МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. БАУМАНА

КАФЕДРА ИУ-3
ИНФОРМАЦИОННЫЕ СИСТЕМЫ И
ТЕЛЕКОММУНИКАЦИИ

ТИХОМИРОВА Е.А.

Курс лекций по дисциплине

ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

Москва, 2015

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Основной проблемой во все времена была проблема адресной доставки информации получателю. И когда зародилась первая почтовая служба, были предприняты попытки решения вопросов технологии адресации сообщений, которые бы гарантировали доставку сообщения именно тому, кому они предназначаются. Например, указанный на конверте письма адрес "На деревню, дедушке" тоже является адресом, но не гарантирует, что письмо будет доставлено именно в ту деревню, и именно тому дедушке, кому оно было написано. Так как в передаче почты участвуют многие составляющие (курьеры, пункты обработки, почтальоны, доставляющие корреспонденцию до адресата и многие другие), то каждый из них должен ясно представлять в каком направление требуется доставка. Это накладывает необходимость четкой структуризации адреса назначения, а также в случае необходимости ответного сообщения, то и адреса отправления.

На сегодняшний день жизнь протекает в информационном обществе. Процесс обмена информацией прочно вошел в жизнь практически каждого человека, и большинство современных людей проводит в системах обмена информации (система обмена электронной почтой, облачные хранилища данных, сервера хранения данных с общим доступом и прочее) достаточно много времени. Любой медицинский прибор, получающий данные от подключенных к пациенту датчиков, должен обязательно знать, от какого датчика получены данные, и в соответствии с этим направить их на рассмотрение именно тому доктору, который на них специализируется. Также невозможно представить себе полет космического аппарата без передачи ему сигналов и команд управления, с целью обеспечения его курса, гарантирующих, что он долетит именно до той планеты, к которой и предназначалось. Таким образом, процесс передачи данных плотно вошел в нашу жизнь во всех ее областях, даже в тех, на которые мы в повседневной жизни не обращаем внимание.

Обмен данными между техническими устройствами во многом схож с обычной почтовой отправкой писем. Например, письмо должно быть изначально написано, упаковано в конверт, опущено в ящик, перевезено на транспорте, опущено в ящик получателя, а затем им прочитано. Каждое устройство, передающее данные, должно четко знать от кого оно поступило, для того, чтобы отправить его обратно в случае невозможности передачи по назначению, а также знать кому они предназначались, что оправить именно в ту деревню и именно тому дедушке, которому надо.

Именно четкая структуризация адреса обеспечивает гарантированную доставку данных получателю, а технические возможности аппаратуры дают возможность сделать эту отправку

быстрой.

Совокупность всех этих составляющих и представляет рассматриваемую в данном курсе инфокоммуникационную систему.

Цель изучения дисциплины – приобретение теоретических знаний и практических навыков в области инфокоммуникационных систем, технологий локальных сетей, маршрутизации, протоколов и сервисов Интернет.

СТРУКТУРА ДИСЦИПЛИНЫ

Дисциплина «Инфокоммуникационные системы и сети» включает три модуля, изучаемые последовательно в течение семестра.

Модуль 1 «Основы инфокоммуникационных систем и технологии локальных сетей» посвящен изучению основных понятий, моделей взаимодействия систем и стандартам локальных сетей.

Модуль 2 «Маршрутизация в сетях» содержит принципы адресации в сетях. В данном модуле рассматриваются основные принципы маршрутизации, а так же базовые протоколы внутренней маршрутизации.

Модуль 3 «Протоколы и сервисы Интернет» содержит информацию о взаимодействии систем для предоставления услуг пользовательским приложениям.

МОДУЛЬ 1

1. ОСНОВЫ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИИ ЛОКАЛЬНЫХ СЕТЕЙ

Работа сетевого инженера связана с глубинными знаниями происходящих процессов внутри инфокоммуникационной системы. Без понимания принципов построения всей системы в целом и взаимодействия отдельно взятых участков проектирование, развертывание и устранение неисправностей невозможно.

В силу описанных выше причин первый модуль дисциплины «Инфокоммуникационные системы и сети» посвящен общим принципам взаимодействия открытых систем, передачи информации по каналам связи, а так же технологиям локальных сетей.

Цели (задачи) изучения 1 модуля

После изучения модуля
«Основы инфокоммуникационных систем и технологии локальных сетей» Вы сможете:

- **определить структуру и принцип действия инфокоммуникационных систем;**
- **объяснить понятие инфокоммуникационной сети и описать принцип действия инфокоммуникационной сети;**
- **перечислить дисциплины передачи информации между узлами и объяснить принцип действия каждой дисциплины;**
- **привести несколько моделей взаимодействия открытых систем, подробно описав функционал**
- **дать описание технологий локальных сетей и для заданной топологии продемонстрировать способы обмена данными между конечными узлами.**

Методика проработки и освоения материала 1 модуля:

1 неделя - получение учебного материала модуля, вводная лекция, постановка целей и задач, представление основных ресурсов Интернета для расширения информации по теме модуля.

2 неделя – изучение ключевых понятий, терминов и классификаций.

3 неделя – изучение принципов взаимодействия систем на основе модели взаимодействия открытых систем OSI/ISO и стека протоколов TCP/IP.

4 неделя – изучение технологий локальных сетей на примере семейства Ethernet.

5 неделя - контрольное мероприятие по оценке освоения модуля: письменное задание по проработанному материалу.

Задачи, выносимые на практические занятия:

- изучить функционал и принципы действия оборудования, применяемого в локальных сетях;
- на примере заданной одноранговой локальной сети продемонстрировать способы обмена данными между конечными узлами.

1.1. Основы инфокоммуникационных систем

В первом разделе модуля «Основы инфокоммуникационных систем и технологии локальных сетей» приведены основные понятия, структура и принципы действия инфокоммуникационных систем.

1.1.1. Структура и принцип действия инфокоммуникационных систем

Инфокоммуникационные системы – совокупность средств и методов, обеспечивающих информационные и телекоммуникационные процессы.

Задачей подобных систем является передача информации от одного удаленного источника инфорамции к другому (рис. 1). Источником информации может являться компьютер, на котором запущено пользовательское приложение, смартфон, планшет, а так же датчики, считывающие показания каких-либо объектов. Задачей инфокоммуникационных систем так же является предоставление информации, удаленно хранящейся на сервере.

Рис. 1. Инфокоммуникационная система

Для организации связи устройств, передающих информацию, необходимо в первую очередь организовать канал связи между ними. По методам передачи данных каналы связи можно подразделить на

- симплексный;
- полудуплексный;
- дуплексный.

Симплексный метод передачи данных подразумевает передачу информации только в одном направлении. Примером подобной передачи данных может служить теле- и радиовещание, когда нет необходимости передавать какие-либо данные на передающую станцию.

Полудуплексный метод передачи информации – передача данных осуществляется в обоих направлениях попеременно, то есть в зависимости от аппаратного решения одновременная передача либо физически невозможна, либо приводит к невозможности восприятия информации. Примером являются переговоры по рации, когда нажатие специальной кнопки переводит рацию в режим передачи, а пользователи рации сообщают об освобождении канала связи специальным кодовым словом (например, «прием»).

Дуплексный режим передачи данных обеспечивает передачу в оба направления одновременно. Данный подход может быть реализован несколькими методами. Например, реализация одного канала связи, который будет исходящим для первого источника и входящим для второго, и второго канала связи – исходящим для второго и входящим для первого. Другим методом может служить применение одного канала связи, при использовании которого принимающая сторона вычитает свой отправленный сигнал из принимаемого и в виде разницы получает принимаемый.

Для обеспечения передачи данных по организованным каналам связи в инфокоммуникационных системах применяются протоколы. **Протокол** – набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами. По сути задача протокола – предоставить передаваемые пользователем (или например, датчиком) данные в приемлемой форме и сопроводить служебной информацией, которая позволит корректно передать по каналам связи и интерпретировать данные на принимаемой стороне.

Существует неколько классификаций протоколов. Основополагающая с точки зрения сетевого инженера (не разработчика) является деление на протоколы с установлением соединения и без установления соединения (рис. 2).

При применении **протоколов с установлением соединения** (рис. 2а) устройство, передающее данные, непосредственно перед передачей устанавливает соединение с принимающей стороной. На данном этапе осуществляется договоренность обеих сторон, учавствующих в передаче, о параметрах передачи.

Достоинствами данного метода является надежность передачи, поскольку посредством данного этапа у передающей стороны появляется уверенность, что принимающая сторона готова к приему, так как она осуществляет подтверждение получения данных.

Недостатком является временные затраты на передачу служебной информации.

Рис. 2. Проколы с установлением соединения (а) и без установления соединения (б)

С **протоколами без установления соединения** (рис. 2б) устройство, передающее данные, сразу отправляет данные в канал связи, не связываясь с устройством-приемником. Таким образом, не тратится время на установление соединения с принимающей стороной. Так же в силу этого обстоятельства не происходит подтверждения доставки данных, что является и достоинством и недостатком одновременно данного типа протоколов. Достоинство – не затрачивается дополнительное время на пересылку служебной информации. Недостаток – ненадежная передача данных. Считать это только как положительной или как отрицательной стороной подобных протоколов стоит в зависимости от поставленной задачи: если необходимо передать данные в режиме реального времени (потоковое видео или аудио) стоит отнести данную характеристику к достоинствам, потому что не будет происходить ненужных задержек, а потеря нескольких отсчетов может быть компенсирована на принимающей стороне за счет восстановления на основе аппроксимации.

1.1.2. Понятие и принцип действия инфокоммуникационной сети

Для обеспечения каналов связи, по которым передаются данные, необходимо в первую очередь продумать физический аспект – необходимые для передачи средства и их физическое соединение, обеспечивающее информационные и телекоммуникационные процессы. Или другими словами – оборудование, которое способно корректно передавать данные от одного удаленного устройства к другому.

Инфокоммуникационная сеть – набор узлов, каким-то образом связанных между собой. Под узлами понимается не только конечные устройства, такие как компьютеры,

планшеты, сервера, принтеры или датчики, но и, в первую очередь, промежуточные устройства (сетевые устройства), обеспечивающие связь конечных устройств между собой.

Сети подразделяются на **локальные** и **глобальные**. **Локальные сети** (Local Area Network, LAN) располагаются на относительно небольшом пространстве в пределах нескольких километров от узла к узлу. **Глобальные** (Wide Area Network, WAN) **сети** располагаются на больших пространствах, по максимуму включая всю планету.

Способ организации связей называется **топологией** сети. То есть топология – конфигурация графа, вершинами которого являются сетевые устройства, а ребрами – связи между ними.

Топологии подразделяются на **физические** и **логические**. Физическая топология описывает физическое соединение сетевых устройств, а логическая – принцип организации пересылки данных.

Независимо от того, физическая или логическая, существует пять основных разновидностей топологий.

1) Шина (рис. 3)

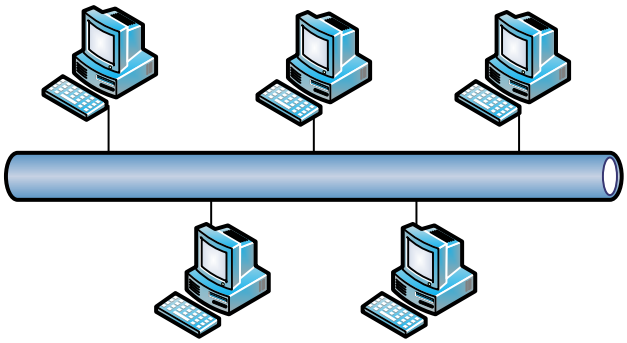


Рис. 3. Топология типа «шина»

Суть топологии заключается в том, что существует общая разделяемая среда передачи данных для всех устройств. Недостатками данной топологии являются конкуренция за общую полосу пропускания (одновременная передача нескольких устройств в данном случае невозможна) и уязвимость всей сети по причине выхода из строя проводника (или в некоторых случаях устройства), соединяющего все устройства сети. При этом данные, посланные одним устройством, будут получены всеми, а обработаны только тем устройством, которому они предназначаются.

2) Кольцо (рис. 4)

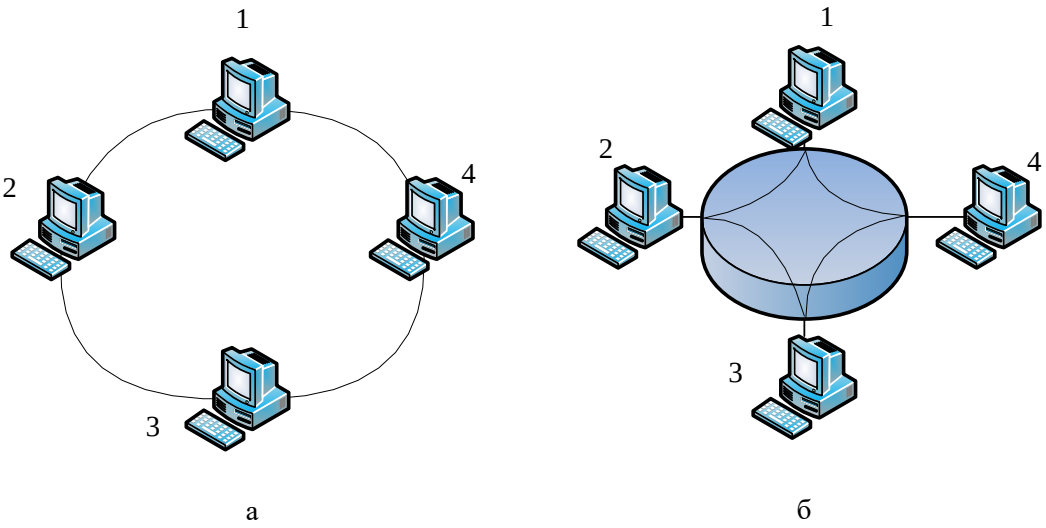


Рис. 4. Топология типа «кольцо»

На рис. 4 представлена физическая реализация топологии типа «кольцо» (б) и логическая организация связи устройств (а) в топологии. В соответствии с логической топологией передача осуществляется только в одну сторону: либо по часовой стрелке, либо против часовой стрелки. Таким образом, для передачи данных, например, от устройства 3 до устройства 2 при условии передачи против часовой стрелки, необходимо, чтобы данные прошли почти все кольцо. То есть устройства могут передавать данные только тем устройствам, которые находятся ниже по потоку, а получать – только от устройств, находящихся выше по потоку. Физически данная топология реализуется другим видом топологии:

3) Звезда (рис. 5)

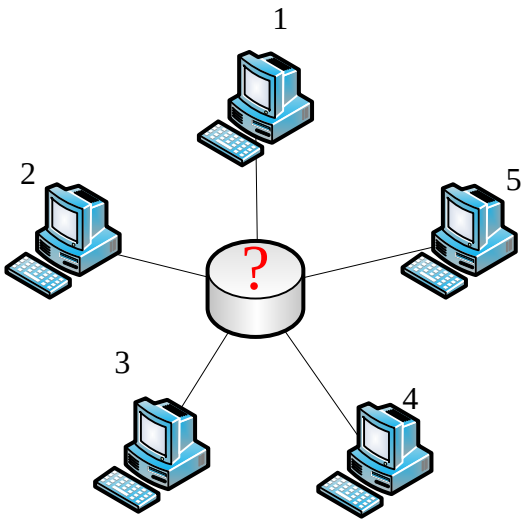


Рис. 5. Топология типа «Звезда»

Несмотря на то, что данная топология реализуется физически как топология типа «кольцо», логика передачи данных отлична. При топологии типа «звезда» подразумевается, что у каждого устройства есть своя полоса пропускания (в отличие от топологии типа «шина») и передавать данные возможно не по кольцу, а непосредственно устройству, которому предназначаются данные (в отличие от топологии типа «кольцо»). Данный функционал предоставляет определенные разновидности сетевого оборудования, соединяющие конечные устройства. Недостатком подобной топологии является уязвимый центральный узел, который и обеспечивает всю логику кольца.

4) Полносвязная (рис. 6)

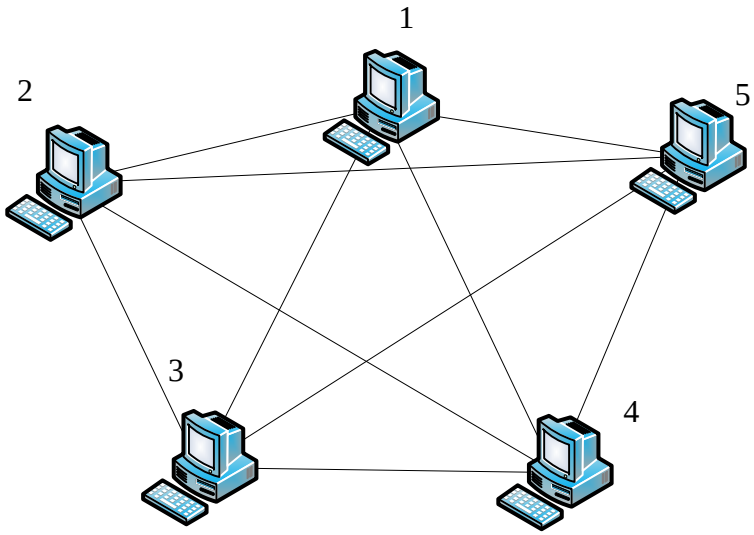


Рис. 6. Полносвязная топология

Полносвязная топология подразумевает связь каждого устройства с каждым. Неоспоримым достоинством данной топологии является неуязвимость: при отказе одной или даже нескольких связей сеть не прекращает функционирование. Недостатками являются плохая расширяемость и сложность физической реализации.

5) Частичносвязная

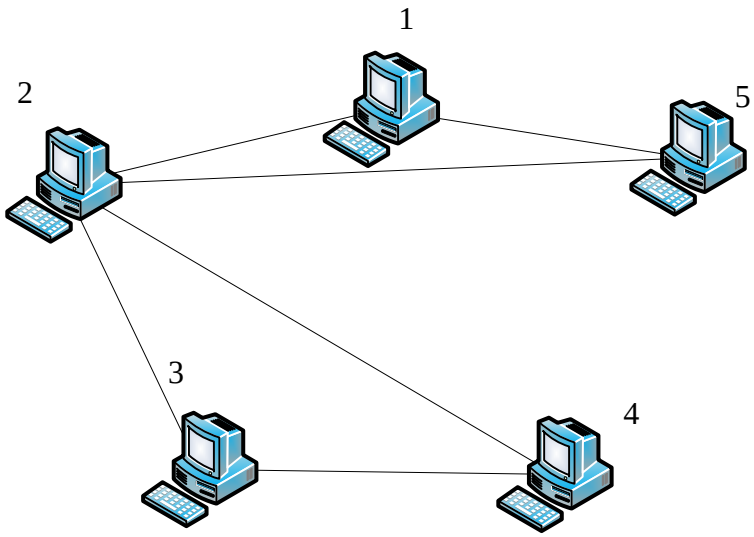


Рис. 7. Частичносвязная топология

Для минимизации недостатков полносвязной топологии прибегают к частичносвязной топологии: связь с одним (наиболее важным) узлом в топологии организуется от всех устройств, а далее устройства соединяются между собой по определенной логике, но не все со всеми.

1.1.3. Дисциплины передачи информации между узлами

Задача разработки инфокоммуникационной сети заключается не только в том, чтобы соединить правильно устройства, но и в обеспечении корректного доступа устройств к среде передачи данных. Для решения поставленной задачи в первую очередь необходимо понимать методы доступа устройства к физической среде, т.е. при каких условиях устройству разрешено передавать данные по предоставленной среде передачи данных.

Правила доступа к физической среде определяют дисциплины передачи информации между узлами, которые по сути регламентируют правила опроса узлов. Дисциплины передачи информации подразделяются на два класса:

- иерархические,
- одноранговые.

При **иерархической** дисциплине передачи данных часть узлов определяют режимы работы других узлов. Иными словами, существует первичный узел, которому подчиняются остальные (вторичные) узлы. При подобном подходе может существовать три сценария работы, два из которых представлены на рис. 8, а третий основывается на первых двух.

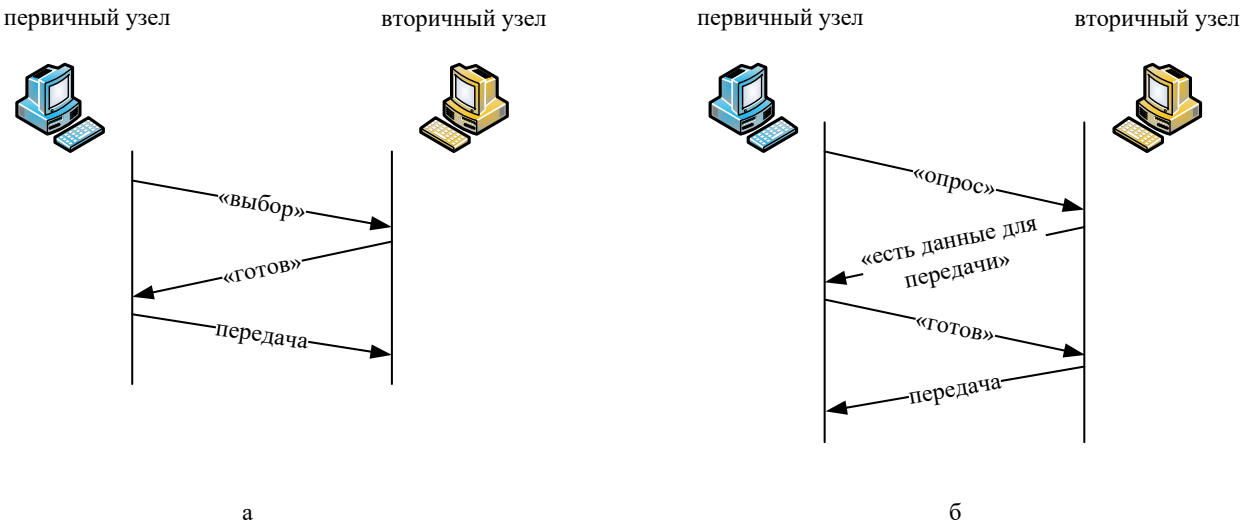


Рис. 8. Иерархическая дисциплина передачи информации между узлами: а – у первичного узла есть данные для передачи, б – у вторичного узла есть данные для передачи

1) У первичного узла есть данные для передачи вторичному узлу (рис. 8а). В этом случае первичный узел должен сообщить вторичному, которому необходимо передать данные, что необходимо передать данные, и дождаться подтверждения от вторичного. Только после получения подтверждения от вторичного узла первичный узел начнет передачу данных.

2) У вторичного узла есть данные для передачи первичному узлу (рис. 8б). Вторичный узел должен дождаться получения команды «опрос», смысл которой заключается в вопросе первичного узла: «У Вас есть данные для передаче мне?». После получения данной команды вторичный узел отправляет информацию о том, что данные для передачи присутствуют в буфере, дожидается команды «готов» от первичного узла, сигнализирующей о готовности в приеме данных, и отправляет данные первичному узлу.

3) У вторичного узла есть данные для передачи другому вторичному узлу. При выполнении данного сценария вторичный узел передает данные первичному узлу в соответствии со вторым сценарием (рис. 8б), затем первичный узел идентифицирует получателя данных, после чего отправляет в соответствии с первым сценарием (рис. 8а) данные вторичному узлу-получателю.

При **одноранговой** дисциплине передачи данных все узлы равны между собой. Это подразумевает в теории одновременные передачи несколькими узлами по одной среде, что вызывает коллизии, то есть наложение двух и более сигналов друг на друга. В случае возникновения **коллизии** в канале связи выделение на принимаемой стороне полезного сигнала не предусмотрено технологиями, поэтому существует несколько алгоритмов, регламинтирующих порядок передачи узлами инфокоммуникационных сетей.

- Случайный доступ к каналу.
- Детерминированный метод.

Логически **случайный доступ к каналу** иллюстрирует топология типа «шина» (рис. 9), когда сигнал распространяется по всей среде передачи данных во всех направлениях и при условии передачи данных несколькими станциями одновременно возникает коллизия.

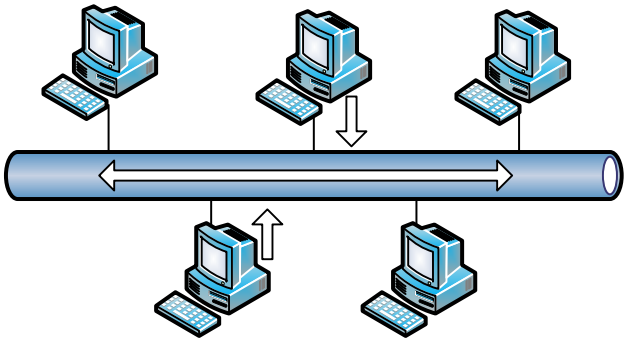


Рис. 9. Случайный доступ к каналу

Для борьбы с коллизиями при подобном алгоритме доступа применяется множественный доступ с контролем несущей и обнаружением коллизий (CSMA/CD – Carrier Sense Multiple Access with Collision Detection). Суть данного метода заключается в проверке канала связи на предмет наличия передачи данных по нему. Такая проверка называется контролем несущей, то есть контроль сигнала, передаваемого по каналу связи на предмет «переноса» им данных. Таким образом, метод CSMA/CD сводится к следующему:

1) прослушивание несущей на предмет передачи по ней данных;
2) передача данных от конечного узла при условии, что несущая свободна;
3) если возникает коллизия, станция, обнаружившая коллизию, передает сигнал специального вида (jam-сигнал), и передача данных прекращается всеми станциями;
4) при обнаружении коллизии на станциях, передающих данные, запускается таймер на случайный промежуток времени, по истечении которого станции снова прослушивают несущую и пытаются осуществить передачу данных.

При **детерминированном методе** применяется логическая топология типа «кольцо». При этом каждому узлу отводится фиксированный промежуток времени для передачи данных, которым он может воспользоваться. Осуществляется этот алгоритм путем перемещения по кольцу специального «маркера», захватив который станция может поместить в него данные и передать адресату.

Таким образом, если при случайном доступе к каналу связи возникновение коллизий возможно и существует метод их обнаружения и коррекции передачи, то при детерминированном методе коллизии возникнуть в принципе не могут.

1.1.4. Модели взаимодействия открытых систем

При обмене данными между конечными устройствами по сети подразумевается, что на конечных устройствах всегда есть некоторые приложения, обменивающиеся данными. Отсюда возникает понимание, что на каждом узле есть необходимость реализации механизмов, которые не только доставят данные до конкретного узла, но и предоставят данные в необходимом приложению виде. Первоначально аппаратное и программное обеспечение были собственностью поставщиков и не работали на оборудовании различных производителей. Для использования аппаратного и программного обеспечения от различных поставщиков было принято решение применения многоуровневого подхода при проектировании систем обмена данными. При этом четко должны быть определены правила взаимодействия между каждым уровнем. Таким образом, одни поставщики могли производить аппаратное и программное обеспечение аппаратных технологий, другие – программное обеспечение для операционных систем, контролирующих обмен данными между узлами.

1.1.4.1. Эталонная модель взаимодействия открытых систем (OSI/ISO)

В 1984 году Международной организацией по стандартам была предложена семиуровневая модель взаимодействия систем передачи данных. По сути модель OSI/ISO (Open System Interconnection/ International Organization for Standardization) содержит набор стандартов, гарантирующих более высокую степень совместимости решений различных поставщиков.

Данная модель состоит из семи уровней (рис. 10): физический, канальный, сетевой, транспортный, сеансовый, представлений и приложений. Каждый из перечисленных уровней обеспечивает определенный функционал в системе передачи данных и ниже лежащий уровень предоставляет услуги выше лежащему уровню.

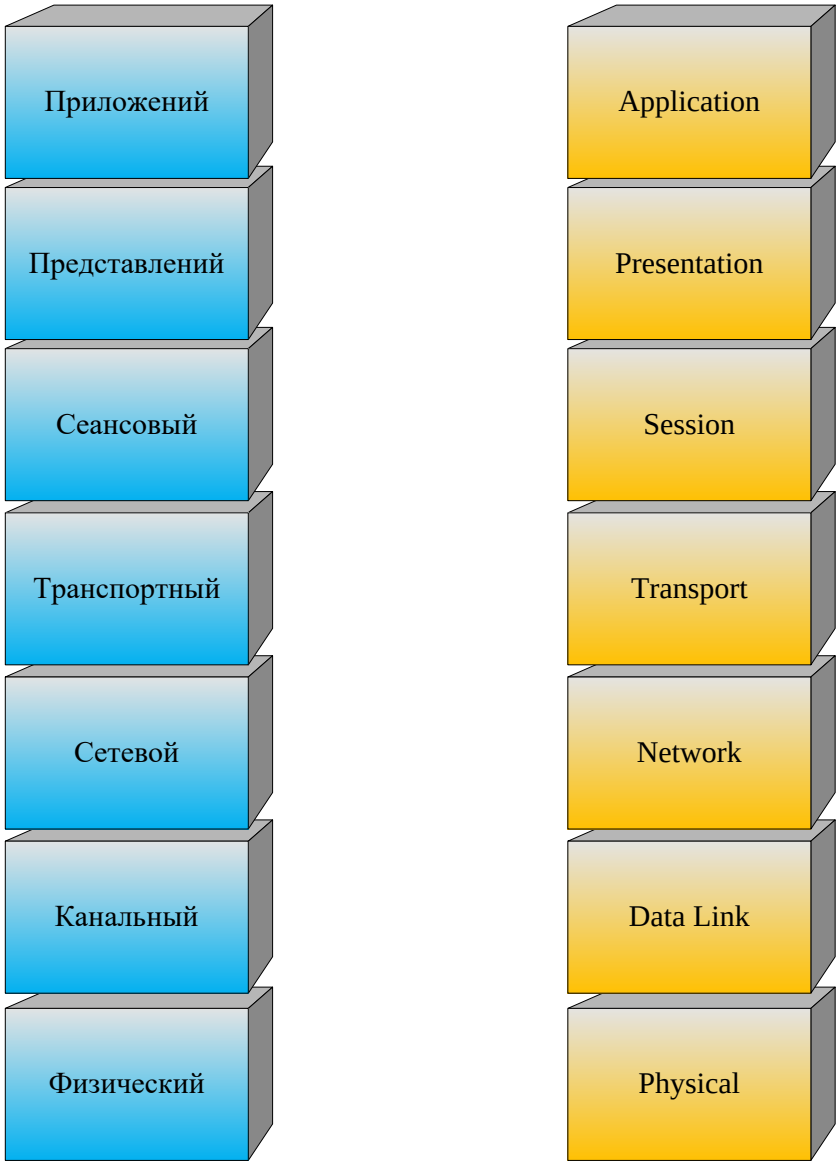


Рис. 10. Модель взаимодействия открытых систем OSI/ISO

На **физическом уровне** определены электрические, механические, процедурные и функциональные характеристики активации, поддержки и отключения физического канала между конечными системами. Примерами могут служить уровень напряжения, синхронизация изменения напряжений, физическая скорость передачи данных, максимальное расстояние передачи данных. Устройствами, работающими на данном уровне, то есть устройствами, предоставляющими описанный выше функционал, являются

- повторитель (ретранслятор) – устройство, усиливыающее сигнал для передачи на более дальнее расстояние;
- концентратор (hub) – устройство, усиливающее и в некоторых модификациях восстанавливающее сигнал для передачи на более дальнее расстояние. Отличительной особенностью концентратора от повторителя является также наличие более двух портов;

- медиаконвертор – устройство, преобразующее сигнал на границе двух сред (например, оптических и медных проводов).

На **канальном уровне** определяется формат данных для передачи и методы контроля доступа к физическим средам. Также данный уровень включает функции обнаружения и коррекции ошибок для обеспечения надежной передачи данных. Устройствами, работающими на данном уровне, являются

- коммутатор (switch),
- мост (bridge).

Оба перечисленные выше устройства предназначены для соединения **сегментов** сети. Сегмент сети – участок сети, устройства которого делят общую полосу пропускания.

Сетевой уровень обеспечивает связь и выбор пути между двумя конечными устройствами, которые могут находиться в сетях, географически удаленных друг от друга. Примером может служить маршрутизатор (router) – устройство, работающее на этом уровне и предназначенное для пересылки пакетов данных между различными сетями.

На **транспортном уровне** выполняется сегментация данных от передающего узла и реорганизация данных в поток на принимающем узле. Вместе с этим решаются задачи, связанные с надежностью передачи данных между узлами: создается, поддерживается и корректно завершается виртуальный канал связи между конечными узлами. Также на данном уровне предусмотрена функция обнаружения и коррекции ошибок.

Транспортный уровень является границей между протоколами приложений и протоколами потока данных, скрывая детали передачи данных от верхних уровней. Протоколами, работающими на данном уровне, являются

- протокол управления передачей (TCP – Transmission Control Protocol),
- протокол пользовательских датаграмм (UDP – User Datagram Protocol).

На **сеансовом уровне** выполняется создание, управление и завершение сеансов между двумя конечными узлами при обмене данными, осуществляется синхронизация диалога между уровнями представлений двух узлов и управлением обменом данными между ними. Примером протокола, работающего на данном уровне является протокол проверки подлинности (PAP – Password Auhtentication Protocol)

Уровень представлений гарантирует, что сведения, переданные на прикладном уровне одной системы, могут быть корректно распознаны на прикладном уровне другой системы. То есть информацию, полученную с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. Пример: видеокодек H.264, аудиокодек G.726 и др.

Уровень приложений предоставляет сетевые услуги приложениям. Примерами протоколов, работающих на данном уровне, являются

- протокол передачи гипертекста (HTTP – Hyper Text Transfer Protocol),
- протокол почтовых сообщений (POP – Post Office Protocol),
- простой протокол передачи почты (SMTP – Simple Mail Transfer Protocol),
- протокол передачи файлов (FTP – Fail Transfer Protocol).

1.1.4.2. Обмен данными между узлами

При обмене данными между узлами протоколы на каждом уровне обмениваются пакетами данных (PDU – Packet Data Unit). Общий вид подобных пакетов данных представлен на рис. 11. Стоит отметить, что добавление хвостовика необязательно, и он присутствует не во всех протоколах.

Рис. 11. Общий вид pdu

На канальном, сетевом и транспортном уровнях данные пакеты получили свои названия (рис. 12). Соблюдая терминологию, появляется представление о каком уровне модели взаимодействия открытых систем идет речь.

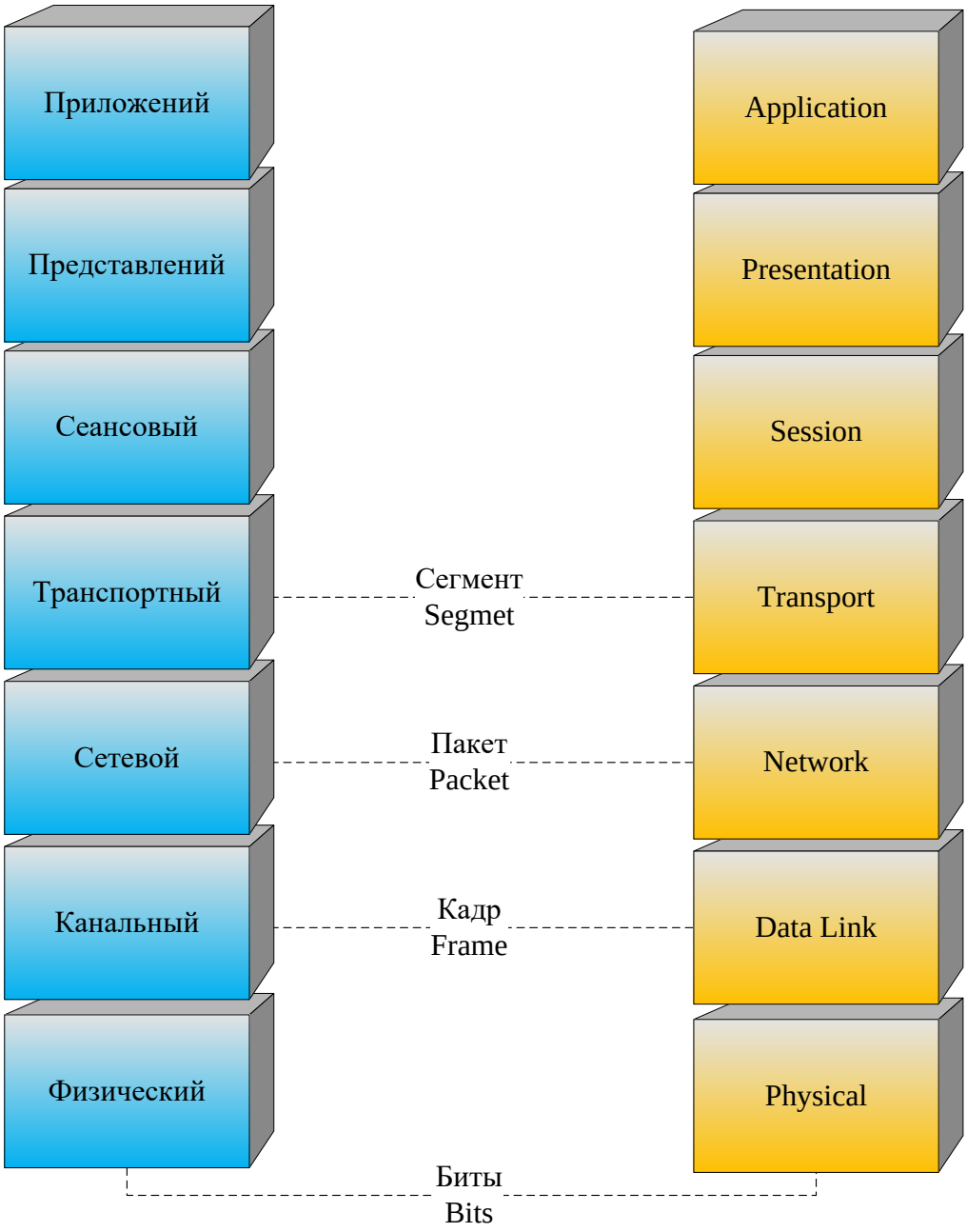


Рис. 12. Обмен данными между узлами

В заголовке и хвостовике содержится служенная информация, позволяющая реализовать функционал каждого из уровней модели взаимодействия открытых систем. За содержимое и правильную интерпретацию служебной информации отвечают протоколы. Таким образом, при передаче данных на передающей стороне данные, передаваемые приложением, спускаясь вниз по всем уровням модели взаимодействия открытых систем, «обрастают» служебной информацией на каждом уровне (рис. 13). Данный процесс называется **инкапсуляцией**. Таким образом, данные на каждом уровне модели содержат не только данные, передаваемые приложением, но и служебную информацию (заголовок и хвостовик), сформированную на вышележащем уровне. Поэтому в поле «Данные» на канальном уровне будут в общем случае

располагаться данные, передаваемые приложением, заголовки и хвостовики пяти вышележащих уровней (рис. 14). Но при этом для канального уровня эти заголовки и хвостовики будут ничем иным как данными, которые необходимо передать по каналу связи и интерпретировать их канальный уровень будет не в состоянии, потому что каждый уровень взаимодействует с заголовком и хвостовиком только своего уровня.

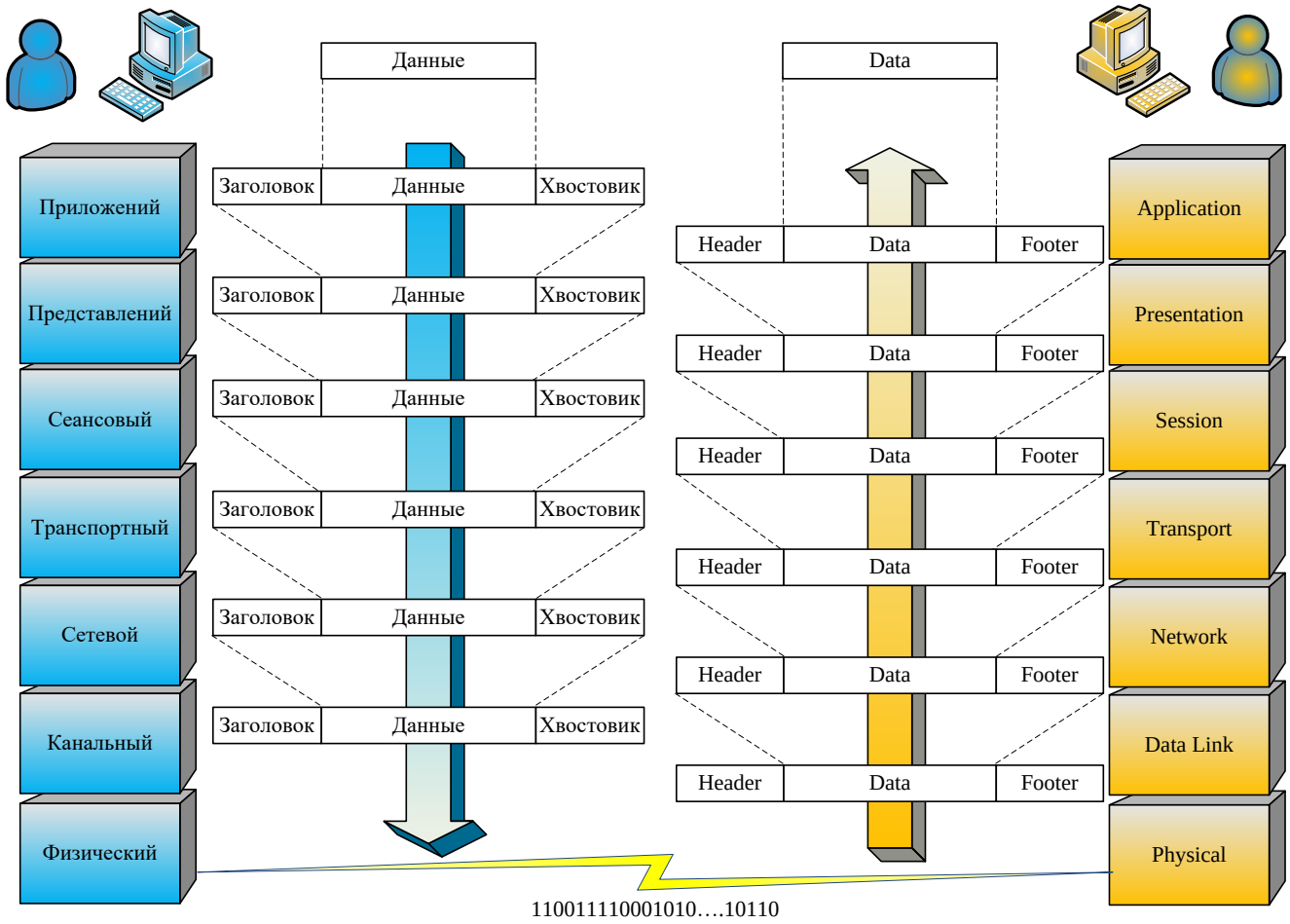


Рис. 13. Инкапсуляция данных на передающей стороне и деинкапсуляция данных на принимающей стороне

Рис. 14. Кадр

Процесс, обратный инкапсуляции и происходящий на принимающей стороне, называется **деинкапсуляцией** и представляет собой процесс отбрасывания заголовка и хвостовика текущего уровня и предоставления содержимого поля «Данные» вышележащему уровню. Перед отбрасыванием служебной информации в виде заголовка и хвостовика на текущем уровне происходит анализ содержимого служебной информации. На этапе анализа,

например, выявляется адресат данного pdu. Если адрес назначения не совпадает с адресом узла, получившего данное pdu на канальном уровне, принимается решение об уничтожении кадра, и дальнейшая деинкапсуляция не происходит. Если, например, подобное выявляется на сетевом уровне – дальнейшая деинкапсуляция также не происходит, но пакет не уничтожается, а в соответствии с функционалом, предоставляющимся сетевым уровнем происходит поиск маршрута, по которому необходимо отправить данный пакет конечному адресату.

По сути, процессы инкапсуляции и деинкапсуляции напоминают процесс отправки посылки по почте. Сначала формируется то, что необходимо отправить, после запаковывается (инкапсулирует), на упаковке указывается адрес получателя, и посылка относится на почту или отдается курьеру. После чего посылка доставляется до конечного адресата, и на приемной стороне при условии совпадения адреса, фамилии, имени и отчества адресата с указанными на посылке, адресат вскрывает посылку (деинкапсулирует) и вынимает содержимое.

1.1.4.3. Стек протоколов TCP/IP

Самой распространенной на данный момент моделью взаимодействия открытых систем является стек протоколов TCP/IP, получивший название от наиболее важных протоколов семейства: TCP и IP (Internet Protocol, межсетевой протокол). Данная модель была разработана по заказу Министерства обороны США, поэтому данную модель также называют моделью DoD (Department of Defence). Структура стека, так же как и модель OSI/ISO содержит многоуровневую структуру и представлена на рис. 15..

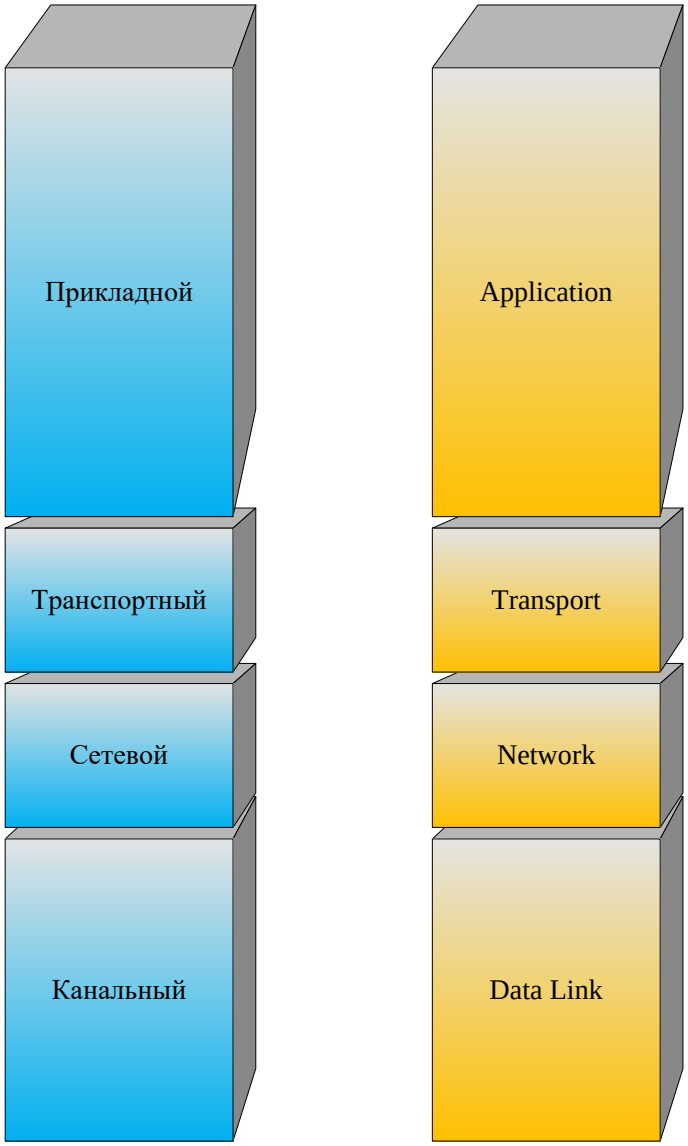


Рис. 15. Стек протоколов TCP/IP

В стеке протоколов TCP/IP реализован тот же функционал, что и в модели OSI/ISO, различие только в уровнях. Сравнение структур модели OSI/ISO и стека протоколов TCP/IP проиллюстрировано на рис. 16.

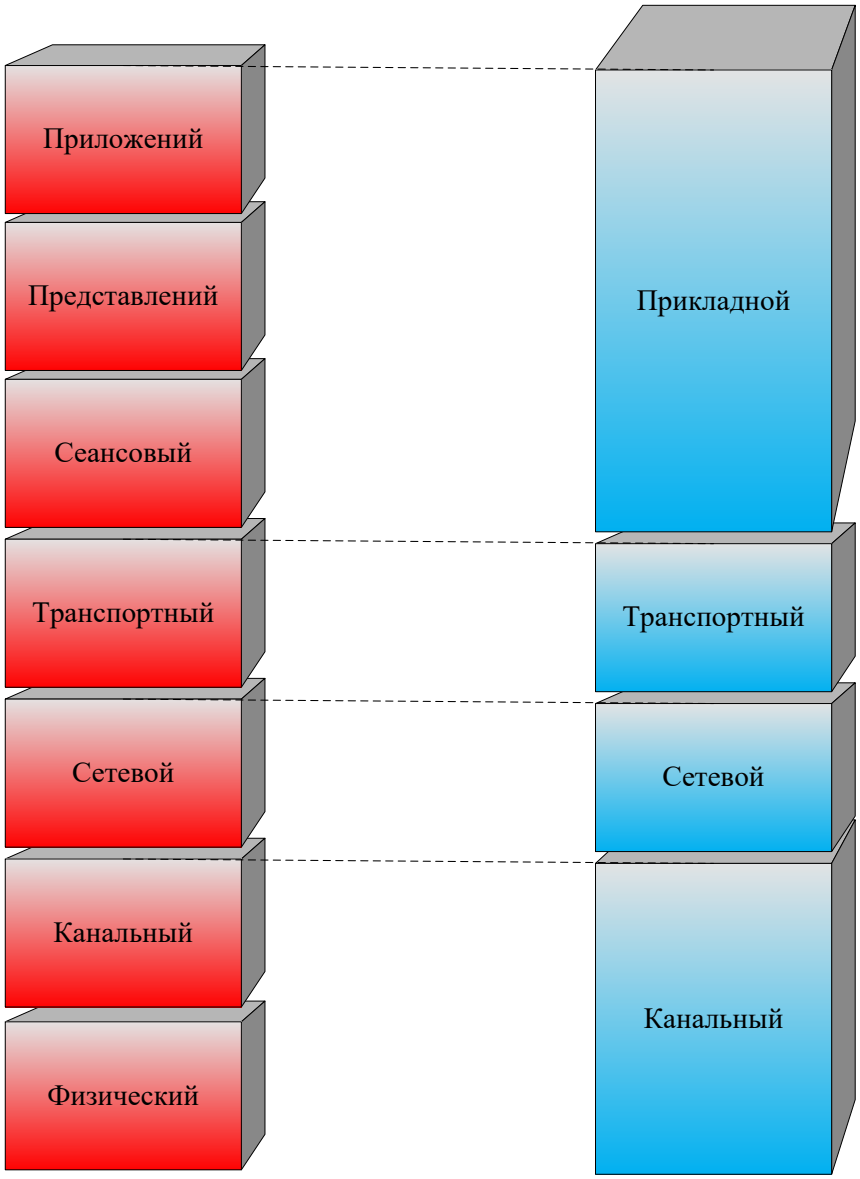


Рис. 16. Сравнение модели OSI/ISO и стека протоколов TCP/IP

Канальный уровень стека протоколов TCP/IP обеспечивает функционал физического и канального уровней модели OSI/ISO. Сетевой и транспортный уровни полностью идентичны в обеих моделях взаимодействия открытых систем. Прикладной уровень предоставляет функционал сеансового уровня и уровней представлений и приложений модели OSI/ISO.

Обмен данными в стеке TCP/IP идентичен обмену данными в модели OSI/ISO: так же формируются pdu на каждом уровне, так же происходит инкапсуляция на передающей стороне и деинкапсуляция на принимающей стороне.

1.1.4.4. Стандарт локальных сетей IEEE 802.2

Стандарт локальных сетей IEEE (Institute of Electrical and Electronics Engineers, Институт инженеров по электротехнике и электронике) определяет методы доступа к каналу связи.

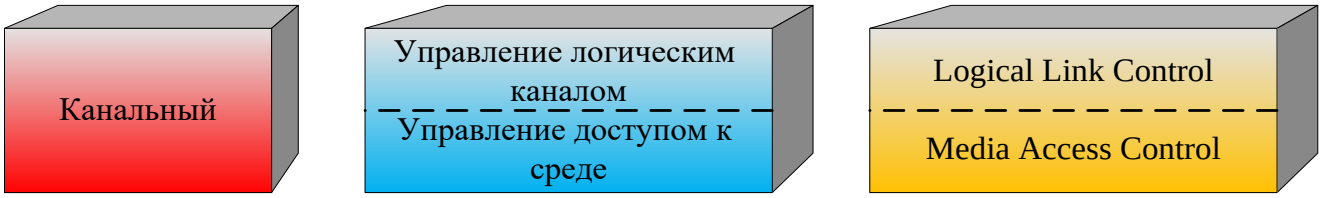


Рис. 17. Стандарт локальных сетей IEEE 802.2

В соответствии с данным стандартом канальный уровень подразделяется на подуровень управление логическим каналом (LLC – Logical Link Control) и подуровень управления доступом к среде (MAC – Media Access Control).

Подуровень LLC обеспечивает независимость канального уровня от существующих технологий. Обеспечивает универсальные услуги для сетевых протоколов верхнего уровня и эффективное взаимодействие с разнообразными технологиями уровня MAC. Данный уровень предоставляет три типа сервиса вышележащим уровням:

- LLC1 – сервис без установления соединения и без подтверждения;
- LLC2 – сервис с установлением соединения и подтверждением;
- LLC3 – сервис без установления соединения, но с подтверждением.

Обычно в стеке протоколов TCP/IP подуровень LLC функционирует в режиме LLC1 по причине предоставления надежной передачи данных более высокими уровнями.

Подуровень MAC отвечает за управление доступом к физической среде и поддерживает таблицы MAC-адресов – адресов, используемых на канальном уровне моделей взаимодействия открытых систем.

1.2. Технологии локальных сетей

Локальная сеть, как было сказано ранее, – это комплекс оборудования, обеспечивающих передачу данных на относительно небольшие расстояния. Основным назначением локальной сети является предоставление общего доступа к данным и оборудованию.

Технология локальных сетей – это набор стандартов, определяющий топологию сети, метод доступа к среде передачи данных, среду передачи данных, формат pdu, тип физического кодирования, а также физическую скорость передачи данных.

В локальных сетях существует несколько разновидностей подобных технологий: Ethernet, ArcNet, Token Ring и FDDI. Все эти технологии реализуются на канальном и физическом уровнях модели OSI/ISO или на канальном уровне стека протоколов TCP/IP.

На данный момент самой распространенной технологией локальных сетей является Ethernet или его модификации: FastEthernet и Gigabit Ethernet.

1.2.1. Семейство Ethernet

Существует несколько форматов кадра Ethernet, но самым распространенным является Ethernet II, который представлен на рис. 18 с отображением занимаемого каждым полем числа байт.

Destination address	Source address	Type	Data	FCS
6	6	2	46 - 1500	4
Адрес получателя	Адрес источника	Тип	Данные	Контрольная сумма

Рис. 18. Формат кадра Ethernet II

Поля «Адрес получателя» и «Адрес источника» – адреса, идентифицирующие отправителя и получателя данных. Данные поля занимают 6 байт. На канальном уровне в роли адреса устройства выступает физический адрес – MAC-адрес (Media Access Control – управление доступом к среде), присваиваемый каждому устройству, работающему в сети, производителем. MAC-адрес, структура которого представлена на рис. 19, занимает 48 бит и отображается в шестнадцатеричной системе счисления.

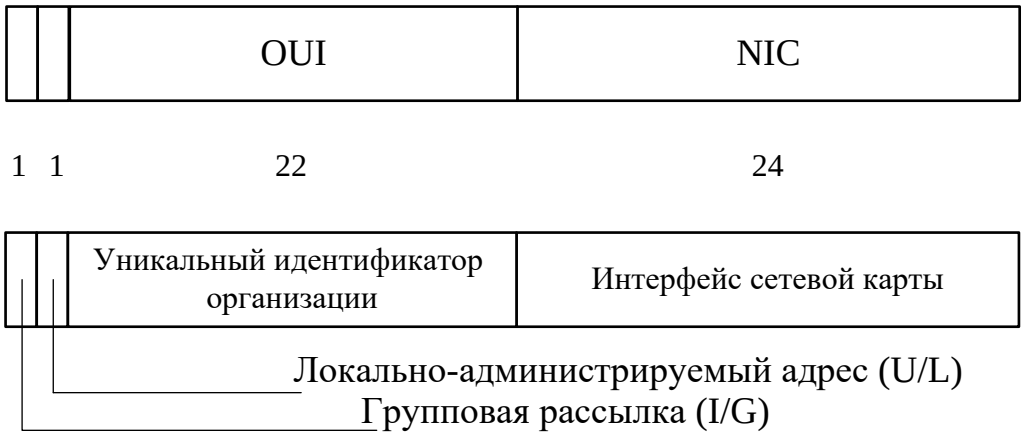


Рис. 19. Структура MAC-адреса

MAC-адрес подразделяется на четыре составляющий. Старший бит MAC-адреса отвечает за тип адреса: если бит установлен в 0, то тип адреса является индивидуальным (I - individual), если в 1 – групповым (G – group). Кадры с групповым типом адреса предназначаются группе устройств, которую идентифицируют остальные 46 бит, а индивидуальные – конкретному устройству. Последующий бит MAC-адреса определяет способ назначения MAC-адреса устройству: производителем (U - universal) или администратором,

отвечающим за конкретную сеть (L - local). Следующие 22 разряда представляют собой код организации, производящей данной устройство. Подобный код присваивается каждой организации комитетом IEEE (Institute of Electrical and Electronics Engineers). Младшие 24 бита адреса – уникальный адрес, присваиваемый организацией-производителем данного устройства (NIC – Network Interface Controller Specific).

Поле «Тип» – идентифицирует протокол вышележащего уровня. То есть данное поле определяет тип (формат) данных, содержащихся в поле «данные» (в соответствии с правилами инкапсуляции в поле «данные» содержится pdu вышележащего уровня (сетевого уровня), и определяет, какой протокол функционирует на сетевом уровня данной системы). Каждому протоколу присвоен цифровой код, например, IP – 0x8000, ARP – 0x0806, RARP – 0x8035 и т.д. Данное поле занимает 2 байта.

Поле «Данные», как уже было описано, содержит pdu вышележащего уровня. Размер данного поля может варьироваться от 46 до 1500 байт.

Поле «Контрольная сумма» – хвостовик, назначение которого состоит в проверке передаваемого кадра на предмет наличия ошибок (FCS – Frame Check Sequence). Данное поле занимает 4 байта.

Методом доступа к физической среде является случайный доступ на основе алгоритма CSMA/CD.

Типом физического кодирования является манчестерский код.

В качестве топологии выбирается пассивная звезда, то есть в центре сети располагается устройство, работающее на физическом или канальном уровнях модели OSI/ISO.

Таким образом, передача кадра от одного устройства другому происходит следующим образом:

1. формируется пакет на сетевом уровне, который инкапсулируется в поле «Данные» кадра Ethernet II;
2. в поле «Тип» кадра записывается код протокола сетевого уровня, который передал данные протоколу Ethernet II;
3. в поле «Адрес источника» записывается MAC-адрес отправителя кадра, в поле «Адрес получателя» - MAC-адрес назначения кадра, по которому оборудование в сети определяет кому предназначается данный кадр;
4. в поле «Контрольная сумма» записывается контрольная сумма, определенная для данного кадра на устройстве-отправителе для дальнейшей проверки кадра на предмет ошибок;
5. на основе алгоритма CSMA/CD устройство-отправитель получает доступ к среде передачи данных и передает в среду кадр, физически закодированный манчестерским кодом;

6. на основе MAC-адреса получателя устройства в сети определяется адрес назначения и данный кадр будет принят только той станцией, чей MAC-адрес записан в поле «Адрес получателя».

Список рекомендуемой литературы

1. В. Олифер, Н. Олифер Компьютерные сети. Принципы, технологии, протоколы. Учебник. Изд-во: Питер, 2014 г. – 944 С.
2. У. Одом Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 640-822 Изд-во: Вильямс, 2012 г. – 720 С.
3. Э. Таненбаум, Д. Уезеролл Компьютерные сети Изд-во: Питер, 2012 г. – 960 С.

Примеры контрольных вопросов для оценки усвоения материала модуля.

1. Перечислите и объясните принципы методов передачи данных.
2. Протокол. Понятие и классификация.
3. Сеть. Понятие и классификация.
4. Топология. Понятие и классификация.
5. Перечислите и объясните принципы дисциплин передачи информации между узлами.
6. Модель OSI/ISO. Задачи, решаемые каждым уровнем и устройства, работающие на физическом и канальном уровне.
7. PDU (Протокольный блок данных). Понятие. Название PDU в зависимости от уровней модели OSI/ISO.
8. Понятия инкапсуляции и деинкапсуляции.
9. Приведите пример типа пользовательских данных, который необходимо передать по надежному соединению между конечными устройствами.
10. При заданных условиях пошагово описать содержимое полей заголовка и хвостовика кадра данных, передаваемого от устройства 1 устройству 4 до момента запуска нового маркера в сеть. Условия задачи:
    a. топология сети тип «кольцо» (рис. 20);
    b. протокол канального уровня – Token Ring с технологией 4 Мбит/с, маркер перемещается в направлении против часовой стрелки;
    c. кадр данных, передаваемый от устройства 1 к устройству 4 (рис. 21);
    d. маркер, захваченный устройством 1, ни разу не проходил по кольцу и только что был создан станцией «Активный монитор».

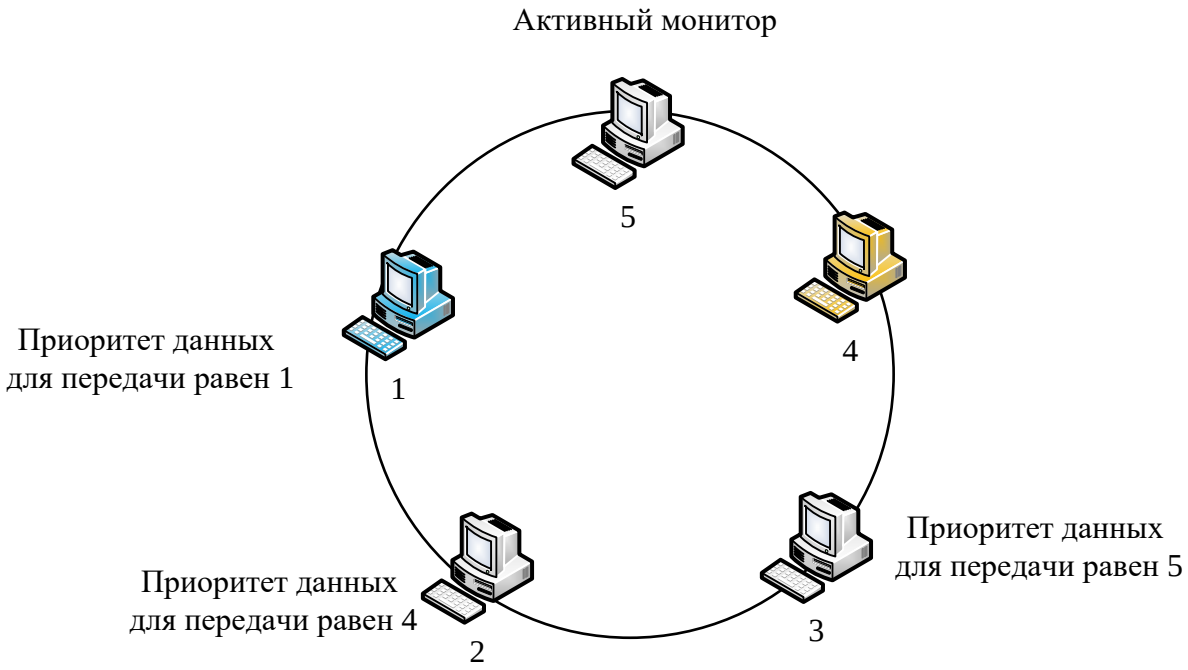


Рис. 20. Топология сети и условия передачи данных устройствами

Рис. 21. Кадр данных формата Token Ring, передаваемый устройством 1 устройству 4

11. Для заданной топологии сети (рис. 22) определите
    a. количество сегментов в сети;
    b. количество доменов коллизий;
    c. количество широковещательных доменов.

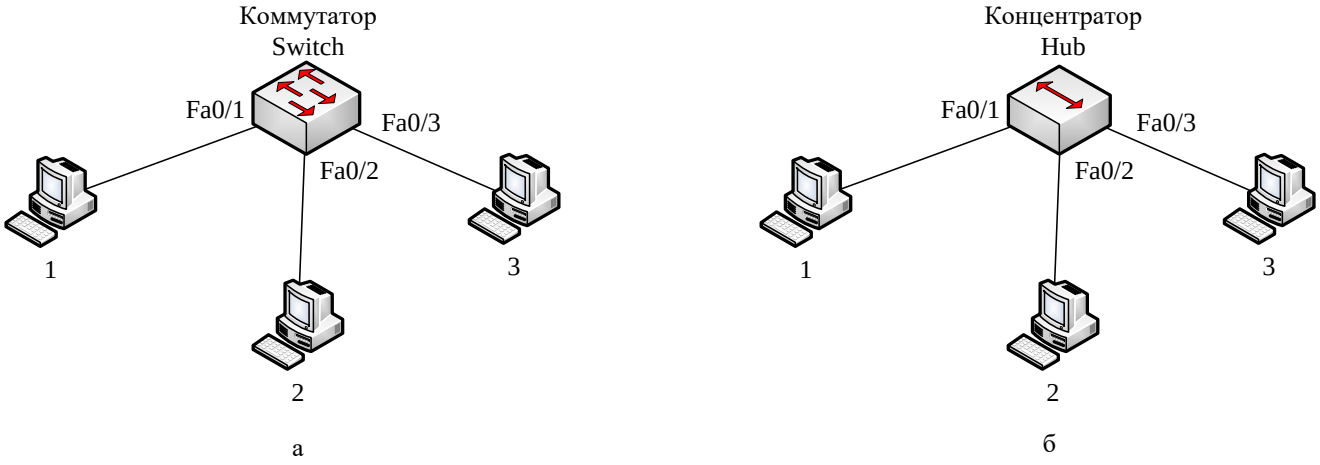


Рис. 22. Топология сети

12. При заданной топологии (рис. 22а) опишите процесс передачи данных коммутатором при следующих сценариях при условии, что таблица коммутации последовательно заполняется:
    a. устройство 1 передает данные устройству 3 при пустой таблице коммутации;
    b. устройство 2 передает данные устройству 3;
    c. устройство 3 передает данные устройству 2;
    d. устройство 1 передает данные с использованием широковещательной рассылки.

МОДУЛЬ 2

2. МАРШРУТИЗАЦИЯ В СЕТЯХ

Технологии локальных сетей, рассматриваемые в модуле 2, выполняют функционал физического и канального уровней модели OSI/ISO. Данные технологии позволяют доставить данные внутри одной локальной сети (до границы ближайшего маршрутизатора). За пределы сети (осуществить переход через маршрутизатор) позволяют технологии, применяемые на сетевом уровне.

Таким образом, второй модуль дисциплины посвящен вопросам адресации и маршрутизации в сетях.

Цели (задачи) изучения 2 модуля

После изучения модуля
«Маршрутизация в сетях» Вы сможете:

- **определить структуру и принцип действия инфокоммуникационных систем;**
- **объяснить понятие инфокоммуникационной сети и описать принцип действия инфокоммуникационной сети;**
- **перечислить дисциплины передачи информации между узлами и объяснить принцип действия каждой дисциплины;**
- **привести несколько моделей взаимодействия открытых систем, подробно описав функционал**
- **дать описание технологий локальных сетей и для заданной топологии продемонстрировать способы обмена данными между конечными узлами.**

Методика проработки и освоения материала 2 модуля:

1 неделя - получение учебного материала модуля, вводная лекция, постановка целей и задач, представление основных ресурсов Интернета для расширения информации по теме модуля.

2 неделя – изучение ключевых понятий, терминов и классификаций.

3 неделя – изучение принципов взаимодействия систем на основе модели взаимодействия открытых систем OSI/ISO и стека протоколов TCP/IP.

4 неделя – изучение технологий локальных сетей на примере семейства Ethernet.

5 неделя - контрольное мероприятие по оценке освоения модуля: письменное задание по проработанному материалу.

Задачи, выносимые на практические занятия:

- изучить функционал и принципы действия оборудования сетевого уровня, применяемого в локальных сетях;
- на примере заданной локальной сети продемонстрировать способы обмена данными между конечными узлами, находящимися в разных сетях;
- изучить принципы бесклассовой адресации.

2.1. Протокол Интернета версии 4 (Internet Protocol version 4, IP ver. 4)

2.1.1. Формат заголовка

Формат заголовка представлен на рис. 23.

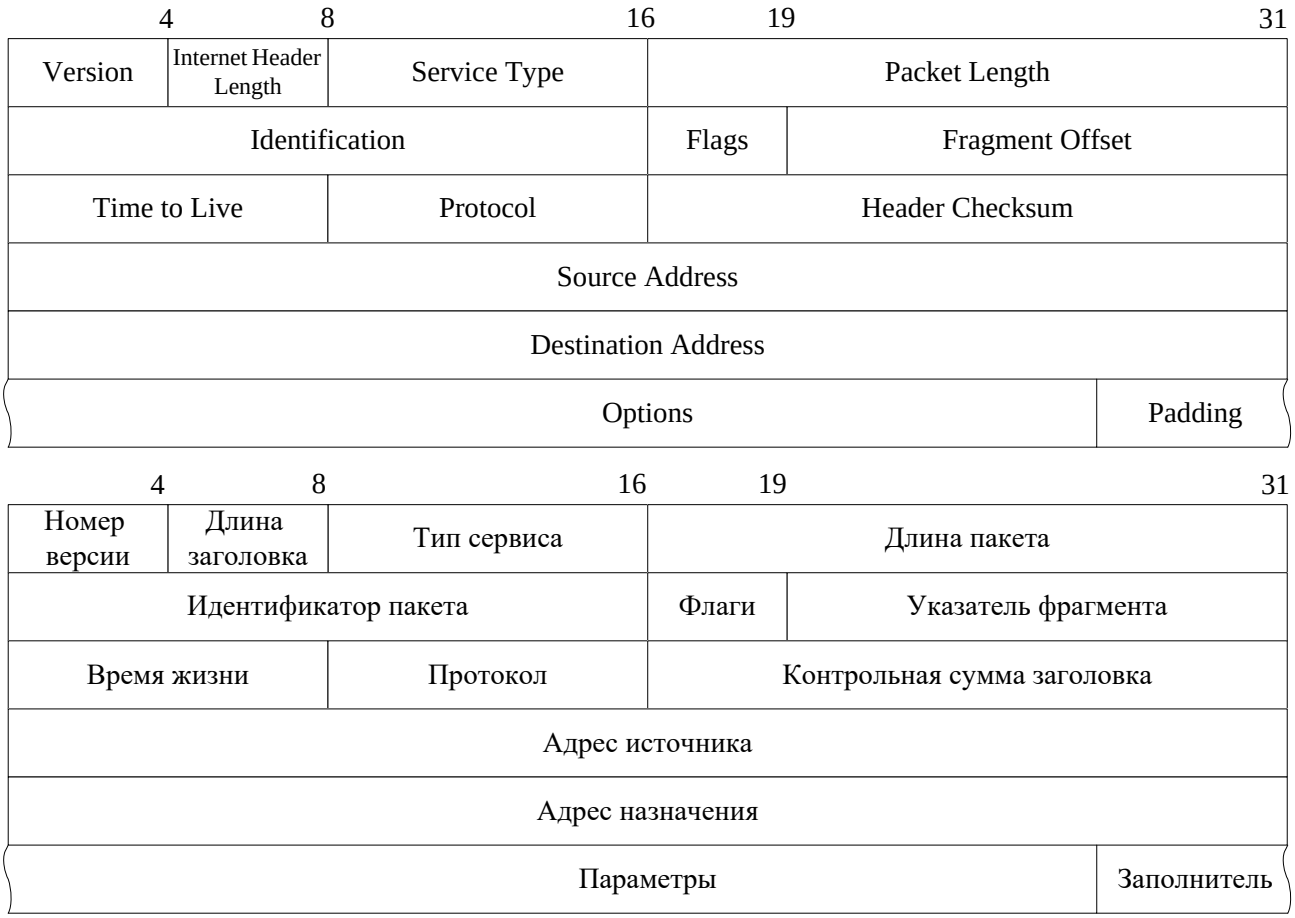


Рис. 23. Формат заголовка протокола IP ver. 4

Поле «Номер версии» указывает версию протокола – 4_{10} (100_2 – соответствует четырем в двоичной системе исчисления). Данное поле занимает 4 бита.

Поле «Длина заголовка» определяет длину заголовка в 32-битных словах (одно 32-битное слово соответствует 4 байтам). Минимальная длина заголовка составляет 5 слов или 20 байт, максимальная – 15 слов или 60 байт. Данное поле является необходимым, поскольку поле «Параметры» является полем переменной длины. Значение длины заголовка определяет границу между заголовком и данными, которые в пакете следуют после заголовка. Данное поле занимает 4 бита.

Поле «Тип сервиса» включает в себя биты, отвечающие за качество передачи данного пакета по каналам связи. Подробная иллюстрация данного поля представлена на рис. 24. Данное поле занимает 8 бит.

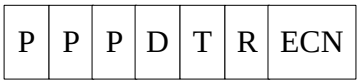


Рис. 24. Поле «Тип сервиса»

Три бита P поля «Тип сервиса» указывают приоритет пакета (precedence – приоритет).

Бит D (delay – задержка) предъявляет требование к минимальной задержке данного IP-пакета.

Бит T (throughout – пропускная способность) предъявляет требование к максимальной пропускной способности канала, по которому будет отправлен данный IP-пакет.

Бит R (reliability – надежность) предъявляет требование к минимальной вероятности ошибки в канале, по которому будет отправлен данный IP-пакет.

Два бита ECN (Explicit Congestion Notification – явное сообщение о задержке) обеспечивает управление IP-потоком. Данный функционал возможен только в случае поддержки обеими сторонами. Результатом возникновения затора в сети является потеря пакетов, потому что оборудование, чьи буферы переполнены, отбрасывают вновь поступающие пакеты. Данный эффект (отбрасывание/потеря пакетов) является показателем возникновения затора в случае, если ECN-сессия не установлена. В случае поддержания устройствами данного функционала, маршрутизаторы путем выставления бит в поле ECN сигнализируют вышележащим по потоку маршрутизаторам, что необходимо либо уменьшить скорость передачи, либо вовсе остановить передачу. По сути, при ECN-сессии маршрутизаторы предупреждаю ситуацию отбрасывания пакетов.

Поле «Длина пакета» содержит информацию о длине пакета (заголовка и данных) в байтах. При этом минимальная длина пакета составляет 20 байт, что соответствует только заголовку (без поля «Параметры»). Максимальный размер – 65535 байт. При этом стоит учитывать, что канал связи, по которому будет происходить передача пакетов, накладывает ограничения на максимальный объем передаваемых данных (MTU – Maximun Transmission

Unit). В случае, если сформированный пакет превышает значение MTU, происходит фрагментация пакета.

Поле «Идентификатор пакета» содержит уникальный идентификатор пакета, который идентифицирует, к какому пакету относится фрагмент: все фрагменты одного пакета содержат одинаковое значение поля «Идентификатор пакета».

Поле «Флаги» состоит из трех бит, которые регламентируют фрагментацию пакета:

- старший бит зарезервирован, и его значение равно нулю;
- средний бит определяет, разрешена ли фрагментация пакета;
- младший бит показывает, является ли данный фрагмент последним в серии или нет.

Стоит отметить следующие важные моменты:

1. если фрагментация не поддерживается, а канал связи, по которому будет происходить передача пакета, предъявляет требование меньшему объему передаваемых данных по сравнению с объемом пакета, пакет будет отброшен, а отправителю будет отправлено сообщение о причине уничтожения пакета;
2. при фрагментации, каждый фрагмент становится самостоятельным пакетом и поле «Длина пакета» содержит не длину первоначального пакета, а длину конкретного фрагмента;
3. при потере какого-либо фрагмента необходима повторная передача всего пакета. Данный функционал обычно реализовывается на транспортном уровне стека протоколов TCP/IP.

Поле «Указатель фрагмента» содержит значение смещения данного фрагмента относительно начала первоначального пакета. Данное поле необходимо для реорганизации пакета из фрагментов на принимающей стороне в силу того обстоятельства, что фрагменты могут приходить на устройство-получатель не в том порядке, в котором были отправлены с устройства-источника. Первый фрагмент имеет нулевое значение.

Поле «Время жизни» определяет число промежуточных устройств сетевого уровня, которое может пройти пакет до момента достижения получателя. Данное поле содержит значение, которое декрементируется на каждом подобном устройстве. В момент достижения нулевого значения данным полем пакет уничтожается, а устройству-источнику отправляется сообщение о причине отбрасывания пакета (истекло время жизни пакета). Максимальное значение данного поля составляет 255.

Поле «Протокол» содержит код вышележащего протокола, которому необходимо передать содержимое поля «Данные». При этом это может быть как протокол транспортного

уровня (TCP, UDP), так и протокол сетевого уровня (ICMP – Internet Control Message Protocol, протокол межсетевых управляющих сообщений).

Поле «Контрольная сумма заголовка» содержит контрольную сумму, с помощью которой каждый узел проверяет целостность заголовка путем сравнения с этим полем контрольной суммы, полученной самостоятельно. Необходимо учитывать, что так как время жизни пакета изменяется на каждом промежуточном узле, работающем на сетевом уровне, контрольная сумма заголовка пересчитывается.

Поля «Адрес источника» и «Адрес назначения» содержат адреса источника и получателя соответственно.

Поле «Параметры» используется для дополнительных опций. Данное поле является не обязательным. Опции подразделяются на 4 класса: 0 – дейтаграммы пользователя или сетевое управление, 1,3 – зарезервированы, 2 – отладка и измерение (диагностика). Примером таких дополнительных опций могут служить жесткая маршрутизация (запись на устройстве-источнике маршрута, по которому необходимо передать пакет, в противном случае маршрут определяется маршрутизаторами), запись маршрута (трассировка) и временных меток.

Поле «Заполнитель» – поле, дополняющее при необходимости поле «Параметры» до целого числа 32-битных слов. Размер поля «Параметры» определен в одном из подполей данного поля.

2.1.2. Адресация в сетях IP версии 4

На сетевом уровне стека протоколов TCP/IP (а также и модели OSI/ISO) адресом является IP-адрес устройства. Данный адрес представляет собой уникальный идентификационный номер устройства, состоящий из 32 бит. Для визуального представления данный адрес записывается в десятичной системе исчисления, разделяя точками 32 бита на 4 октета (по 8 бит каждый). Примером подобного адреса может служить адрес 192.168.1.1.

В общем случае IP-адрес состоит из адреса сети и номера конечного узла (хоста), как показано на рис. 25.

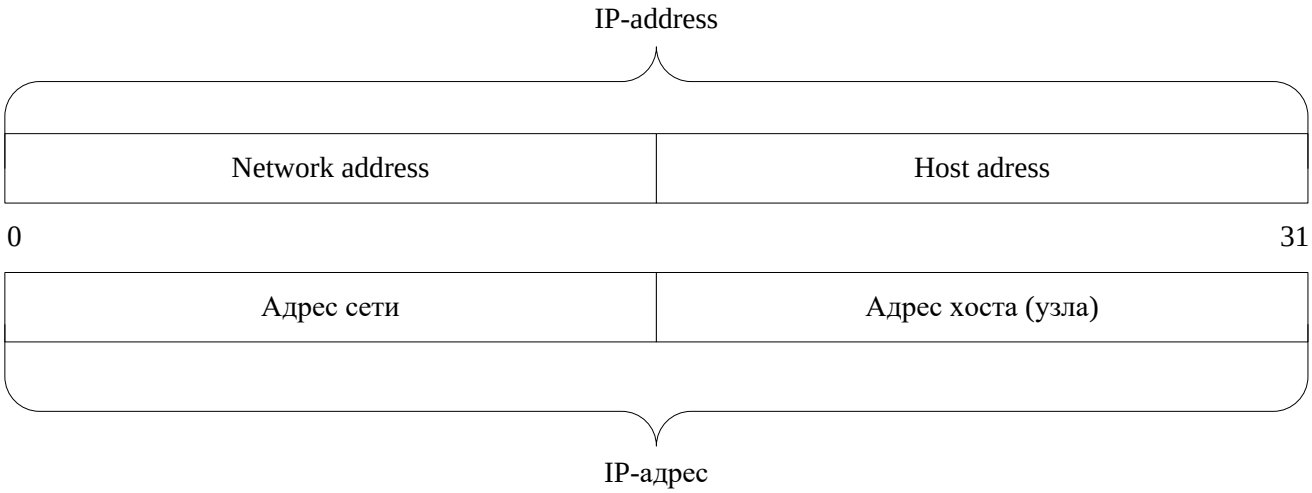


Рис. 25. Структура IP-адреса версии 4

Адрес сети – уникальный идентификатор сети, к которой относится данный хост. По сути, адрес сети – номер многоквартирного дома, а адрес хоста – номер квартиры, если конечным пунктом назначения является квартира.

Определение границы между адресом сети и адресом хоста в IP-адресе может осуществляться на основе

- классовой адресации;
- бесклассовой адресации.

При классовой адресации местоположение данной границы определено классом сети. Выделяется 5 классов сетевых адресов: A, B, C, D, E. Классы A, B, C различаются числом октетов, выделяемых под адрес сети/хоста (рис. 26). Класс D предназначен для групповой рассылки, а класс E зарезервирован и не используется.

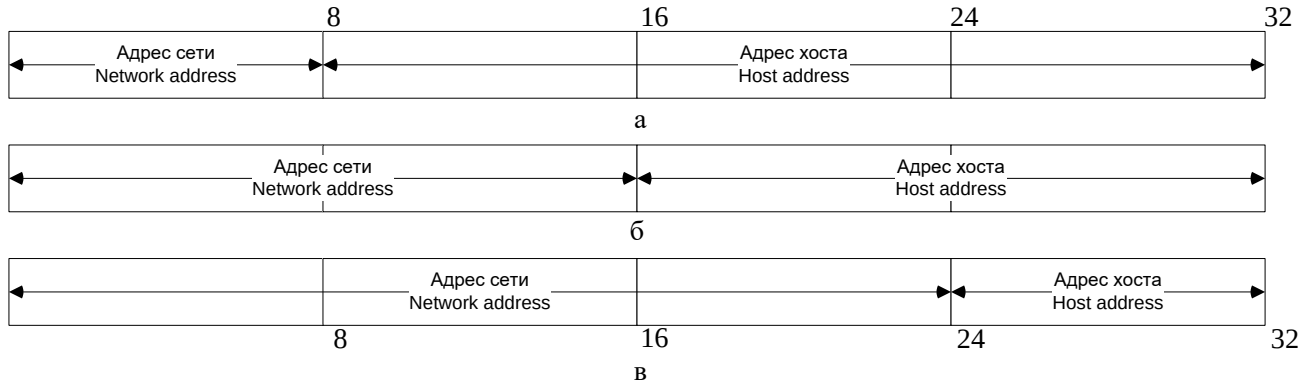


Рис. 26. Классы IP-адресов: A(а), B(б), C(в)

Определение, к какому классу относится IP-адрес, осуществляется на основе значения, записанного в первом октете адреса (табл. 1).

Табл. 1.

Соотнесение классу сети первого октета адреса сети

Класс сети	Значение первого октета. Двоичная система исчисления	Значение первого октета. Десятичная система исчисления
A	0xxxxxxx	0-127
B	10xxxxxx	128-191
C	110xxxxx	191-223
D	1110xxxx	223-239
E	1111xxxx	240-255

Стоит отметить, что в данных диапазонах существуют зарезервированные адреса

- адрес сети – все биты, отведенные под адреса хостов, заполнены нулями (например, 192.168.1.0);
- локальный широковещательный адрес – все биты IP-адреса заполнены единицами (255.255.255.255);
- направленный широковещательный адрес – все биты, отведенные под адреса хостов, заполнены единицами (например, 192.168.1.255);
- неопределенный адрес – все биты IP-адреса заполнены нулями (0.0.0.0);
- адрес возвратной петли (localhost) – сеть 127.0.0.0;
- адрес автоконфигурации – сеть 169.254.0.0.

Назначение широковещательных адресов заключается в рассылке пакета с данным типом адреса назначения всем устройствам, находящимся в сети. Неопределенный адрес имеет устройство, которое только подключилось к сети и находится на стадии назначения IP-адреса. Адрес возвратной петли подразумевает связь устройства с самим собой. Адрес автоконфигурации – адрес, присваиваемый устройству в случае невозможности получения IP-адреса устройству автоматически.

Таким образом, количество сетей в классе, включая зарезервированные адреса сети, определяется как 2^n (где n – число бит, выделяемое под адрес сети), а количество хостов в сети – как (2^m-2) (где m – число бит, выделяемое под адрес узла), поскольку из комбинации 2^n необходимо вычесть адрес сети и направленный широковещательный адрес.

Так же существует разбиение IP-адресов на частные и публичные адреса. Частные адреса обслуживаются только в пределе своей локальной сети, публичные – во всех публичных сетях (сеть Internet). Подобная необходимость связана с нехваткой адресов при условии, что каждое

сетевое устройство в сети (т.е. в мире) должно иметь уникальный IP-адрес. Диапазоны частных адресов приведены в табл. 2.

Таблица 2.

Диапазон частных адресов IP ver. 4.

Класс сети	Диапазон адресов
A	10.0.0.0 – 10.255.255.255
B	172.16.0.0 – 172.31.255.255
C	192.168.0.0 – 192.168.255.255

Ip-адрес (версия 4) устройство может получить

- путем ручной настройки на устройстве администратором сети;
- путем автоматического получения от специализированного сервера – DHCP-сервер (Dynamic Host Configuration Protocol – протокол динамического конфигурирования хостов).

2.1.3. DHCP для IP версии 4

DHCP – протокол прикладного уровня модели OSI/ISO (стека протоколов TCP/IP), регламентирующий обмен сообщениями между клиентом (хостом), которому необходимо назначить IP-адрес в автоматическом режиме (без ручного назначения администратором сети каждому клиенту), и сервером, предоставляющим подобные настройки.

Первоначально, когда клиент только подключился к сети, при условии выставленной настройки «получить IP-адрес автоматически», он имеет неопределенный IP-адрес, т.е. 0.0.0.0. Естественно, при таком IP-адресе устройство не может функционировать в сети. Для получения IP-адреса и настроек сети устройство начинает процесс переговоров с DHCP-сервером как это представлено на рис. 27.

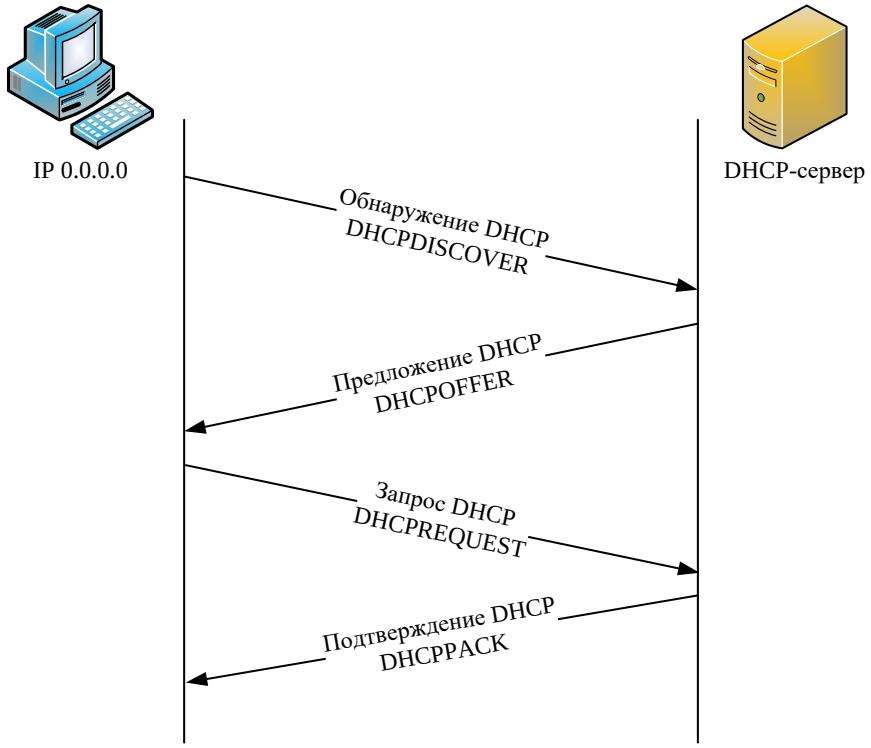


Рис. 27. Процесс обмена сообщениями между хостом и DHCP-сервером с целью получения IP-адреса и настроек сети

Сообщение «Обнаружение DHCP» (DHCPDISCOVER) – первый этап процесса получения IP-адреса устройством, предназначенный для обнаружения доступных DHCP-серверов. В силу природы сообщения (найти устройство, не имея адреса устройства) адрес получателя является локальным широковещательным (сеть, в которой работает вновь появившееся устройство, неизвестна).

Сообщение «Предложение DHCP» (DHCPOFFER) является предложением DHCP-сервера о предоставляемой конфигурации клиенту. Поскольку IP-адрес у клиента отсутствует (является неопределенным), адрес назначения данного сообщения является также широковещательным. Стоит отметить, что возможна ситуация нахождения нескольких DHCP-серверов, которые предлагают клиенту свои конфигурации.

Сообщение «Запрос DHCP» (DHCPREQUEST) – ответ клиента на предложение DHCP-сервера о принятии предложения. Поскольку возможна ситуация нескольких DHCP-серверов, данное сообщение рассылается на широковещательной основе и с включением в тело сообщения идентификатора DHCP-сервера, предложение которого было принято.

Сообщение «Подтверждение DHCP» (DHCPPACK) – подтверждение DHCP-сервера получения запроса от клиента (DHCPREQUEST), рассылаемое широковещательно. И только по получению подтверждения клиент настраивает сетевой интерфейс в соответствии с полученными конфигурациями.

DHCP-сервер предоставляет конфигурацию клиенту, исходя из трех возможных сценариев настройки самого сервера:

- «ручное назначение статических адресов» – на сервере администратором настроены пары «MAC-адрес» - «IP-адрес», регламентирующие какой IP-адрес выдать устройству с известным MAC-адресом;
- «автоматическое назначение статических адресов» – на сервере администратор настаивает пул IP-адресов, из которого происходит выдача адресов клиентам, записывая (запоминая) автоматически пары «MAC-адрес» - «IP-адрес», и впоследствии строго следуя записанным парам, как это выполняется в первом сценарии;
- «автоматическое распределение динамических адресов» – на сервере администратором настраивается пул IP-адресов, из которого происходит выдача адресов клиентам на время, заданное администратором (время аренды), по истечении которого клиенту необходимо отправить запрос на продление использования полученного IP-адреса (DHCPREQUEST).

2.1.4. ICMP

ICMP (Internet Control Message Protocol) – протокол межсетевых управляющих сообщений, работающий на сетевом уровне модели OSI/ISO (стека протоколов TCP/IP).

Особенностью данного протокола является исключение из правила «протокол вышележащего уровня инкапсулируется в протокол нижележащего уровня»: ICMP инкапсулируется в протокол сетевого уровня (IP ver. 4).

Данный протокол не выполняет определенные задачи, а регламентирует механизм, на основе которого могут передаваться и приниматься управляющие сообщения: сообщения об ошибках и информационные сообщения. Под информационными сообщениями понимаются сообщения диагностики и тестирования работоспособности сети.

Формат ICMP-сообщения представлен на рис. 28.

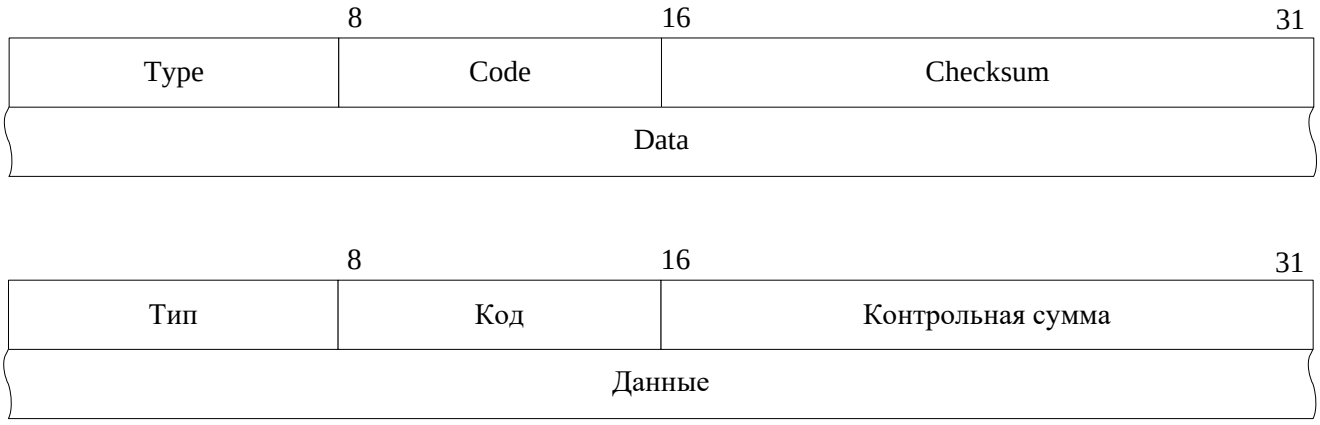


Рис. 28. Формат ICMP-сообщения

Поле «Тип» регламентирует общее предназначение сообщения.

Поле «Код» вносит дополнительный уровень детализации.

Примером заполнения подобных полей может служить следующая ситуация: поле «Тип» содержит значение «3», что соответствует ситуации «Адресат недоступен», поле «Код» – «9», что соответствует «Сеть административно запрещена», т.е. указывает причину, по которой адресат отправленной датаграммы недоступен.

Поле «Контрольная сумма» содержит значение контрольной суммы всего ICMP-сообения, предназначенное для контроля целостности и корректности пакета.

Формат и содержание поля «Данные» зависит от значений полей «Тип» и «Код». Например, при запросе метки времени («Тип» = 13, «Код» = 0) в поле «Данные» помещается, в том числе, время приема и отправки датаграммы.

Правила формирования ICMP-пакетов:

- ICMP-пакет не формируется на потерянный ICMP-пакет;
- ICMP-пакет не формируется в ответ на ICMP-пакет, имеющий широковещательный или групповой адрес рассылки для того, чтобы не вызывать перегрузки сети;
- ICMP-пакет не формируется на непервый поврежденный/потерянный фрагмент фрагментированного IP-пакета.

2.2. Протокол Интернета версии 6 (Internet Protocol version 4, IP ver. 6)

В связи с нехваткой адресного пространства при использовании адресации IP версии 4 была разработана адресация следующей версии (IP версии 6), главным достоинством которой является увеличенное адресное пространство за счет использования 128 битного адреса.

Так же основными достоинствами нового протокола является

- отсутствие широковещательной рассылки;
- автоконфигурация адреса устройства;
- более простой заголовок;
- мобильность – возможность мобильным пользователям (беспроводных подключений) перемещаться между сетями, которая была реализована в IP версии 4 в виде дополнительной конфигурации;
- безопасность – стандарт IPsec является обязательным в IP версии 6.

2.2.1. Формат заголовка

Формат заголовка пакета протокола IP версии 6 представлен на рис. 29.

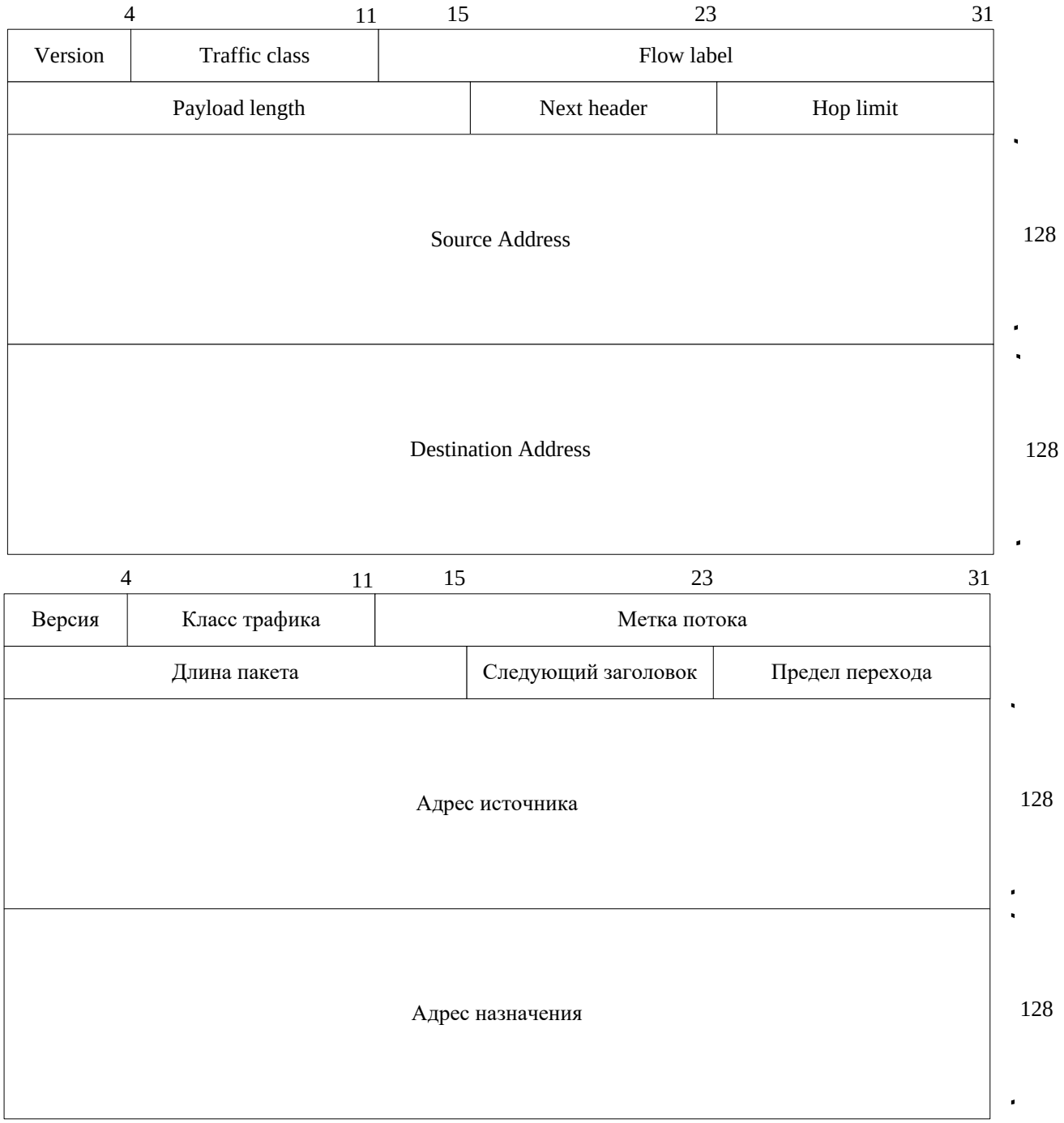


Рис. 29. Формат заголовка IP версии 6

Поле «Номер версии» указывает версию протокола – 6_{10} (110_2 – соответствует шести в двоичной системе исчисления). Данное поле занимает 4 бита.

Поле «Класс трафика» показывает приоритет пакета. Данное поле определяет класс передаваемого трафика в соответствии с качеством обслуживания (Quality of Service, качество обслуживания).

Поле «Метка потока» – это псевдослучайное 20-ти битное число, задаваемое отправителем пакета. При обработке потока пакетов IP версии 6 маршрутизаторы анализируют

дополнительные поля и запоминают результаты обработки в локальном КЭШе. Ключем к такой записи служит комбинация «адрес-отправитель» - «метка потока». При наличии такой записи время на обработку пакета сокращается, что является достоинством протокола IP версии 6. Стоит отметить, что поскольку запись является динамической, время ее жизни в локальном КЭШе маршрутизатора ограничено (порядка 6 секунд по умолчанию).

Поле «Длина пакета» определяет длину пакета в байтах. Стоит отметить, что заголовок пакета протокола IP версии 6 фиксированной длины, в отличие от заголовка пакета протокола IP версии 4.

Поле «Следующий заголовок» идентифицирует по коду протокол, которому следует передать содержимое поле «Данные» по завершению деинкапсуляции.

Поле «Предел перехода» по функционалу полностью идентичен полю «Время жизни» протокола IP версии 4, то есть определяет в единицах время жизни пакета.

Поля «Адрес источника» и «Адрес получателя» определяют соответствующие адреса в формате протокола IP версии 6, то есть в 128 битном размере, записанном в шестнадцатеричной системе счисления, разделенном двоеточием 16-битных сегментов.

2.2.2. Адресация в сетях IP версии 6

На сетевом уровне стека протоколов TCP/IP (а также и модели OSI/ISO) адресом является IP-адрес устройства. Данный адрес представляет собой уникальный идентификационный номер устройства, состоящий из 128 бит. Для визуального представления данный адрес записывается в шестнадцатеричной системе исчисления, разделяя двоеточием 128 бита каждые 16 бит. Примером подобного адреса может служить адрес 2013:0000:120F:0000:0000:09C0:876A :130B.

В общем случае IP-адрес состоит из префикса сети (который может включать префикс подсети) и идентификатора интерфейса, как показано на рис. 30.

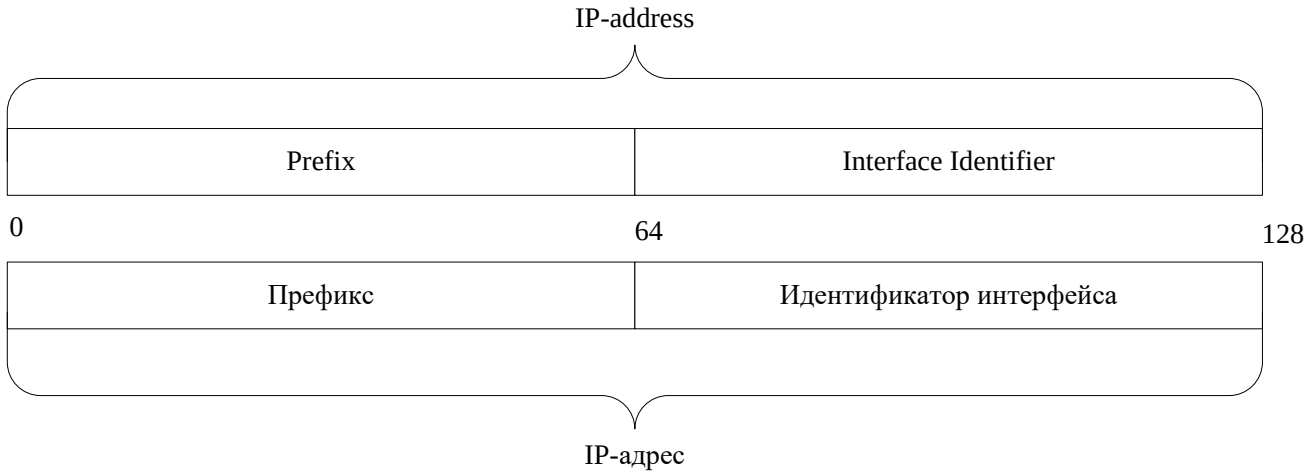


Рис. 30. Структура IP-адреса версии 6

В силу длины адреса были приняты правила представления IP-адреса версии 6:

- пропуск старших 0 в шестнадцатеричной записи (например, 09C0 соответствует 9C0),
- двойное двоеточие (::) может заменить любую или несколько следующих друг за другом шестнадцатеричных сегментов, состоящих их нулей. Двойное двоеточие может применяться в адресе только один раз.

Таким образом, основываясь на описанных правилах представления IP-адреса версии 6, приведенный в примере адрес 2013:0000:120F:0000:0000:09C0:876A :130B можно записать как 2013:0:120F::9C0:876A:130B.

Адреса в протоколе IP версии 6 подразделяются на

- индивидуальный (unicast) – один к одному,
- групповой (multicast) – один ко многим,
- альтернативный (anycast) – один к ближайшему.

Одно из отличий протокола IP версии 6 от версии 4 заключается в отсутствии широковещательного типа адреса (broadcast), одного ко всем, который порождает широковещательные штормы в сети.

Индивидуальный тип адреса в свою очередь подразделяется на

- глобальный (по функционалу соответствует публичному адресу в протоколе IP версии 4), начальный префикс которого выглядит следующим образом: 2000::/3, что соответствует 0010_2 (рис. 31),
- зарезервированный – зарезервированные для определенных групп устройств адреса,
- частный:
  - локальный адрес площадки (Unique Local) – аналог частных адресов протокола IP версии 4. Данный тип адреса маршрутизируется в пределах внутренней сети (анонсируется во всех локальных сетях внутренней сети) (рис. 32). Префикс данного типа адреса равен FD::/8.
  - локальный адрес канала (Link Local) – адреса, используемые для взаимодействия в пределах одной сети в основном служебными протоколами (например, для обнаружения соседей). Данный тип адреса не маршрутизируется за пределы локальной сети. Префикс данного типа адреса равен FE80::/10 (рис. 33).
- адрес возвратной петли (loopback) – аналогичен по функционалу адресу возвратной петли протокола IP версии 4, с одной разницей: в протоколе IP версии 4 под адрес

возвратной петли разработчиками отдана целая сеть 127.0.0.0/8, тогда как в протоколе IP версии 6 – только один адрес (::1),

- неопределенный адрес – адрес, который имеют все устройства, которые только что подключились к существующей сети и не имеют настроенного IP-адреса версии 6 (::).

На рис. 31 представлен формат глобального индивидуального IP-адреса протокола IP версии 6. Данный адрес состоит из пяти адресных пространств:

- регистратора (Registry) – адресное пространство, предоставленное IANA (Internet Assigned Numbers Authority, Администрация адресного пространства Интернет) региональному интернет-регистратору (Regional Internet Registry),
- провайдера (Internet Service Provider) – адресное пространство, предоставляемое региональным интернет-регистратором конкретному интернет-провайдеру,
- организация (Site, площадка) – адресное пространство, предоставляемое интеренет-провайдером конкретной организации,
- подсеть (Subnet) – адресное пространство, которым располагает организация для создания 65536 подсетей,
- идентификатор интерфейса (Interface Identifier) – уникальный идентификатор хоста в локальной сети.

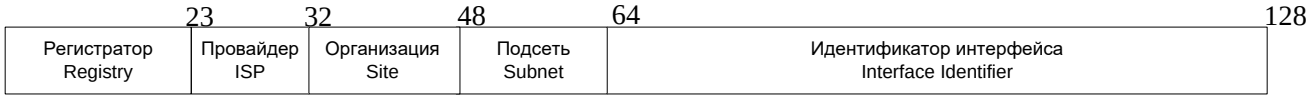


Рис. 31. Формат глобального индивидуального IP-адреса протокола IP версии 6

На рис. 32 представлен формат локального IP-адреса площадки протокола IP версии 6. Данный адрес состоит из четырех адресных пространств:

- FD – префикс, идентифицирующий локальный адрес площадки,
- идентификатор организации (Global ID, глобальный идентификатор) – адресное пространство, описывающее внутреннюю сеть организации,
- подсеть (subnet) – адресное пространство, описывающее подсеть в пределах внутренней сети организации,
- идентификатор интерфейса (Interface Identifier) – уникальный идентификатор хоста в локальной сети.

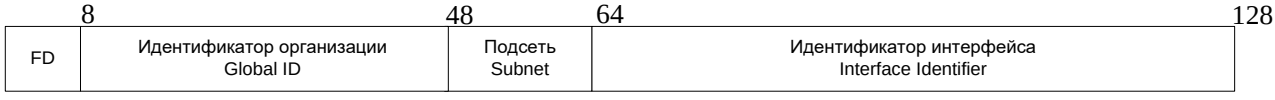


Рис. 32. Формат локального IP-адреса площадки протокола IP версии 6

На рис. 33 представлен формат локального IP-адреса канала протокола IP версии 6. Данный адрес состоит из трех адресных пространств:

- FE80 – префикс, идентифицирующий локальный адрес канала,
- 0 – адресное пространство, выраженное нулями и занимающее 54 бита, поскольку данный тип адреса не маршрутизируется за пределы локальной сети,
- идентификатор интерфейса (Interface Identifier) – уникальный идентификатор хоста в локальной сети.

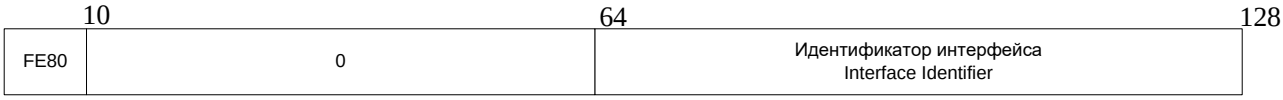


Рис. 33. Формат локального IP-адреса канала протокола IP версии 6

2.2.3. Формирование IP-адреса

IP-адрес (версия 6) устройство может получить

- путем ручной настройки на устройстве администратором сети;
- путем автоматического получения от специализированного сервера – DHCP-сервер (Dynamic Host Configuration Protocol – протокол динамического конфигурирования хостов);
- путем автоконфигурации (новый функционал по сравнению с протоколом IP версии 4).

Выбор технологии, на основании которой устройство получает IP-адрес зависит от того, необходимо ли сохранение полученных настроек. Если подобное условие необходимо выполнить, необходимо применение либо протокола DHCP версии 6, либо ручной настройки. В противном случае, допустимо использование функции автоконфигурации.

Протокол DHCP версии 6 не значительно отличается от предшественника (DHCP версии 4). Основное отличие, которое вытекает из типов адресов, используемых в протоколе IP версии 6, заключается в применении не широковещательной рассылки, а групповой на DHCP-сервера.

Функция автоконфигурации позволяет автоматически настроить как локальный адрес канала, так и локальный адрес площадки на устройстве.

Локальный адрес канала в своем составе имеет только один неопределенный параметр для устройства, только подключившегося к сети – идентификатор интерфейса. Для автоматической генерации данного параметра применяется модификация EUI-64 (Extended Unique Identifier, расширенный уникальный идентификатор). Данная модификация заключается в

1. в добавлении между двумя составляющими MAC-адреса устройства вставки FFFE,
2. инвертировании бита U/L (локально-администрируемый адрес).

Таким образом, в силу уникальности MAC-адреса устройства генерируется локальный адрес канала, так как префикс сети заранее известен (рис. 33).

Автокофигурация также позволяет автоматически настроить локальный адрес площадке на устройстве, адрес шлюза по-умолчанию, адрес DNS-сервера и многих других параметров. Данный функционал реализовывается за счет отправки ICMP версии 6 сообщения (Router Solicitation) на групповой адрес маршрутизаторов. Маршрутизатор в ответ на получение данного сообщения ответит ICMP сообщением Router Advertisement, содержащим префикс сети, в которой находится устройство и множество параметров, необходимых для маршрутизирования адреса устройства за пределы локальной сети (в пределах внутренней сети).

2.3. Маршрутизация

2.3.1. Введение в маршрутизацию

Основной задачей сетевого уровня стека протоколов TCP/IP является маршрутизирование пакетов между сетями географически удаленными друг от друга.

Возникает вопрос причины подобной необходимости при наличии адресации уже на канальном уровне. Причина заключается в том, что априори устройство, функционирующее в сети, не имеет информации о MAC-адресах других устройств. Данное знание появляется посредствам изучения сети (на основе функционирования протокола сетевого уровня Address Resolution Protocol, ARP, протокола разрешения адресов). Суть данного изучения заключается в получении ответа на посланный широковещательный (на канальном уровне) запрос о том, у какого устройства какой MAC-адрес.

Далее необходимо рассмотреть два типа топологии: один – локальная сеть, состоящая из одной сети (рис. 34а), другой – локальная сеть, состоящая из нескольких сетей/подсетей (рис. 34б).

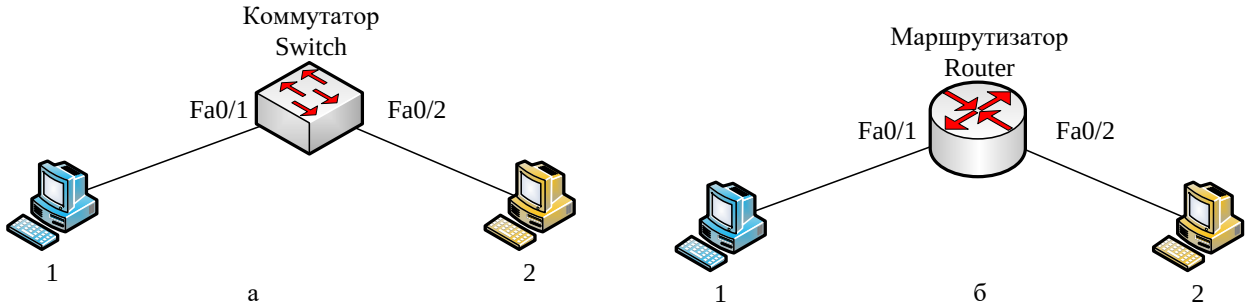


Рис. 34. Топологии сети

Отличие данных топологий заключается в том, что в первом случае сетевым устройством, соединяющим конечные устройства, обменивающиеся запросами о MAC-адресах,

является коммутатор (или любое другое устройство не выше второго уровня модели OSI/ISO), распространяющий широковещательные кадры, а во-втором, - маршрутизатор, устройство третьего уровня модели OSI/ISO, ограничивающее широковещательные домены, а, следовательно, подавляющее всякую широковещательную рассылку как на канальном, так и на сетевом уровнях. Следовательно, запросы на получение информации о MAC-адресах устройств не могут быть доставлены до адресата (протокол, предназначенный для разрешения IP-адреса в MAC-адрес, называется ARP, Address Resolution Protocol).

Отсюда вытекает необходимость более глобальной адресации устройств на сетевом уровне. А, следовательно, и необходимость предоставления информации о местонахождении устройств с конкретными адресами. Но первоначально ставится вопрос о местонахождении сети, в которой находится адресат (как местонахождение улицы, на которой расположен жилой дом). В этом и заключается задача маршрутизации: распространение информации о местонахождении сетей, а, следовательно, и построение маршрутов до каждой сети.

Приведем в пример топологию, представленную на рис. 35, в которой конечные устройства соединены уже не одним маршрутизатором (как на рис. 34б), а несколькими с несколькими возможными маршрутами по сети. Как раз, благодаря работе протоколов маршрутизации, все маршрутизаторы сети будут знать необходимую информацию о каждой сети и построят на ее основе маршрут до конкретной сети.

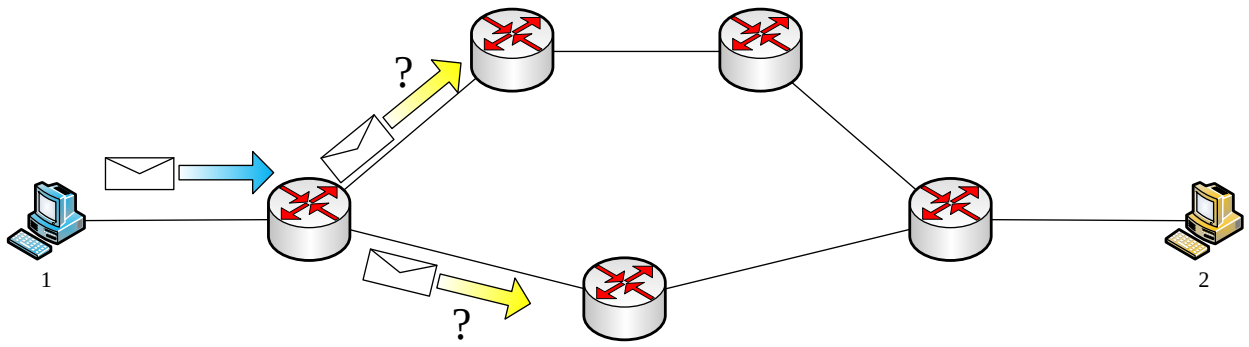


Рис. 35. Топология сети

2.3.2. Параметры протоколов маршрутизации

При разработке протоколов маршрутизации стараются учитывать следующие параметры.

1. Оптимальность алгоритма – характеризует способность алгоритма маршрутизации выбирать наилучший маршрут, который зависит от показателей и от «веса» этих показателей, используемых при расчете маршрута.

2. Низкие непроизводительные затраты. Алгоритмы маршрутизации разрабатываются как можно более простыми. То есть алгоритм маршрутизации должен

эффективно обеспечивать свои функциональные возможности с минимальными затратами программного обеспечения.

3. Стабильность работы. Алгоритмы маршрутизации должны обладать устойчивостью в работе, то есть они должны четко функционировать в случае неординарных или непредвиденных обстоятельств, таких как:

- отказ аппаратуры,
- условия высокой нагрузки,
- некорректные реализации.

Причина заключается в том, что маршрутизаторы расположены в узловых точках сети и их отказ может вызвать значительные проблемы работы сети.

4. Быстрая сходимость алгоритма. **Сходимость** – это процесс соглашения между всеми маршрутизаторами сети об оптимальных маршрутах.

Когда какое-нибудь событие в сети приводит к тому, что маршруты или отвергаются или становятся доступными, маршрутизаторы рассылают сообщения об обновлении маршрутов. Сообщения об обновлении маршрутов пронизывают сети, стимулируя пересчет оптимальных маршрутов и в конечном итоге вынуждая все маршрутизаторы прийти к соглашению по этим маршрутам.

Алгоритмы маршрутизации, которые сходятся медленно, могут привезти к образованию петель маршрутизации или выходам из строя сети.

5. Гибкость алгоритма. Алгоритм маршрутизации должен быстро и точно адаптироваться к различным обстоятельствам в сети. Например, предположим, что сегмент сети отвергнут. Алгоритм маршрутизации, узнав об этой проблеме, быстро выбирает следующий наилучший маршрут для всех маршрутизаторов.

Алгоритмы маршрутизации могут быть запрограммированы таким образом, чтобы они могли адаптироваться к изменениям полосы пропускания канала связи, размеров очереди к маршрутизатору, величины задержки сети и другим переменным.

2.3.3. Классы протоколов маршрутизации

Протоколы маршрутизации можно квалифицировать по следующим классам.

1. Статические и динамические.

Статические алгоритмы представляют собой свод правил работы со статическими таблицами маршрутизации, которые настраиваются администратором сети до начала маршрутизации. Данные таблицы не меняются, если только администратор не изменит их. Эти алгоритмы просты для разработки и хорошо работают в окружении, где трафик сети относительно предсказуем, а топология сети проста.

Динамические алгоритмы маршрутизации подстраиваются к изменяющимся обстоятельствам в сети в масштабе реального времени. Они выполняют это путем анализа поступающих сообщений об обновлении маршрутов. Если в сообщении указывается, что имело место изменение в сети, программа маршрутизации пересчитывает маршруты и рассылает новые сообщения о корректировке маршрута. Такие сообщения приводят к изменению таблиц маршрутизации во всех маршрутизаторах сети.

Статические и динамические протоколы маршрутизации дополняют друг друга, где это уместно.

2. Одномаршрутные и многомаршрутные.

В **одномаршрутных** протоколах для каждой сети существует единственный маршрут. В **многомаршрутных** таких путей может быть несколько. При этом данные пути близки по оптимальности.

Когда протокол многомаршрутный, возможна балансировка нагрузки, то есть пакеты до одного и того же получателя отправляются разными маршрутами.

3. Одноуровневые и иерархические.

Данные протоколы отличаются способами взаимодействия между маршрутизаторами сети.

В **одноуровневых** протоколах все маршрутизаторы равноправны, а в **иерархических** – маршрутизаторы делятся на основные (базовые) и вспомогательные. Базовые маршрутизаторы составляют основу маршрутизации сети. Вспомогательные доставляют пакеты до ближайшего базового маршрутизатора и из последнего базового маршрутизатора до сети назначения.

4. Алгоритмы с маршрутизацией от источника и без маршрутизации от источника

В случае **маршрутизации от источника** маршрут задается отправителем, и маршрутизаторы действуют просто как устройства хранения и пересылки.

В случае **без маршрутизации от источника** отправитель ничего не знает о маршруте. При использовании такого рода алгоритма маршрутизаторы определяют к сети, базируясь на собственных расчетах.

5. Внутренние и внешние (внутридоменные и междоменные) алгоритмы

Внутренние протоколы работают внутри доменов (автономных систем).

Внешние – как внутри, так и между доменами (областями).

6. Алгоритмы состояния канала и дистанционно-векторные

Алгоритмы **состояния канала** направляют потоки маршрутной информации во все маршрутизаторы сети. Однако каждый маршрутизатор посылает только ту часть таблицы маршрутизации, которая описывает состояние его собственных каналов. Таким образом, каждый маршрутизатор имеет представление о всей сети в целом и на основе этих знаний

рассчитывает маршруты. При использовании этих алгоритмов, как правило, отсутствуют регулярные рассылки о состояниях каналов. Рассылки отправляются лишь в начале работы сети и в случае каких-либо изменений в сети.

В **дистанционно-векторных** протоколах соседние (и только соседние) маршрутизаторы регулярно обмениваются между собой таблицами маршрутизации и на основе этих сообщений вносят изменения в таблицу маршрутизации.

Отличаясь более быстрой сходимостью, алгоритмы состояния канала менее склоны к образованию маршрутных петель, но требуют больших вычислительных ресурсов.

2.3.4. Таблица маршрутизации

Задачей протоколов маршрутизации, как было показано ранее, является построение маршрутов до сети назначения. Данная информация отражается в таблицах маршрутизации в виде **записей**. Записи подразделяются на статические и динамические. Статические записи формируются администратором вручную, источниками динамических записей могут быть сети, подключенные напрямую к маршрутизатору, а также информация, полученная с помощью динамических протоколов маршрутизации. В общем случае подобная запись включает в себя следующую информацию:

- протокол, на основе работы которого получена информация;
- адрес сети назначения;
- административное расстояние;
- метрика;
- адрес следующего маршрутизатора;
- интерфейс маршрутизатора, с которого необходимо отправить пакет до сети назначения.

Административное расстояние (Administrative Distance, AD) – целое число от 0 до 255, показывающее приоритет источника маршрута. Маршрут с более низким AD считается более надежным. Таким образом, административное расстояние – степень доверия к источнику, предоставившему информацию о маршруте. Примером ситуации, в которой административное расстояние играет существенную роль, является ситуация, когда информация об одной и той же сети назначения предоставляется двумя и более протоколами маршрутизации (будет выбран маршрут, предоставленный протоколом с наименьшим административным расстоянием). В табл. 3 представлены значения административного расстояния по умолчанию для некоторых протоколов.

Таблица 3.

Значения административного расстояния по умолчанию.

Источник информации о сети	Административное расстояние
Подключенная на прямую	0
Статический маршрут	1

EIGRP	90
OSPF	110
RIP	120

Метрика – расстояние от точки сети, в которой в данный момент находится пакет, до сети назначения. Вычисляется метрика на основе различных параметров (число промежуточных маршрутизаторов, пропускная способность, надежность канала, задержка и т.п.) в зависимости от используемого протокола маршрутизации.

На рис. 36 представлен пример таблицы маршрутизации для топологии, изображенной на рис. 35, функционирующей на основе протокола маршрутизации RIP ver. 2.

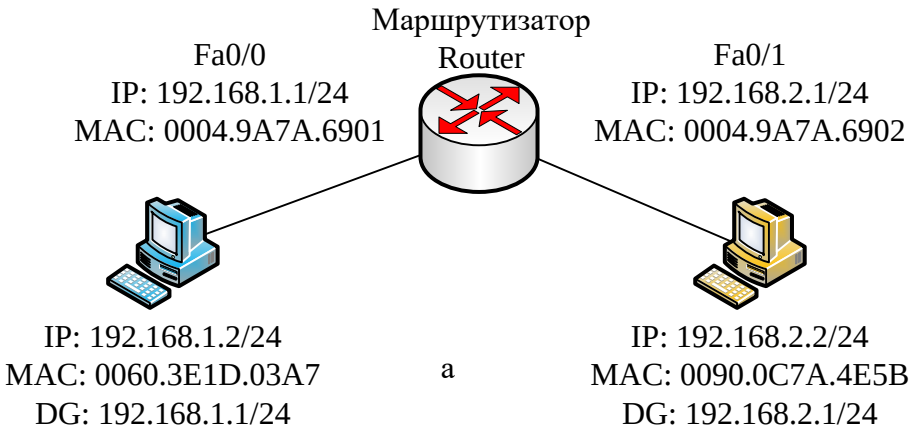
```
Router#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
R    192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:18, FastEthernet0/1
R    192.168.4.0/24 [120/2] via 192.168.2.2, 00:00:18, FastEthernet0/1
                    [120/2] via 192.168.5.2, 00:00:07, FastEthernet1/0
C    192.168.5.0/24 is directly connected, FastEthernet1/0
R    192.168.6.0/24 [120/1] via 192.168.5.2, 00:00:07, FastEthernet1/0
R    192.168.7.0/24 [120/2] via 192.168.5.2, 00:00:07, FastEthernet1/0
```

Рис. 36. Пример таблицы маршрутизации

Исходя из соображения, что протоколы маршрутизации функционируют на маршрутизаторах сети, конечные устройства не имеют информацию о местоположении всех сетей. В силу данного обстоятельства необходимо обеспечить доставку пакета от конечного устройства до ближайшего маршрутизатора, имеющего представление о всей сети в целом. Данная задача решается путем настройки **шлюза по умолчанию** (default gateway, dg). Шлюз по умолчанию – адрес ближайшего интерфейса маршрутизатора. По сути, шлюз последней надежны. Конечное устройство при идентификации адреса получателя как адреса, не принадлежащего сети, к которой принадлежит данное конечное устройство, отправляет пакет по адресу шлюза по умолчанию, на котором принимается решение, куда именно следует отправить данный пакет. Осуществляется адресация до шлюза по умолчанию на канальном уровне. То есть на канальном уровне фигурирует MAC-адрес шлюза по умолчанию, а на сетевом уровне IP-адрес получателя, как показано на рис. 37.



а

192.168.2.2	192.168.1.2	...	...

IP-пакет

0004.9A7A.690	0060.3E1D.03A7	...	...	...

Ethernet-кадр

б

Рис. 37. Пример заполнения адресной информации в заголовках кадра и пакета: а – топология сети с описанием конфигурации, б – адресная информация в заголовках кадра и пакета

2.3.5. Протокол маршрутизации RIP

Протокол маршрутизации RIP (Routing Information Protocol) – дистанционно-векторный протокол маршрутизации. Следовательно, его алгоритм работы заключается в регулярных обменах между соседними маршрутизаторами полными таблицами маршрутизации. Интервал данной регулярной рассылки составляет 30 секунд.

Метрикой для данного протокола служит количество промежуточных маршрутизаторов до сети назначения (число прыжков, число hop'ов). Максимальное значение метрики – 15, то есть сеть назначения не может находится на расстоянии 16 и более hop'ов, данная сеть будет считаться недостижимой.

Для корректной работы динамических протоколов маршрутизации предусмотрены таймеры, регламентирующие время актуальности и жизни записи в таблице маршрутизации. Для протокола RIP по умолчанию установлено два таймера (кроме таймера регулярной рассылки обновлений, update):

- invalid, равный по умолчанию 180 секунд;
- таймер уборки мусора (garbage collection, flush timer) – 240 секунд.

Таймер таймаут запускается в момент занесения записи в таблицу маршрутизации и сбрасывается каждый раз, когда маршрутизатор получает обновление, содержащее информацию о данной записи. Если запись не обновляется в течение 180 секунд, то маршрутизатор считает, что запись более непригодна: метрика становится равной 16 (что соответствует недостижимому маршруту для данного протокола), и запускается таймер уборки мусора. Если в течение работы таймера уборки мусора обновления о маршруте не поступает, запись удаляется. В противном случае – восстанавливается и таймер invalid сбрасывается.

2.3.5.1. Версия 1

На рис. 38 представлен формат сообщения регулярного обновления протокола RIP версии 1. При этом рассылка организовывается широковещательно. На сетевом и канальном уровнях стека протоколов TCP/IP.

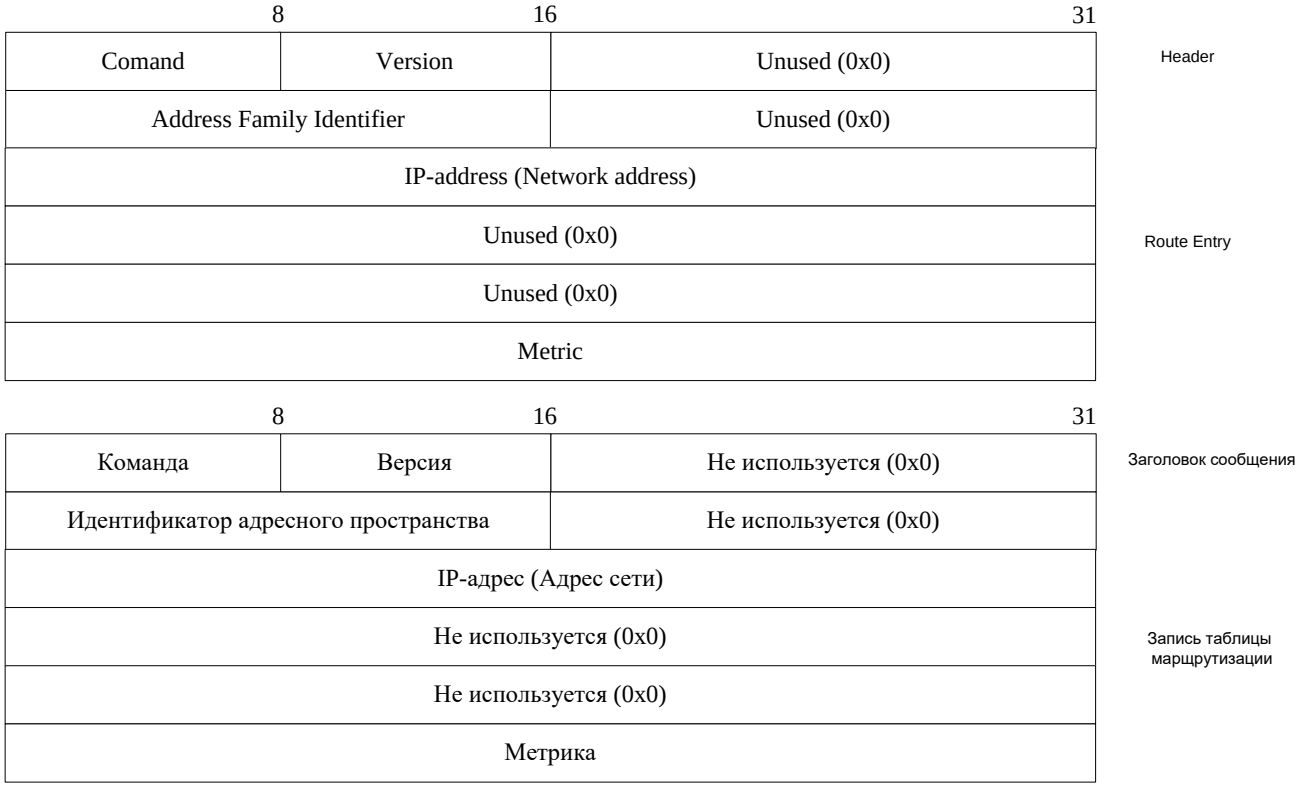


Рис. 38. Формат сообщения протокола RIP версии 1

Поле «Команда» в заголовке сообщения указывает функциональное назначение сообщения:

1. запрос – запрос, распространяемый какой-либо системой для получения полной таблицы маршрутизации;
2. ответ – сообщение, содержащее полную таблицу маршрутизации. Данный тип сообщения может быть ответом на запрос или может рассылаться на регулярной основе.

Поле «Версия» содержит номер версии протокола.

Поле «Идентификатор адресного семейства» (Address Family Identifier, AFI)– тип используемого адреса. В случае использования адреса типа IP версии 4 поддерживается запись AF_INET, которая равна 2. В случае запроса полной таблицы маршрутизации данное поле имеет значение, равное нулю.

Поле «IP-адрес» содержит адрес сети назначения.

Поле «Метрика» содержит значение метрики до сети назначения для того маршрутизатора, который является получателем данного сообщения.

Максимальное число записей в одном сообщении – 25.

2.3.5.2. Версия 2

Протокол RIP версии 2 является только расширением протокола RIP версии 1, так как не внес каких-либо серьезных изменений в механизм или формат сообщения, а только обеспечивает передачу дополнительной информации для исправления недостатков:

1. классовая адресация – в силу отсутствия поля в сообщении RIP версии 1, содержащего маску IP-адреса, возможно использования только информации о классовых сетях, то есть информация о подсетях рассылаться не будет.
2. широковещательная рассылка, которая является всегда недостатком в силу «захламления» сети ненужными данными и возможного возникновения широковещательных штормов.

На рис. 39 представлен формат сообщения регулярной рассылки протокола RIP версии 2.

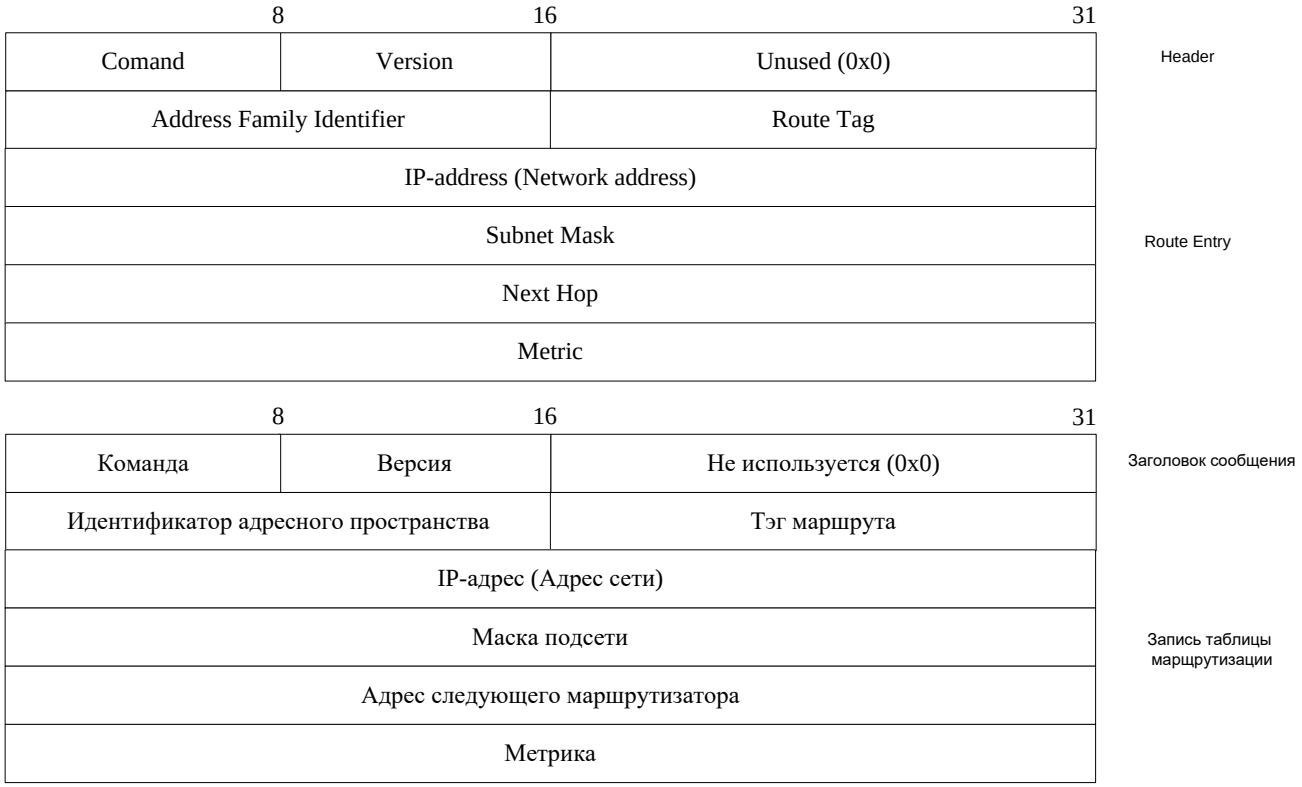


Рис. 39. Формат сообщения протокола RIP версии 2

В дополнение к полям, определенным в формате сообщения протокола RIP версии 1 в формате сообщения протокола RIP версии 2 добавлены:

- поле «Тэг маршрута», которое используется для идентификации источника информации о маршруте с точки зрения протокола маршрутизации: маршруты, изученные с помощью протокола RIP (внутренние маршруты), или маршруты, изученные с помощью других протоколов маршрутизации (внешние маршруты);
- поле «Маска подсети», идентифицирующую бесклассовые сети (деление на подсети);
- поле «Адрес следующего маршрутизатора», содержащее IP-адрес интерфейса маршрутизатора, с которого пришло сообщение.

Максимальное число записей в сообщении составляет 25, как и в сообщении протокола RIP версии 1. Но сами сообщения рассылаются не на широковещательный адрес назначения, а на групповой – 224.0.0.9.

2.3.6. Недостатки работы протоколов маршрутизации и способы их решения

Недостатки или проблемы, которые могут возникать при работе протоколов маршрутизации рассмотрим на примере работы конкретной топологии, представленной на рис. 40.

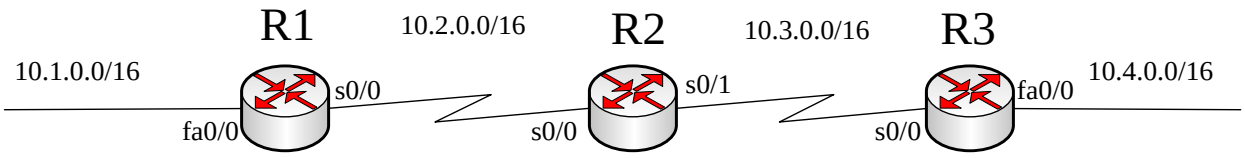


Рис. 40. Топология сети

До начала обмена регулярными обновлениями каждый маршрутизатор сети обладает записями в таблице маршрутизации, представленными на рис. 41.

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.1.0.0/16	fa0/0	0
10.2.0.0/16	s0/0	0

а

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.2.0.0/16	s0/0	0
10.3.0.0/16	s0/1	0

б

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.3.0.0/16	s0/0	0
10.4.0.0/16	fa0/0	0

в

Рис. 41. Таблицы маршрутизации маршрутизаторов R1(а), R2(б), R3(в) до обмена сообщениями протокола маршрутизации

В результате работы протокола RIP версии 2 происходит обмен регулярными сообщениями, содержащими полные таблицы маршрутизации. После обновлений таблицы маршрутизации будут иметь записи, представленные на рис. 42.

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.1.0.0/16	fa0/0	0
10.2.0.0/16	s0/0	0
10.3.0.0/16	s0/0	1
10.4.0.0/16	s0/0	2

а

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.2.0.0/16	s0/0	0
10.3.0.0/16	s0/1	0
10.1.0.0/16	s0/0	1
10.4.0.0/16	s0/1	1

б

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.3.0.0/16	s0/0	0
10.4.0.0/16	fa0/0	0
10.1.0.0/16	s0/0	2
10.2.0.0/16	s0/0	1

в

Рис. 42. Таблицы маршрутизации маршрутизаторов R1(а), R2(б), R3(в) после обмена сообщениями протокола маршрутизации

Далее делаем два допущения, возможные для реальной жизни: первое – сеть 10.4.0.0/16 выходит из строя и второе – обновление маршрутизатора R2 приходит на маршрутизатор R3 раньше обновления маршрутизатора R3 на R2, информирующее об отсутствии сети 10.4.0.0/16. Таким образом, таблица маршрутизации маршрутизатора R3 будет выглядеть, как показано на рис. 43, и образуется петля маршрутизации, из которой пакеты будут выходить только по истечению времени жизни пакетам (когда TTL становится равным нулю, пакет уничтожается, а источнику отправляется ICMP сообщение, информирующее о причине уничтожения пакета), потому что маршрутизатор R3 будет считать, что маршрутизатор R2 знает маршрут до сети 10.4.0.0/16. Иллюстрация петли маршрутизации представлена на рис. 44.

Таблица маршрутизации		
Сеть назначения	Выходной интерфейс	Метрика
10.3.0.0/16	s0/0	0
10.4.0.0/16	**s0/0**	**2**
10.1.0.0/16	s0/0	2
10.2.0.0/16	s0/0	1

Рис. 43. Таблица маршрутизации маршрутизатора R3 после получения обновления от маршрутизатора R2

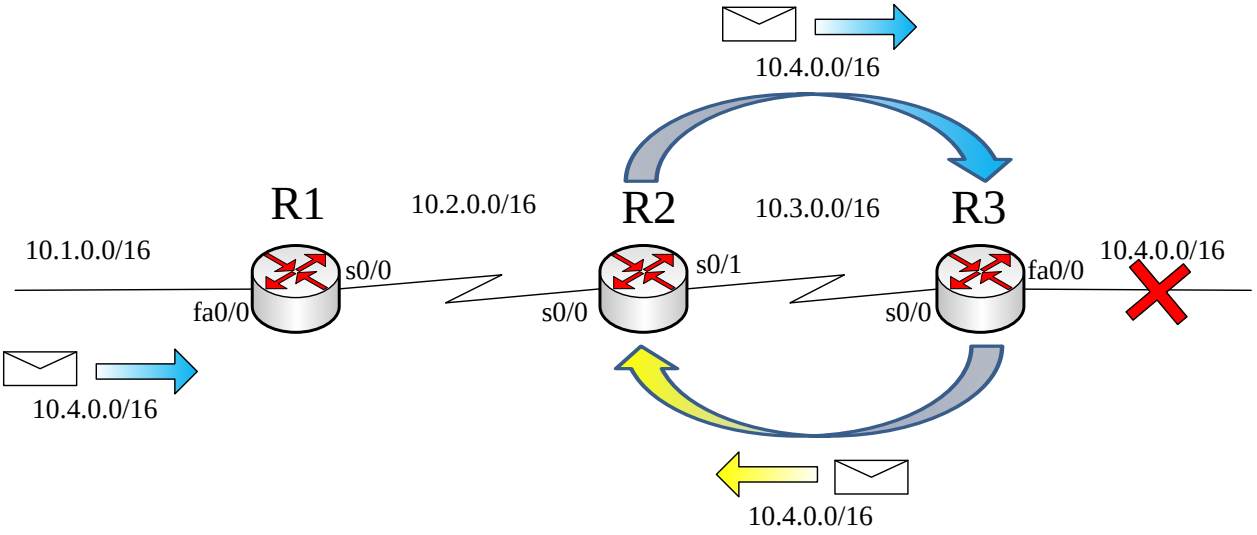


Рис. 44. Петля маршрутизации

Существует четыре метода предотвращения появления петель маршрутизации:

- расщепление горизонта;
- отравление маршрута;
- обратное отравление;
- таймер удержания:
- триггерные сообщения.

К примеру, протокол RIP не зависимо от версии использует по умолчанию таймер удержания и расщепление горизонта с обратным отравлением.

2.3.6.1. Расщепление горизонта

Суть метода «Расщепление горизонта» заключается в том, что не следует посылать информацию в обратном направлении, то есть к источнику исходных данных.

Таким образом, для рассматриваемого выше примера применение данного метода ликвидирует возможность появления петли маршрутизации, потому что маршрутизатор R2 не включит информацию о маршруте 10.4.0.0/16 в сообщение обновления для маршрутизатора R3.

Следовательно, если при обмене сообщениями дистанционно-векторные протоколы включают в сообщение все записи своей таблицы маршрутизации, то при применении расщепления горизонта сообщения, которыми будут обмениваться маршрутизаторы будут включать записи, представленные на рис.45.

0x2	0x2	0x0
0x2		0x0
10.1.0.0		
255.255.0.0		
10.2.0.1		
0x1		

а

0x2	0x2	0x0
0x2		0x0
10.4.0.0		
255.255.0.0		
10.3.0.2		
0x1		

б

0x2	0x2	0x0
0x2		0x0
10.3.0.0		
255.255.0.0		
10.2.0.2		
0x1		
0x2		0x0
10.4.0.0		
255.255.0.0		
10.2.0.2		
0x2		

в

0x2	0x2	0x0
0x2		0x0
10.2.0.0		
255.255.0.0		
10.3.0.1		
0x1		
0x2		0x0
10.1.0.0		
255.255.0.0		
10.3.0.1		
0x2		

г

Рис. 45. Сообщение регулярной рассылки протокола RIP версии 2 маршрутизатора R1 к маршрутизатору R2 (а), маршрутизатора R3 к маршрутизатору R2 (б), маршрутизатора R2 к маршрутизатору R1 (в), маршрутизатора R2 к маршрутизатору R3 (г)

Конфигурация интерфейсов маршрутизаторов представлена в таблице 4.

Таблица 4

Конфигурация интерфейсов маршрутизаторов

Маршрутизатор	Интерфейс	IP-адрес/маска
R1	fa0/0	10.1.0.1/16
	s0/0	10.2.0.1/16
R2	s0/0	10.2.0.2/16
	s0/1	10.3.0.1/16
R3	s0/0	10.3.0.2/16
	fa0/0	10.4.0.1/16

2.3.6.2. Отравление маршрута

Функция «отравление маршрута» на маршрутизаторе создает запись в таблице маршрутизации, которая поддерживает согласованность состояния сети, пока на других маршрутизаторах сети выполняется конвергенция изменения топологии. Суть данной функции состоит в рассылке обновления о маршруте с бесконечной метрикой (для протокола RIP равной 16) в случае недостижимости сети.

Рассмотрим на описанном выше примере: сеть 10.4.0.0/16 становится недостижимой, маршрутизатор R3 устанавливает бесконечную метрику для данного маршрута (рис. 46), затем отправляет обновление об отравлении маршрута маршрутизатору R2, который в свою очередь отправляет обновление маршрутизатору R1.

0x2	0x2	0x0
0x2		0x0
10.4.0.0		
255.255.0.0		
10.3.0.2		
0x10		

Рис. 46. Сообщение регулярной рассылки протокола RIP версии 2 маршрутизатора R3 к маршрутизатору R2 с отравленным маршрутом

2.3.6.3. Обратное отравление

«Обратное отравление» – механизм, отправления обновления о маршруте на тот интерфейс, с которого была получена информация о маршруте, с бесконечной метрикой. В результате данного обновления маршрутизатор убеждается, что маршрут действительно недостижим через соседний маршрутизатор, который прислал обновление с бесконечной метрикой.

Таким образом, при применении отравления маршрута в совокупности с обратным отравлением возможно предотвращение петель маршрутизации.

Для примера, рассматриваемого выше, маршрутизатор R3 отправит обновление маршрутизатору R2 о том, что сеть 10.4.0.0/16 недостижима, то есть имеет метрику 16 (для протокола RIP), а маршрутизатор R2 отправит два обновления: первое – на маршрутизатор R3 («обратное отравление») о том, что через него маршрут 10.4.0.0/16 также недостижим (тем самым на маршрутизаторе R3 не может появиться метрика 3 для данного маршрута, как было показано ранее), а второе – на маршрутизатор R1 с информацией, что маршрут 10.4.0.0/16 более не достижим, в ответ на который маршрутизатор R1 отправит так же обновление «обратное отравление».

Применение механизма «обратное отравление» в совокупности с механизмом «расщепление горизонта» переопределяет принцип «расщепления горизонта», включая в обновления маршруты, полученные от соседнего маршрутизатора, помечая их как недостижимые, то есть устанавливая бесконечную метрику.

2.3.6.4. Таймер удержания

Данный механизм применяется для предотвращения регулярных обновлений, указывающих на маршрут, который может быть недостижим. Данный таймер не позволяет маршрутизатору применять изменение маршрута в течение определенного периода времени. Период удержания зависит от протокола маршрутизации, но как правило равен трем интервалам периодического обновления протокола.

Алгоритм действия таймера удержания следующий.

1. Когда маршрутизатор получает обновление от соседнего маршрутизатора, указывающего, что доступная сеть более недоступна, маршрутизатор отмечает маршрут как «предположительно недоступным» (рис. 47) и запускает таймер удержания.

```
Router#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/16 is subnetted, 4 subnets
R       10.1.0.0 [120/1] via 10.2.0.1, 00:00:12, FastEthernet0/0
C       10.2.0.0 is directly connected, FastEthernet0/0
C       10.3.0.0 is directly connected, FastEthernet0/1
R       10.4.0.0 is possibly down, routing via 10.3.0.2, FastEthernet0/1
Router#
```

Рис. 47. Таблица маршрутизации R3 с «предположительно недоступным» маршрутом

2. Если от соседнего маршрутизатора приходит обновление с лучшей метрикой, по сравнению с исходной метрикой маршрута (который является «предположительно недоступным»), маршрутизатор отмечает маршрут как «доступный», записывает новую метрику и адрес источника маршрута и удаляет таймер удержания.

3. Если во время работы таймера от соседнего маршрутизатора приходит обновление данного маршрута с худшей или равной метрикой, обновление игнорируется. Пропуск обновления с худшей или такой же метрикой во время работы таймера удержания дает больше времени для распространения изменения на всю сеть.

2.3.6.5. Триггерные сообщения

Триггерные обновления таблицы маршрутизации отправляются немедленно в ответ на определенное изменение. Маршрутизатор, обнаруживший изменение, немедленно отправляет обновление смежным маршрутизаторам, которые в свою очередь генерируют триггерные сообщения , оповещающие их соседей об изменении.

Триггерные сообщения были бы достаточной мерой, если бы гарантировали своевременную доставку оповещений всем маршрутизаторам сети. Однако существует две вероятности, негативно сказывающиеся на работе сети в данном случае.

1. Пакеты обновлений могут быть отброшены или повреждены одним из каналов связи.

2. Триггерные обновления не применяются мгновенно. Существует вероятность того, что маршрутизатор, который еще не получил триггерное обновление, отправит регулярное обновление в неудачное время, что приведет к повторной установке маршрута в соседнем маршрутизаторе, который уже получил триггерное обновление.

Удачной комбинацией для разрешения описанных ситуаций могут служит триггерные сообщения и таймеры удержания, которые не позволят регулярным обновлениям отменить результаты триггерных.

2.3.7. Протокол маршрутизации OSPF

Протокол маршрутизации OSPF (Open Shortest Path First) – протокол состояния канала. Следовательно, маршрутизаторы обмениваются информацией о своих каналах связи со всеми маршрутизаторами сети. На основе полученной информации каждый маршрутизатор имеет представление о сети в целом и на основе алгоритма Дейкстры осуществляет поиск наикратчайшего пути до пункта назначения. Метрика в протоколе OSPF высчитывается как соотношение эталонной пропускной способности к пропускной способности канала:

$$M = \frac{10^8,\text{бти/с}}{\text{пропускная способность канала,бит/с}}.$$

2.3.7.1. Алгоритм поиска наикратчайшего пути (алгоритм Дейкстры)

Алгоритм Дейкстры – алгоритм поиска наикратчайшего маршрута по графу. Применительно к протоколу маршрутизации в виде графа изображается топология сети, где вершинами являются маршрутизаторы сети, а ребрами – связи между ними. Каждое ребро графа имеет вес, определенный на основе способа расчета метрики, применяемого в протоколе маршрутизации.

Данный граф строит каждый маршрутизатор сети на основе знаний о всей сети в целом.

Рассмотрим работу алгоритма Дейкстры на примере. Дана сеть, топология которой изображена на рис. 48 в виде графа.

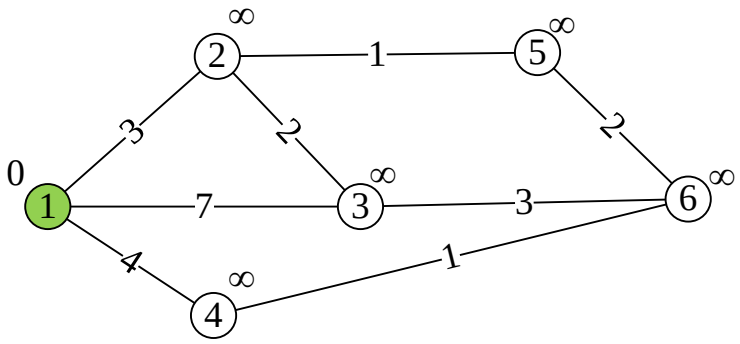


Рис. 48. Начальная топология сети

Задача алгоритма состоит в нахождении кратчайшего пути до всех вершин графа. Точкой отчета является вершина 1, которая является тем маршрутизатором, на котором производится расчет.

Шаг 1. Всем вершинам, кроме начальной, присваивается бесконечное расстояние от исходной вершины. Начальной – нулевое.

Шаг 2. Определяется расстояние от вершины, в которой алгоритм находится в данный момент, до всех соседних вершин. Расстояние равно сумме расстояния данной вершины от исходной (вершина 0) вершины и веса ребра, по которому возможно пройти до вершины. Если расстояние меньше расстояния, ассоциируемого с данной вершиной, новое расстояние присваивается вершине. Таким образом, расстояние от вершины 1 до вершины 3 равно 3, от вершины 1 до вершины 3 – 7, от вершины 1 до вершины 4 – 4.

Шаг 3. Осуществляется переход в соседнюю непосещенную вершину от исходной с наименьшим расстоянием. Предыдущая вершина помечается, как посещенная (рис. 49).

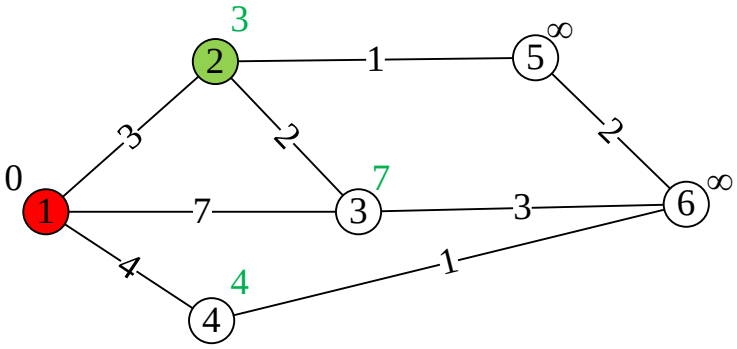


Рис. 49. Переход в соседнюю вершину с наименьшим расстоянием

Шаг 4. Повторяются шаги 2, 3 (рис. 50).

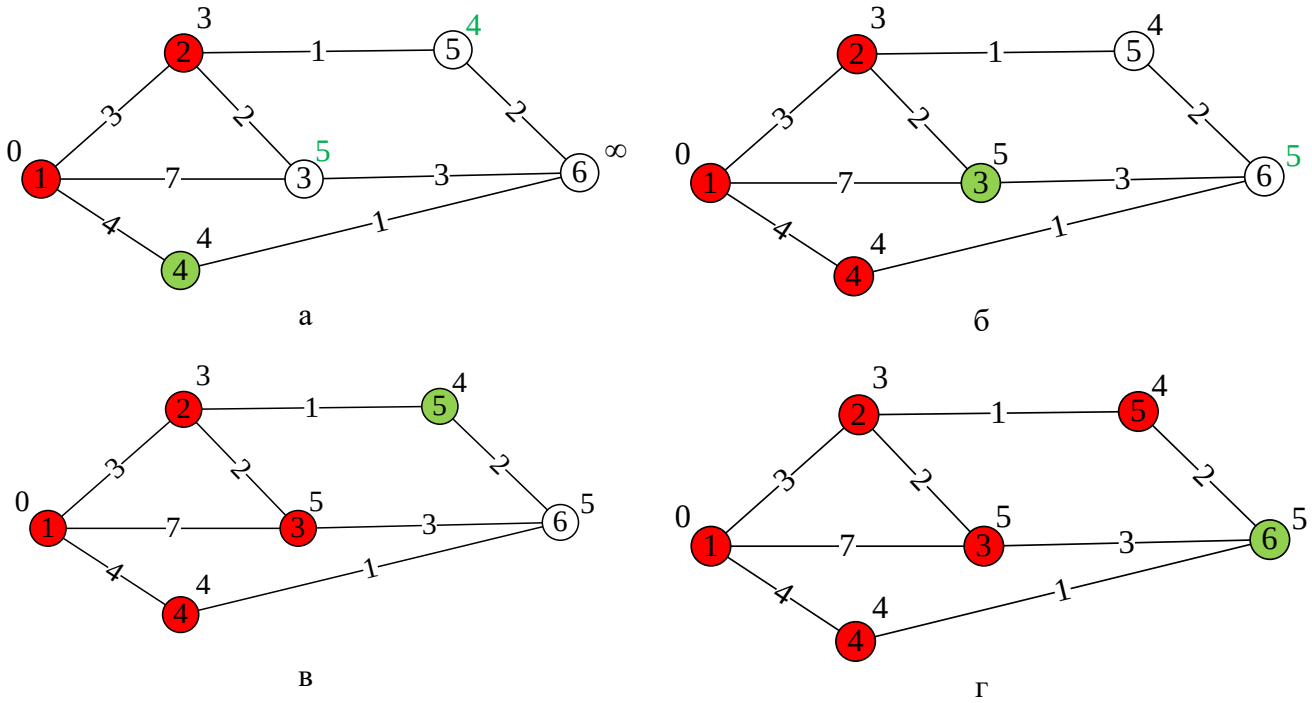


Рис. 50. Пошаговый поиск наикратчайшего маршрута на основе алгоритма Дейкстры

Таким образом, например, до вершины 6 из вершины 1 наикратчайший маршрут проходит через вершины 1-4-6.

Стоит отметить, что в случае существования маршрутов с равным расстоянием у алгоритма Дейкстры нет четкого критерия выбора маршрута. Этот вопрос решается непосредственно самим алгоритмом маршрутизации. Например, протокол OSPF поддерживает **балансировку маршрутов**, заключающуюся в существовании нескольких, близких по стоимости, маршрутов до пункта назначения. По умолчанию число подобных маршрутов равно 4, но технологии поддерживают до 16.

Балансировка маршрутов быть как **потоковая**, так и **пакетная**. Потоковая балансировка обеспечивает пересылку потоков (поток пакетов от одного и того же источника к одному и тому же получателю), а пакетная – отдельных пакетов. Таким образом, в первом случае при передаче, например, от одного и того же источника потока пакетов к двум разным получателям один поток будет направлен по одному маршруту, второй – по другому. При пакетной балансировке пакеты (а не потоки) будут направляться по разным маршрутам, не зависимо от адреса получателя.

2.3.7.2. Сообщения протокола OSPF

Формат заголовка сообщений OSPF представлен на рис.51. Сообщения протокола OSPF инкапсулируются непосредственно в пакет протокола IP версии 4 и отправляются на групповой адрес 224.0.0.5.

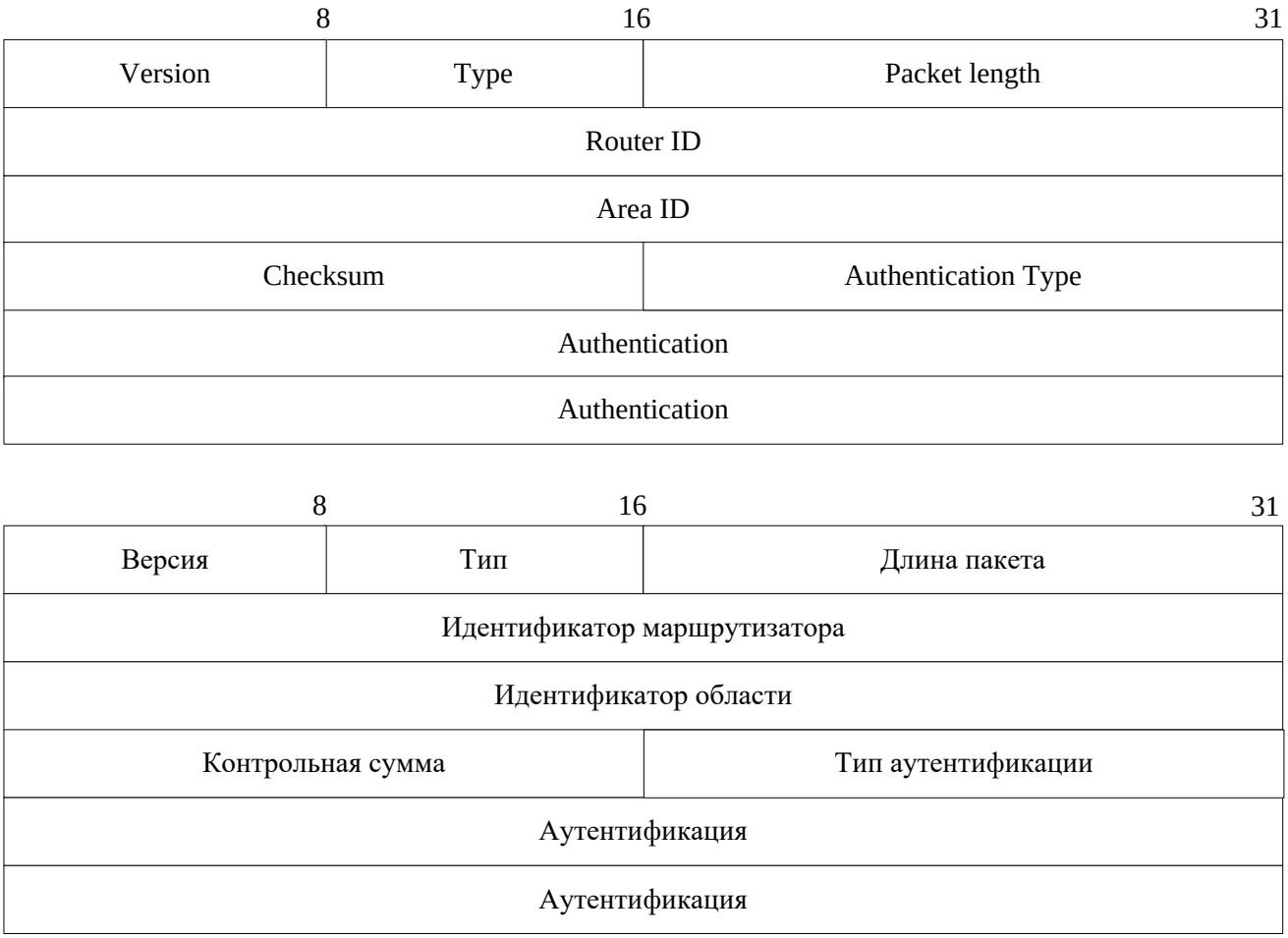


Рис. 51. Формат заголовка сообщения протокола OSPF

Поле «Версия» содержит версию протокола OSPF. На данный момент функционирует 2 версия протокола и, следовательно, в данном поле записано число 2.

Поле «Тип» определяет сообщения, идентифицируя тем самым функционал сообщения:

1. «Hello» – используется для проверки доступности маршрутизатора;
2. «Data State Description» – описание топологической базы данных;
3. «Link State Request» – запрос состояния канала;
4. «Link State Update» – изменение состояния канала;
5. «Link State Acknowledgment» – подтверждение получения сообщения о статусе канала.

Поле «Длина пакета» определяет длину сообщения, включая заголовок.

Поле «Идентификатор маршрутизатора» включает уникальный 32-битный код маршрутизатора, отправившего сообщения, идентифицирующий маршрутизатор в пределах автономной системы.

Поле «Идентификатор области» – 32-битный код идентифицирующий область. Протокол OSPF является иерархическим протоколом, использующим двухуровневую иерархию: **автономную систему** и **область**.

Область – группа смежных сетей, логические разделы автономной системы.

Автономная системы – совокупность сетей с общим управлением и общей стратегией маршрутизации.

Использование подобной иерархии (рис. 52) позволяет уменьшить размер топологической базы данных состояния канала и таблиц маршрутизации.

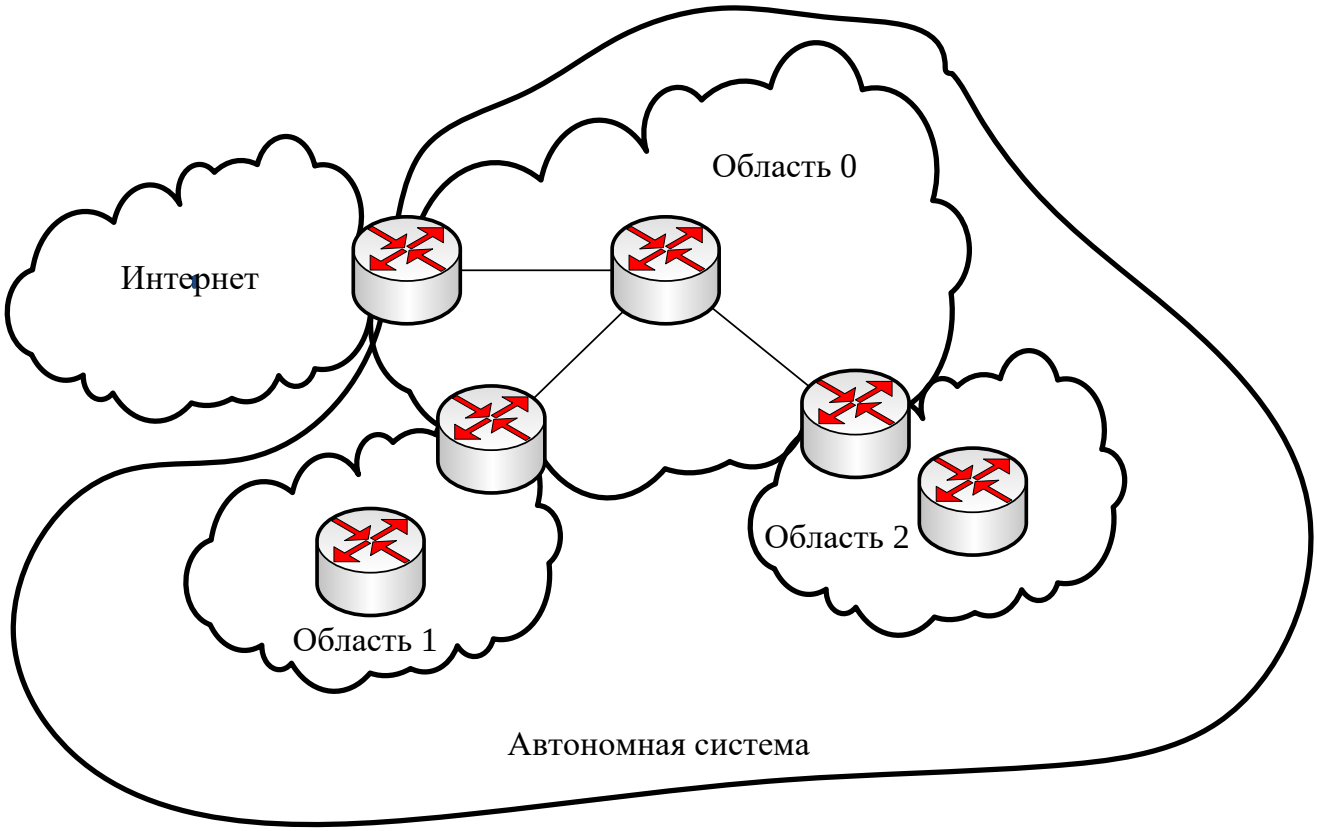


Рис. 52. Понятие автономной системы и области в протоколе OSPF

Поле «Контрольная сумма» включает контрольную сумму всего сообщения.

Поле «Тип аутентификации» определяет наличие и тип аутентификации (0 – отсутствие контроля доступа, 1 – аутентификация открытым текстом, 2 – аутентификация MD5).

Поле «Аутентификация» – поле данных аутентификации.