv4 в FE, манипулировать этими пакетами и повторно передавать их. Правило межсетевого экрана является первым из числа вставляемых для активизации перенаправления пакетов. CPC информирует FEC о своем желании получать метаданные для пакета или реальные данные из него, а также сообщает максимальный размер данных, которые будут перенаправляться. Перенаправленные пакеты по-прежнему сохраняются в FEC, ожидая решения о своей судьбе от CPC. Решение может быть простой командой на восприятие или отбрасывание пакета (в этом случае решение применяется к пакету, все еще находящемуся в FEC) или включать измененный пакет, который должен быть передан взамен исходного.

Существует два типа сообщений, передаваемых от CPC к FEC - Mode (режим) и Verdict (решение). Сообщения типа Mode незамедлительно передаются FEC и сообщают о том, что CPC желает принимать от FEC. Сообщения типа Verdict передаются FEC после принятия решения о дальнейшей судьбе полученного пакета. Формат сообщений рассматривается ниже.

Опишем сначала сообщение, указывающее режим.

Mode - 8 битов

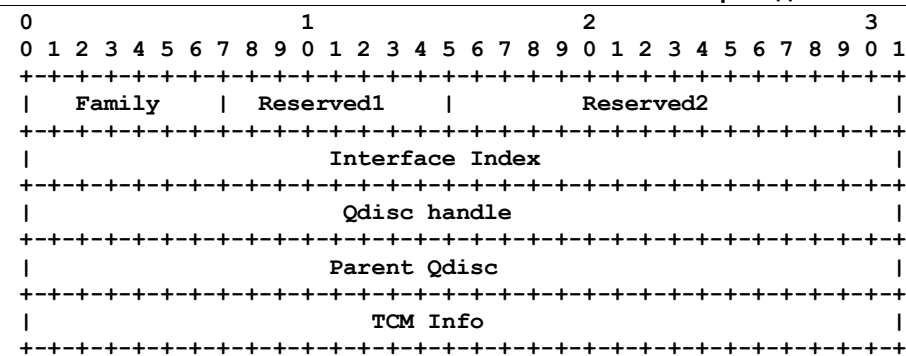
Определяет тип информации в пакетах, отправляемых CPC:

IPQ_COPY_META – копировать в CPC только метаданные для пакета.

IPQ_COPY_PACKET – копировать в CPC метаданные и содержимое поля данных пакета.

Range - 32 бита

В режиме **IPQ_COPY_PACKET** это значение определяет максимальный размер копируемых данных.



²⁵ root qdisc

	0	1								2								3																						
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Пакет и связанные с ним метаданные, полученные из пользовательского пространства, показаны на рисунке справа.	+++++																																							
Packet ID - 32 бита	Packet ID																																							
Уникальный идентификатор пакета, передаваемый CPC от FEC.	+++++																																							
Mark - 32 бита	Mark																																							
Значение внутренних метаданных, установленное для описания правила, в котором был взят пакет.	+++++																																							
timestamp_m - 32 бита	timestamp_m																																							
Время прибытия пакета (в секундах)	+++++																																							
timestamp_u - 32 бита	timestamp_u																																							
Время прибытия пакета (микросекунды, добавляемые к timestamp_m)	+++++																																							
hook - 32 бита	hook																																							
Модуль межсетевое экрана, из которого был взят пакет.	+++++																																							
indev_name - 128 битов	indev_name																																							
Имя приемного интерфейса (строка ASCII).	+++++																																							
outdev_name - 128 битов	outdev_name																																							
Имя выходного интерфейса (строка ASCII).	+++++																																							
hw_protocol - 16 битов	hw_protocol																hw_type																							
Аппаратный протокол (в сетевом порядке битов).	+++++																																							
hw_type - 16 битов	hw_type																Reserved																							
Тип оборудования.	+++++																																							
hw_addr - 64 бита	hw_addr																																							
Размер аппаратного адреса.	+++++																																							
hw_addr - 64 бита	hw_addr																																							
Аппаратный адрес.	+++++																																							
data_len - 32 бита	data_len																																							
Размер данных в пакете.	+++++																																							
Payload – размер задается полем data_len	Payload . . .																																							
Данные из полученного пакета.	+++++																																							

	0	1								2								3																						
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Формат сообщений типа Verdict показан на рисунке справа.	+++++																																							
Value - 32 бита	Value																																							
Решение, принятое по отношению к пакету, который по-прежнему находится в FEC. Поддерживаются значения:	+++++																																							
NF_ACCEPT – принять пакет для дальнейшей обработки.	Packet ID																																							
NF_DROP - отбросить (Drop) пакет.	Data Length																																							
Packet ID - 32 бита	Payload . . .																																							
Уникальный идентификатор пакета, передаваемый CPC от FEC.	+++++																																							
Data Length - 32 бита	Data Length																																							
Размер данных в измененном пакете (в байтах). Если пакет не был изменен, это поле имеет значение 0.	+++++																																							
Payload – размер определяется значением поля Data Length.	Payload . . .																																							

3.3. Служба NETLINK_ARPD

Этот сервис используется CPC для поддержки таблицы соседей в FE. Формат сообщений, передаваемых между FEC и CPC, описан параграфе, посвященном службе учета соседей (стр. 10).

Предполагается, что сервис CPC принимает участие в работе протоколов организации соседских отношений (neighbor solicitation protocol).

Сообщение типа RTM_NEWNEIGH передается CPC от FE для информирования CPC об изменениях, которые могут произойти с записью для этого соседа.

Сообщения RTM_GETNEIGH используются для получения информации о конкретном соседе.

4. Литература

4.1. Нормативные документы

- [1] Braden, R., Clark, D. and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.
- [2] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812²⁶, June 1995.
- [3] Blake, S., Black, D., Carlson, M., Davies, E, Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [4] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R. and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.
- [5] Moy, J., "OSPF Version 2", STD 54, RFC 2328²⁶, April 1998.
- [6] Case, J., Fedor, M., Schoffstall, M. and C. Davin, "Simple Network Management Protocol (SNMP)", STD 15, RFC 1157²⁶, May 1990.
- [7] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and B. Thomas, "LDP Specification", RFC 3036, January 2001.
- [8] Bernet, Y., Blake, S., Grossman, D. and A. Smith, "An Informal Management Model for DiffServ Routers", RFC 3290, May 2002.

4.2. Дополнительная литература

- [9] G. R. Wright, W. Richard Stevens. "TCP/IP Illustrated Volume 2, Chapter 20", June 1995.
- [10] <http://www.netfilter.org>
- [11] <http://diffserv.sourceforge.net>

5. Вопросы безопасности

Netlink работает в безопасной среде (trusted environment) на одном хосте с разделением ядра и пользовательского пространства. Средствами Linux обеспечивается возможность открывать сокет только для процессов с флагом возможностей CAP_NET_ADMIN (обычно процессы, запущенные пользователем root).

6. Благодарности

- 1) **Andi Kleen** за страницы руководства (man pages) для netlink и rtnetlink.
- 2) **Alexey Kuznetsov** за добавление модели службы доставки IP в Netlink. Исходный вариант символического устройства Netlink создал Alan Cox.
- 3) **Jeremy Ethridge** за исполнение роли "непонимающего Netlink" и обзор документа с точки зрения его восприятия.

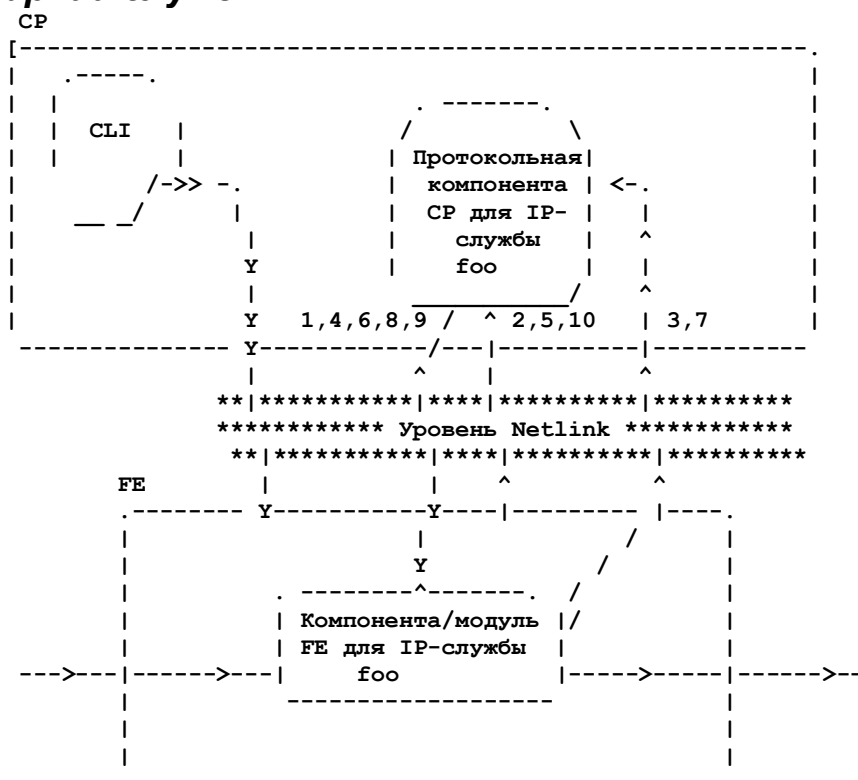
Приложение 1: Пример иерархии служб

На рисунке справа показан пример единичного IP-сервиса **foo** и взаимодействие компонент CP и FE для этой службы (метки 1-3).

Эта схема используется так же как пример адресации CP->FE. В этом приложении иллюстрируется только семантика адресации. В Приложении 2 эта схема рассматривается с точки зрения протокольного взаимодействия между компонентами CPC и FEC сервиса (метки 4-10).

Протокол плоскости управления для IP-службы **foo** выполняет перечисленные ниже операции для подключения к FE (нумерация в списке соответствует номерам на рисунке).

- 1) Подключение к IP-сервису **foo** через сокет. Обычно соединение организуется с помощью вызова `socket(AF_NETLINK, SOCK_RAW, NETLINK_FOO)`.
- 2) Привязка с целью прослушивания специфических асинхронных событий для сервиса **foo**.
- 3) Привязка с целью прослушивания специфических асинхронных событий FE.



²⁶ На сайте www.protocols.ru имеется перевод этого документа на русский язык. Прим. перев.

Приложение 2: Пример протокола для IP-службы Foo

В этом примере IP-сервис foo используется теперь для демонстрации простого управления сервисом IP с использованием Netlink.

Этапы этого управления осуществляются после операций, указанных в Приложении 1 и списки используют общую нумерацию.

- 4) Запрашивается текущая конфигурация компоненты FE.
- 5) Принимается отклик на запрос (4) через канал, организованный на этапе (3).
- 6) запрашивается текущее состояние IP-сервиса foo.
- 7) Принимается отклик на запрос (6) через канал (2).
- 8) регистрируются связанные с протоколом пакеты, которые хочется получать от FE.
- 9) Передаются специфические для данной службы команды foo и (при необходимости) принимаются отклики на них.

Приложение 2а: Взаимодействие с другими службами IP

На схеме в Приложении 1 показана другая компонента, которая может конфигурировать тот же сервис. В данном случае это фирменный командный интерфейс CLI²⁷. Интерфейс CLI может или не может использоваться Netlink для взаимодействия с компонентами foo. Если CLI дает команды, которые оказывают влияние на политику FEC для сервиса foo компонента CPC получает уведомления об этом. На основе этих уведомлений может приниматься решение. Например, если FE позволяет другому сервису удалять правила, установленные иной службой и установленные foo правила были удалены сервисом bar, может возникнуть необходимость распространить это всем партнерам службы foo.

Приложение 3: Примеры

В этом примере рассматривается простое конфигурационное сообщение Netlink, передаваемое от TC CPC выходной очереди TC FIFO. Этот алгоритм управления очередью основан на учете пакетов и отбрасывании пакетов при достижении порогового значения 100. Предполагается, что очередь находится в иерархии с родителем Parent = 100:0 и Classid = 100:1 и размещается на устройстве с ifindex = 4.

Адреса авторов

Jamal Hadi Salim

Znyx Networks

Ottawa, Ontario

Canada

EMail: hadi@znyx.com

Hormuzd M Khosravi

Intel

2111 N.E. 25th Avenue JF3-206

Hillsboro OR 97124-5961

USA

Phone: +1 503 264 0334

EMail: hormuzd.m.khosravi@intel.com

Andi Kleen

SuSE

Stahlgruberring 28

81829 Muenchen

Germany

EMail: ak@suse.de

Alexey Kuznetsov

INR/Swsoft

Moscow

Russia

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Length (52)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type (RTM_NEWQDISC)           | Flags (NLM_F_EXCL |                   |
|                               | NLM_F_CREATE | NLM_F_REQUEST) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Sequence Number (произвольное значение) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Process ID (0)                             |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Family (AF_INET) | Reserved1 | Reserved1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Interface Index (4)                       |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Qdisc handle (0x1000001)                   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Parent Qdisc (0x1000000)                   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               TCM Info (0)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Type (TCA_KIND) | Length (4) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Value ("pfifo")                             |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Type (TCA_OPTIONS) | Length (4) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Value (limit=100)                           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

²⁷ Command Line Interface

E-Mail: kuznet@ms2.inr.ac.ru

Перевод на русский язык

Николай Малых

BiLiM Systems

Russia

nmalykh@bilim.com

Полное заявление авторских прав

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Подтверждение

Финансирование функций RFC Editor в настоящее время обеспечивается Internet Society.