**Министерство науки и образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования**

**«Донской государственный технический университет»**

**(ДГТУ)**

**Сети и телекоммуникации**

Учебное пособие

Ростов-на-Дону

2016

**УДК 004.7**

**Г 15**

|  |  |
| --- | --- |
| **Рецензенты** | декан факультета «Энергетика и системы коммуникаций» Донского государственного технического университета, доктор технических наук, профессор Цыгулёв Н.И.  старший научный сотрудник АО «КБ «Связь», доктор технических наук, профессор А.И. Долгов |

**Автор** кандидат технических наук, доцент кафедры «Вычислительные системы и информационная безопасность» Галушка В.В.

|  |  |
| --- | --- |
| **Г 15** | **Сети и системы передачи данных: учебное пособие / Галушка В.В. — Ростов н/Д: Издательский центр ДГТУ, 2016. — 105 с.** |

**ISBN 978-5-7890-1117-1**

Учебное пособие содержит теоретические и практические сведения о методах и средствах построения компьютерных сетей. Рассматриваются базовые топологии сетей, методы коммутации, мультиплексирования, модуляции и кодирования. Особое внимание уделено уровням модели сетевого взаимодействия TCP/IP. Для каждого из них даётся описание наиболее распространённых сетевых протоколов, поясняются принципы функционирования сетевого оборудования. Отдельно описываются наиболее важные технологии: маршрутизация, трансляция сетевых адресов и построения виртуальных частных сетей. Пособие предназначено для студентов технических направлений подготовки и специальностей, изучающих дисциплины, связанные с проектированием и разработкой информационных сетей.

**УДК 004.7**

**Печатается по решению редакционно-издательского совета Донского государственного технического университета**

|  |  |
| --- | --- |
| **Научный редактор** | канд. техн. наук, доцент кафедры «Вычислительные системы и информационная безопасность» А.Р. Айдинян (г. Ростов-на-Дону) |

**© Галушка В.В.**

**ISBN 978-5-7890-1117-1 © Издательский центр ДГТУ, 2016**

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc446355952)

[1 ОБЩИЕ СВЕДЕНИЯ О КОМПЬЮТЕРНЫХ СЕТЯХ 6](#_Toc446355953)

[1.1 Определение и классификация сетей 6](#_Toc446355954)

[1.2 Базовые топологии сетей 8](#_Toc446355955)

[1.3 Сети с коммутацией каналов и сети с коммутацией пакетов 11](#_Toc446355956)

[2 МОДЕЛИ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ 14](#_Toc446355957)

[2.1 Эталонная модель взаимодействия открытых систем (OSI — Open System Interconnection) 14](#_Toc446355958)

[2.2 Модель DOD. Стек протоколов TCP/IP 17](#_Toc446355959)

[3 ФИЗИЧЕСКИЙ УРОВЕНЬ 18](#_Toc446355960)

[3.1 Типы линий связи 19](#_Toc446355961)

[3.2 Модуляция и кодирование 23](#_Toc446355962)

[3.3 Оборудование физического уровня 26](#_Toc446355963)

[3.4 Режимы передачи данных 26](#_Toc446355964)

[4 КАНАЛЬНЫЙ УРОВЕНЬ 27](#_Toc446355965)

[4.1 Подуровень MAC (Media Access Control — управления доступом к среде) 28](#_Toc446355966)

[4.2 Метод доступа к разделяемой среде CSMA/CD 29](#_Toc446355967)

[4.3 MAC-адрес 31](#_Toc446355968)

[4.4 Подуровень LLC (Logical Link Control — управление логической связью) 33](#_Toc446355969)

[4.5 Оборудование канального уровня 35](#_Toc446355970)

[4.6 Сетевые технологии канального уровня 36](#_Toc446355971)

[4.7 Семейство стандартов IEEE 802.3 (Ethernet) 38](#_Toc446355972)

[5 СЕТЕВОЙ УРОВЕНЬ 42](#_Toc446355973)

[5.1 Адресация узлов сети 43](#_Toc446355974)

[5.2 Формат IP-адреса 44](#_Toc446355975)

[5.3 Классовая и бесклассовая адресация. Маска подсети 44](#_Toc446355976)

[5.4 Особые IP-адреса 47](#_Toc446355977)

[5.5 Порядок назначения IP-адресов 49](#_Toc446355978)

[5.4 Протокол ARP 51](#_Toc446355979)

[5.5 Протокол IPv6 53](#_Toc446355980)

[6 МАРШРУТИЗАЦИЯ 55](#_Toc446355981)

[6.1 Протокол маршрутизации RIP 57](#_Toc446355982)

[6.2 Протокол маршрутизации OSPF 61](#_Toc446355983)

[7 ТРАНСПОРТНЫЙ УРОВЕНЬ 64](#_Toc446355984)

[7.1 Понятие порта 64](#_Toc446355985)

[7.2 Протокол UDP 66](#_Toc446355986)

[7.3 Протокол TCP 67](#_Toc446355987)

[7.4 Метод скользящего окна 68](#_Toc446355988)

[8 ТРАНСЛЯЦИЯ СЕТЕВЫХ АДРЕСОВ 71](#_Toc446355989)

[8.1 Виды NAT 72](#_Toc446355990)

[8.2 Технология перенаправления портов 74](#_Toc446355991)

[9 ВИРТУАЛЬНЫЕ ЧАСТНЫЕ СЕТИ (VPN) 76](#_Toc446355992)

[9.1 Туннелирование 78](#_Toc446355993)

[9.2 Шифрование в виртуальных сетях 80](#_Toc446355994)

[10 ПРИКЛАДНОЙ УРОВЕНЬ 83](#_Toc446355995)

[10.1 Система доменных имён 83](#_Toc446355996)

[10.2 Схема работы DNS 86](#_Toc446355997)

[10.3 Протоколы RDP и VNC 88](#_Toc446355998)

[10.4 Протокол DHCP 89](#_Toc446355999)

[КОНТРОЛЬНЫЕ ВОПРОСЫ 91](#_Toc446356000)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 93](#_Toc446356001)

# ВВЕДЕНИЕ

Сети и системы передачи информации — это обширная область знаний, лежащая на границе между информационными технологиями, электроникой и радиотехникой, которая включает в себя физические линии связи, методы передачи по ним сигналов, алгоритмы поиска путей через сложную сеть, кодирование и шифрование информации и многие другие аспекты.

В процессе построения сетей возникает множество задач, и, за сравнительно недолгую историю развития сетевых технологий, для каждой такой задачи было предложено множество решений, подробное описание которых если и возможно, то займёт не одну сотню страниц. Поэтому цель данного пособия — дать представления об общих принципах построения и функционирования компьютерных сетей, основных используемых на практике технологиях и оборудовании передачи данных, протоколах информационного обмена.

В учебном пособии обсуждаются как теоретические, так и практические вопросы построения компьютерных сетей. Его структура во многом соответствует эталонной модели сетевого взаимодействия (OSI), каждому из наиболее важных уровней которой посвящена отдельная глава. При этом рассматриваются не только общие методы решения задач передачи данных на каждом уровне, но и их конкретные реализации в распространённых сетевых протоколах, а также особенности работы сетевого оборудования. Некоторым технологиям, знание которых важно с практической точки зрения, уделено отдельное внимание. Это касается маршрутизации и трансляции сетевых адресов без которых не обходится сегодня ни одна крупная сеть, а также виртуальным частным сетям, получившим широкое распространение в территориально распределённых организациях.

Пособие предназначено для студентов технических направлений подготовки, изучающих дисциплины, касающиеся методов и средств связи, способов построения компьютерных сетей и передачи данных по ним. Приведенные в пособии сведения могут использоваться студентами при дипломном проектировании, если его тема связана как непосредственно с сетями, так и с их безопасностью. Пособие также может быть использовано специалистами в области разработки систем и средств передачи данных.

# 1 ОБЩИЕ СВЕДЕНИЯ О КОМПЬЮТЕРНЫХ СЕТЯХ

## 1.1 Определение и классификация сетей

|  |
| --- |
| Компьютерная сеть (вычислительная сеть, сеть передачи данных) — совокупность компьютеров, соединенных в единую систему с помощью каналов связи и средств коммутации. |

Два компьютера соединенных друг с другом кабелем или беспроводным каналом связи, уже представляют собой простейшую компьютерную сеть. Это позволяет им использовать свои ресурсы совместно.

Главная цель объединения компьютеров в сеть — разделение ресурсов, таких как:

* периферийные устройства (принтеры, сканеры);
* данные, хранящиеся в оперативной памяти или на дисках;
* вычислительная мощность.

В зависимости от признака, существует несколько вариантов классификации компьютерных сетей.

По территориальной распространенности:

* PAN (Personal Area Network) — персональная сеть, предназначенная для взаимодействия различных устройств, принадлежащих одному владельцу.
* LAN (Local Area Network) — локальные сети, сосредоточенные в пределах одного или нескольких ближайших зданий. Строятся с использованием высококачественных линий связи, которые позволяют, применяя простые методы передачи данных, достигать высоких скоростей. Локальные сети являются сетями закрытого типа, доступ к ним разрешен только ограниченному кругу пользователей, для которых работа в такой сети непосредственно связана с их профессиональной деятельностью.
* MAN (Metropolitan Area Network) — городские сети между учреждениями в пределах одного или нескольких городов, связывающие много локальных вычислительных сетей.
* WAN (Wide Area Network) — глобальная сеть, покрывающая большие географические регионы, включающие в себя как локальные сети, так и прочие телекоммуникационные сети и устройства. Часто используют уже существующие не очень качественные линии связи. Более низкие, чем в локальных сетях, скорости передачи данных (десятки килобит в секунду) ограничивают набор предоставляемых услуг передачей файлов, преимущественно не в оперативном, а в фоновом режиме, с использованием электронной почты. Для устойчивой передачи дискретных данных применяются более сложные методы и оборудование, чем в локальных сетях. Глобальные сети являются открытыми и ориентированы на обслуживание любых пользователей.
* Термин «корпоративная сеть» также используется в литературе для обозначения объединения нескольких сетей, каждая из которых может быть построена на различных технических, программных и информационных принципах.

Хронологически первыми появились глобальные сети. Именно при их построении были впервые предложены и отработаны многие основные идеи, лежащие в основе современных вычислительных сетей. Глобальные компьютерные сети очень многое унаследовали от других, гораздо более старых и распространенных глобальных сетей — телефонных. Их развитие началось в 1969 году, когда министерство обороны США инициировало работы по объединению в единую сеть суперкомпьютеров оборонных и научно-исследовательских центров. Эта сеть, получившая название ARPANET (Advanced Research Projects Agency Network), стала отправной точкой для создания первой и самой известной ныне глобальной сети — Интернет.

В свою очередь, прообразом локальных сетей послужили многотерминальные системы, состоящие из одного центрального компьютера и подключённых к нему пользовательских терминалов, а реальное их развитие началось с появлением в пределах одного здания нескольких компьютеров. Несмотря на разную историю появления, для современного этапа характерно сближение технологий локальных и глобальных сетей.

Следующий вариант классификации — по типу функционального взаимодействия. При нём выделяют следующие типы сетей:

* одноранговая, в которой все участники равноправны;
* многоранговая сеть, в которой участники имеют различные права;
* клиент-серверная, в которой существуют явно выделенные устройства, предоставляющие ресурсы для общего пользования — серверы и устройства, осуществляющий доступ к ресурсам сервера — клиенты;
* смешанная сеть, сочетающая в себе свойства нескольких перечисленных выше.

По типу среды передачи сети делятся на:

* проводные (используют телефонный провод, коаксиальный кабель, витую пара, волоконно-оптический кабель);
* беспроводные (основаны на передаче информации по радиоволнам в определенном частотном диапазоне).

Также существует классификация сетей по скорости передачи данных, при которой они условно делятся на следующие типы:

* низкоскоростные (до 10 Мбит/с);
* среднескоростные (до 100 Мбит/с);
* высокоскоростные (свыше 100 Мбит/с).

## 1.2 Базовые топологии сетей

Объединяя в сеть несколько (больше двух) компьютеров, необходимо решить, каким образом соединить их друг с другом, другими словами, выбрать конфигурацию физических связей, или топологию.

|  |
| --- |
| Сетевая топология — схема расположения и соединения сетевых устройств, описываемая конфигурацией графа, вершинам которого соответствуют конечные узлы сети и коммуникационное оборудование, а рёбрам — физические или информационные связи между устройствами. |

Можно соединять каждый компьютер с каждым или же связывать их последовательно, предполагая, что они будут общаться, передавая сообщения друг другу «транзитом». Транзитные узлы должны быть оснащены специальными средствами, позволяющими им выполнять эту специфическую посредническую операцию. В качестве транзитного узла может выступать как универсальный компьютер, так и специализированное устройство.

От выбора топологии связей существенно зависят характеристики сети. Например, наличие между узлами нескольких путей повышает надежность сети и делает возможным распределение загрузки между отдельными каналами. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи.

Полносвязная топология (рис. 1.1) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными.



Рис. 1.1 — Полносвязная топология.

Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, в таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с, каждым из остальных компьютеров сети.

Все другие варианты основаны на неполносвязных топологиях, когда для обмена данными между двумя компьютерами может потребоваться транзитная передача через другие узлы сети.

В сетях с кольцевой топологией (рис. 1.2) данные передаются по кольцу от одного компьютера к другому.



Рис. 1.2 — Топология кольцо.

Главным достоинством кольца является то, что оно по своей природе обеспечивает резервирование связей. Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против нее. Кроме того, кольцо представляет собой очень удобную конфигурацию для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату.

Звезда (рис. 1.3) образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству — концентратору.



Рис. 1.3 — Топология звезда.

В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. К недостаткам звездообразной топологии относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

В топологии общая шина (рис. 1.4) используется один кабель, к которому подключены все компьютеры сети. В каждый момент времени вести передачу может только один компьютер. Данные передаются всем компьютерам сети, однако информацию принимает только тот компьютер, чей адрес соответствует адресу получателя.



Рис. 1.4 — Топология общая шина.

Основными преимуществами такой схемы являются ее дешевизна и простота присоединения новых узлов к сети, а недостатками — низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

Базовые топологии редко применяются в реальных сетях в своём оригинальном виде, чаще используются смешанные топологии (рис. 1.5), представляющие собой комбинацию нескольких базовых.



Рис. 1.5 — Смешанная топология.

Среди смешанных наиболее популярной является топология иерархическая звезда (рис. 1.6), по которой построены подавляющее большинство современных локальных сетей.



Рис. 1.6 — Иерархическая звезда или дерево.

При соединении в сеть большого числа компьютеров возникают несколько проблем, среди которых основные — адресация и коммутация.

## 1.3 Сети с коммутацией каналов и сети с коммутацией пакетов

|  |
| --- |
| Коммутация — это процесс соединения узлов сети через транзитные узлы. |

При коммутации каналов сеть образует между конечными узлами непрерывный составной физический канал из последовательно соединенных коммутаторами промежуточных участков. Перед передачей данных всегда необходимо выполнить процедуру установления соединения, в процессе которой и создается составной канал.

Как правило, операцию коммутации сопровождает также операция мультиплексирования и обратная ей — демультиплексирование.

|  |
| --- |
| Мультиплексирование — образование из нескольких отдельных потоков общего агрегированного потока, который передается по одному физическому каналу связи. |

В сетях применяются два основных вида мультиплексирования:

* Частотное, когда каждому сигналу отводится определенная доля всей частотной полосы носителя, так что на одном носителе существуют одновременно сразу несколько сигналов (рис. 1.7).
* Временное, когда сигналу каждого приложения выделяется вся полоса носителя, но на короткий промежуток времени — тайм-слот, так что мультиплексный сигнал представляется в виде последовательности сменяющих друг друга тайм-слотов, ответственных за разные приложения.



Рис. 1.7 — Частотное мультиплексирование.

Коммутация каналов имеет ряд преимуществ:

* постоянная и известная скорость передачи данных по установленному между конечными узлами каналу;
* низкий и постоянный уровень задержки передачи данных через сеть. Это позволяет качественно передавать данные, чувствительные к задержкам, называемые также трафиком реального времени (голос, видео).

Однако, рассматриваемому способу коммутации свойственны и некоторые недостатки:

* возможность отказа сети в установлении соединения из-за полной загруженности части составного канала. Отказ может случиться и на конечном участке составного канала — например, если абонент способен поддерживать только одно соединение, что характерно для многих телефонных сетей.
* обязательная задержка перед передачей данных из-за фазы установления соединения.
* нерациональное использование пропускной способности физических каналов.

Сетевые приложения генерируют трафик очень неравномерно, с высоким уровнем пульсации скорости передачи данных. Примером может служить просмотр web-страниц, когда на их загрузку тратится очень небольшое количество времени по сравнению со временем, затрачиваемым на просмотр, когда не имеет смысла держать открытым составной канал связи расходуя на него часть пропускной способности. Для эффективной передачи компьютерного трафика используется коммутация пакетов (рис. 1.8).

При данном виде коммутации все передаваемые пользователем сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами (кадрами, фреймами, ячейками). Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета на узел назначения, а также номер пакета, который будет использоваться узлом назначения для сборки исходного сообщения. Пакеты транспортируются по сети как независимые информационные блоки. Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге — узлу назначения.



Рис. 1.8 — Коммутация пакетов.

Коммутаторы пакетной сети отличаются от коммутаторов каналов тем, что они имеют внутреннюю буферную память для временного хранения пакетов, если выходной порт коммутатора в момент принятия пакета занят передачей другого пакета. В этом случае пакет находится некоторое время в очереди пакетов в буферной памяти выходного порта, а когда до него дойдет очередь, он передается следующему коммутатору. Такая схема передачи данных позволяет сглаживать пульсацию трафика на магистральных каналах связи между коммутаторами и тем самым наиболее эффективно использовать их для повышения пропускной способности сети в целом.

Достоинства коммутации пакетов.

* Высокая общая пропускная способность сети при передаче пульсирующего трафика.
* Возможность динамически перераспределять пропускную способность физических каналов связи между абонентами в соответствии с реальными потребностями их трафика.

Недостатки.

* Неопределенность скорости передачи данных между абонентами сети, обусловленная тем, что задержки в очередях буферов коммутаторов сети зависят от общей загрузки сети.
* Переменная величина задержки пакетов данных, которая может быть достаточно продолжительной в моменты мгновенных перегрузок сети.

# 2 МОДЕЛИ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ

## 2.1 Эталонная модель взаимодействия открытых систем (OSI — Open System Interconnection)

Открытой может быть названа любая система (компьютер, вычислительная сеть, операционная система, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями, то есть формализованным описанием аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации и других характеристик.

Эталонная модель OSI представляет собой 7-уровневую сетевую иерархию (рис. 2.1), разработанную международной организацией по стандартам. Каждый уровень отвечает за строго определенный аспект взаимодействия сетевых устройств.



Рис. 2.1 — Модель OSI.

Уровень 1 — физический

Физический уровень получает пакеты данных от вышележащего канального уровня и преобразует их в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока. Эти сигналы посылаются через среду передачи на приемный узел. Механические и электрические/оптические свойства среды передачи определяются на физическом уровне и включают:

* тип кабелей и разъемов;
* разводку контактов в разъемах;
* схему кодирования сигналов для значений 0 и 1.

Уровень 2 — канальный

Обеспечивает создание, передачу и прием кадров данных посредством выполнения следующий функций:

* установление логического соединения между взаимодействующими узлами;
* согласование в рамках соединения скоростей передатчика и приемника информации;
* обеспечение надежной передачи, обнаружение и коррекция ошибок.

Уровень 3 — сетевой

Средства данного уровня отвечают за выполнение следующих функций:

* адресация узлов сети;
* поиск узла по его адресу;
* продвижение данных через промежуточные узлы сети к конечному устройству.

Уровень 4 — транспортный

Предназначен для обеспечения надёжной передачи данных от отправителя к получателю через составную сеть.

Уровень 5 — сеансовый

Отвечает за организацию и поддержание сеансов обмена данными между оконечными устройствами, включая синхронизацию процесса обмена данными.

Уровень 6 — представительский

Обеспечивает преобразование данных (кодирование, компрессия и т.п.) прикладного уровня в поток информации для транспортного уровня. Уровень представления отвечает за возможность диалога между приложениями на разных машинах.

Уровень 7 — прикладной

Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами с помощью стандартизированных форматов сообщений. Эти правила принято называть интерфейсом.

|  |
| --- |
| Интерфейс определяет последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на соседних уровнях в одном узле. Интерфейс определяет набор услуг, предоставляемый данным уровнем соседнему уровню. |

|  |
| --- |
| Сетевой протокол — набор правил и действий (очерёдности действий), позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами на одном уровне модели OSI. |

Средства каждого уровня должны отрабатывать, во-первых, собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

## 2.2 Модель DOD. Стек протоколов TCP/IP

Модель OSI является эталонной, однако, существующие на практике средства организации сетевого взаимодействия, редко в точности соответствуют этой модели, что обусловлено исторически сложившимися особенностями развития сетевых технологий. Так, например, существующие сегодня способы построения сетей берут своё начало с сети ARPANET, созданной, агентством передовых оборонных исследовательских проектов министерства обороны США, которое разработало и применило при создании ARPANET собственную модель (DOD — Department Of Defense).

В отличие от модели OSI, модель DOD состоит из четырёх уровней (сверху вниз) (рис. 2.2):

* Уровня приложений (Прикладной уровень) (англ. Process/Application), соответствующего трем верхним уровням модели OSI.
* Транспортного уровня (англ. Transport), соответствующего транспортному уровню модели OSI.
* Межсетевого уровня (англ. Internet), соответствующего сетевому уровню модели OSI.
* Уровня сетевого доступа (англ. Network Access), соответствующего двум нижним уровням модели OSI.



Рис. 2.2 — Соответствие уровней моделей OSI и DOD.

Более того был разработан собственный набор протоколов для каждого уровня, соответствующий этой модели. Самыми известными из них являются IP — Internet Protocol и TCP — Transmission Control Protocol.

|  |
| --- |
| Согласованный иерархический набор протоколов разных уровней, достаточный для организации межсетевого взаимодействия, называется стеком протоколов. |

Самый распространённый на сегодня стек протоколов называется TCP/IP и полностью соответствует именно модели DOD, однако помимо протоколов, указанных в его названии, он включает в себя и другие, каждый из которых относится к определённому уровню.

Уровень приложений:

* HTTP (HyperText Transfer Protocol) — протокол передачи HTML-страниц,
* SMTP (Simple Mail Transfer Protocol) — протокол передачи почты,
* FTP (File Transfer Protocol) — протокол передачи файлов,
* и многие другие.

Транспортный:

* TCP (Transmission Control Protocol),
* UDP (User Datagram Protocol).

Сетевой:

* IP (Internet Protocol),
* ICMP (Internet Control Message Protocol) — служебный протокол для проверки и анализа состояния сети,
* Протоколы маршрутизации RIP (Routing Information Protocol), OSPF (Open Shortest Path First), BGP (Border Gateway Protocol) и другие.

Уровень доступа к среде:

* Ethernet,
* 802.11 Wi-Fi,
* ATM.

# 3 ФИЗИЧЕСКИЙ УРОВЕНЬ

К данному уровню относятся линии связи и методы представления данных для передачи по этим линиям. Единица измерения, используемая на физическом уровне — биты, то есть двоичные нули и единицы, а точнее, их представление в виде сигналов, соответствующих используемой среде передачи. Часть сетевого оборудования работает исключительно на физическом уровне.

## 3.1 Типы линий связи

Выделяют 3 типа линий связи: проводные, кабельные и беспроводные.

Проводные линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплёток.

Кабельные линии имеют более сложную конструкцию. Кабель состоит из проводников, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической и, возможно, климатической. Кроме того, кабель может быть оснащен разъёмами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных (и телекоммуникационных) сетях применяются три основных типа кабеля:

* коаксиальные кабели с медной жилой;
* кабели на основе скрученных пар медных проводов — неэкранированная витая пара (Unshielded Twisted Pair, UTP) и экранированная витая пара (Shielded Twisted Pair, STP);
* волоконно-оптические кабели.

Коаксиальный (от англ. Co-Axis — соосный) кабель состоит из несимметричных пар проводников. Каждая пара представляет собой внутреннюю медную жилу и соосную с ней внешнюю жилу, которая может быть полой медной трубой или оплеткой, отделенной от внутренней жилы диэлектрической изоляцией (рис. 3.1). Внешняя жила играет двоякую роль — по ней передаются информационные сигналы, и она является экраном, защищающим внутреннюю жилу от внешних электромагнитных полей.

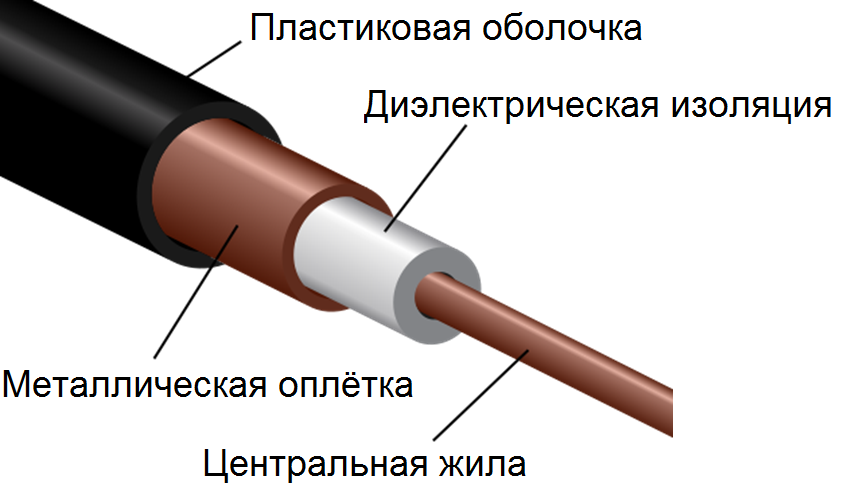


Рис. 3.1 — Коаксиальный кабель.

Существует несколько типов коаксиального кабеля, отличающихся характеристиками и областями применения:

* «Тонкий» коаксиальный кабель имеет диаметр примерно 6 мм и значительную гибкость, которая позволяет прокладывать его практически в любых местах.
* «Толстый» коаксиальный кабель с диаметром 12 мм и более толстой центральной жилой, обеспечивающий передачу данных на большее по сравнению с тонким коаксиальным кабелем расстояние.
* Телевизионный кабель. Существуют стандарты локальных сетей, позволяющие использовать такой кабель для передачи данных.

Витой парой называется скрученная пара проводов. Этот вид среды передачи данных очень популярен и составляет основу большого количества как внутренних, так и внешних кабелей. Кабель может состоять из нескольких скрученных пар (внешние кабели иногда содержат до нескольких десятков таких пар). Скручивание проводов снижает влияние внешних и взаимных помех на полезные сигналы, передаваемые по кабелю.

Симметричный кабель на основе витой пары может быть как экранированным (Shielded Twisted Pair — STP), так и неэкранированным (Unshielded Twisted Pair — UTP). Кабель на основе неэкранированной витой пары, используемый для проводки внутри здания, разделяется в международных стандартах на 7 категорий, наиболее распространённой из которых является 5 категория. На её использование ориентированы большинство современных стандартов локальных сетей.

Все кабели UTP независимо от их категории выпускаются в 4-парном исполнении. Каждая из четырех пар кабеля имеет определенный цвет и шаг скрутки (рис. 3.2).

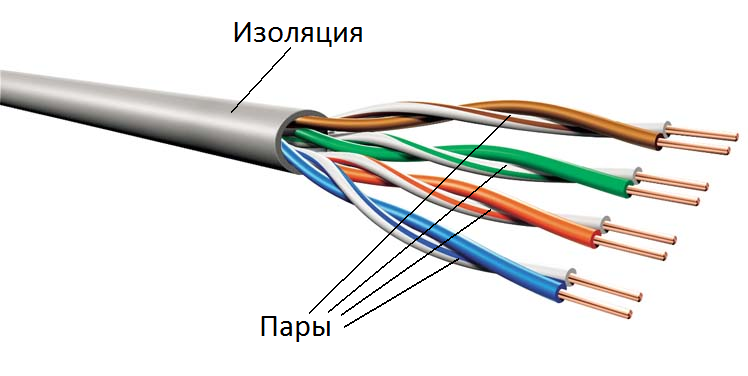


Рис. 3.2 — Витая пара

Волоконно-оптический кабель состоит из тонких гибких стеклянных волокон (волоконных световодов), по которым распространяются световые сигналы. Это наиболее качественный тип кабеля — он обеспечивает передачу данных с очень высокой скоростью (до 10 Гбит/с и выше) и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех (в силу особенностей распространения света такие сигналы легко экранировать).

Каждый световод состоит из центрального проводника света (сердцевины) — стеклянного волокна, и стеклянной оболочки, обладающей меньшим показателем преломления, чем сердцевина. Распространяясь по сердцевине, лучи света не выходят за её пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и величины диаметра сердечника различают:

* многомодовое волокно со ступенчатым изменением показателя преломления (рис. 3.3 а);
* многомодовое волокно с плавным изменением показателя преломления (рис. 3.3 б);
* одномодовое волокно (рис. 3.3 в).

Мода (в оптике) — это стабильное состояние электромагнитного поля внутри световода. Условно моду световода иногда определяют и как траекторию, по которой распространяется свет.

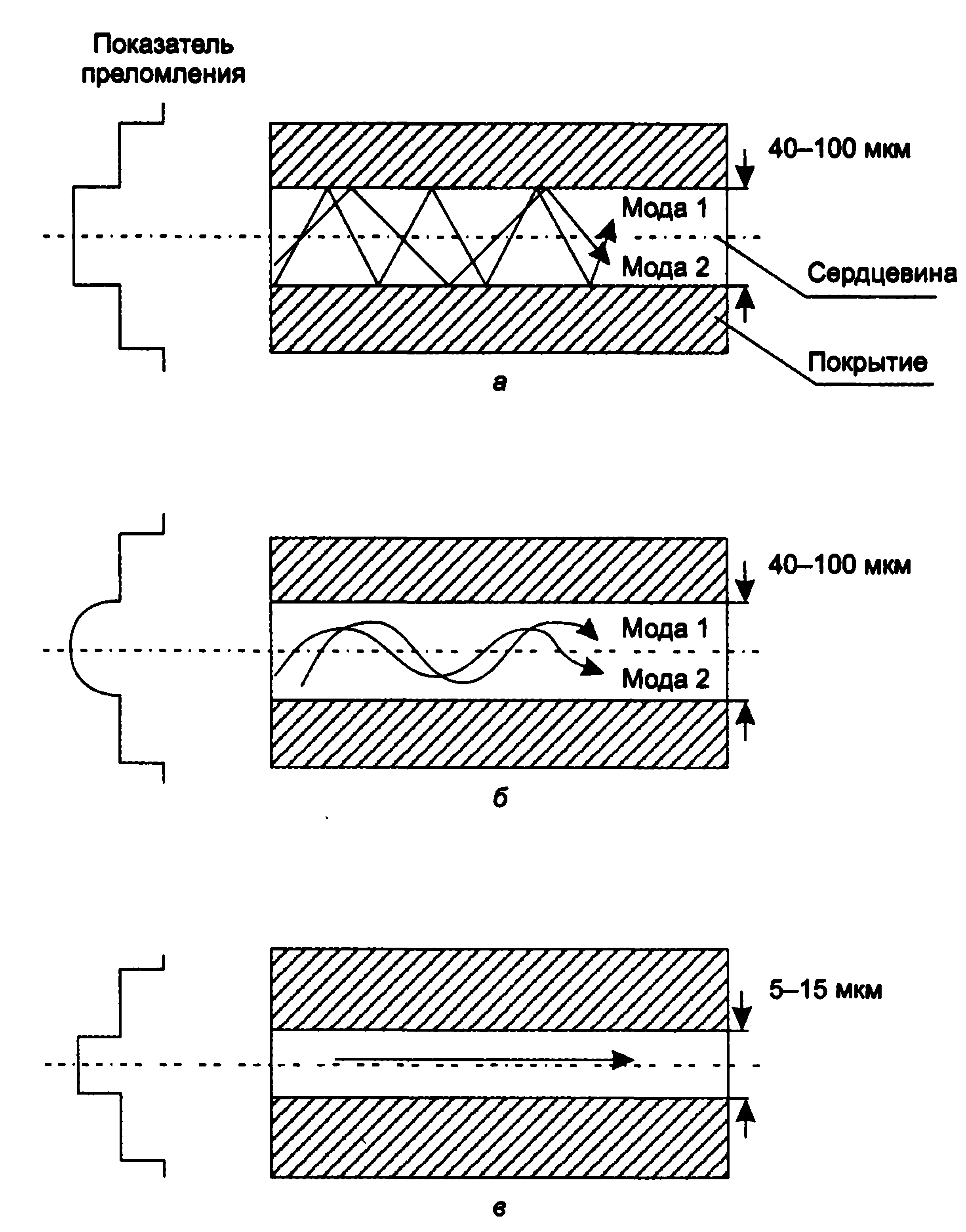


Рис. 3.3 — Виды волоконно-оптических кабелей

В компьютерных сетях сегодня применяются практически все описанные типы физических сред передачи данных. Хорошие возможности предоставляют волоконно-оптические кабели, обладающие широкой полосой пропускания и низкой чувствительностью к помехам. На них сегодня строятся как магистрали крупных территориальных и городских сетей, так и высокоскоростные локальные сети. Популярной средой является также витая пара, которая характеризуется отличным отношением качества к стоимости, а также простотой монтажа. Беспроводные каналы используются чаще всего в тех случаях, когда кабельные линии связи применить нельзя, например, при прохождении канала через малонаселенную местность или же для связи с мобильными пользователями сети.

## 3.2 Модуляция и кодирование

Проводные и беспроводные среды, предоставляют только потенциальную возможность передачи дискретной информации. Для того чтобы передатчик и приемник, соединенные некоторой средой, могли обмениваться информацией, им необходимо договориться о том, какие сигналы будут соответствовать двоичным единицам и нулям дискретной информации.

Для представления дискретной информации в среде передачи данных применяются сигналы двух типов: прямоугольные импульсы и синусоидальные волны. В первом случае используют термин «кодирование», во втором — «модуляция».

Модуляция используется при передаче данных по проводным (не кабельным) и беспроводным каналам связи, то есть тем, которые изначально не предназначались для передачи цифровой информации. Кодирование — при передаче по кабельным каналам.

|  |
| --- |
| Модуляция — это процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала. |

В зависимости от выбранного для изменения параметра существуют 3 базовых вида модуляции: амплитудная, частотная и фазовая (рис. 3.4). Для повышения скорости передачи данных также возможно их комбинирование. Более подробное рассмотрение способов модуляции выходит за рамки данного учебного пособия, однако может быть изучено отдельно с использованием литературы [1, 3].

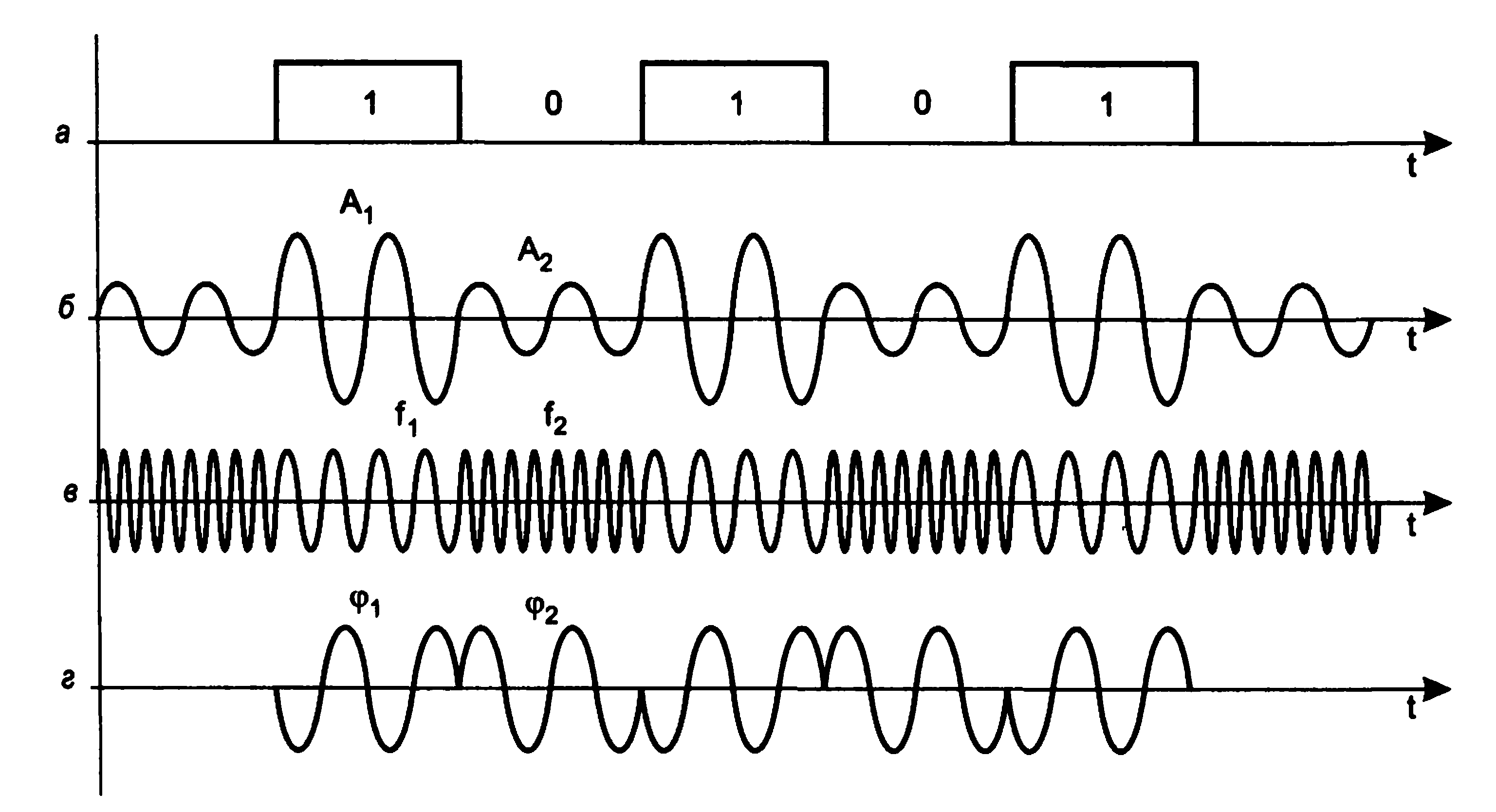


Рис. 3.4 — Виды модуляции дискретных сигналов.

|  |
| --- |
| Кодирование информации — процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки. |

При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей:

* минимизировать ширину спектра сигнала, полученного в результате кодирования;
* обеспечивать синхронизацию между передатчиком и приемником;
* обеспечивать устойчивость к шумам;
* обнаруживать и по возможности исправлять битовые ошибки;
* минимизировать мощность передатчика.

На рисунке 3.5 представлено несколько способов кодирования двоичных сигналов. Самым простым из них является метод потенциального кодирования, называемый также кодированием без возвращения к нулю (Non Return to Zero — NRZ), при котором, в отличие от других методов, при передаче последовательности единиц сигнал не возвращается к нулю в течение такта.

Одной из модификаций метода NRZ является метод биполярного кодирования с альтернативной инверсией (Alternate Mark Inversion — AMI). В этом методе применяются три уровня потенциала — отрицательный, нулевой и положительный. Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

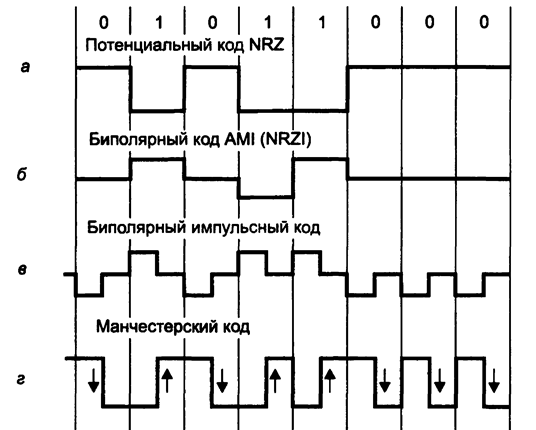


Рис. 3.5 — Методы кодирования.

Помимо потенциальных кодов в сетях используются и импульсные коды, в которых данные представлены полным импульсом или же его частью — фронтом. Наиболее простым кодом такого рода является биполярный импульсный код, в котором единица представляется импульсом одной полярности, а ноль — другой.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль — обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд.

Следует отметить, что биполярный импульсный код и манчестерский код обладают свойством самосинхронизации благодаря тому, что сигнал в них изменяется по крайней мере один раз за такт, то есть даже при передаче длинной последовательности нулей или единиц всегда можно определить, где закончилась передача одного бита и началась передача следующего.

## 3.3 Оборудование физического уровня

К оборудованию сетевого уровня относятся:

* модем;
* хаб;
* повторитель;
* медиаконвертер.

Устройство, которое выполняет функцию модуляции несущей частоты на передающей стороне и обратную функцию демодуляции на приемной стороне, носит название модема (модулятор-демодулятор).

Хаб (сетевой концентратор) — сетевое устройство 1-го (физического) уровня сетевой модели OSI, ретранслирующее входящий сигнал с одного из портов в сигнал на все остальные (подключённые) порты и реализующее, таким образом топологию общая шина.

Повторитель (репитер) — сетевое устройство, выполняющее регенерацию и ресинхронизацию сетевых сигналов на битовом уровне, что позволяет передавать их по среде на большее расстояние.

Медиаконвертер — это устройство, преобразующее сигналы в физической среде одного типа в сигналы другой физической среды. В современной терминологии медиаконвертер работает как связующее звено только между двумя средами — оптическим и медным кабелями.

## 3.4 Режимы передачи данных

Важной характеристикой среды передачи является возможность одновременной передачи информации в одну или обе стороны. Даже в том случае, если пользователю кажется, что он только передаёт или только получает информацию, обмен, на самом деле происходит в двух направлениях: в одном передаются интересующие пользователя данные, а в противоположном — служебная информация, включающая в себя подтверждения о получении этих данных.

Физические каналы связи делятся на несколько типов в зависимости от того, могут они передавать информацию в обоих направлениях или нет:

* симплексный;
* дуплексный;
* полудуплексный.

Симплекс (Simplex) — режим передачи данных, при котором передача ведется только в одном направлении по общему каналу связи. Передача в обратном направлении физически невозможна. Такой способ передачи данных используется для передачи сигнала по телевидению или радио. В этом случае используется один передатчик и много приемников, объединенных общим каналом связи.

Полудуплекс (Half Duplex) — режим передачи данных, при котором передача между устройствами ведется по общему каналу связи в любом направлении, но с разделением по времени.

При таком режиме передачи данных каждое из устройств, подключенное к общему каналу связи должно быть способно попеременно принимать и передавать сигналы, однако, при этом может возникнуть конфликт (коллизия), когда два и более устройства начнут одновременно передавать сигналы по общему каналу. При возникновении коллизии сигналы в канале связи перемешиваются, и другие устройства не способны воспринимать какой-либо из них по отдельности. Поэтому дальнейшая передача теряет смысл, и информация должна быть отправлена заново, что существенно снижает производительность. Такой вид связи используется в основном при обмене информацией некоторого (небольшого) количества устройств по общему каналу связи.

Полный дуплекс (Full Duplex) — режим передачи данных, при котором передача может вестись одновременно в двух направлениях по разным подканалам связи. Одним из вариантов организации дуплексного режима является использование двух независимых линий связи, каждая из которых работает в симплексном режиме.

# 4 КАНАЛЬНЫЙ УРОВЕНЬ

Канальный уровень использует определенные службы физического уровня для отправки и получения битов по каналам связи, и у него есть ряд собственных специфических функций. К ним относятся:

* обеспечение строго определённого служебного интерфейса для сетевого уровня;
* обработка ошибок передачи данных;
* управление потоком данных, исключающее «затопление» медленных приемников быстрыми передатчиками.

В отличие от физического уровня, единицей передачи данных на котором являются биты, единицей данных на канальном уровне является кадр (или фрейм, frame). В каждом кадре содержится заголовок, поле данных и концевик (рис. 4.1). Управление кадрами — это основа деятельности средств канального уровня.



Рис. 4.1 — Структура кадра.

В любом, даже самом надёжном канале связи иногда случаются ошибки при передаче данных. Кроме того, скорость передачи данных ограничена, а время распространения сигнала отлично от нуля. Все эти ограничения оказывают серьезное влияние на эффективность передачи данных и требуют использования протоколов, учитывающих все перечисленные факторы.

## 4.1 Подуровень MAC (Media Access Control — управления доступом к среде)

Все неполносвязные сетевые топологии подразумевают передачу по одному физическому каналу, данных от нескольких устройств.

|  |
| --- |
| Совместно используемый несколькими сетевыми интерфейсами физический канал называют разделяемой средой. |

Существуют различные способы организации совместного доступа к разделяемым линиям связи. В одних случаях используют централизованный подход, когда доступом управляет отдельно выделенное устройство, в других — децентрализованный. Внутри компьютера проблемы разделения линий связи между различными модулями также существуют — примером может служить доступ к системной шине, которым управляет либо процессор, либо специальный арбитр шины. В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по линиям связи. Из-за этого процедуры согласования доступа к линии связи могут занимать слишком много времени и приводить к значительному снижению производительности сети.

Использование разделяемых сред позволяет упростить логику работы узлов сети. Действительно, поскольку в каждый момент времени выполняется только одна передача, отпадает необходимость в буферизации кадров в транзитных узлах и, как, следствие, в самих транзитных узлах. Соответственно, отпадает необходимость в сложных процедурах управления потоком и борьбы с перегрузками.

Основной недостаток разделяемой среды — плохая масштабируемость. Этот недостаток является принципиальным, так как независимо от метода доступа к среде ее пропускная способность делится между всеми узлами сети.

Подуровень MAC появился из-за существования в локальных сетях разделяемой среды передачи данных. Именно этот подуровень обеспечивает корректное совместное использование общей среды, предоставляя её в соответствии с определенным алгоритмом в распоряжение той или иной станции сети. После того как доступ к среде получен, ею могут пользоваться более высокие уровни.

Алгоритм доступа к разделяемой среде является одним из главных факторов, определяющих эффективность совместного использования среды конечными узлами локальной сети.

Метод доступа — это способ "захвата" передающей среды, способ определения того, какая из рабочих станций сети может следующей использовать ресурсы сети.

## 4.2 Метод доступа к разделяемой среде CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) — множественный доступ с контролем несущей и обнаружением коллизий.

Все компьютеры в сети на разделяемой среде имеют возможность немедленно (с учетом задержки распространения сигнала в физической среде) получать данные, которые любой из компьютеров начал передавать в общую среду. То есть среда, к которой подключены все станции, работает в режиме коллективного доступа (Multiple Access).

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая еще называется несущей частотой. Признаком «незанятости» среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5 — 10 МГц в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

Если среда свободна, то узел имеет право начать передачу кадра.

Однако механизм прослушивания среды и пауза между кадрами не гарантируют исключения ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит коллизия, так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации.

Коллизия — это нормальная ситуация в работе сетей. Для возникновения коллизии не обязательно, чтобы несколько станций начали передачу абсолютно одновременно, такая ситуация маловероятна. Более вероятна ситуация, когда один узел начинает передачу, а через некоторое (короткое) время другой узел, проверив среду и не обнаружив несущую (сигналы первого узла еще не успели до него дойти), начинает передачу своего кадра. Таким образом, возникновение коллизии является следствием распределения узлов сети в пространстве.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется факт обнаружения коллизии (Collision Detection).

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему алгоритму:

*Пауза* = *L* × *(интервал отсрочки)*.

В технологии Ethernet интервал отсрочки выбран равным значению 512 битовых интервалов. Битовый интервал соответствует времени между появлением двух последовательных битов данных на кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс или 100 нс.

*L* представляет собой целое число, выбранное с равной вероятностью из диапазона [0, 2×*N*], где *N* — номер повторной попытки передачи данного кадра. После 10-й попытки интервал, из которого выбирается пауза, не увеличивается.

Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр. Описанный алгоритм носит название усеченного экспоненциального двоичного алгоритма отсрочки.

При увеличении нагрузки значительно возрастает время ожидания доступа к среде. Единственный способ обеспечения приемлемых характеристик функционирования сети с данным методом доступа — работа в недогруженном режиме.

Надежное распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных передан ею верно, этот кадр будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией из-за несовпадения контрольной суммы.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

*Tmin* > *RTT*.

Здесь *Tmin* — время передачи кадра минимальной длины, a *RTT* — время оборота, то есть время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. В худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а в обратном направлении — сигнал, уже искаженный коллизией).

При выполнении этого условия передающая станция должна успеть обнаружить коллизию, которую вызвал переданный ее кадр, еще до того, как она закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от минимальной длины кадра и скорости передачи данных протокола, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле.

Протокол множественного доступа не используется в коммутируемых полнодуплексных сетях, таких, как используемые сегодня коммутируемые сети Ethernet, но частично доступен в оборудовании для обеспечения совместимости.

## 4.3 MAC-адрес

На канальном уровне, который обеспечивает доступ к среде и передачу кадра, для идентификации сетевых интерфейсов узлов сети используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые МАС-адресами (рис. 4.2). Обычно МАС-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточиями, например:

00-0F-3D-F2-15-A1.

Каждый сетевой адаптер имеет, по крайней мере, один МАС-адрес.



Рис. 4.2 — Структура MAC-адреса.

Старшие 3 байта MAC-адреса представляют собой так называемый уникальный идентификатор организации (Organizationally Unique Identifier, OUI), который централизованно выделятся производителям сетевого оборудования IEEE (Институтом инженеров электроники и электротехники). На сайте IEEE имеется возможность поиска информации о производителе по значению OUI.

Младшие 3 байта представляют собой организационно уникальный адрес (Organizationally Unique Address, OUA), который назначается производителем каждому выпущенному им контроллеру сетевого интерфейса.

Помимо отдельных интерфейсов, МАС-адрес может определять группу интерфейсов или даже все интерфейсы сети. Первый (младший) бит старшего байта адреса назначения является признаком того, является адрес индивидуальным или групповым. Если он равен 0, то адрес является индивидуальным, то есть идентифицирует один сетевой интерфейс, а если 1, то групповым.

Второй бит старшего байта адреса определяет способ назначения адреса — централизованный или локальный. Если этот бит равен 0 (что бывает почти всегда в стандартной аппаратуре Ethernet), это говорит о том, что адрес назначен централизованно по правилам стандарта IEEE 802.

Mac-адрес сетевых адаптеров компьютера под управлением операционной системы Windows можно посмотреть выполнив команду:

ipconfig /all

## 4.4 Подуровень LLC (Logical Link Control — управление логической связью)

Уровень LLC отвечает за передачу кадров данных между узлами с различной степенью надежности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем.

Средства канального уровня могут использоваться для предоставления различного сервиса, который можно разделить на 3 класса:

* сервис без уведомления и без соединения;
* сервис с уведомлением и без соединения;
* сервис с уведомлением и с соединением.

Сервис без уведомления и без соединения не предполагает, что прием переданного кадра должен подтверждаться, и что до начала передачи должно устанавливаться соединение, которое после передачи должно разрываться. Если в результате помех на физическом уровне кадр будет потерян, то никаких попыток его восстановить на канальном уровне не будет. Этот класс сервиса используется там, где физический уровень обеспечивает высокую надежность при передаче так, что восстановление при потере кадров можно оставить на верхние уровни. Он также применяется при передаче данных в реальном времени там, где лучше потерять часть данных, чем увеличить задержку в их доставке.

Сервис с уведомлением и без соединения. В этом классе получение каждого посланного кадра должно быть подтверждено. Если подтверждения не пришло в течение определенного времени, то кадр должен быть послан опять. Этот класс сервиса используется над ненадежной физической средой передачи.

Сервис с уведомлением и с соединением предполагает, что до начала передачи между машинами устанавливается соединение, и данные передаются по этому соединению. Каждый передаваемый кадр нумеруется и канальный уровень гарантирует, что он будет обязательно получен и только один раз и все кадры будут получены в надлежащей последовательности.

Подуровень управления логической связью отвечает также за выполнение такой функции канального уровня как обработка ошибок передачи данных, которая подразумевает их обнаружение и, по возможности, исправление. Для этого применяется помехоустойчивое кодирование.

Методы обнаружения ошибок основаны на передаче в составе блока данных избыточной служебной информации, называемой контрольной суммой. Контрольная сумма вычисляется как функция от основной информации, причем не обязательно путем суммирования. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае её совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть корректно. Существует несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Самым простым методом является контроль чётности. Суть его заключается в том, чтобы в каждом передаваемом байте информации количество единиц всегда было чётным. Для этого один (обычно последний) бит выделяется под значение контрольной суммы, в которую записывается 0 если количество единиц и так было чётным или 1, если количество единиц было нечётным. Он позволяет обнаруживать только одиночные ошибки в проверяемых данных. Так как на каждый байт (8 бит) приходится 1 бит служебной информации, то коэффициент избыточности данного метода составляет 1/8. Метод редко используется в компьютерных сетях из-за значительной избыточности и невысоких диагностических возможностей.

Циклический избыточный контроль (Cyclic Redundancy Check, CRC) является в настоящее время наиболее популярным методом контроля в вычислительных сетях (и не только в сетях, например, этот метод широко применяется при записи данных на гибкие и жесткие диски). Метод основан на представлении исходных данных в виде одного многоразрядного двоичного числа. Контрольной информацией считается остаток от деления этого числа на известный делитель. Обычно в качестве делителя выбирается семнадцати- или тридцатитрёхразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байт) или 32 разряда (4 байт). При получении кадра данных снова вычисляется остаток от деления на тот же делитель, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

Метод CRC позволяет обнаруживать все одиночные ошибки, двойные ошибки и ошибки в нечетном числе битов. Метод обладает также невысокой степенью избыточности. Например, для кадра размером 1024 байт контрольная информация длиной 4 байт составляет только 0,4 %.

Техника кодирования, которая позволяет приемнику не только понять, что присланные данные содержат ошибки, но и исправить их, называется прямой коррекцией ошибок. Коды, которые обеспечивают прямую коррекцию ошибок, требуют введения большей избыточности в передаваемые данные, чем коды, только обнаруживающие ошибки. Для того чтобы оценить количество дополнительных битов, требуемых для исправления ошибок, нужно знать так называемое расстояние Хемминга между разрешенными комбинациями кода. Расстоянием Хемминга называется минимальное число битовых разрядов, в которых отличается любая пара разрешенных кодов. Можно доказать, что если получить избыточный код с расстоянием Хемминга, равным *n*, то такой код будет в состоянии распознавать (*n*–1)-кратные ошибки и исправлять (*n*–1)/2-кратные ошибки.

Коды Хемминга эффективно обнаруживают и исправляют изолированные ошибки, то есть отдельные искаженные биты, которые разделены большим количеством корректных битов. Однако при появлении длинной последовательности искаженных битов (пульсации ошибок) коды Хемминга не работают.

## 4.5 Оборудование канального уровня

Мост, сетевой мост, бридж (англ. bridge) — сетевое устройство 2 уровня модели OSI, предназначенное для объединения сегментов компьютерной сети разных топологий и архитектур.

Сетевой коммутатор (жарг. свитч от англ. switching hub, switch — переключатель) — устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети. Коммутаторы были разработаны с использованием мостовых технологий и часто рассматриваются как многопортовые мосты.

В отличие от концентратора, который распространяет трафик от одного подключенного устройства ко всем остальным, коммутатор передаёт данные только непосредственно получателю (исключение составляет широковещательный трафик всем узлам сети и трафик для устройств, для которых не известен исходящий порт коммутатора). Это повышает производительность и безопасность сети, избавляя остальные сегменты сети от необходимости (и возможности) обрабатывать данные, которые им не предназначались.

Коммутатор хранит в памяти таблицу коммутации, в которой указывается соответствие MAC-адреса узла порту коммутатора. При включении коммутатора эта таблица пуста, и он работает в режиме обучения. В этом режиме поступающие на какой-либо порт данные передаются на все остальные порты коммутатора. При этом коммутатор анализирует кадры (фреймы) и, определив MAC-адрес хоста-отправителя, заносит его в таблицу на некоторое время. Впоследствии, если на один из портов коммутатора поступит кадр, предназначенный для хоста, MAC-адрес которого уже есть в таблице, то этот кадр будет передан только через порт, указанный в таблице. Если MAC-адрес хоста-получателя не ассоциирован с каким-либо портом коммутатора, то кадр будет отправлен на все порты, за исключением того порта, с которого он был получен. Со временем коммутатор строит таблицу для всех активных MAC-адресов, то есть узнаёт обо всех устройствах, подключённых к сети, и в результате трафик локализуется.

## 4.6 Сетевые технологии канального уровня

В процессе развития компьютерных сетей существовало большое количество протоколов канального уровня, которые реализовывали совершенно разные технологии. У каждой из них были свои достоинства и недостатки, которые определяли её популярность и степень распространённости. Однако, наличие большого числа различных, а значит несовместимых, технологий усложняло процесс объединения сетей, который неизбежно привёл к тому, что сети Ethernet к настоящему времени вытеснили все остальные технологии и стали по сути стандартом. При этом некоторые виды сетей оказали значительное влияние на развитие в своей области, а реализованные в них подходы оригинальны и заслуживают отдельного рассмотрения.

Сеть Token-Ring была предложена фирмой IBM и рассчитана на скорость обмена 4.16 Мбит/c при числе сегментов до 250. Token-Ring имеет топологию «кольцо», хотя внешне она больше напоминает «звезду». Это связано с тем, что отдельные абоненты (компьютеры) присоединяются к сети не напрямую, а через специальные концентраторы или многостанционные устройства доступа (MSAU или MAU — Multistation Access Unit). Поэтому физически сеть образует звездно-кольцевую топологию (рис. 5.3). В действительности же абоненты объединяются все-таки в кольцо, то есть каждый из них передает информацию одному соседнему абоненту, а принимает информацию от другого соседнего абонента.



Рис. 4.3 — Схема подключения абонентов к сети Token Ring.

Более распространённой технологией, основанной на топологии кольцо является FDDI (Fiber Distributed Data Interface — волоконно-оптический интерфейс передачи данных). Это стандарт локальной сети с двойным кольцом и передачей маркера, которая использует в качестве среды передачи волоконно-оптический кабель. В FDDI предполагается применение двойных кольцевых сетей. Трафик по этим кольцам движется в противоположных направлениях. В физическом выражении кольцо состоит из двух или более двухточечных соединений между смежными станциями. Одно из двух колец FDDI называется первичным кольцом, другое — вторичным кольцом. Первичное кольцо используется для передачи данных, в то время как вторичное кольцо обычно является дублирующим.

При обрывах оптоволокна возможно частичное (при двух обрывах) или полное (при одном обрыве) восстановление связности сети (рис. 4.4). Поскольку такое дублирование повышает надёжность системы, данный стандарт с успехом применяется в магистральных каналах связи.



Рис. 4.4 — Варианты связей в случае обрывов волокон в сети FDDI.

Технологии Token Ring и FDDI используют метод маркерного доступа, основанный на передаче от узла к узлу особого кадра — маркера (токена) доступа.

Сети с передачей маркера перемещают от узла к узлу небольшой блок данных, называемый маркером. Владение этим маркером даёт право передачи. Если узел, принимающий маркер, не имеет информации для отправки, он просто переправляет маркер к следующей конечной станции. Каждая станция может удерживать маркер в течение определенного максимального времени (по умолчанию — 10 мс). Однако сети Token Ring и FDDI используют сложную систему приоритетов, позволяющую некоторым станциям пользоваться сетью чаще остальных, например, для передачи чувствительного к задержкам траффика.

В сетях с маркерным методом доступа к общей среде возможна реализация различных механизмов для обнаружения и предотвращения влияния сбоев и ошибок. Например, пусть одна из станций выбрана в качестве активного монитора. Эта станция работает как центральный источник синхронизации для других станций сети и выполняет ряд других функций. Одной из таких функция является удаление из кольца бесконечно циркулирующих кадров (маркеров), которые могут появиться если отправитель ошибся, установив, например, ошибочный адрес места назначения, что приводит к зацикливанию кадра.

Более детерминированный характер доступа технологий Token Ring и FDDI предопределил более эффективное использование разделяемой среды, чем у технологии Ethernet, но одновременно и усложнил оборудование, сделав невозможным широкое распространение данных технологий.

## 4.7 Семейство стандартов IEEE 802.3 (Ethernet)

Ethernet — это наиболее распространённая на сегодняшний день технология организации локальных сетей. Она описывает реализацию двух первых уровней модели OSI — проводные соединения и электрические сигналы (физический уровень), а также форматы блоков данных и протоколы управления доступом к сети (канальный уровень).

Название Ethernet произошло от двух английских слов — ether (эфир) и net (сеть). Ethernet использует концепцию общего эфира. Каждый компьютер посылает данные в этот эфир и указывает, кому они адресованы. Данные могут дойти до всех компьютеров сети, но обрабатывает их только тот, которому они предназначены. Остальные узлы чужие данные игнорируют. Такая работа аналогична эфиру радиостанций и была заимствована из существовавших на момент разработки Ethernet технологий радиосетей с пакетной коммутацией.

В ранних версиях Ethernet коаксиальный кабель являлся носителем общего электромагнитного эфира. Компьютеры подключались к общему кабелю с помощью специальных коннекторов по топологии общая шина. Каждый компьютер отправлял в шину электрические сигналы, а все остальные узлы их получали. При получении, компьютер должен был определить, кому реально этот сигнал адресован, и, соответственно, свои сигналы обработать, а чужие — проигнорировать.

Первая версия Ethernet была рассчитана на скорость 3Мбит/с. Она использовала толстый коаксиальный кабель, метод управления доступом — CSMA/CD и полудуплексный режим работы, то есть узел не мог одновременно передавать и принимать информацию. Размер пакета от 72 до 1526 байт Количество узлов в одном разделяемом сегменте сети ограничено предельным значением в 1024 рабочих станции. Однако сеть, построенная на одном разделяемом сегменте, становится неэффективной задолго до достижения предельного значения количества узлов, в основном по причине полудуплексного режима работы.

Несмотря на то, что Ethernet на коаксиальном кабеле уже давно не используется, механизм адресации данных и концепция общего эфира сохранились без изменений.

К сегодняшнему дню разработано большое количество стандартов Ethernet сетей, которые отличаются скоростью передачи, типами кабелей и оборудованием. Их можно разделить на несколько групп:

* 10 Мбит Ethernet;
* Fast Ethernet;
* Gigabit Ethernet;
* Прочие (перспективные) реализации Ethernet.

Все они описаны в стандартах семейства IEEE 802.3, разработанных институтом инженеров по электротехнике и электронике (англ. Institute of Electrical and Electronics Engineers).

Наиболее известным представителем стандартов 10 Мбит Ethernet является стандарт IEEE 802.3a имеющий название 10BASE2. Оно происходит от некоторых физических свойств применяемой передающей среды. Число 10 означает максимальную скорость передачи данных в 10 Мбит/с. Слово BASE является сокращением от baseband (принцип передачи данных без модуляции), а двойка является первой цифрой числа 200 — округлённой максимальной длины сегмента сети, в качестве которого используется тонкий коаксиальный кабель.

Общей характеристикой стандартов Fast Ethernet является скорость передачи данных до 100 Мбит/с, сохранение метода случайного доступа CSMA/CD и формата кадра, принятого в ранних версиях Ethernet. Однако в отличие от предыдущих стандартов, в Fast Ethernet возможно использование в качестве среды передачи данных только витой пары и волоконно-оптического кабеля.

Наиболее известным стандартом Fast Ethernet является 100Base-TX. Буква T в названии означает, что он рассчитан на использование витой пары (англ. Twisted Pair), а именно двухпарного неэкранированного кабеля не ниже 5 категории. По одной из пар ведётся передаче в одном направлении, по другой — в обратном, то есть сеть работает в режиме полного дуплекса. Буква X обозначает применяемую в стандарте топологию звезда.

Вариант Fast Ethernet для построения с использованием двух многомодовых оптических волокон называется соответственно 100BASE-FX, где буква F означает Fiber — волокно, а остальные характеристики аналогичны 100BASE-TX.

Независимо от стандарта, кадры Ethernet всегда имеют одинаковый формат. (рис. 4.5).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 байт | 1 байт | 6 байт | 6 байт | 2 байта | 46-1500 байт | | 4 байт |
| преамбула | ограничитель начала кадра | адрес назначения | адрес источника | длина/тип | данные | наполнитель | контрольная сумма |

Рис. 4.5 — Формат кадра Ethernet.

Ethernet-кадр состоит из полей перечисленный ниже.

Преамбула: 7 байт, каждый из которых представляет чередование единиц и нулей 10101010. Преамбула позволяет установить битовую синхронизацию на приемной стороне.

Ограничитель начала кадра (SFD, start frame delimiter): 1 байт, последовательность 10101011. Указывает, что далее последуют, информационные поля кадра. Этот байт можно относить к преамбуле.

Адрес назначения (DA, destination address): 6 байт, указывает МАС-адрес станции (МАС-адреса станций), для которой (которых) предназначен этот кадр. Это может быть единственный физический адрес (unicast), групповой адрес (multicast) или широковещательный адрес (broadcast).

Адрес отправителя (SA, source address): 6 байт, указывает МАС-адрес станции, которая посылает кадр.

Поле типа или длины кадра (Т or L, type or length): 2 байта, в разных версиях протокола используется либо для указания протокола более высокого уровня, чью данные помещены в кадр, либо для указания длины самого кадра.

Дополнительное поле (pad — наполнитель) — заполняется только в том случае, когда поле данных невелико, с целью удлинения длины кадра до минимального размера 64 байта — преамбула не учитывается. Ограничение снизу на минимальную длину кадра необходимо для правильного разрешения коллизий.

Контрольная последовательность кадра (FCS, frame check sequence): 4-байтовое поле, в котором указывается контрольная сумма, вычисленная с использованием циклического избыточного кода по полям кадра, за исключением преамбул.

По аналогии с Fast Ethernet, стандарты Gigabit Ethernet описывают несколько сетевых технологий с использованием как медных кабелей (1000BASE-T, 1000BASE-TX), так и оптического волокна (1000BASE-SX, 1000BASE-LX и другие). Интересно отметить, что стандарты Gigabit Ethernet позволяют, используя те же самые провода, что и Fast Ethernet (витую пару 5-ой категории) получать скорость до 1 Гбит/с.

Для уменьшения частоты сигнала до величин, позволяющих передавать данные с такой скоростью по витым парам категории 5, данные в линии представляются в так называемом коде PAM-5 (рис. 4.6). В нем передаваемый сигнал имеет набор из пяти фиксированных уровней {–2, –1, 0, +1, +2}. Четыре из них используются для кодирования информационных битов, а пятый предназначен для коррекции ошибок. На наборе из четырех фиксированных уровней одним дискретным состоянием сигнала можно закодировать сразу два информационных бита, поскольку комбинация из двух бит имеет четыре возможные комбинации (так называемые дибиты) — 00, 01, 10 и 11.

Переход к дибитам позволяет в два раза повысить битовую скорость. Чтобы различать битовую, или информационную, скорость и скорость различных дискретных состояний сигнала, вводят понятие бодовой скорости. Бод — это количество различных дискретных состояний сигнала в единицу времени. Поэтому, если в одном дискретном состоянии кодируется два бита, битовая скорость в два раза больше бодовой, то есть 1 Бод = 2 бит/с.

Если учесть, что кабель 5-й категории рассчитан на частоту 125 МГц, то есть способен работать с бодовой скоростью 125 МБод, то информационная скорость по одной витой паре составит 250 Мбит/с. Вспомним, что в кабеле имеется четыре витые пары, поэтому если задействовать все четыре пары, то можно повысить скорость передачи до 250 Мбит/с × 4 = 1000 Мбит/с, то есть достичь желаемой скорости.



Рис. 4.6 — Метод кодирования PAM-5.

Существуют стандарты Ethernet, позволяющие передавать данные со скоростью 5, 10, 40 и даже 100 Гбит/с, однако они, как правило либо имеют очень жесткие ограничения на максимальную длину сегмента сети, либо очень требовательны к линям связи и оборудованию, в связи с чем их распространение минимально.

# 5 СЕТЕВОЙ УРОВЕНЬ

Средствами рассмотренного ранее канального уровня можно объединить в сеть несколько компьютеров, однако его особенности, такие как применение плоской адресации, активное использование широковещательных пакетов и некоторые другие накладывают жёсткие ограничения как на количество объединённых компьютеров, так и на расстояние между ними. При построении глобальных сетей, которые являются объединением нескольких более простых, возникают новые задачи, такие как:

* адресация узлов сложной составной сети;
* продвижение пакетов между сетями через транзитные узлы;
* маршрутизация.

Решение этих задач выполняется средствами сетевого уровня.

## 5.1 Адресация узлов сети

Проблемой, которую нужно учитывать при объединении трех и более компьютеров, является их адресация, точнее адресация их сетевых интерфейсов, так как один компьютер может иметь несколько сетевых интерфейсов.

По количеству адресуемых интерфейсов адреса можно классифицировать следующим образом:

* уникальный адрес (unicast) используется для идентификации отдельных интерфейсов;
* групповой адрес (multicast) идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;
* данные, направленные по широковещательному адресу (broadcast), должны быть доставлены всем узлам сети;
* адрес произвольной рассылки (anycast), определенный в новой версии протокола IPv6, так же, как и групповой адрес, задает группу адресов, однако данные, посланные по этому адресу, должны быть доставлены не всем адресам данной группы, а любому из них.

Адреса могут быть числовыми (например, 129.26.255.255 или 81.1A.FF.FF) и символьными (site.domen.ru).

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется адресным пространством. Адресное пространство может иметь плоскую (линейную) организацию или иерархическую организацию.

При плоской организации множество адресов никак не структурировано. Примером плоского числового адреса является МАС-адрес, предназначенный для однозначной идентификации сетевых интерфейсов в локальных сетях. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного числа, например, 008b05е24а8f. При задании МАС-адресов не требуется выполнение ручной работы, так как они обычно встраиваются в аппаратуру компанией-изготовителем, поэтому их называют также аппаратными адресами (hardware address). Использование плоских адресов является жестким решением — при замене аппаратуры, например, сетевого адаптера, изменяется и адрес сетевого интерфейса компьютера.

При иерархической организации адресное пространство структурируется в виде вложенных друг в друга подгрупп, которые, последовательно сужая адресуемую область, в конце концов, определяют отдельный сетевой интерфейс. Примером иерархической адресации являются IP-адреса.

На практике обычно применяют сразу несколько схем адресации, так что сетевой интерфейс компьютера может одновременно иметь несколько адресов-имен. Каждый адрес задействуется в той ситуации, когда соответствующий вид адресации наиболее удобен. А для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, которые называют протоколами разрешения адресов.

## 5.2 Формат IP-адреса

В заголовке IP-пакета для хранения IP-адресов отправителя и получателя отводятся два поля, каждое из которых имеет фиксированную длину 4 байта (32 бита). IP-адрес состоит из двух логических частей — номера сети и номера узла в сети.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30

Этот же адрес может быть представлен в двоичном формате:

10000000.00001010.00000010.00011110

А также в шестнадцатеричном формате:

80.0A.02.1D

Заметьте, что запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла. Вместе с тем при передаче пакета по сети часто возникает необходимость разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор, получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети.

Существует несколько способов, которыми определяют, какая часть из 32 бит, отведенных под IP-адрес, относится к номеру сети, а какая — к номеру узла.

## 5.3 Классовая и бесклассовая адресация. Маска подсети

Простейший отделения адреса сети от адреса узла состоит в использовании фиксированной границы. При этом все 32-битное поле адреса заранее делится на две части не обязательно равной, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, в другой — номер узла. Поскольку поле, которое отводится для хранения номера узла, имеет фиксированную длину, все сети будут иметь одинаковое максимальное число узлов. Если, например, под номер сети отвести один первый байт, то все адресное пространство распадется на сравнительно небольшое (28) число сетей огромного размера (224 узлов). Если границу передвинуть дальше вправо, то сетей станет больше, но все равно все они будут одинакового размера. Очевидно, что такой жесткий подход не позволяет дифференцированно удовлетворять потребности отдельных предприятий и организаций. Именно поэтому он не нашел применения, хотя и использовался на начальном этапе существования технологии TCP/IP.

Второй способ, основан на классах адресов. Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса (табл. 5.1).

Таблица 5.1 Классы сетей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Первые биты** | **Наименьший номер сети** | **Наибольший номер сети** | **Максимальное число узлов** |
| A | 0 | 1.0.0.0 | 126.0.0.0 | 224 |
| B | 10 | 128.0.0.0 | 191.255.0.0 | 216 |
| C | 110 | 192.0.0.0 | 223.255.255.0 | 28 |
| D | 1110 | 224.0.0.0 | 239.255.255.255 | Групповые адреса |
| E | 11110 | 240.0.0.0 | 247.255.255.255 | Зарезервировано |

К классу А относится адрес, в котором старший бит имеет значение 0. В адресах класса А под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Сети, все IP-адреса которых имеют значение первого байта в диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса А. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей. Сетей класса А сравнительно немного, зато количество узлов в них может достигать 224, то есть 16 777 216 узлов.

К классу В относятся все адреса, старшие два бита которых имеют значение 10. В адресах класса В под номер сети и под номер узла отводится по 2 байта. Сети, значения первых двух байтов адресов которых находятся в диапазоне от 128.0 (10000000 00000000) до 191.255 (10111111 11111111), называются сетями класса В. Ясно, что сетей класса В больше, чем сетей класса А, а размеры их меньше. Максимальное количество узлов в сетях класса В составляет 216 (65 536).

К классу С относятся все адреса, старшие три бита которых имеют значение 110. В адресах класса С под номер сети отводится 3 байта, а под номер узла — 1 байт. Сети, старшие три байта которых находятся в диапазоне от 192.0.0 (11000000 00000000 00000000) до 223.255.255 (11011111 11111111 11111111), называются сетями класса С. Сети класса С наиболее распространены, и наименьшее максимальное число узлов в них равно 28 (256).

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый групповой адрес (multicast address). В то время как адреса классов А, В и С служат для идентификации отдельных сетевых интерфейсов, то есть являются индивидуальными адресами (unicast address), групповой адрес идентифицирует группу сетевых интерфейсов, которые в общем случае могут принадлежать разным сетям. Интерфейс, входящий в группу, получает наряду с обычным индивидуальным IP-адресом еще один групповой адрес. Если при отправке пакета в качестве адреса назначения указан адрес класса D, то такой пакет должен быть доставлен всем узлам, которые входят в группу.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу Е. Адреса этого класса зарезервированы для будущих применений.

Чтобы получить из IP-адреса номер сети и номер узла, требуется не только разделить адрес на две соответствующие части, но и дополнить каждую из них нулями до полных 4 байт. Возьмем, например, адрес класса В 129.64.134.5. Первые два байта идентифицируют сеть, а последующие два — узел. Таким образом, номером сети является адрес 129.64.0.0, а номером узла — адрес 0.0.134.5.

Ещё один подход к определению адреса сети и адреса узла основан на использовании маски, которая позволяет максимально гибко устанавливать границу между этими двум частями IP-адреса. При таком подходе адресное пространство можно использовать для создания множества сетей разного размера.

|  |
| --- |
| Маска — это число, применяемое в паре с IP-адресом, двоичная запись которого содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностями единиц и нулей в маске соответствует границе между номером сети и номером узла в IP-адресе. |

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации.

Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде IP-адрес 129.64.134.5 — это:

10000001.01000000.10000110.00000101,

а маска 255.255.128.0 в двоичном виде выглядит так:

11111111.11111111.1000000.00000000.

Если игнорировать маску и интерпретировать адрес 129.64.134.5 на основе классов, то номером сети является 129.64.0.0, а номером узла — 0.0.134.5 (поскольку адрес относится к классу В).

Если же использовать маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес 129.64.134.5, делят его на две части, номер сети:

10000001.01000000.1

и номер узла:

0000110.00000101.

В десятичной форме записи номера сети и узла, дополненные нулями до 32 бит, выглядят соответственно как 129.64.128.0 и 0.0.6.5.

Наложение маски можно интерпретировать как выполнение логической операции И (AND). Так, в предыдущем примере номер сети из адреса 129.64.134.5 является результатом выполнения логической операции AND с маской 255.255.128.0.

Для стандартных классов сетей маски имеют следующие значения:

класс А — 11111111.00000000.00000000.00000000 (255.0.0.0);

класс В — 11111111.11111111.00000000.00000000 (255.255.0.0);

класс С — 11111111.11111111.11111111.00000000 (255.255.255.0).

Для записи масок используются и другие форматы. Например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде: FF.FF.00.00 — маска для адресов класса В. Еще чаще встречается обозначение 185.23.44.206/16 — данная запись говорит о том, что маска для этого адреса содержит 16 единиц или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

## 5.4 Особые IP-адреса

Независимо от способа адресации, реальное количество адресов в подсети не равно количеству возможных узлов. В TCP/IP существуют ограничения при назначении IP-адресов, а именно номера сетей и номера узлов не могут состоять из одних двоичных нулей или единиц. Отсюда следует, что максимальное количество узлов, приведенное в табл. 5.1 для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса С под номер узла отводится 8 бит, которые позволяют задать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса С не может превышать 254, так как адреса 0 и 255 запрещены для адресации сетевых интерфейсов. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса А состоит из одних двоичных единиц.

Итак, некоторые IP-адреса интерпретируются особым образом (табл. 5.2):

* Если IP-адрес состоит только из двоичных нулей, то он называется неопределенным адресом и обозначает адрес того узла, который сгенерировал этот пакет. Адрес такого вида в особых случаях помещается в заголовок IP-пакета в поле адреса отправителя.
* Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет. Такой адрес также может быть использован только в качестве адреса отправителя.
* Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется ограниченным широковещательным (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной сети не при каких условиях.
* Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем узлам сети 192.190.21.0. Такой тип адреса называется широковещательным (broadcast).

Таблица 5.2 Особые IP-адреса

|  |  |  |
| --- | --- | --- |
| **Номер сети** | **Номер узла** | **Что означает** |
| Все биты установлены в "0" | | Данное устройство |
| Номер сети | Все биты равны 0 | Данная IP-сеть |
| Все биты равны 0 | Номер хоста | Устройство в данной IP-сети |
| Все биты установлены в "1" | | Все устройства в данной IP-сети |
| Номер сети | Все биты равны 1 | Все устройства в указанной сети |
| 127 (десятичное) | Что-нибудь (обычно 1) | Адрес обратной связи |

В протоколе IP нет понятия широковещания в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам сети. Как ограниченный, так и обычный варианты широковещательной рассылки имеют пределы распространения в составной сети: они ограничены либо сетью, которой принадлежит источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из подсетей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является внутренним адресом стека протоколов компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети, однако, если для этого будет использовать адрес реального сетевого интерфейса компьютера, это приведёт к избыточным передачам пакетов в сеть, к тому же компьютер вообще может не иметь сетевого интерфейса. Решением этой проблемы является применение внутреннего адреса 127.0.0.0.

В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со значения 127. Когда программа посылает данные по IP-адресу 127.х.х.х, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется адресом обратной петли (loopback).

## 5.5 Порядок назначения IP-адресов

По определению схема IP-адресации должна обеспечивать уникальность нумерации сетей, а также уникальность нумерации узлов в пределах каждой сети, следовательно, процедуры назначения номеров как сетям, так и узлам сетей должны быть централизованными.

Когда дело касается сети, являющейся частью Интернета, уникальность нумерации может быть обеспечена только усилиями специально созданных для этого центральных органов. В небольшой же автономной IP-сети условие уникальности номеров сетей и узлов может быть выполнено силами сетевого администратора.

В этом случае в распоряжении администратора имеется все адресное пространство, так как совпадение IP-адресов в не связанных между собой сетях не вызовет никаких отрицательных последствий. Администратор может выбирать адреса произвольным образом, соблюдая лишь синтаксические правила и учитывая ограничения на особые адреса.

Однако при таком подходе исключена возможность в будущем подсоединить данную сеть к Интернету. Действительно, произвольно выбранные адреса данной сети могут совпасть с централизовано назначенными адресами Интернета. Для того чтобы избежать коллизий, связанных с такого рода совпадениями, в стандартах Интернета определено несколько диапазонов так называемых частных адресов, рекомендуемых для автономного использования:

* в классе А — сеть 10.0.0.0;
* в классе В — диапазон из 16 номеров сетей (172.16.0.0-172.31.0.0);
* в классе С — диапазон из 255 сетей (192.168.0.0-192.168.255.0).

Заданный диапазон IP-адресов может быть разбит на подсети.

Предположим, что организация получила сеть класса C 193.1.1.0, и ей необходимо сформировать шесть подсетей. Наибольшая подсеть должна поддерживать 25 хостов. На первом шаге определяется число битов, требуемых для определения необходимых шести подсетей. Поскольку найти число, при котором два в какой-либо степени равнялось шести, невозможно, то в данном примере администратор должен определить восемь подсетей (23 = 8), т. е. для выделения подсетей будут использованы три бита из выделенного адреса.

В данном примере организации выделен адрес класса С, с записью расширенного сетевого префикса как /24, а значит, полученный после выделения подсетей, расширенный сетевой префикс будет записан как /27 (24 + 3 = 27). Этот расширенный сетевой префикс имеет эквивалентное значение маски подсети 255.255.255.224.

Для определения какой-либо подсети, сетевой администратор помещает двоичное представление номера этой подсети (а в нашем случае для восьми подсетей это может быть число от 0 до 7) в битовое поле номера подсети. Например, для определения подсети #4 администратор просто помещает двоичное представление числа 4 (1002) в трехбитовое поле номера подсети. В Таблице 5.3 приведены все восемь возможных вариантов подсетей для рассматриваемого примера.

Таблица 5.3 Разделение сети на подсети

|  |  |  |
| --- | --- | --- |
| **Точечно-десятичная нотация** | | **Двоичное представление адреса** |
| Базовая сеть | 193.1.1.0 /24 | 11000001.00000001.00000001.00000000 |
| Подсеть #0 | 193.1.1.0 /27 | 11000001.00000001.00000001.00000000 |
| Подсеть #1 | 193.1.1.32 /27 | 11000001.00000001.00000001.00100000 |
| Подсеть #2 | 193.1.1.64 /27 | 11000001.00000001.00000001.01000000 |
| Подсеть #3 | 193.1.1.96 /27 | 11000001.00000001.00000001.01100000 |
| Подсеть #4 | 193.1.1.128 /27 | 11000001.00000001.00000001.10000000 |
| Подсеть #5 | 193.1.1.160 /27 | 11000001.00000001.00000001.10100000 |
| Подсеть #6 | 193.1.1.192 /27 | 11000001.00000001.00000001.11000000 |
| Подсеть #7 | 193.1.1.224 /27 | 11000001.00000001.00000001.11100000 |

## 5.4 Протокол ARP

Как уже было сказано, никакой функциональной зависимости между физическим адресом устройства и его IP-адресом не существует, следовательно, единственный способ установления соответствия — ведение таблиц. В результате конфигурирования сети каждый интерфейс «знает» свои IP-адрес и физический адрес, что можно рассматривать как таблицу, состоящую из одной строки. Проблема состоит в том, как организовать обмен имеющейся информацией между узлами сети.

Для определения локального адреса по IP-адресу используется протокол разрешения адресов (Address Resolution Protocol, ARP). Протокол разрешения адресов реализуется различным образом в зависимости от того, работает ли в данной сети протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещания или же какой-либо из протоколов глобальной сети (Frame Relay, ATM), которые, как правило, не поддерживают широковещательный доступ.

Рассмотрим работу протокола ARP в локальных сетях с широковещанием. Каждый сетевой интерфейс имеет IP-адрес и МАС-адрес. Пусть в какой-то момент IP-модуль узла A направляет пакет узлу B. Протокол IP узла A определил IP-адрес интерфейса следующего маршрутизатора — это IP1. Теперь, прежде чем упаковать пакет в кадр Ethernet и направить его маршрутизатору, необходимо определить соответствующий МАС-адрес. Для решения этой задачи протокол IP обращается к протоколу ARP. Протокол ARP поддерживает на каждом интерфейсе сетевого адаптера или маршрутизатора отдельную ARP-таблицу, в которой в ходе функционирования сети накапливается информация о соответствии между IP-адресами и МАС-адресами других интерфейсов данной сети. Первоначально, при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

На первом шаге происходит передача от протокола IP протоколу ARP примерно такого сообщения: «Какой МАС-адрес имеет интерфейс с адресом IP1?». Работа протокола ARP начинается с просмотра собственной ARP-таблицы. Предположим, что среди содержащихся в ней записей отсутствует запрашиваемый IP-адрес. В этом случае исходящий IP-пакет, для которого оказалось невозможным определить локальный адрес из ARP-таблицы, запоминается в буфере, а протокол ARP формирует ARP-запрос, вкладывает его в кадр протокола Ethernet и широковещательно рассылает (использую MAC-адрес FF:FF:FF:FF:FF:FF). Все интерфейсы сети получают ARP-запрос и направляют его «своему» протоколу ARP. ARP сравнивает указанный в запросе адрес IP1 с IP-адресом интерфейса, на который поступил этот запрос.

Протокол ARP узла, который констатировал совпадение, формирует ARP-ответ, в котором указывает локальный адрес (MAC1) своего интерфейса и отправляет его запрашивающему узлу (в данном примере узлу A), используя его локальный адрес. Широковещательный ответ в этом случае не требуется, так как формат ARP-запроса предусматривает поля локального и сетевого адресов отправителя.

ARP-таблица пополняется не только за счет поступающих на данный интерфейс ARP-ответов, но и в результате извлечения полезной информации из широковещательных ARP-запросов.

Просмотр ARP-таблицы в операционной системе Windows выполняется из командной строки командой arp –a (см. рис. 5.2).

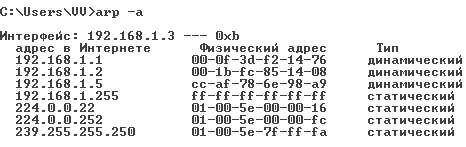


Рис. 5.2 — Пример ARP-таблицы.

## 5.5 Протокол IPv6

Версия IPv4, используемая в настоящий момент в сети Интернет, позволяет использовать около 4,23 миллиарда адресов. Их распределением занимается IANA (от англ. Internet Assigned Numbers Authority — «Администрация адресного пространства Интернет»), которая делегирует свои полномочия по распределению IP-адресов региональным регистраторам в виде диапазонов класса A. Региональные регистраторы, в свою очередь, делегируют более мелкие диапазоны интернет-провайдерам. В феврале 2011 года IANA выделила региональным интернет-регистраторам последние пять оставшихся блоков из своего адресного пространства, при этом у регионального регистратора азиатско-тихоокеанского региона (APNIC — Asia-Pacific Network Information Center), выделенные ему IP-адреса закончились.

Для решения этой и других проблем был разработан протокол IPv6. В протоколе IPv6 под IP-адрес отведено 16 байт. Это дает возможность пронумеровать огромное количество узлов (около 6 000 на каждого жителя Земли). Однако главной целью изменения системы адресации было не механическое увеличение адресного пространства, а повышения эффективности работы стека TCP/IP в целом.

Вместо прежних двух уровней иерархии адреса (номер сети и номер узла) в IPv6 имеется 4 уровня, из которых три уровня используются для идентификации сетей, а один — для идентификации узлов сети. Для записи IPv6-адресов используется шестнадцатеричную форма:

fedc:0a98:0:0:0:0:7654:3210.

Большое количество нулевых групп может быть пропущено с помощью двойного двоеточия (fe80::200:f8ff:fe21:67cf). Такой пропуск должен быть единственным в адресе.

В новой версии IPv6 предусмотрено три основных типа адресов: индивидуальные адреса, групповые адреса и адреса произвольной рассылки. Назначение этих типов адресов уже описывалось ранее. Тип адреса определяется значением нескольких старших битов адреса, которые названы префиксом формата. Индивидуальные адреса делятся на несколько подтипов. Основным подтипом индивидуального адреса является глобальный агрегируемый уникальный адрес, структура которого приведена на рис. 5.3, Рассмотрим назначение его полей.

* Префикс формата (Format Prefix, FP) для этого типа адресов имеет размер 3 бита и значение 001.
* Поле TLA (Top-Level Aggregation, TLA) предназначено для идентификации сетей самых крупных поставщиков услуг. Конкретное значение этого поля представляет собой общую часть адресов, которыми располагает данный поставщик услуг. Сравнительно небольшое количество разрядов, отведенных под это поле (13), выбрано специально для ограничения размера таблиц маршрутизации в магистральных маршрутизаторах самого верхнего уровня Интернета. Это поле позволяет перенумеровать 8196 сетей поставщиков услуг верхнего уровня, а значит, число записей, описывающих маршруты между этими сетями, также будет ограничено значением 8196, что ускорит работу магистральных маршрутизаторов. Следующие 8 разрядов зарезервированы на будущее для расширения при необходимости поля TLA.
* Поле NLA (Next-Level Aggregation, NLA) предназначено для нумерации сетей средних и мелких поставщиков услуг. Значительный размер поля NLA позволяет путем агрегирования адресов отразить многоуровневую иерархию поставщиков услуг.
* Поле SLA (Site-Level Aggregation, SLA) предназначено для адресации подсетей отдельного абонента, например подсетей одной корпоративной сети.
* Идентификатор интерфейса является аналогом номера узла в IPv4. Отличием версии IPv6 является то, что в общем случае идентификатор интерфейса просто совпадает с его локальным (аппаратным) адресом, а не представляет собой произвольно назначенный администратором номер узла. Идентификатор интерфейса имеет длину 64 бита, что позволяет поместить туда МАС-адрес (48 бит), адрес конечного узла ATM (48 бит) или номер виртуального соединения ATM (до 28 бит), а также, вероятно, даст возможность использовать локальные адреса технологий, которые могут появиться в будущем. Такой подход делает ненужным протокол ARP, поскольку процедура отображения IP-адреса на локальный адрес становится тривиальной — она сводится к простому отбрасыванию старшей части адреса. Кроме того, в большинстве случаев отпадает необходимость ручного конфигурирования конечных узлов, так как младшую часть адреса — идентификатор интерфейса — узел узнает от аппаратуры (сетевого адаптера и т. п.), а старшую — номер подсети — ему сообщает маршрутизатор.



Рис. 5.3 — Структура глобального агрегируемого уникального адреса

Одной из основных целей изменения формата заголовка протокола IPv6 было снижение накладных расходов, то есть уменьшение объема служебной информации, передаваемой с каждым пакетом. Для этого в новом протоколе IP были введены понятия основного и дополнительных заголовков. Основной заголовок присутствует всегда, а необязательные дополнительные заголовки могут содержать, например, информацию о фрагментации исходного пакета, полный маршрут следования пакета при маршрутизации от источника, информацию, необходимую для защиты передаваемых данных.

# 6 МАРШРУТИЗАЦИЯ

|  |
| --- |
| Маршрутизация — процесс определения маршрута следования информации в сетях связи. Протоколы маршрутизации обеспечивают поиск и фиксацию маршрутов продвижения данных через составную сеть TCP/IP. |

Наиболее простым способом передачи пакетов по сети является так называемая лавинная маршрутизация, когда каждый маршрутизатор передает пакет всем своим непосредственным соседям, исключая тот, от которого его получил. Однако, это очень нерациональный способ, так как пропускная способность сети используется крайне расточительно, тем не менее такой подход работоспособен (именно так мосты и коммутаторы локальных сетей поступают с кадрами, имеющими неизвестные адреса).

Еще одним видом маршрутизации, не требующим наличия таблиц маршрутизации, является маршрутизация от источника. В этом случае отправитель помещает в пакет информацию о том, какие промежуточные маршрутизаторы должны участвовать в передаче пакета к сети назначения. На основе этой информации каждый маршрутизатор считывает адрес следующего маршрутизатора, и если он действительно является адресом его непосредственного соседа, передает ему пакет для дальнейшей обработки. Вопрос

о том, как отправитель узнает точный маршрут следования пакета через сеть, остается открытым. Маршрут может задавать либо вручную администратор, либо автоматически узел-отправитель, но в этом случае ему нужно поддерживать какой-либо протокол маршрутизации, который сообщит ему о топологии и состоянии сети. Маршрутизация от источника является одним из стандартных режимов продвижения пакетов в протоколе IPv6.

Однако, большинство протоколов маршрутизации нацелено на создание таблиц маршрутизации.

Выбор рационального маршрута может осуществляться на основании различных критериев. Сегодня в IP-сетях применяются протоколы маршрутизации, в которых маршрут выбирается по критерию кратчайшего расстояния. При этом расстояние измеряется в различных метриках. Чаще всего используется простейшая метрика — количество хопов, то есть количество маршрутизаторов, которые нужно преодолеть пакету до сети назначения. В качестве метрик применяются также пропускная способность и надежность каналов, вносимые ими задержки и любые комбинации этих метрик.

Протокол маршрутизации должен обеспечить создание на маршрутизаторах согласованных друг с другом таблиц маршрутизации, то есть таких таблиц, которые обеспечат доставку пакета от исходной сети в сеть назначения за конечное число шагов. Современные протоколы маршрутизации поддерживают согласованность таблиц, однако это их свойство не абсолютно — при изменениях в сети, например при отказе каналов передачи данных или самих маршрутизаторов, возникают периоды нестабильной работы сети, вызванной временной несогласованностью таблиц разных маршрутизаторов. Протоколу маршрутизации обычно нужно некоторое время, которое называется временем конвергенции, чтобы после нескольких итераций обмена служебной информацией все маршрутизаторы сети внесли изменения в свои таблицы и в результате таблицы снова стали согласованными.

Различают протоколы, выполняющие статическую и адаптивную (динамическую) маршрутизацию.

При статической маршрутизации все записи в таблице имеют неизменяемый, статический статус, что подразумевает бесконечный срок их жизни. Записи о маршрутах составляются и вводятся в память каждого маршрутизатора вручную администратором сети. При изменении состояния сети администратору необходимо срочно отразить эти изменения в соответствующих таблицах маршрутизации, иначе может произойти их рассогласование, и сеть будет работать некорректно.

При адаптивной маршрутизации все изменения конфигурации сети автоматически отражаются в таблицах маршрутизации благодаря протоколам маршрутизации. Эти протоколы собирают информацию о топологии связей в сети, что позволяет им оперативно отрабатывать все текущие изменения. В таблицах маршрутизации при адаптивной маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют временем жизни (TTL) маршрута. Если по истечении времени жизни существование маршрута не подтверждается протоколом маршрутизации, то он считается нерабочим, пакеты по нему больше не посылаются.

Протоколы адаптивной маршрутизации бывают распределенными и централизованными. При распределенном подходе все маршрутизаторы сети находятся в равных условиях, они находят маршруты и строят собственные таблицы маршрутизации, работая в тесной кооперации друг с другом, постоянно обмениваясь информацией о конфигурации сети. При централизованном подходе в сети существует один выделенный маршрутизатор, который собирает всю информацию о топологии и состоянии сети от других маршрутизаторов. На основании этих данных выделенный маршрутизатор (который иногда называют сервером маршрутов) строит таблицы маршрутизации для всех остальных маршрутизаторов сети, а затем распространяет их по сети, чтобы каждый маршрутизатор получил собственную таблицу и в дальнейшем самостоятельно принимал решение о продвижении каждого пакета.

Применяемые сегодня в IP-сетях протоколы маршрутизации относятся к адаптивным распределенным протоколам, которые, в свою очередь, делятся на две группы:

* дистанционно-векторные алгоритмы (Distance Vector Algorithm, DVA);
* алгоритмы состояния связей (Link State Algorithm, LSA).

## 6.1 Протокол маршрутизации RIP

Протокол RIP (Routing Information Protocol — протокол маршрутной информации) является внутренним протоколом маршрутизации дистанционно-векторного типа.

Для измерения расстояния до сети стандарты протокола RIP допускают различные виды метрик: хопы, значения пропускной способности, вносимые задержки, надежность сетей, а также любые комбинации этих метрик. Метрика должна обладать свойством аддитивности — метрика составного пути должна быть равна сумме метрик составляющих этого пути.

В большинстве реализаций RIP используется простейшая метрика — количество хопов, то есть количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим пример работы протокола RIP на примере составной сети, изображённой на рис. 6.1.



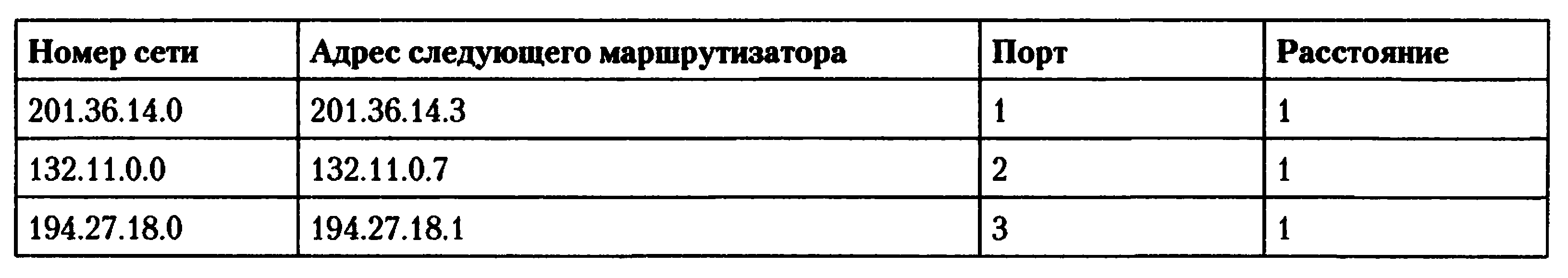
Рис. 6.1 — Пример составной сети.

Процесс построения таблицы маршрутизации включает 5 этапов.

Этап 1 — создание минимальной таблицы. Данная составная сеть включает восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами: Rl, R2, R3 и R4. Маршрутизаторы, работающие по протоколу RIP, могут иметь идентификаторы, однако для протокола они не являются необходимыми. В RIP-сообщениях эти идентификаторы не передаются. В исходном состоянии на каждом маршрутизаторе программным обеспечением стека TCP/IP автоматически создается минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединенные сети (табл. 6.1).

Таблица 6.1.

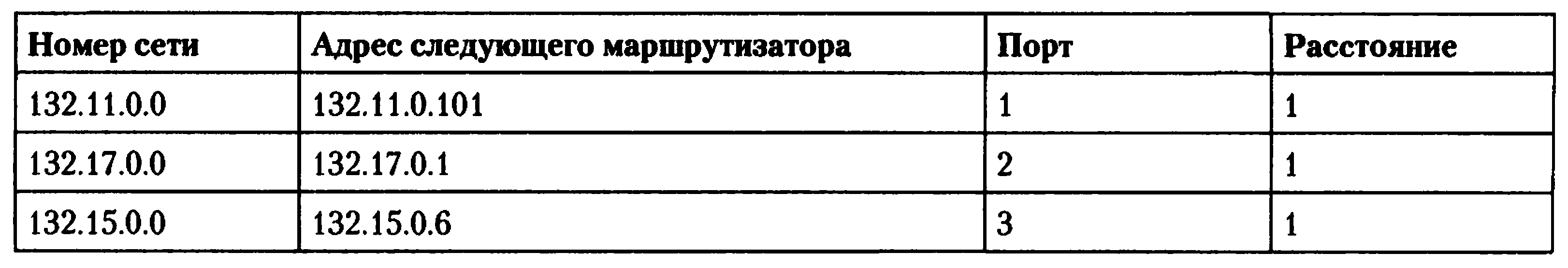
Минимальная таблица маршрутизации маршрутизатора R1.



Минимальные таблицы маршрутизации в других маршрутизаторах будут выглядеть соответственно (табл. 6.2).

Таблица 6.2.

Минимальная таблица маршрутизации маршрутизатора R2.



Этап 2 — рассылка минимальной таблицы соседям. После инициализации каждый маршрутизатор начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица. RIP-сообщения передаются в дейтаграммах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние до нее от передающего сообщение маршрутизатора.

По отношению к любому маршрутизатору соседями являются те маршрутизаторы, которым данный маршрутизатор может передать IP-пакет по какой-либо своей сети, не пользуясь услугами промежуточных маршрутизаторов. Например, для маршрутизатора R1 соседями являются маршрутизаторы R2 и R3, а для маршрутизатора R4 — маршрутизаторы R2 и R3.

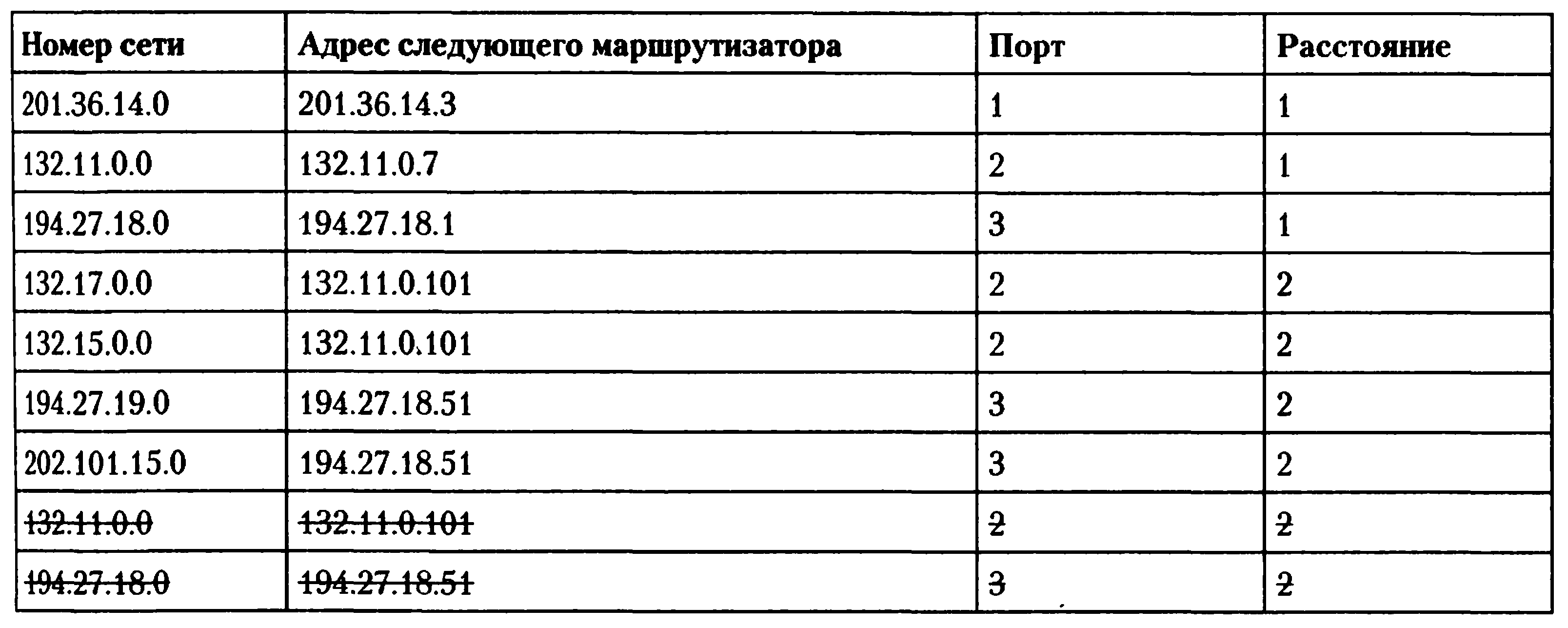
Таким образом, маршрутизатор R1 передает маршрутизаторам R2 и R3 следующие сообщения:

* сеть 201.36.14.0, расстояние 1;
* сеть 132.11.0.0, расстояние 1;
* сеть 194.27.18.0, расстояние 1.

Этап 3 — получение RIP-сообщений от соседей и обработка полученной информации. После получения аналогичных сообщений от маршрутизаторов R2 и R3 маршрутизатор R1 наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора станет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации (табл. 6.3).

Таблица 6.3.

Таблица маршрутизации маршрутизатора R1.



Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (с меньшим расстоянием в хопах), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остается только одна запись; если же имеется несколько записей, равнозначных в отношении путей к одной и той же сети, то все равно в таблице остается одна запись, которая пришла в маршрутизатор первая по времени. Для этого правила существует исключение — если худшая информация о какой-либо сети пришла от того же маршрутизатора, на основании сообщения которого была создана данная запись, то худшая информация замещает лучшую.

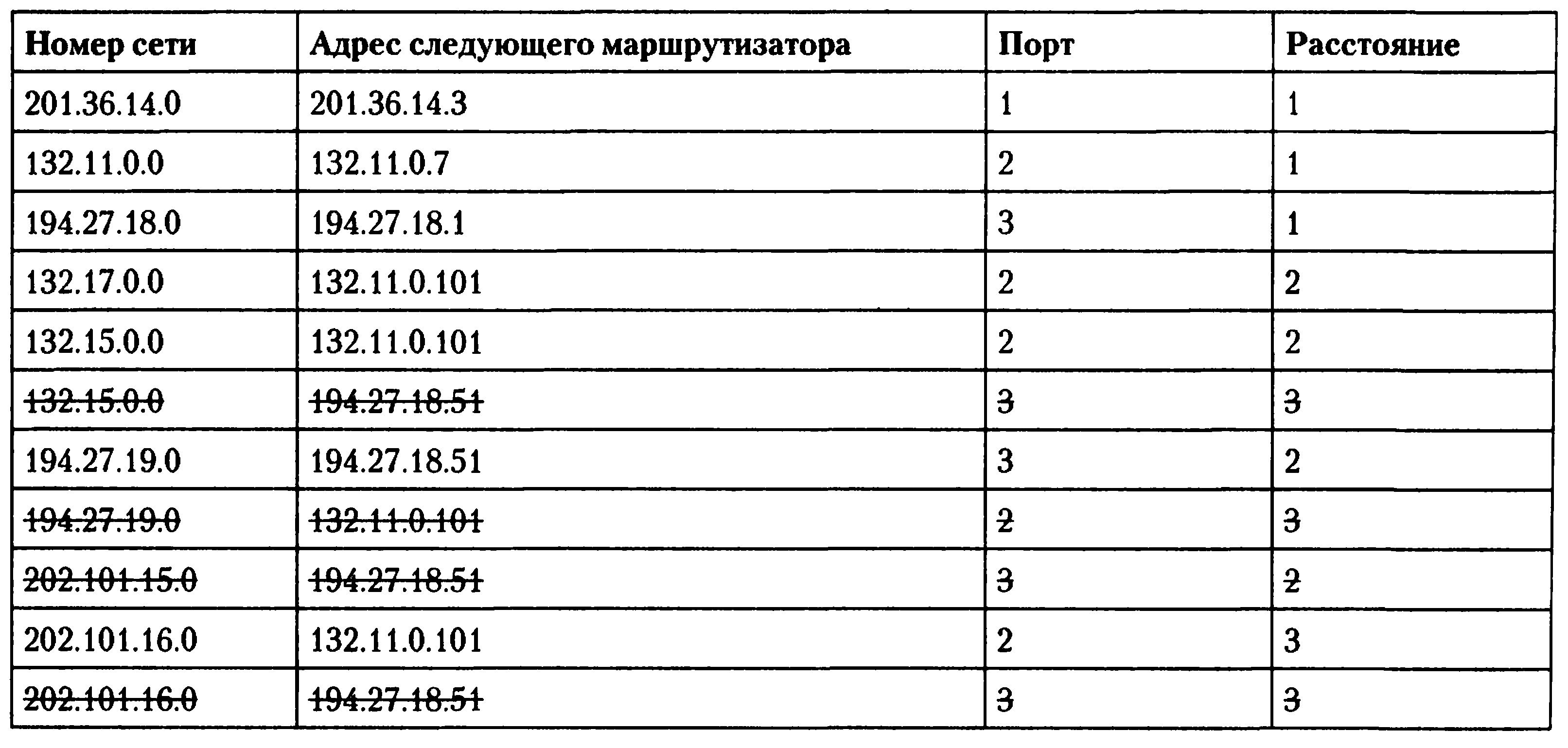
Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4 — рассылка новой таблицы соседям. Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные обо всех известных ему сетях: как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5 — получение RIP-сообщений от соседей и обработка полученной информации. Этап 5 повторяет этап 3 — маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации (табл. 6.4).

Таблица 6.4.

Таблица маршрутизации маршрутизатора R1.



О сети 202.101.16.0 маршрутизатор R1 узнает на этом этапе впервые, причем данные о ней приходят от двух соседей — от R3 и R4. Поскольку метрики в этих сообщениях указаны одинаковые, то в таблицу попадают данные, пришедшие первыми.

Если маршрутизаторы периодически повторяют этапы рассылки и обработки RIP-сообщений, то за конечное время в сети установится корректный режим маршрутизации. Под корректным режимом маршрутизации здесь понимается такое состояние таблиц маршрутизации, когда все сети достижимы из любой сети с помощью некоторого рационального маршрута.

## 6.2 Протокол маршрутизации OSPF

Протокол OSPF (Open Shortest Path First — выбор кратчайшего пути первым) является последним принятым протоколом, основанном на алгоритме состояния связей, и обладает многими особенностями, ориентированными на применение в больших гетерогенных (составных) сетях.

OSPF разбивает процедуру построения таблицы маршрутизации на два этапа, к первому относится построение и поддержание базы данных о состоянии связей сети, ко второму — нахождение оптимальных маршрутов и генерация таблицы маршрутизации.

Связи сети могут быть представлены в виде графа, в котором вершинами являются маршрутизаторы и подсети, а ребрами — связи между ними. Каждый маршрутизатор обменивается со своими соседями той информацией о графе сети, которой он располагает к данному моменту. Этот процесс похож на процесс распространения векторов расстояний до сетей в протоколе RIP, однако сама информация качественно иная — это информация о топологии сети. Сообщения, с помощью которых распространяется топологическая информация, называются объявлениями о состоянии связей (Link State Advertisement, LSA) сети. При транзитной передаче объявлений LSA маршрутизаторы не модифицируют информацию, как это происходит в дистанционно-векторных протоколах, в частности в RIP, а передают ее в неизменном виде. В результате все маршрутизаторы сети сохраняют в своей памяти идентичные сведения о текущей конфигурации графа связей сети.

Для контроля состояния связей и соседних маршрутизаторов OSPF-маршрутизаторы передают друг другу особые HELLO-сообщения каждые 10 секунд. Небольшой объем этих сообщений делает возможным частое тестирование состояния соседей и связей с ними. В том случае, когда сообщения HELLO перестают поступать от какого-либо непосредственного соседа, маршрутизатор делает вывод о том, что состояние связи изменилось с работоспособного на неработоспособное и вносит соответствующие коррективы в свою топологическую базу данных. Одновременно он отсылает всем непосредственным соседям объявление LSA об этом изменении, те также вносят исправления в свои базы данных и, в свою очередь, рассылают данное объявление LSA своим непосредственным соседям.

Следующая задача — нахождения оптимального пути на графе является достаточно сложной и трудоемкой. В протоколе OSPF для ее решения используется итеративный алгоритм Дейкстры. Каждый маршрутизатор сети, действуя в соответствии с этим алгоритмом, ищет оптимальные маршруты от своих интерфейсов до всех известных ему подсетей. В каждом найденном таким образом маршруте запоминается только один шаг — до следующего маршрутизатора. Данные об этом шаге и попадают в таблицу маршрутизации.

Если состояние связей в сети изменилось и произошла корректировка графа сети, каждый маршрутизатор заново ищет оптимальные маршруты и корректирует свою таблицу маршрутизации. Аналогичный процесс происходит и в том случае, когда в сети появляется новая связь или новый сосед, объявляющий о себе с помощью своих сообщений HELLO.

При работе протокола OSPF конвергенция таблиц маршрутизации к новому согласованному состоянию происходит достаточно быстро, быстрее, чем в сетях, в которых работают дистанционно-векторные протоколы. Это время состоит из времени распространения по сети объявления LSA и времени работы алгоритма Дейкстры, который обладает быстрой сходимостью. Однако вычислительная сложность этого алгоритма предъявляет высокие требования к мощности процессора маршрутизатора.

Когда состояние сети не меняется, то объявления о связях не генерируются, топологические базы данных и таблицы маршрутизации не корректируются, что экономит пропускную способность сети и вычислительные ресурсы маршрутизаторов. Однако у этого правила есть исключение: каждые 30 минут OSPF-маршрутизаторы обмениваются всеми записями базы данных топологической информации, то есть синхронизируют их для более надежной работы сети. Так как этот период достаточно большой, то данное исключение незначительно сказывается на загрузке сети.

В протоколе OSPF применяется понятие зон (area), разбивающих сеть на меньшие логические участки, в которых топологическая база информации о подсетях будет меньшего размера. Маршрутизаторы в какой-либо зоне знают о сетях и устройствах своей зоны, но могут абсолютно ничего не знать о топологии остальных зон. Поскольку база топологической информации становится намного меньше при таком подходе, процессоры устройств загружены меньше, меньше времени требуется на выполнение алгоритма SPF и меньший объем оперативной памяти будет занят.

Протоколы, основанные на состоянии связей, к которым относится OSPF, обладают рядом преимуществ по отношению к дистанционно-векторным протоколам маршрутизации. Основными из них являются более высокая скорость работы и меньший объём передаваемого по сети траффика, а главным недостатком служит их вычислительная сложность, которая при современном уровне развития аппаратных средств не оказывает значительного влияния на общую эффективность алгоритма. Более подробное сравнение этих двух видов протоколов приведено в таблице 6.5.

Таблица 6.5 Сравнение протоколов маршрутизации

|  |  |
| --- | --- |
| **Маршрутизация по вектору расстояния** | **Маршрутизация с учетом состояния канала связи** |
| Видит топологию сети глазами соседних маршрутизаторов | Получает общий вид топологии всей сети |
| Суммирует вектор расстояния от одного маршрутизатора к другому | Вычисляет кратчайший путь до других маршрутизаторов |
| Частые периодические обновления топологической информации, медленная сходимость | Обновления инициируются фактом изменения топологии; быстрая сходимость |
| Передает копии таблицы маршрутизации только соседним маршрутизаторам | Передает пакеты с информацией об актуальном состоянии канала связи всем другим маршрутизаторам |

# 7 ТРАНСПОРТНЫЙ УРОВЕНЬ

Средства сетевого уровня обеспечивают доставку данных между устройствами в составной сети (компьютерами, маршрутизаторами и т.д). Однако не следует забывать, что на одном узле может функционировать параллельно несколько программ, которым требуется доступ к сети. Следовательно, данные внутри компьютерной системы должны распределяться между программами. Поэтому, при передаче данных по сети недостаточно просто адресовать конкретный узел. Необходимо также идентифицировать программу-получателя, что невозможно осуществить средствами сетевого уровня.

Другой серьезной проблемой протоколов сетевого уровня является отсутствие средств, позволяющих передавать большие массивы данных. Когда исходные данные превышают максимально допустимый размер пакета сетевого уровня, то эти данные должны быть разбиты на порции, каждая из которых передается в сеть отдельным пакетом. Однако каждый пакет сетевого уровня передается по сети как единый, независимый от других блок данных. В случае если какие-либо пакеты "потерялись", то модуль сетевого протокола на принимающей стороне не сможет обнаружить потерю, и, следовательно, – обнаружить нарушение целостности общего массива данных. Поэтому средства транспортного уровня обеспечивают отсутствие потерь информации. Такой режим передачи данных получил название гарантированной доставки.

Таким образом, средства транспортного уровня представляют собой функциональную надстройку над сетевым уровнем и решают две основных задачи:

* обеспечение доставки данных между конкретными программами, функционирующими, в общем случае, на разных узлах сети;
* обеспечение гарантированной доставки массивов данных произвольного размера.

## 7.1 Понятие порта

|  |
| --- |
| Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне числом, которое называется номером порта. |

Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер порта (или проще — порт).

Диапазоны чисел, из которых выделяются номера TCP- и UDP-портов, совпадают: от 0 до 1023 для назначенных и от 1024 до 65 535 для динамических. Однако никакой связи между назначенными номерами TCP- и UDP-портов нет. Даже если номера TCP- и UDP-портов совпадают, они идентифицируют разные приложения. Например, одному приложению может быть назначен ТСР-порт 1750, а другому — UDP-порт 1750.

Назначенные номера из диапазона от 0 до 1023 являются уникальными в пределах Интернета и закрепляются за приложениями централизованно.

Для тех приложений, которые еще не стали столь распространенными, номера портов назначаются локально разработчиками этих приложений или операционной системой в ответ на поступление запроса от приложения. На каждом компьютере операционная система ведет список занятых и свободных номеров портов. При поступлении запроса от приложения, выполняемого на данном компьютере, операционная система выделяет ему первый свободный номер. Такие номера называют динамическими. В дальнейшем все сетевые приложения должны адресоваться к данному приложению с указанием назначенного ему динамического номера порта. После того как приложение завершит работу, его номер возвращается в список свободных и может быть назначен другому приложению. Динамические номера являются уникальными в пределах каждого компьютера, но при этом обычной является ситуация совпадения номеров портов приложений, выполняемых на разных компьютерах. Как правило, клиентские части известных приложений (DNS, WWW, FTP, telnet и др.) получают динамические номера портов от ОС.

Каждый компьютер может выполнять несколько процессов, более того, прикладной процесс тоже может иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных. Поэтому после того, как пакет средствами протокола IP доставлен на сетевой интерфейс компьютера-получателя, данные необходимо переправить конкретному процессу получателю.

Существует и обратная задача: пакеты, которые отправляют в сеть разные приложения, работающие на одном конечном узле, обрабатываются общим для них протоколом IP. Следовательно, в стеке должно быть предусмотрено средство «сбора» пакетов от разных приложений для передачи протоколу IP. Эту работу выполняют протоколы TCP и UDP.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их операционных систем.

## 7.2 Протокол UDP

Протокол UDP (User Datagram Protocol) — протокол транспортного уровня, входящий в стек протоколов TCP/IP, обеспечивающий негарантированную доставку данных без установления виртуального соединения.

При работе на хосте-отправителе данные от приложений поступают протоколу UDP через порт в виде сообщений (рис. 7.1). Протокол UDP добавляет к каждому отдельному сообщению свой 8-байтный заголовок, формируя из этих сообщений собственные протокольные единицы, называемые UDP-дейтаграммами, и передает их нижележащему протоколу IP.



Рис. 7.1 — Работа протокола UDP на хосте отправителе.

Каждая дейтаграмма переносит отдельное пользовательское сообщение. Сообщения могут иметь различную длину, не превышающую однако длину поля данных протокола IP, которое, в свою очередь, ограничено размером кадра технологии нижнего уровня. Поэтому если буфер UDP переполняется, то сообщение приложения отбрасывается.

Заголовок UDP состоит из четырех 2-байтных полей (рис. 7.2) после которых идёт поле данных переменной длины (минимум 8 байт).

|  |  |  |  |
| --- | --- | --- | --- |
| порт отправителя | порт получателя | длина дейтаграммы | контрольная сумма |

Рис. 7.2 — Структура заголовка UDP-дейтаграммы.

Поле «контрольная сумма» является необязательным и используется для повышения надёжности. Она рассчитывается с использованием метода контроля чётности (см. п. 4.4) и позволяет только выявлять (но не исправлять) ошибки. В случае, если контрольная сумма дейтаграммы, вычисленная на принимающей стороне, не совпадает с контрольной суммой, указанной в заголовке отправителем, UDP-пакет отбрасывается.

Основная область применения протокола UDP — приложения, для которых основным требованием является минимизация задержек передачи данных, например, трансляция аудио или видео потока.

## 7.3 Протокол TCP

Протокол TCP (Transmission Control Protocol) — транспортный протокол стека TCP/IP, обеспечивающий гарантированную доставку данных с установлением виртуального соединения.

В отличие от протокола UDP, который создает свои дейтаграммы на основе логически обособленных единиц данных — сообщений, генерируемых приложениями, протокол TCP делит поток данных на сегменты без учета их смысла или внутренней структуры. Для надёжной передачи сегментов протокол TCP использует метод продвижения данных с установлением логического соединения, определяя при этом параметры процедуры обмена данными. Как было сказано ранее, логическое соединение дает возможность участникам обмена следить за тем, чтобы данные не были потеряны, искажены или продублированы, а также чтобы они пришли к получателю в том порядке, в котором были отправлены.

Для обеспечения гарантированной доставки протокол TCP использует механизм отправки подтверждения (квитирование), при котором отправитель отсылает данные и ждет, пока к нему не придет квитанция, подтверждающая, что его данные благополучно дошли до адресата.

К реализации этого механизма существуют 2 различных подхода:

* метод простоя источника,
* метод скользящего окна.

Метод простоя источника требует, чтобы источник, пославший кадр (в данном случае не имеет значения, какое название используется для единицы передаваемых данных), дожидался от приемника квитанции, извещающей о том, что исходный кадр получен и данные в нем корректны, и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция в течение тайм-аута не пришла, то кадр (или квитанция) считается утерянным, и его передача повторяется (рис. 7.3).



Рис. 7.3 — Метод простоя источника.

Достаточно очевидно, что при использовании данного метода производительность обмена данными ниже потенциально возможной — передатчик мог бы посылать следующий кадр сразу же после отправки предыдущего, но он обязан ждать прихода квитанции.

## 7.4 Метод скользящего окна

В этом методе для повышения скорости передачи данных источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе еще до получения на эти кадры квитанций (рис. 7.4). Количество кадров, которые разрешается передавать таким образом, называется размером окна.

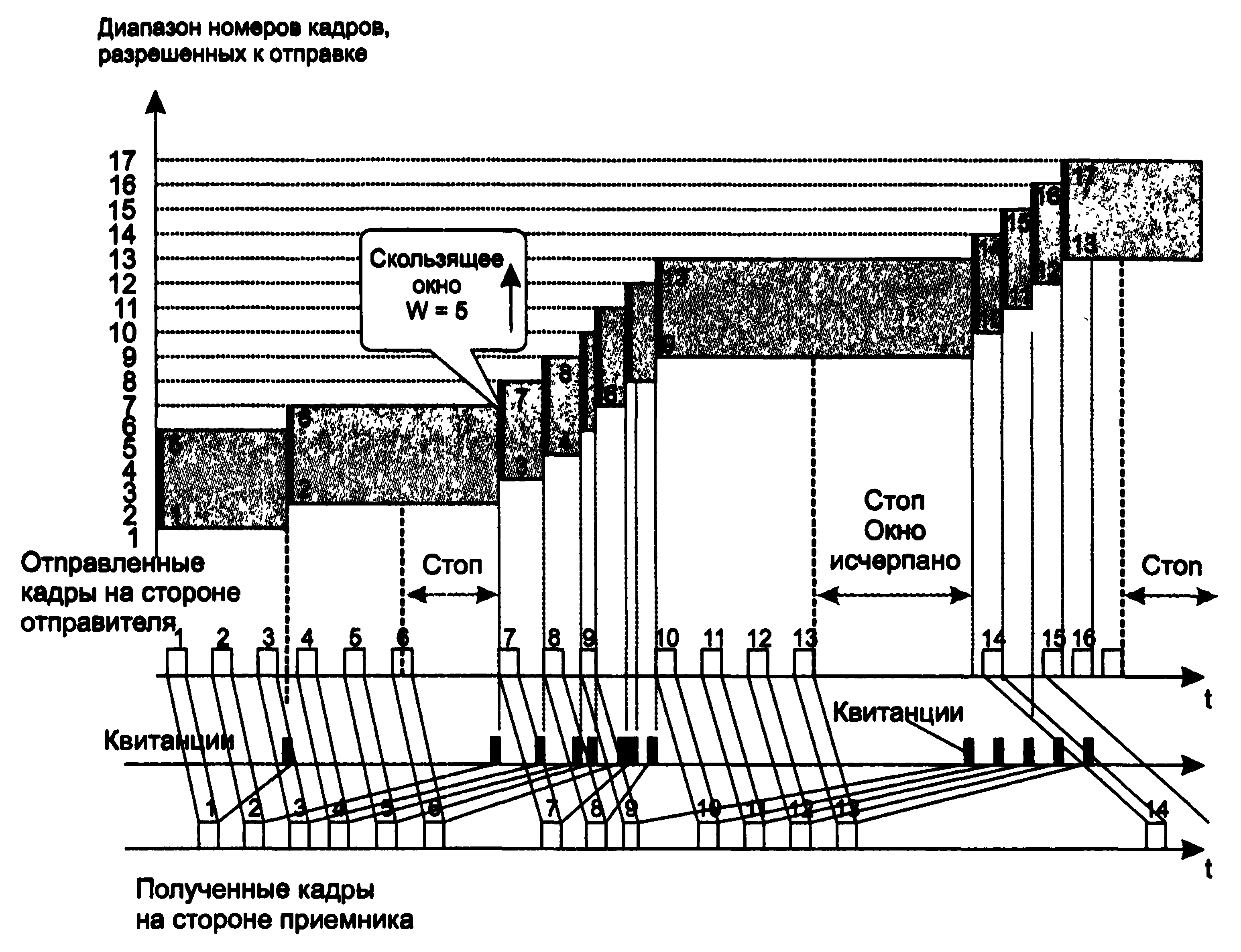


Рис. 7.4 — Метод скользящего окна.

Реализация метода скользящего окна в протоколе TCP подразумевает управление потоком передачи данных, которое включает в себя определение следующих параметров: тайм-аут ожидания квитанции и размер окна. Правильный подбор данных параметров позволяет практически полностью исключить простои в сети и достичь большей производительности по сравнению с методом простоя источника.

При выборе величины тайм-аута должны учитываться скорость и надежность линий связи, их протяженность и многие другие факторы. Тайм-аут не должен быть слишком коротким, чтобы по возможности исключить избыточные повторные передачи, снижающие полезную пропускную способность системы, но он не должен быть и слишком длинным, чтобы избежать длительных простоев, связанных с ожиданием несуществующей или потерянной квитанции.

В протоколе TCP тайм-аут определяется с помощью достаточно сложного адаптивного алгоритма, идея которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времени оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент.

Размер окна приема связан с наличием в данный момент места в буфере данных у принимающей стороны. Поэтому в общем случае окна приема на разных концах соединения имеют разный размер. Например, можно ожидать, что сервер, вероятно обладающий большим буфером, пошлет клиентской станции окно приема большее, чем клиент серверу. В зависимости от состояния сети то одна, то другая стороны могут объявлять новые значения окон приема, динамически уменьшая и увеличивая их.

Варьируя величину окна, можно влиять на загрузку сети. Чем больше окно, тем большая порция неподтвержденных данных может быть послана в сеть. Но если пришло большее количество данных, чем может быть принято модулем TCP, данные отбрасываются. Это ведет к излишним пересылкам информации и ненужному росту нагрузки на сеть и модуль TCP. В то же время окно малого размера может ограничить передачу данных скоростью, которая определяется временем передачи по сети каждого посылаемого сегмента.

В общем случае метод скользящего окна более сложен в реализации, чем метод простоя источника, так как передатчик должен хранить в буфере копии всех кадров, на которые пока не получены квитанции. Кроме того, при использовании данного метода требуется отслеживать несколько параметров алгоритма, таких как размер окна, номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции и многие другие, однако гарантия доставки пакетов узлу назначения является зачастую решающим критерием, который определяет широкое использование протокола TCP в сетевых приложения.

# 8 ТРАНСЛЯЦИЯ СЕТЕВЫХ АДРЕСОВ

При подключении к сети интернет, провайдеры, как правило, выделяют одному клиенту одну физическую линию и один IP-адрес, однако на практике почти всегда возникает необходимость обеспечить выход в интернет одновременно нескольким устройствам. Это может быть компьютер и смартфон в домашней сети или несколько сотен компьютеров в локальной сети организации. Ещё большей проблемой является то, что в нескольких сетях, подключённых, имеющих выход в интернет могут быть выбраны одинаковые схемы адресации, что делает в принципе невозможным их соединение с использованием стандартных методов маршрутизации рассмотренных ранее (рис. 8.1).

Для решения этой проблемы используется технология трансляции сетевых адресов (Network Address Translation — NAT), которая преобразовывает IP-адреса транзитных пакетов при их прохождении через какое-либо устройство (маршрутизатор, шлюз, сервер доступа). При этом сети, между которыми находится это устройство, отделяются друг от друга. В этом случае говорят, что одна сеть является внешней, а другая — внутренней. Если трансляция сетевых адресов используется при подключении локальной сети к интернету, то он является внешней сетью, а локальная сеть — внутренней.



Рис. 8.1 — Типичная схема подключения к интернету нескольких

локальных сетей

Для использования во внутренних сетях, выделены 3 диапазона частных адресов:

10.0.0.0 — 10.255.255.255 (8-битной маской);

172.16.0.0 — 172.31.255.255 (12-битной маской);

192.168.0.0 — 192.168.255.255 (16-битной маской).

Ни один из этих адресов не является маршрутизируемым в интернете. Если попытаться выполнить команду ping, направленную на частный адрес, в ответ будут получены пакеты network unreachable (сеть недоступна). Использование частных адресов, а также их произвольных комбинаций предоставляет организации широкие возможности по разработке своей внутренней схемы адресации.

Технология NAT также оказывается полезной, когда организация из соображений безопасности желает скрыть адреса узлов своей сети, чтобы не дать возможности злоумышленникам составить представление о структуре и масштабах корпоративной сети, а также о структуре и интенсивности исходящего и входящего трафиков.

## 8.1 Виды NAT

В зависимости от способа преобразования внутреннего IP-адреса во внешний и наоборот выделяют несколько видов NAT:

* статический,
* динамический,
* динамический с перегрузкой.

Статический NAT задает однозначное соответствие одного адреса другому, которое хранится неограниченно долго (табл. 8.1). Особенно полезен данный способ, когда устройство должно быть доступным из внешней сети. В примере, приведённом в табл. 8.1, компьютер с внутренним адресом 192.168.1.2 будет всегда транслироваться в адрес 31.23.192.75 и наоборот, запрос из внешней сети на адрес 31.23.192.75 будет передаваться во внутреннюю сеть на адрес 192.168.1.2.

Таблица 8.1. Соответствие IP-адресов при статическом NAT.

|  |  |
| --- | --- |
| **Внутренний IP** | **Внешний IP** |
| 192.168.1.2 | 31.23.192.75 |
| 192.168.1.3 | 31.23.192.76 |
| 192.168.1.4 | 31.23.192.77 |

Динамический NAT подразумевает, что при прохождении пакета через маршрутизатор, новый адрес выбирается динамически из некоторого диапазона адресов, называемого пулом. Запись о трансляции хранится некоторое время, чтобы ответные пакеты могли быть доставлены адресату. Если в течение некоторого времени трафик по этой трансляции отсутствует, трансляция удаляется и адрес возвращается в пул. Если требуется создать трансляцию, а свободных адресов в пуле нет, то пакет отбрасывается. Иными словами, необходимо, чтобы число внутренних адресов было не намного больше числа адресов в пуле, иначе высока вероятность проблем с доступом во внешнюю сеть.

Динамический NAT с перегрузкой (Overload NAT) — форма динамического NAT, который отображает несколько внутренних адресов в единственный внешний IP адрес, используя различные порты (рис. 8.3). Известен также как PAT (Port Address Translation).



Рис. 8.3 — Схема трансляции адресов.

При перегрузке, каждый компьютер в частной сети транслируется в тот же самый адрес (для примера на рисунке это 172.16.0.2), но с различным номером порта. NAT-перегрузка использует особенность стека протокола TCP/IP, такую как мультиплексирование, которая позволяет компьютеру поддерживать несколько параллельных подключений с удаленным компьютером, используя различные TCP или UDP порты.

При этом выделяют несколько основных типов адресов (рис. 7.3):

Inside local — внутренний локальный адрес узла (как правило частный адрес) который посылает пакет во внешнюю сеть;

Inside global — внутренний глобальный адрес — разрешенный адрес из набора адресов NAT, который заменяет частный адрес при выходе пакета в открытую сеть;

Outside local — внешний локальный адрес — открытый адрес интерфейса конечной сети в которой находится получатель пакета (адресуемый узел).

Outside global — внешний глобальный адрес — адрес конечного узла которому отправляется пакет.

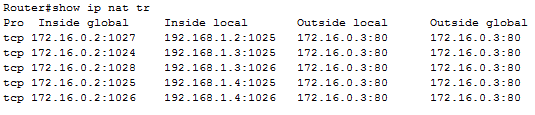


Рис. 8.4 — NAT-таблица на маршрутизаторе Cisco.

NAT прозрачен для компьютеров источника и приемника, то есть никто из них не знает, что это имеет дело с третьим устройством.

Реализация динамического NAT автоматически создает межсетевую защиту между внутренней сетью и внешними сетями или интернетом. Динамический NAT позволяет только подключения, которые порождаются в локальной сети. По существу, это означает, что компьютер во внешней сети не может соединиться с компьютером во внутренней (частной) сети, если тот сам не начал соединение.

## 8.2 Технология перенаправления портов

Описанная выше особенность с одной стороны действительно повышает безопасность, но с другой существенно ограничивает возможности по использованию сетевых технологий выполнением только клиентских функций. В случае необходимости обеспечить возможность подключения из внешней сети к компьютерам или серверам во внутренней сети используют перенаправление портов (port forwarding).

Для его настройки в маршрутизаторе необходимо задать соответствие между портами, используемыми определенными приложениями, и IP-адресами тех серверов внутренней сети, на которых эти приложения работают. В результате любой запрос из внешней сети на IP-адрес внешнего порта маршрутизатора по указанному порту, будет автоматически перенаправлен на указанный внутренний сервер.

Маршрутизаторы позволяют создавать несколько статических перенаправлений портов. То есть, можно открыть несколько портов на одном внутреннем сервере или создать несколько серверов с различными IP-адресами во внутренней сети. Однако при статическом перенаправлении портов нельзя перенаправлять один порт на несколько IP-адресов, то есть порт может соответствовать только одному IP-адресу.

Большинство моделей маршрутизаторов позволяют также задавать статическое перенаправление группы портов, то есть ставить в соответствие IP-адресу внутреннего сервера сразу несколько портов. Такая возможность полезна в том случае, если необходимо обеспечить работу приложений, использующих большое количество портов, например игр или аудио/видеоконференций.

Более полезной может оказаться возможность перенаправления внешних пакетов, приходящих на разные порты, разным внутренним серверам, но на одинаковый порт. В примере на рисунке 8.5 показана ситуация, когда имеется 2 сервера, выполняющих одинаковые функции — web-серверы. Соответственно, они оба ожидают подключения на стандартный для web-сервера порт 80, однако, как говорилось выше, нельзя перенаправлять один порт на несколько IP-адресов, но можно перенаправить внешний порт, отличный от стандартного (в примере это 8080) на стандартный (80). В этом случае внешние клиенты для обращения к 1-му web-серверу должны будут указать только внешний адрес маршрутизатор — 87.117.93.42, а для обращения ко 2-му web-серверу добавить после адреса новый порт, то есть 87.117.93.42:8080.



Рис. 8.5 — Перенаправление портов

# 9 ВИРТУАЛЬНЫЕ ЧАСТНЫЕ СЕТИ (VPN)

|  |
| --- |
| VPN (англ. Virtual Private Network — виртуальная частная сеть) — обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (рис. 9.1). |

В зависимости от применяемых протоколов, VPN может обеспечивать соединения трёх видов: узел-узел, узел-сеть и сеть-сеть.



Рис. 9.1 — Пример организации VPN-сети

Технология VPN находит применение в различных сферах деятельности для решения широкого круга задач по организации взаимодействия различных участников корпоративной сети через интернет. При этом можно выделить несколько основных вариантов использования виртуальных частный сетей.

Intranet VPN используется для объединения в единую защищённую сеть нескольких территориально распределённых филиалов одной организации, обменивающихся данными по открытым каналам связи. При организации такой схемы подключения требуется наличие VPN серверов равное количеству связываемых офисов. В качестве VPN сервера может выступать маршрутизатор.

Данный способ целесообразно использовать как для обыкновенных филиалов, так и для мобильных офисов, которые будут иметь доступ к ресурсам главного офиса компании, а также без проблем обмениваться данными между собой.

Remote Access VPN используется для создания защищённого канала между сегментом корпоративной сети (центральным офисом или филиалом) и одиночным пользователем, который, работая дома, подключается к корпоративным ресурсам с домашнего компьютера, корпоративного ноутбука, смартфона с использованием программного VPN-клиента.

Client/Server VPN обеспечивает защиту передаваемых данных между двумя узлами (не сетями) корпоративной сети. Особенность данного варианта в том, что VPN строится между узлами, находящимися, как правило, в одном сегменте сети, например, между рабочей станцией и сервером. Такая необходимость очень часто возникает в тех случаях, когда в одной физической сети необходимо создать несколько логических сетей.

Extranet VPN используется для сетей, к которым подключаются «внешние» пользователи (например, заказчики или клиенты). Уровень доверия к ним намного ниже, чем к сотрудникам компании, поэтому требуется обеспечение специальных «рубежей» защиты, предотвращающих или ограничивающих доступ последних к особо ценной, конфиденциальной информации. В этом случае для удаленных клиентов будут очень урезаны возможности по использованию корпоративной сети, фактически они будут ограничены доступом к тем ресурсам компании, которые необходимы при работе со своими клиентами, например, сайта с коммерческими предложениями, а VPN используется в этом случае для безопасной пересылки конфиденциальных данных.

Internet VPN используется для предоставления доступа к интернету провайдерами, обычно в случае если по одному физическому каналу подключаются несколько пользователей.

В основе работы VPN лежат два фундаментальных понятия: туннелирование и шифрование.

## 9.1 Туннелирование

|  |
| --- |
| Туннелирование (от англ. tunnelling — «прокладка туннеля) — процесс, в ходе которого создается логическое соединение между двумя конечными точками посредством инкапсуляции различных протоколов. |

|  |
| --- |
| Инкапсуляция — это процесс передачи данных с верхнего уровня приложений вниз (по стеку протоколов) к физическому уровню. При продвижении пакета данных по уровням сверху вниз каждый новый уровень добавляет к пакету свою служебную информацию. |

Служебная информация предназначается для объекта того же уровня на удаленном компьютере, её формат и интерпретация определяются протоколом данного уровня. Разумеется, данные, приходящие с верхнего уровня, могут на самом деле представлять собой пакеты с уже инкапсулированными данными еще более верхнего уровня.

От обычных многоуровневых сетевых моделей (таких как OSI или TCP/IP) (рис. 9.2 а) туннелирование отличается тем, что инкапсулируемый протокол относится к тому же или более низкому уровню, чем используемый в качестве тоннеля (рис. 9.2 б).



а)



б)

Рис. 9.2 — Инкапсуляция пакетов: а) — стандартная, б) — при туннелировании.

Суть туннелирования состоит в том, чтобы упаковать передаваемую порцию данных, вместе со служебными полями, в новый «конверт» для обеспечения конфиденциальности и целостности всей передаваемой порции, включая служебные поля. Для выполнения этих функций чаще всего используется протокол GRE (Generic Routing Encapsulation — общая инкапсуляция маршрутов), а также некоторые другие. Туннелирование может применяться на сетевом и на прикладном уровнях.

Для транспортировки данных по открытой сети используются открытые поля заголовка внешнего пакета. Для внешних пакетов используются адреса пограничных маршрутизаторов, установленных в начале и конце туннеля, а внутренние адреса конечных узлов содержатся во внутренних исходных пакетах в защищенном виде. По прибытии в конечную точку защищенного канала из внешнего пакета извлекают и расшифровывают внутренний исходный пакет и используют его восстановленный заголовок для дальнейшей передачи по внутренней сети.

VPN соединение всегда состоит из канала типа точка-точка (туннеля). Туннель создаётся в незащищённой сети, в качестве которой чаще всего выступает Интернет. Соединение точка-точка подразумевает, что оно всегда устанавливается между двумя компьютерами, которые называются узлами или пирами (от англ. peer — равный). Каждый конечный узел отвечает за шифрование данных до того, как они попадут в туннель и расшифровке этих данных после того, как они туннель покинут.

Хотя VPN-туннель всегда устанавливается между двумя точками, каждый конечный узел может устанавливать дополнительные туннели с другими узлами. Для примера, когда трём удалённым станциям необходимо связаться с одним и тем же офисом, будет создано три отдельных VPN туннеля к этому офису.

## 9.2 Шифрование в виртуальных сетях

Узел, обеспечивающий подключения к частной сети называется VPN шлюзом, а сеть за ним — доменом шифрования (рис. 9.3). Использование шлюзов даёт несколько преимуществ. Во-первых, все пользователи должны пройти через одно устройство, которое облегчает задачу управления политикой безопасности и контроля входящего и исходящего трафика сети. Во-вторых, если устанавливать персональные туннели к каждой рабочей станции, к которой необходимо получить доступ, их число будет расти очень быстро, и они станут неуправляемыми. При наличии шлюза, пользователь устанавливает соединение с ним (то есть с одним устройством), после чего открывается доступ ко всей сети (домену шифрования).



Рис. 9.3 — Домен шифрования.

Интересно отметить, что внутри домена шифрования самого шифрования не происходит. Причина в том, что эта часть сети считается безопасной и находящейся под непосредственным контролем в противоположность интернету. Это справедливо и при соединении офисов с помощью VPN шлюзов. Таким образом гарантируется шифрование только той информации, которая передаётся по небезопасному каналу между офисами. Рисунок 2 показывает VPN соединяющую два офиса.

Существует много вариантов VPN шлюзов и VPN клиентов. Это может быть аппаратное VPN устройство или программное VPN обеспечение, которое устанавливается на маршрутизаторах или на ПК.

Независимо от используемого ПО, все VPN работают по следующим принципам:

* Каждый из узлов идентифицирует друг друга перед созданием туннеля, чтобы удостовериться, что шифрованные данные будут отправлены на нужный узел.
* Оба узла требуют заранее настроенной политики, указывающей какие протоколы могут использоваться для шифрования и обеспечения целостности данных.
* Узлы сверяют политики, чтобы договориться об используемых алгоритмах; если это не получается, то туннель не устанавливается.
* Как только достигнуто соглашение по алгоритмам, создаётся ключ, который будет использован в симметричном алгоритме для шифрования/расшифровки данных.

Основным средством обеспечения безопасности VPN соединений является набор протоколов IPSec (IP Security), использующийся для обеспечения сервисов приватности и аутентификации на сетевом уровне модели OSI.

Гарантии целостности и конфиденциальности данных в IPsec обеспечиваются за счет использования механизмов аутентификации и шифрования которые, в свою очередь, основаны на предварительном согласовании сторонами информационного обмена так называемого "контекста безопасности" или политики безопасности — конкретных криптографических алгоритмов, алгоритмов управления ключевой информацией и их параметров. Спецификация IPsec предусматривает возможность поддержки сторонами информационного обмена различных протоколов и параметров аутентификации и шифрования пакетов данных, а также различных схем распределения ключей.

При этом существуют 2 режима функционирования IPSec.

Транспортный режим используется для шифрования только поля данных IP пакета. Его недостатком является отсутствие механизмов скрытия конкретных отправителя и получателя пакета, а также возможность проведения анализа трафика. Результатом такого анализа может стать информация об объемах и направлениях передачи информации, области интересов абонентов, расположение руководителей.

Туннельный режим предполагает шифрование всего пакета, включая заголовок сетевого уровня. Туннельный режим применяется в случае необходимости скрытия информационного обмена организации с внешним миром. При этом, адресные поля заголовка сетевого уровня пакета, использующего туннельный режим, заполняются устройством, выполняющим шифрование (например VPN-шлюзом), и не содержат информации о конкретном отправителе пакета.

Задача настройки IPsec сводится к тому, чтобы выбрать необходимые механизмы и алгоритмы и настроить соответствующим образом устройства участвующие в создании безопасного соединения.

Другими протоколами, поддерживающими шифрование и установление соединения являются: PPP (Point-to-Point Protocol) и его подвиды: PPPoE (Point-to-Point Protocol over Ethernet) и PPPoA (Point-to-Point Protocol over ATM).

PPPoE — это туннелирующий протокол, который позволяет передавать IP-пакеты через соединения Ethernet, но с программными возможностями PPP соединений. Это позволяет применять традиционное PPP-ориентированное программное обеспечение для настройки соединения, которое использует не последовательный канал, а пакетно-ориентированную сеть, чтобы организовать классическое соединение с логином, паролем для подключения к интернет. Также, IP-адрес по другую сторону соединения назначается только когда PPPoE соединение открыто, позволяя динамическое использование IP адресов.

# 10 ПРИКЛАДНОЙ УРОВЕНЬ

Прикладной уровень представляет собой набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые web-страницы, а также организуют свою совместную работу. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением. Существует очень большое разнообразие служб прикладного уровня и все они, в основном, решают практические задачи пользователей. Однако к данному уровню также относятся несколько служб и протоколов, необходимых для обеспечения работы сети. Рассмотрим их более подробно.

## 10.1 Система доменных имён

В операционных системах, которые первоначально разрабатывались для локальных сетей, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, применялись так называемые плоские имена, состоящие из последовательности символов, не разделенных на части. Для установления соответствия между символьными именами и МАС-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные операционные системы.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным. В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества составных частей. Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен (рис. 10.1) начинается с корня, обозначаемого точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети.

Рис. 10.1 — Древовидная структура доменных имён

В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяются друг от друга точкой. Например, в имени home.microsoft.com составляющая home является именем одного из компьютеров в домене microsoft.com. Совокупность имен, у которых несколько старших составных частей совпадают, образуют домен имен.

Разделение имени на части позволяет разделить административную ответственность за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного рис. 10.1, одна организация может нести ответственность за то, чтобы все имена с окончанием «ru» имели уникальную следующую вниз по иерархии часть. Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

Корневой домен управляется центральными органами Интернета, в частности организацией ICANN (Internet Corporation for Assigned Names and Numbers — Корпорация по управлению доменными именами и IP-адресами). Домены верхнего уровня назначаются для каждой страны, а также для различных типов организаций. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций, например, следующие обозначения:

* com — коммерческие организации;
* edu — образовательные организации;
* gov — правительственные организации;
* org — некоммерческие организации;
* net — сетевые организации.

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой делегированы полномочия по распределению имен доменов.

Доменная система имен реализована в Интернете, но она может работать и как автономная система имен в любой крупной корпоративной сети, которая хотя и использует стек TCP/IP, никак не связана с Интернетом.

Имя и IP-адрес не тождественны — один IP-адрес может иметь множество имён, что позволяет поддерживать на одном компьютере множество веб-сайтов. Обратное тоже справедливо — одному имени может быть сопоставлено множество IP-адресов. Компьютеры, имена которых относятся к одному и тому же домену, могут иметь абсолютно независимые друг от друга IP-адреса, принадлежащие различным сетям и подсетям. Для сопоставления доменного имени и IP-адреса используется служба доменных имён — DNS (Domain Name Service).

## 10.2 Схема работы DNS

Широковещательный способ установления соответствия между символьными именами и локальными адресами, подобный протоколу ARP, хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой способ разрешения символьных имен. Хорошей альтернативой широковещательной рассылке является применение централизованной службы, поддерживающей соответствие между различными типами адресов всех компьютеров сети.

В сетях TCP/IP соответствие между доменными именами и IP-адресами может устанавливаться средствами как локального хоста, так и централизованной службы.

На раннем этапе развития Интернета на каждом хосте вручную создавался текстовый файл с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP-адрес», однако по мере роста интернета файлы hosts.txt также увеличивались в объеме, и стало необходимым создание масштабируемого решения для разрешения имен. Таким решением является система доменных имён (Domain Name System) — DNS, поддерживаемая сегодня большинством сетевых устройств и операционных систем.

Для каждого домена имен создается свой DNS-сервер. На серверах применяют два подхода к распределению имен. В первом случае сервер может хранить отображения «доменное имя — IP-адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка ка этот сервер может превысить его возможности. Чаще используется другой подход, когда сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена.

Распределённая база данных имён интернета поддерживается с помощью иерархии DNS-серверов. Каждый DNS-сервер помимо таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых широко известны.

Существует две основные схемы сопоставления доменного имени и IP-адреса. В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

* 1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.
  2. DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени.
  3. DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и так далее, пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая процедура разрешения имени называется нерекурсивной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Эта схема загружает клиента достаточно сложной работой, и она применяется редко.

Во втором варианте реализуется рекурсивная процедура:

1. DNS-клиент запрашивает локальный DNS-сервер, то есть сервер, обслуживающий поддомен, которому принадлежит имя клиента.
2. Далее возможны два варианта действий:

* если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту (это может произойти, когда запрошенное имя входит в тот же поддомен, что и имя клиента, или когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше);
* если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в предыдущем варианте, а получив ответ, передает его клиенту, который все это время просто ждет его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, именно поэтому схема называется рекурсивной, или косвенной. Практически все DNS-клиенты используют рекурсивную процедуру.

Для ускорения поиска IP-адресов DNS-серверы широко применяют кэширование проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней.

Динамический DNS — технология, позволяющая информации на DNS-сервере обновляться в реальном времени, и (по желанию) в автоматическом режиме. Она применяется для назначения постоянного доменного имени устройству (компьютеру, сетевому накопителю) с динамическим IP-адресом. Это может быть IP-адрес, полученный по DHCP или с использованием PPP-соединения (например, при удалённом доступе через модем). Другие компьютеры в интернете могут устанавливать соединение с этим компьютером по доменному имени и даже не знать, что IP-адрес изменился.

Время устаревания для динамической записи делается очень маленьким (не более двух-трёх минут), иначе другие DNS-серверы поместят её в свой кэш, а когда она изменится — их клиенты долго будут получать устаревшую информацию. Динамический DNS также часто применяется в локальных сетях, где клиенты получают IP-адрес по DHCP, а потом регистрируют свои имена в локальном DNS-сервере.

## 10.3 Протоколы RDP и VNC

Remote Desktop Protocol (протокол удалённого рабочего стола) — протокол прикладного уровня, обеспечивающий удаленный доступ через сеть к рабочему столу компьютера под управлением ОС Windows.

В качестве графического вывода может выступать как точная копия экрана, передаваемая как изображение, так и команды на отрисовку графических примитивов (прямоугольник, линия, эллипс, текст и др.). Передача вывода с помощью примитивов является приоритетной для протокола RDP, так как значительно экономит трафик; а изображение передается лишь в том случае, если иное невозможно по каким-либо причинам.

Серверные версии Windows поддерживают одновременно два удаленных подключения и один локальный вход в систему, в то время как клиентские — только один вход (локальный или удаленный). Для разрешения удаленных подключений требуется включить удаленный доступ к рабочему столу в свойствах рабочей станции.

VNC — это система удаленного управления рабочим столом компьютера. Пользователь VNC клиента видит изображение рабочего стола VNC сервера и управляет им мышью и клавиатурой так же, как своим собственным компьютером.

VNC состоит из двух частей: клиента и сервера. Сервер — программа, предоставляющая доступ к экрану компьютера, на котором она запущена. Клиент (или viewer) — программа, получающая изображение экрана с сервера и взаимодействующая с ним по протоколу RFB (Remote FrameBuffer, удалённый кадровый буфер).

Протокол RFB позволяет клиенту и серверу «договориться» о том, какая кодировка будет использована. Самый простой метод кодирования, поддерживаемый всеми клиентами и серверами — «raw encoding» (рус. сырое кодирование), при котором пиксели передаются в порядке слева-направо, сверху-вниз, и после передачи первоначального состояния экрана передаются только изменившиеся пиксели. Этот метод работает очень хорошо при незначительных изменениях изображения на экране (движения указателя мыши по рабочему столу, набор текста под курсором), но загрузка канала становится очень высокой при одновременном изменении большого количества пикселей, например, при просмотре видео в полноэкранном режиме.

Протоколы RDP и VNC широко применяются в задачах администрирования локальных сетей, так как позволяют получить доступ к компьютеру или серверу независимо от его местоположения с минимальными затратами времени.

## 10.4 Протокол DHCP

Для нормальной работы сети каждому сетевому интерфейсу компьютера или маршрутизатора должен быть назначен IP-адрес. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса, для компьютера сводящейся, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие еще свободны.

При конфигурировании клиенту назначается не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например маска и IP-адрес шлюза по умолчанию, IP-адрес DNS-сервера, доменное имя компьютера и т. п. Даже при не очень большом размере сети на выполнение этой работы потребуется довольно много времени.

Протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP) автоматизирует процесс конфигурирования сетевых интерфейсов, обеспечивая отсутствие дублирования адресов за счет централизованного управления их распределением.

Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие конфигурационные параметры.

При этом сервер DHCP может работать в разных режимах, включая:

* ручное назначение статических адресов;
* автоматическое назначение статических адресов;
* автоматическое распределение динамических адресов.

Во всех режимах работы администратор при настройке DHCP-сервера указывает один или несколько диапазонов IP-адресов, причем все эти адреса относятся к одной сети, то есть имеют одно и то же значение в поле номера сети.

В ручном режиме администратор, помимо диапазона доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов. DHCP-сервер, пользуясь этой информацией, всегда выдаст определенному DHCP-клиенту один и тот же назначенный ему администратором IP-адрес (а также набор других конфигурационных параметров). Такой подход мало отличается от ручного задания сетевых настроек непосредственно на компьютере и, как следствие, практически не применяется.

В режиме автоматического назначения статических адресов DHCP-сервер самостоятельно без вмешательства администратора произвольным образом выбирает клиенту IP-адрес из множества доступных. Адрес дается клиенту в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес. Данный подход целесообразно использовать в локальных сетях организаций, состоящих преимущественно из стационарных компьютеров.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, называемое сроком аренды. Когда компьютер, являющийся DHCP-клиентом, удаляется из подсети, назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс.

Это дает возможность впоследствии повторно использовать один и тот же IP-адрес для назначения другому компьютеру. Таким образом, помимо основного преимущества DHCP — автоматизации рутинной работы по конфигурированию стека TCP/IP на каждом компьютере, режим динамического распределения адресов в принципе позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении IP-адресов. Такой подход подходит для использования в любых сетях, однако его особенности наиболее полно проявляются там, где количество и состав компьютерной техники постоянно меняются, например, для временного подключения к сети посетителей какого-либо заведения.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое компьютерная сеть?
2. Какие Вы знаете виды сетей?
3. Перечислите базовые сетевые топологии и их отличительные особенности.
4. В чём разница между технологиями коммутации каналов и коммутации пакетов?
5. Что такое мультиплексирование?
6. Перечислите уровни модели OSI.
7. Что такое интерфейс?
8. Что такое протокол?
9. В чём отличие модели DOD(TCP/IP) от модели OSI?
10. Что такое стек протоколов?
11. Назовите основные типы линий связи.
12. Назовите основные виды кабелей, применяемых при построении компьютерных сетей.
13. Что такое модуляция?
14. Что такое кодирование?
15. В каких линиях связи применяется модуляция, а в каких кодирование?
16. Что означает свойство самосинхронизации кодов?
17. Почему несмотря на то, что скорость распространения сигнала одинакова для всех линий связи (и равна скорости света), скорость передачи данных по ним разная?
18. Какое сетевое оборудование работает на физическом уровне?
19. Какие функции выполняются на подуровне MAC?
20. Как работает метод доступа к разделяемой среде CSMA/CD?
21. Насколько эффективен метод доступа к разделяемой среде CSMA/CD?
22. Что представляет собой MAC-адрес? Чему и кем он присваивается?
23. Какие функции выполняются на подуровне LLC?
24. Опишите принцип работы сетевого коммутатора (свитча)? В чём его отличие от концентратора (хаба)?
25. Какие преимущества имеются у технологии FDDI?
26. Перечислите наиболее распространённые стандарты Ethernet.
27. Каким образом протокол Gigabit Ethernet обеспечивают передачу данных со скоростью в 10 раз больше чем Fast Ethernet по тем же линиям связи?
28. В чём разница между плоской и иерархической схемами адресации? В каких случаях применяют каждую из них?
29. Из каких частей состоит IP-адрес?
30. В чём разница между классовой и безклассовой адресацией?
31. Что такое маска подсети?
32. Какие функции выполняет протокол ARP?
33. В чём отличия IPv6 от IPv4?
34. Что такое маршрутизация?
35. Как работают дистанционно-векторные протоколы маршрутизации?
36. Как работают протоколы маршрутизации, основанные на состоянии связей?
37. Что такое порт?
38. В чём разница между протоколами TCP и UDP?
39. Как работает алгоритм скользящего окна?
40. Что такое NAT? В каких случаях он применяется?
41. Какая сеть при трансляции адресов называется внутренней, а какая внешней?
42. Как работает перенаправление портов и для чего оно используется?
43. Что такое VPN?
44. Что такое туннелирование?
45. Что означает термин инкапсуляция применительно к сетевым технологиям?
46. Что такое домен шифрования?
47. Каким образом в компьютерных сетях осуществляется преобразование символьного названия (например адреса сайта) в числовой IP-адрес?
48. Каким образом обеспечивается уникальность доменных имён?
49. Для чего предназначены протоколы VNC и RDP?
50. Как работает протокол DHCP?

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. — СПб.: Питер, 2010. — 944 с.
2. Дуглас Камер Э. Сети TCP/IP Принципы, протоколы, структура. — М.: Вильямс, 2003. — 851 с.
3. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. — СПб.: Питер, 2012. — 960 с.
4. Гаранин М.В., Журавлёв В.И., Кунегин В.И. Системы и сети передачи информации: Учебное пособие. — М.: Радио и связь, 2003. — 336 с.
5. Головин Ю.А., Суконщиков А.А., Яковлев С.А. Информационные сети: Учебник для вузов. — М.: Академия, 2011. — 384 с.
6. Кузин А.В. Компьютерные сети (3-е издание). — М.: ФОРУМ: ИНФРА-М, 2011. — 192 с.
7. Алиев Т. И. Сети ЭВМ и телекоммуникации: учебное пособие. — СПб.: СПбГУ ИТМО, 2011. — 399 с.

Учебное издание

Галушка Василий Викторович

СЕТИ И СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Учебное пособие

Редактирование осуществлено авторами

Компьютерная обработка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

В печать 16.05.2016.

Объем 6,6 усл.п.л. Офсет. Формат 60x84/16.

Заказ № 152 Тираж 80 экз. Цена свободная

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия:

344000, г. Ростов-на-Дону, пл. гагарина, 1.