7. Протокол IP

7.1. Основные функции протокола IP

Описание протокола IP (Internet Protocol) дано в документе RFC 791. IP является базовым протоколом всего стека TCP/IP.

Название данного протокола — Intrenet Protocol — отражает его суть: он должен передавать пакеты между сетями. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель.

Протокол IP относится к протоколам без установления соединений. Перед IP не ставится задача надежной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как независимую единицу, не имеющую связи ни с какими другими IP-пакетами. В протоколе IP нет механизмов, обычно применяемых для увеличения достоверности конечных данных: отсутствует квитирование — обмен подтверждениями между отправителем и получателем, нет процедуры упорядочивания, повторных передач или других подобных функций. Если во время продвижения пакета произошла какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен по причине истечения времени жизни или из-за ошибки в контрольной сумме, то модуль IP не пытается заново послать испорченный или потерянный пакет. Все вопросы обеспечения надежности доставки данных по составной сети в стеке TCP/IP решает протокол TCP, работающий непосредственно над протоколом IP. Именно TCP организует повторную передачу, когда в этом возникает необходимость.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов (например, от сетевого протокола IPX), является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными, максимально допустимыми значениями поля данных кадров MTU. Свойство фрагментации во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

Имеется прямая связь между функциональной сложностью протокола и сложностью заголовка пакетов, которые этот протокол использует. Это объясняется тем, что основные служебные данные, на основании которых протокол выполняет то или иное действие, переносятся между двумя модулями, реализующими этот протокол на разных машинах, именно в полях заголовков пакетов. Поэтому, очень полезно изучить назначение каждого поля заголовка IP-пакета, что даст не только формальные знания о структуре пакета, но и объяснит все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

Дейтаграмма состоит из заголовка и поля данных, которое следует сразу за заголовком. Длина поля данных определяется полем «Общая длина» в заголовке. На рис. 7.1 показан формат заголовка IP-дейтаграммы.

<table>
<tr><td>Номер версии (4 бита)</td><td>Длина заголовка (4 бита)</td><td>Тип сервиса (8 бит)</td><td colspan="2">Общая длина (16 бит)</td></tr>
<tr><td colspan="3">Идентификатор (16 бит)</td><td>Флаги (3 бита)</td><td>Смещение фрагмента (13 бит)</td></tr>
<tr><td colspan="2">Время жизни (8 бит)</td><td>Протокол (8 бит)</td><td colspan="2">Контрольная сумма заголовка (16 бит)</td></tr>
<tr><td colspan="5">Адрес отправителя (32 бита)</td></tr>
<tr><td colspan="5">Адрес получателя (32 бита)</td></tr>
<tr><td colspan="2">Опции (поле переменной длины)</td><td colspan="3">Выравнивание до 32-битной границы</td></tr>
</table>

Рис. 7.1. Формат заголовка дейтаграммы протокола IP

Поле «Номер версии» указывает на версию используемого протокола IP. В настоящее время распространена версия 4, но постепенно осуществляется переход к версии 6. Связь между абонентами гарантируется только в том случае, если все они работают с одной версией протокола IP.

Поле «Длина заголовка»- определяет длину заголовка в 32-битовых словах. Заголовок

может иметь минимальный размер 5 слов. При увеличении объема служебной информации эта длина может быть увеличена за счет поля «Опции».

Поле «Тип сервиса» определяет способ обслуживания дейтаграммы. Первые три бита (0-2) этого поля задают приоритет дейтаграммы. Возможные значения приоритета — от 0 (обычная дейтаграмма) до 7 (управляющая дейтаграмма). Устройства в сети учитывают приоритет дейтаграммы и обрабатывают в первую очередь более важные. Информация в остальных битах поля используется протоколами маршрутизации OSPF и BGP. Протоколы маршрутизации отвечают за вычисление наилучшего маршрута к получателю, основываясь на понятии «стоимость пути». Ею может быть скорость, надежность и т. д. Четвертый бит определяет вид задержки: 0 — нормальная задержка, 1 — малая задержка. Этот бит учитывается различными алгоритмами управления перегрузкой сети. Пятый бит определяет пропускную способность (нормальная или высокая). Шестой бит определяет надежность доставки. Седьмой и восьмой биты зарезервированы (0). Отметим, что программное обеспечение большинства рабочих станций и маршрутизаторов игнорирует тип сервиса.

Протокол IP обрабатывает каждую дейтаграмму независимо от других. При этом используются четыре основных механизма: установка типа сервиса, установка времени жизни, установка опций и вычисление контрольной суммы заголовка. Тип сервиса характеризует набор услуг, которые требуются от маршрутизаторов в распределенной сети. Эти параметры должны использоваться для управления выбором реальных рабочих характеристик при передаче дейтаграмм. В некоторых случаях передача дейтаграммы осуществляется с установкой приоритета, который дает данной дейтаграмме по сравнению с остальными некоторые преимущества при обработке. Тип сервиса определяется тремя показателями: малой задержкой при передаче, большой пропускной способностью и высокой достоверностью.

Поле «Время жизни». При определенных условиях IP-дейтаграммы могут попасть в замкнутый логический контур, образованный некоторой группой маршрутизаторов. Иногда такие логические контуры существуют в течение короткого промежутка времени, порой они оказываются достаточно долговечными. Чтобы избавить сеть от дейтаграмм, циркулирующих в таких логических контурах слишком долго, протоколом IP устанавливается предельный срок пребывания дейтаграммы в сети. Он задается в поле «Время жизни» — TTL (Time To Live). Его содержимое уменьшается на единицу при прохождении дейтаграммы через маршрутизатор; при обнулении поля TTL дейтаграмма отбрасывается.

Первоначально спецификации IP включали еще одно требование: поле TTL должно уменьшаться, по крайней мере, один раз в секунду. Поскольку поле TTL является восьмиразрядным, это означает, что дейтаграмма могла находиться в сети не более 4,26 мин. На практике требование ежесекундного уменьшения поля TTL игнорируется, тем не менее, в спецификациях многих протоколов следующих уровней (TCP) по-прежнему предполагается, что максимальное время жизни дейтаграммы в сети составляет лишь две минуты.

Поле - «Общая длина»- указывает общую длину дейтаграммы (заголовок и поле данных). Максимальный размер дейтаграммы может составлять 65535 байт. В подавляющем большинстве сетей столь большой размер дейтаграмм не используется. По стандарту все устройства в сети должны быть готовы принимать дейтаграммы длиной 576 байт. Эти ограничения необходимы для передачи дейтаграмм в физических кадрах. Передача дейтаграммы в кадре называется инкапсуляцией. С точки зрения низших уровней дейтаграмма выглядит так же, как и любое другое сообщение в сети. Сетевое оборудование канального уровня не работает с дейтаграммами, поэтому дейтаграмма является частью области данных кадра (рис. 7.2).

<table>
<tr><td rowspan="2">Заголовок кадра канального уровня</td><td>Заголовок IP-дейтаграммы</td><td>Область данных IP-дейтаграммы</td><td rowspan="2">Контрольная сумма</td></tr>
<tr><td colspan="2">Область данных кадра</td></tr>
</table>

Рис. 7.2. Инкапсуляция дейтаграммы в кадр

Функции фрагментации и сборки также возложены на протокол IP. Фрагментация — это разделение большой дейтаграммы на несколько небольших частей. В большинстве локальных и глобальных сетей есть ограничения на максимальный размер кадра. Эту величину называют

максимальной единицей передачи (Maximum Transmission Unit, MTU). Например, в сетях Ethernet данная величина составляет 1500 байт, а в сетях FDDI — 4096 байт.

Когда маршрутизатор переправляет дейтаграмму из одной сети в другую, может оказаться, что ее размер окажется недопустимым в новой сети. Спецификация IP предусматривает следующее решение этой проблемы: маршрутизатор может разбить дейтаграмму на более мелкие фрагменты, приемлемые для выходной среды, а в пункте назначения эти фрагменты будут вновь объединены в дейтаграмму исходного вида. Формируемые маршрутизатором фрагменты идентифицируются смещением относительно начала исходной дейтаграммы. Дейтаграмма идентифицируется по отправителю, пункту назначения, типу протокола верхнего уровня и 16-разрядному полю «Идентификатор». Все это в совокупности должно образовывать уникальную комбинацию.

Следует подчеркнуть связь между полями «Время жизни» и «Идентификатор». Действительно, во избежание смешивания фрагментов двух разных дейтаграмм источник IP-данных обязан исключить ситуацию, когда в один пункт назначения по одному и тому же протоколу в течение жизненного цикла дейтаграммы будут отправлены две дейтаграммы с совпадающими идентификаторами. В связи с тем, что идентификатор 16-разрядный, а наибольшее время жизни дейтаграммы будем считать равным 2 мин., получаем скорость передачи — 546 дейтаграмм в секунду. В случае её превышения возможно смешивание фрагментов.

Проблема совпадения битов идентификатора может быть решена, если конечная система установит в заголовке IP-дейтаграммы бит DF (Don't Fragment — не фрагментировать), запрещающий фрагментацию. Однако при этом отправители должны заранее узнать величину MTU на пути от отправителя до получателя (используя технологию MTU Path Discovery) и отправлять дейтаграммы, не превышающие её.

Рассмотрим пример фрагментации. Предположим, отправителю необходимо передать сообщение длиной 5600 байт. Отправитель работает в сети, у которой значение MTU составляет 4096 байт. При поступлении пакета на сетевой уровень, протокол IP делит его на две равные дейтаграммы по 2800 байт, устанавливая в первой дейтаграмме признак фрагментации и присваивая пакету уникальный идентификатор. Бит фрагментации во второй дейтаграмме равен нулю, что указывает на последний фрагмент сообщения. Таким образом, дейтаграммы укладываются в кадр физического уровня данной сети (2800 байт данных + 20 байт заголовка меньше 4096 байт).

После маршрутизатора дейтаграммы необходимо передать в сеть с MTU, равным 1500 байт. Для этого маршрутизатор делит поступающие дейтаграммы пополам. Он формирует новые дейтаграммы, каждая из которых имеет размер 1400+20 байт, чтобы уложиться в MTU второй сети. Необходимо отметить, что маршрутизатор не собирает фрагменты в более крупные дейтаграммы, даже если на пути встречается сеть, допускающая такое укрупнение.

Фрагментация и сборка производятся автоматически, не требуя от отправителя специальных действий. Каждая фрагментированная часть исходной дейтаграммы имеет тот же формат. Использование фрагментации повышает вероятность потери исходной дейтаграммы, так как потеря даже одного фрагмента приводит к потере всей дейтаграммы. Сборка дейтаграммы осуществляется на месте назначения. Такой метод позволяет маршрутизировать фрагменты независимо.

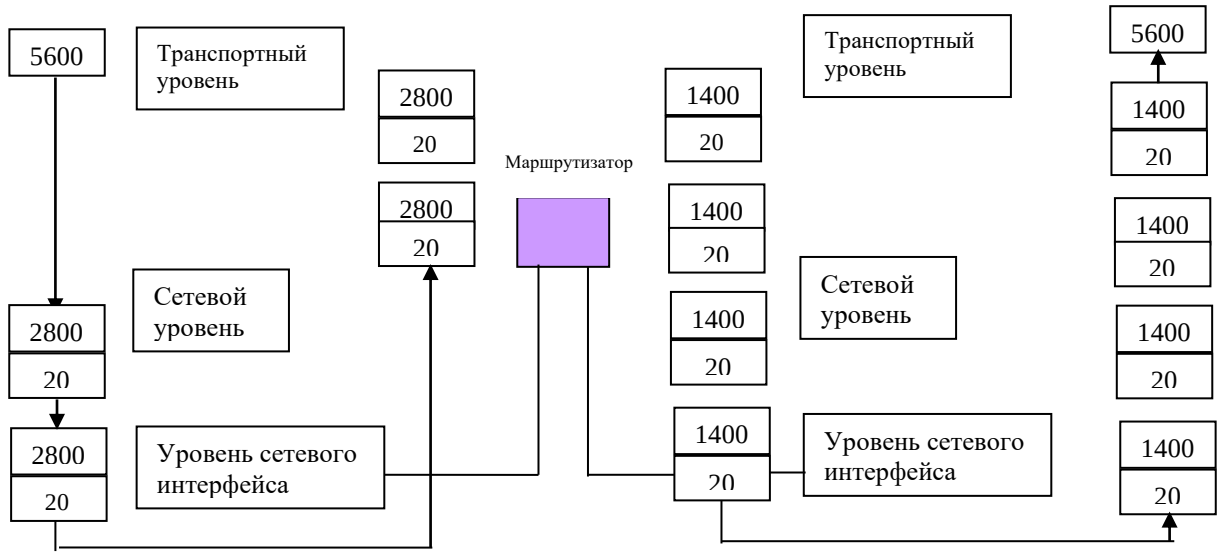

Рис. 7.3. Фрагментация дейтаграммы

Поля «Идентификатор», «Флаги» и «Смещение фрагмента» управляют фрагментацией и сборкой дейтаграммы.

Рассмотрим еще один пример фрагментации дейтаграммы с приведением конкретных значений полей заголовка. На рис. 7.4 показан исходный заголовок дейтаграммы общей длиной 472 байта, пришедшей на входной порт маршрутизатора, для которой MTU на выходном порту был равен 280 байт.

Номер версии = 4	Длина заголовка = 5	Тип сервиса	Общая длина = 472
Идентификатор = 111	Флаги 0 0 0	Смещение фрагмента = 0	
Время жизни = 123	Протокол = 6	Контрольная сумма заголовка	
Адрес отправителя			
Адрес получателя			

Рис. 7.4. Заголовок дейтаграммы до фрагментации

На рис. 7.5 показаны поля заголовков двух полученных в результате фрагментации дейтаграмм, при этом максимальный размер поля данных для них равен 256 байтам.

Поле «Флаги» используется при фрагментации. Первый бит зарезервирован(=0); второй бит DF (0 – можно фрагментировать, 1 – нельзя); третий бит MF указывает на последний фрагмент дейтаграммы (если =0).

Поле «Смещение фрагмента» используется для указания смещения данных во фрагменте относительно начала исходной дейтаграммы в 8-байтных блоках. Чтобы получить смещение в байтах, надо значение этого поля умножить на 8. Первый фрагмент всегда имеет нулевое смещение.

Заголовок фрагмента дейтаграммы #1

Номер версии = 4	Длина заголовка = 5	Тип сервиса	Общая длина = 276
Идентификатор = 111	Флаги 0 0 1	Смещение фрагмента = 0	
Время жизни = 122	Протокол = 6	Контрольная сумма заголовка	
Адрес отправителя			
Адрес получателя			

Заголовок фрагмента дейтаграммы #2

Номер версии = 4	Длина заголовка = 5	Тип сервиса	Общая длина = 216
Идентификатор = 111	Флаги 0 0 0	Смещение фрагмента = 32	
Время жизни = 122	Протокол = 6	Контрольная сумма заголовка	
Адрес отправителя			
Адрес получателя			

Рис. 7.5. Содержание заголовков двух дейтаграмм после фрагментации

Поле «Протокол» показывает, какому протоколу верхнего уровня принадлежит дейтаграмма. При поступлении дейтаграммы это поле указывает, какому приложению следует ее передать. В табл. 7.1 содержится перечень (неполный) протоколов.

Таблица 7.1. Значения поля «Протокол»

Значение	Протокол	Пояснение
0	резерв	
1	ICMP	Internet Control Message Protocol протокол управляющих сообщений
2	IGMP	Internet Group Management Protocol протокол управления группами
4	IP	Инкапсуляция IP в IP
б	TCP	Transmission Control Protocol протокол управления передачей
8	EGP	Exterior Gateway Protocol внешний шлюзовый протокол
17	UDP	User Datagram Protocol протокол пользовательских дейтаграмм
88	IGRP	Interior Gateway Routing Protocol внутренний протокол маршрутизации
89	OSPF	Open Shortest Path First «первый кратчайший путь»

Поле «Контрольная сумма» рассчитывается по всему заголовку. Так как некоторые поля заголовка меняют свое значение, например время жизни, при прохождении дейтаграммы через маршрутизаторы контрольная сумма проверяется и повторно рассчитывается при каждой модификации заголовка. Определение контрольной суммы заголовка обеспечивает безошибочность передачи дейтаграммы через сеть. Перед отправкой дейтаграммы вычисляется контрольная сумма, которая вносится в ее заголовок. При получении дейтаграммы вычисляется ее контрольная сумма, которая сравнивается со значением контрольной суммы в ее заголовке. При обнаружении ошибки в контрольной сумме дейтаграмма отбрасывается.

Поля «Адрес отправителя и «Адрес получателя» имеют одинаковые длину и структуру. Поля содержат 32-битные IP-адреса отправителя и получателя дейтаграммы.

Поле «Опции» необязательно и обычно используется при настройке сети. В поле могут быть указаны точный маршрут прохождения дейтаграммы в распределенной сети, данные о безопасности, различные временные отметки и т. д. Поле не имеет фиксированной длины, поэтому для выравнивания заголовка дейтаграммы по 32-битной границе предусмотрено следующее поле — поле «Выравнивание». Выравнивание осуществляется нулями.

Длина поля «Опции» меняется в зависимости от того, какие опции были выбраны. Опции в дейтаграмме размещаются друг за другом, без разделителей. Каждая опция состоит из кода опции (1 байт), за которым могут следовать длина опции (1 байт) и байты данных этой опции. На рис. 7.6 показан формат байта кода опции.

0	1	2	3	4	5	б	7
Копировать	Класс опции		Номер опции				

Рис. 7.6. Формат байта кода опции

Байт кода опции делится на три поля: флаг «Копировать», «Класс опции» и «Номер опции». Флаг «Копировать» управляет тем, как маршрутизаторы учитывают опции при фрагментации дейтаграммы. Если бит установлен, опции должны копироваться во все фрагменты дейтаграммы. Если флаг не установлен, опцию нужно скопировать только в первый фрагмент. Поля «Класс опции» и «Номер опции» указывают класс опции и номер опции внутри этого класса (табл. 7.2 и 7.3).

Таблица 7.2. Значения поля «Класс опции»

Значение поля	Пояснение
0	Управление дейтаграммами или сетью
1	Зарезервировано
2	Отладка сети
3	Зарезервировано

Из класса 2 применяется опция с номером 4. В нее записываются межсетевые временные метки. Они используются при протоколировании следования дейтаграммы по маршруту.

В настоящее время некоторые опции практически не используются. Например, опция

«Безопасность» с номером 2 (из класса 0) была разработана исключительно для нужд министерства обороны США, и в гражданских сетях не используется. Опция «Идентификатор потока» использовалась только в экспериментах с сетями Satnet и сейчас не встречается.

Таблица 7.3. Номера опций класса 0

Номер опции	Длина	Пояснение
0	-	Конец списка опций. Используется, если опция не заканчивается в конце заголовка
1	-	Нет операций. Используется для выравнивания по 32-битной границе в списке опций
2	11	Безопасность
3	Переменная	Используется для маршрутизации дейтаграммы с учетом информации, предоставленной отправителем (маршрут однозначно не определен)
7	Переменная	Запись маршрута
8	4	Идентификатор маршрута. Используется для поддержки идентификации потока
9	Переменная	Используется для маршрутизации дейтаграммы с учетом информации, предоставленной отправителем (маршрут определен однозначно)
Другой	-	Не используется

7.2.Классификация протоколов маршрутизации.

Существует два подхода к выбору маршрута:

- одношаговый подход;
- маршрутизация от источника.

Согласно методу одношаговой маршрутизации каждый маршрутизатор и конечный узел принимает участие в выборе только одного шага передачи дейтаграммы. В каждой строке таблицы маршрутизации указывается не весь маршрут (в виде последовательности IP-адресов маршрутизаторов, через которые должна пройти дейтаграмма), а только один IP-адрес следующего маршрутизатора (маршрутизатора на том пути, по которому нужно передать дейтаграмму). Вместе с дейтаграммой этому маршрутизатору передается и ответственность за выбор следующего шага. Такой подход распределяет задачу выбора маршрута и снимает ограничение на максимальное количество маршрутизаторов в пути. Кроме того, за счет использования маршрутизатора по умолчанию (который обычно занимает в таблице маршрутизации последнюю строку) существенно сокращается объем таблицы. Все дейтаграммы, номера сетей которых отсутствуют в таблице маршрутизации, передаются маршрутизатору по умолчанию. Подразумевается, что маршрутизатор по умолчанию передает дейтаграмму в магистральную сеть, а маршрутизаторы, подключенные к магистральной сети, имеют полную информации о ее топологии.

Существуют различные алгоритмы построения таблиц для одношаговой маршрутизации. Их делят на три класса:

- **Алгоритмы фиксированной маршрутизации.** Они применяются в сетях с простой топологией и основаны на составлении таблиц маршрутизации «вручную» администратором сети.
- **Алгоритмы простой маршрутизации.** Они разделяются на три подкласса:
  - ✓ случайная маршрутизация (дейтаграммы передаются в любом случайном направлении, кроме исходного);
  - ✓ лавинная маршрутизация (дейтаграммы передаются во всех направлениях, кроме исходного);
  - ✓ по предыдущему опыту (таблица маршрутизации составляется на основании данных, содержащихся в проходящих через маршрутизатор дейтаграммах).
- **Алгоритмы адаптивной маршрутизации.** Основные алгоритмы, применяемые в современных сетях. Маршрутизаторы периодически обмениваются между собой информацией о сетевой топологии. Подразделяются на два подкласса:
  - ✓ дистанционно-векторные (например, Routing Information Protocol (RIP));
  - ✓ состояния связей (например, Open Shortest Path First Protocol (OSPF)).

При маршрутизации от источника выбор маршрута производится конечным узлом или первым маршрутизатором на пути следования дейтаграммы. Все остальные маршрутизаторы

только отрабатывают выбранный маршрут. Этот метод в сетях IP применяется только в целях отладки.

Управление таблицей маршрутизации на маршрутизаторах в большой распределенной сети является сложной задачей. Таблицы маршрутизации для отображения текущей сетевой топологии должны быть динамическими.

Вспомним алгоритм обработки маршрутизатором IP-дейтаграммы:

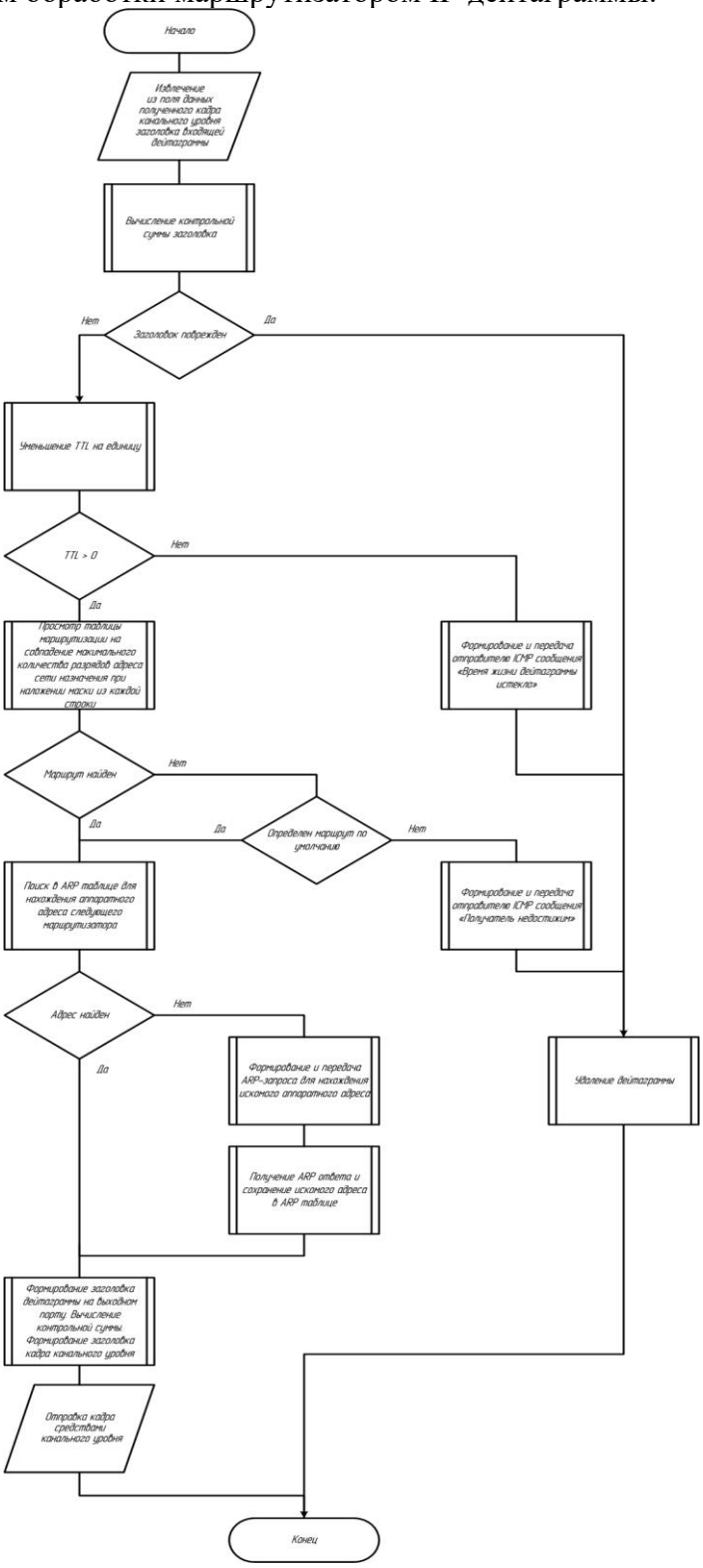

Рисунок 7.7. Алгоритм обработки IP-дейтаграммы маршрутизатором

7.3.Технология бесклассовой междоменной маршрутизации CIDR

С момента начала работы Internet многое изменилось: резко возросло число узлов и

сетей, повысилась интенсивность трафика, изменился характер передаваемых данных. Из-за несовершенства протоколов маршрутизации обмен сообщениями об обновлении таблиц стал иногда приводить к сбоям магистральных маршрутизаторов из-за перегрузки при обработке большого объема служебной информации. Так, в 1994 году таблицы магистральных маршрутизаторов в Internet содержали до 70 000 маршрутов.

На решение этой проблемы была направлена, в частности, и технология бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR), впервые о которой было официально объявлено в 1993 году, когда были опубликованы RFC 1517, RFC 1518, RFC 1519 и RFC 1520.

Суть технологии CIDR заключается в следующем. Каждому поставщику услуг Internet должен назначаться непрерывный диапазон в пространстве IP-адресов. При таком подходе адреса всех сетей каждого поставщика услуг имеют общую старшую часть — префикс, поэтому маршрутизация на магистралях Internet может осуществляться на основе префиксов, а не полных адресов сетей. Агрегирование адресов позволит уменьшить объем таблиц в маршрутизаторах всех уровней, а следовательно, ускорить работу маршрутизаторов и повысить пропускную способность Internet.

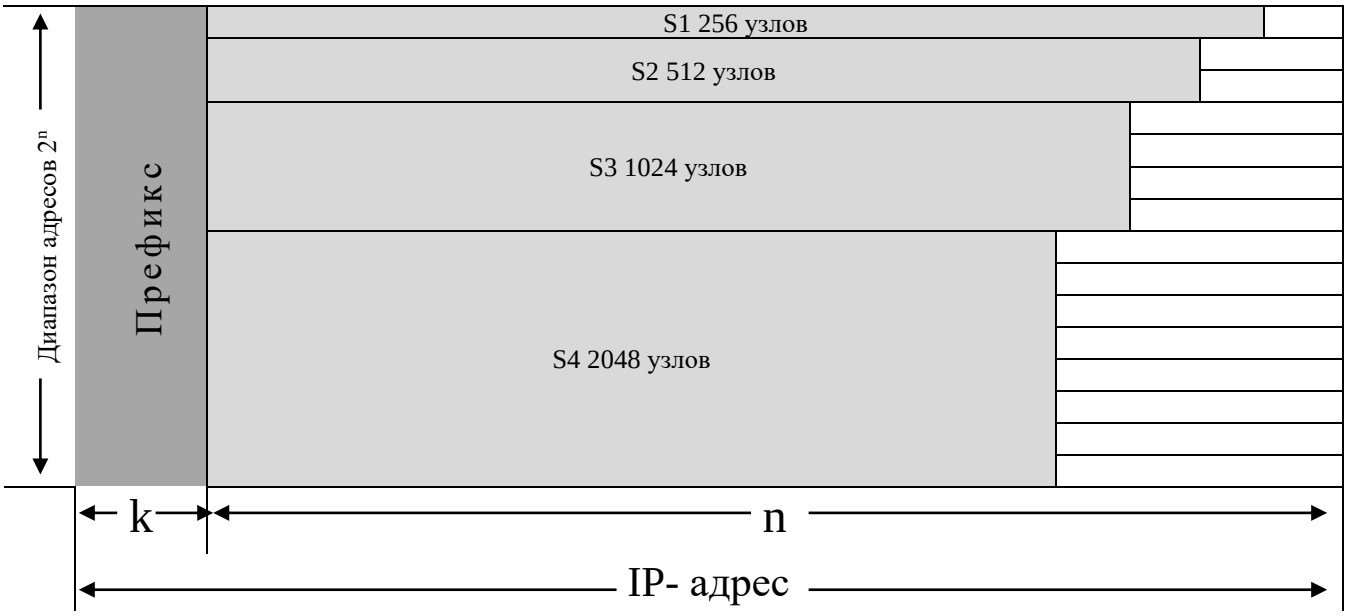

Рис. 7.8. Технология CIDR

Деление IP-адреса на номер сети и номер узла в технологии CIDR происходит не на основе нескольких старших бит, определяющих класс сети (A, B или C), а на основе маски переменной длины, назначаемой поставщиком услуг. На рис. 7.8 показан пример некоторого пространства IP-адресов, которое имеется в распоряжении гипотетического поставщика услуг. Все адреса имеют общую часть в k старших разрядах — префикс. Оставшиеся n разрядов используются для дополнения неизменяемого префикса переменной частью адреса. Диапазон имеющихся адресов в таком случае составляет 2n. Когда потребитель услуг обращается к поставщику услуг с просьбой о выделении ему некоторого количества адресов, то в имеющемся пуле адресов «вырезается» непрерывная область Sl, S2, S3 или S4 соответствующего размера. Причем границы этой области выбираются такими, чтобы для нумерации требуемого числа узлов хватило некоторого числа младших разрядов, а значения всех оставшихся (старших) разрядов было одинаковым у всех адресов данного диапазона. Таким условиям могут удовлетворять только области, размер которых кратен степени двойки. А границы выделяемого участка должны быть кратны требуемому размеру.

Рассмотрим пример. Пусть поставщик услуг Internet располагает пулом адресов в диапазоне 193.20.0.0-193.23.255.255 (11000001.00010100.00000000.00000000-11000001.00010111.11111111.11111111) с общим префиксом 193.20 (11000001.000101) и маской, соответствующей этому префиксу 255.252.0.0

Если абоненту этого поставщика услуг требуется совсем немного адресов, например 13,

то поставщик мог бы предложить ему различные варианты: например, сеть 193.20.30.0, сеть 193.20.30.16 или сеть 193.21.204.48, все с одним и тем же значением маски 255.255.255.240. Во всех случаях в распоряжении абонента для нумерации узлов имеются 4 младших бита.

Рассмотрим другой вариант, когда к поставщику услуг обратился крупный заказчик, сам, возможно собирающийся оказывать услуги по доступу в Internet. Ему требуется блок адресов в 4000 узлов. В этом случае поставщик услуг мог бы предложить ему, например, диапазон адресов 193.22.160.0-193.22.175.255 с маской 255.255.240.0. Агрегированный номер сети (префикс) в этом случае будет равен 193.22.160.0.

Администратор маршрутизатора М2 (рис. 7.9) поместит в таблицу маршрутизации только по одной записи на каждого клиента, которому был выделен пул адресов, независимо от количества подсетей, организованных клиентом. Если клиент, получивший сеть 193.22.160.0, через некоторое время разделит ее адресное пространство в 4096 (на самом деле 4096-2 т.к. адреса из всех нулей и единиц в поле адреса узла не используются) адресов на 8 подсетей, то в маршрутизаторе М2 первоначальная информация о выделенной ему сети не изменится.

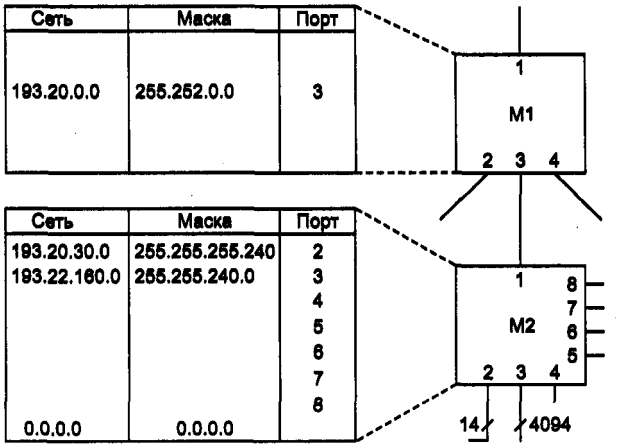

Рис. 7.9. Выигрыш в количестве записей в маршрутизаторе при использовании технологии CIDR

Для поставщика услуг верхнего уровня, поддерживающего клиентов через маршрутизатор Ml, усилия поставщика услуг нижнего уровня по разделению его адресного пространства также не будут заметны. Запись 193.20.0.0 с маской 255.252.0.0 полностью описывает сети поставщика услуг нижнего уровня в маршрутизаторе Ml.

Итак, внедрение технологии CIDR позволяет решить две основные задачи:

• Более экономное расходование адресного пространства. Действительно, получая в свое распоряжение адрес сети, например, класса С, некоторые организации не используют весь возможный диапазон адресов просто потому, что в их сети имеется гораздо меньше 255 узлов. Технология CIDR позволяет получать в пользование столько адресов, сколько реально необходимо. Благодаря технологии CIDR поставщики услуг получают возможность «нарезать» блоки из выделенного им адресного пространства в точном соответствии с требованиями каждого клиента, при этом у клиента остается пространство для маневра на случай его будущего роста.

• Уменьшение числа записей в таблицах маршрутизаторов за счет объединения маршрутов — одна запись в таблице маршрутизации может представлять большое количество сетей. Действительно, для всех сетей, номера которых начинаются с одинаковой последовательности цифр, в таблице маршрутизации может быть предусмотрена одна запись (см. рис. 7.9). Так, маршрутизатор М2 установленный в организации, которая использует технику CIDR для выделения адресов своим клиентам, должен поддерживать в своей таблице маршрутизации все 8 записей о сетях клиентов. А маршрутизатору Ml достаточно иметь одну запись о всех этих сетях, на основании которой он передает пакеты с префиксом 193.20 маршрутизатору М2, который их и распределяет по нужным портам. Особенно заметный выигрыш будет достигаться в магистральных маршрутизаторах.

Использование CIDR в сетях IPv4 в общем случае требует перенумерации сетей.

Поскольку эта процедура сопряжена с определенными временными и материальными затратами, для ее проведения пользователей нужно каким-либо образом стимулировать. В качестве таких стимулов рассматривается, например, введение оплаты за строку в таблице маршрутизации или же за количество узлов в сети с реальными IP адресами.

7.4. Протокол IPv6

Работа по расширению протокола IP была начата в 1992 году. Необходимость этого диктовалось тем, что практически все ресурсы старой версии протокола IP (IPv4) были исчерпаны. Быстрый рост сети Internet привел к появлению дефицита IP-адресов. Возросший трафик начал вызывать перегрузки магистральных маршрутизаторов. Изменился и характер передаваемого трафика. Все большую долю в нем стали занимать мультимедийные данные.

Новая версия протокола IP — версия 6 (IPv6) — была принята организацией IETF в 1995 году. Она описана в документе RFC 1752. В настоящее время осуществляется постепенный переход к протоколу IPv6. Существует несколько фрагментов сети Internet, в которых маршрутизаторы поддерживают обе версии IP. Эти фрагменты объединены между собой и образуют так называемую «шестую» магистраль (6 bone). Для того чтобы передать дейтаграммы протокола IPv6, магистраль 6 bone инкапсулирует их в дейтаграмму IP и передает через те части сети Internet, которые не поддерживают новую версию протокола. Этот процесс называется туннелированием. Следует помнить, что появление дополнительного заголовка при туннелировании ведет к росту сетевого трафика. Документ RFC 1933 определяет четыре конфигурации туннелей между рабочими станциями и маршрутизаторами:

- маршрутизатор—маршрутизатор;
- рабочая станция—маршрутизатор;
- рабочие станции – маршрутизатор;
- маршрутизатор – рабочая станция.

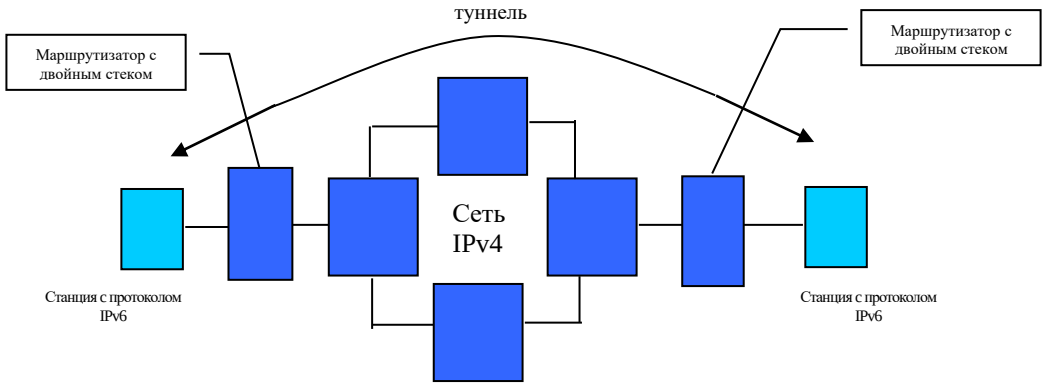

Рис. 7.10. Туннелирование дейтаграммы IPv6, инкапсулированной в дейтаграмму IPv4.

Другим методом, позволяющим осуществить плавный переход на новую версию, является использование двойных стеков. Двойные стеки позволяют узлу в сети IP поддерживать обе версии протокола. Такие узлы называются IPv6/IPv4-узлами. Использование двойного стека позволяет раздельно переводить на протокол IPv6 каждое устройство в сети. При этом необходимо задействовать дополнительные ресурсы такого устройства, изменить его конфигурационную информацию и провести ряд других операций. Нужно учитывать, что маршрутизаторам может потребоваться дополнительная оперативная память, так как таблицы маршрутизации протокола IPv6 больше по объему. На рис. 7.11 показано распределение уровней узла с двойным стеком IPv4/IPv6.

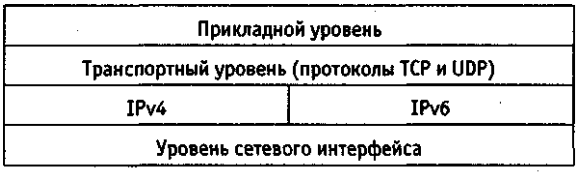

Рисунок 7.11. Уровни двойного стека TCP/IP

Протокол IPv6 поддерживается всеми современными операционными системами и производителями сетевого оборудования. Он уменьшил объем маршрутной и служебной информации. Кроме этого произведены изменения, которые напрямую влияют на загрузку маршрутизаторов:

- реализована гибкая схема разделения адресного пространства с использованием технологий CIDR и масок подсетей переменной длины. Изменение адресной схемы позволяет сократить объем таблиц маршрутизации и ускорить их просмотр и обновление;
- введено повсеместное использование физического адреса устройства в качестве номера узла. При этом снижается нагрузка на сеть за счет отказа от протокола ARP;
- проведение фрагментации перенесено на конечные узлы. Узлы, поддерживающие протокол IPv6, сами определяют размер MTU, который устраивает все транзитные узлы и каналы на пути следования дейтаграммы.
- с целью повышения производительности и с расчётом на то, что современные технологии канального и транспортного уровней обеспечивают достаточный уровень обнаружения ошибок, заголовок не имеет поля для подсчета контрольной суммы.

Схема адресации IPv6 существенно отличается от схемы адресации протокола IP. Адреса получателя и отправителя в протоколе IPv6 задаются 128 битами. Такая длина адресного пространства позволяет на достаточно большой период времени снять проблему дефицита адресов в сети Internet. Основным механизмом, заложенным в схему адресации протокола IPv6, является введение иерархического разделения адресного пространства на уровни. Вместо прежних двух уровней — номера сети и номера устройства, — в протоколе IPv6 используется пять уровней, включая два уровня идентификации провайдеров и три уровня идентификации абонентов в сети (рис. 7.12).

Префикс	Идентификатор провайдера	Идентификатор абонента	Идентификатор подсети	Идентификатор узла

Рисунок 7.12. Уровни адресации протокола IPv6

Префикс определяет тип используемого адреса. Приведем пример адреса с идентификацией провайдера. Такой адрес имеет префикс 010. Этот префикс выбран согласно табл. 7.4, в которой приведено исходное распределение адресов протокола IPv6.

Таблица 7.4. Исходное распределение адресов IPv6.

Назначение блока адресов	Двоичный префикс	Доля адресного пространства
Резервный	0000 0000	1/256
Незанятый	0000 000	11/258
Зарезервирован для IPX	0000 010	1/128
Незанятый	0000 011	1/128
Незанятый	0000 1	1/32
Незанятый	0001	1/16
Незанятый	001	1/8
Адреса идентификации провайдера	010	1/8
Незанятый	011	1/8
Зарезервирован для адресов по географической принадлежности	100	1/8
Незанятый	101	1/8
Незанятый	110	1/8
Незанятый	1110	1/16
Незанятый	11110	1/32
Незанятый	1111 10	1/64
Незанятый	1111 110	1/128
Незанятый	1111 1110 0	1/512
Локальные адреса для линии	1111 1110 10	1/1024
Локальные адреса для узла	1111 1110 11	1/1024
Групповые адреса	1111 1111	1/256

На рисунке 7.13 показан формат адреса в случае идентификации провайдера

Пре-фикс	Идентификатор организации	Идентификатор провайдера	Зарезервировано	Идентификатор абонента	Зарезервировано	Адрес сети и устройства
010	5 бит	16 бит	8 бит	24 бита	8 бит	64 бита

Рисунок 7.13 формат адреса с префиксом 010

Поле «Идентификатор организации» определяет организацию, ответственную за выделение адресов провайдерам (регионального интернет-регистратора, аккредитованного ICANN: ARIN, RIPE NCC, APNIC, LACNIC, AfriNIC). Поле «Идентификатор провайдера» определяет непосредственно провайдера. Провайдер назначает поле «Идентификатор абонента». Данное поле идентифицирует конкретную организацию предприятие или частное лицо, получившую непрерывный диапазон адресов для своих подсетей. За полями «Идентификатор провайдера» и «Идентификатор абонента» следуют резервные поля, необходимые для будущего расширения. Оставшиеся 64 бита в адресе по принадлежности к провайдеру определяют номер сети и номер устройства. Данное поле предоставляет достаточно пространства для разбиения выделенного блока адресов на адреса подсетей и рабочих станций в каждой подсети. Учитывая, что в качестве адреса узла используется MAC адрес, для адресации подсетей остается ещё 16 разрядов

Такая структура адреса по принадлежности к провайдеру значительно упрощает маршрутизацию. Поле «Идентификатор провайдера» сразу определяет сеть другого провайдера. После определения сети провайдера маршрутизатор анализирует поле «Идентификатор абонента» и определяет непосредственного абонента, которому должна быть передана информация. Далее маршрутизация осуществляется в сети самого абонента, чтобы доставить пакет конкретному узлу.

В протоколе IPv6 отменено разделение адресов на классы. В основе распределения адресного пространства лежит технология CIDR. При этом адреса сетей каждого провайдера имеют одинаковое значение сетевого префикса и все устройства в этой сети поддерживают его передачу. Деление IP-адреса на адрес подсети и адрес устройства производится на основе маски подсети переменной длины и уже не зависит от класса адреса.

Протокол IPv6 вводит несколько типов адресов:

- Unicast — индивидуальный (единичный) адрес. Адрес определяет отдельное устройство в сети или порт маршрутизатора. В свою очередь, индивидуальный адрес подразделяется на:
  - ❖ Global — глобальный. Основной тип адресов в Internet;
  - ❖ Link-Local — локальный адрес для линии. Адреса используются в сетях, не связанных с Internet. Поэтому поля идентификаторов заполняются нулями, число которых определяется требуемым остатком бит после задания адреса линии или узла. Термин «Link» относится к сетям Frame Relay и ATM, то есть к прямой выделенной линии или соединению с сетью Ethernet, FDDI и т. д. С использованием этих адресов можно впоследствии подключать к Internet сети без присвоения им новых адресов. Локальный адрес для линии действителен только в пределах сетевого сегмента канального уровня и используется, в основном, для обмена информационными ICMPv6 пакетами. Он присваивается самим узлом при первом включении с целью автоконфигурации. Узел может запросить информацию о настройках сети у ближайшего к нему маршрутизатора, отправив ICMPv6 сообщение «Router Solicitation» на групповой адрес маршрутизаторов. Маршрутизатор, получивший это сообщение, отвечает ICMPv6 сообщением «Router Advertisement», в котором может содержаться информация о сетевом префиксе, адресе шлюза, адресах DNS серверов, MTU и множестве других параметров. Объединяя сетевой префикс и идентификатор интерфейса, узел получает новый глобальный адрес.
  - ❖ Unic-Local соответствуют внутренним IP адресам, которыми в версии IPv4 являлись 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16. Начинаются с цифр FC00 и FD00.

На рис. 7.14 показан пример локального адреса для линии. В формате локального адреса для линии поле уникального адреса линии содержит физический адрес локальной сети. Префикс имеет длину 10 бит, а оставшаяся часть — 118 бит. Если локальной адрес для линии используется для подключения к сети Ethernet, то физический адрес займет 48 бит (6 байт).

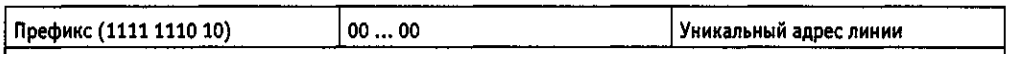

Рисунок 7.14. Адрес для линии.

- Multicast — адрес набора узлов (групповой адрес). В протоколе IPv6 отсутствует понятие широковещательного адреса. Широковещательная адресация заменена поддержкой групповой передачи данных. Такой механизм необходим протоколу IPv6 для регулирования пропускной способности сети при распространении мультимедийного трафика;
- Anycast — адрес набора узлов. Этот тип адресов используется для обеспечения прохождения определенного трафика через маршрутизаторы отдельных провайдеров. В отличие от групповых адресов, такая дейтаграмма должна быть доставлена любому члену группы. В протоколе IPv6 широко используется маршрутизация от источника. Этот вид маршрутизации освобождает маршрутизаторы от функции анализа своих таблиц маршрутизации, уменьшает время задержки дейтаграммы для ее обработки и, естественно, повышает пропускную способность сети в целом. При назначении адресов каждому порту маршрутизатора вместе с физическим адресом присваивается еще один адрес, общий для всех портов всех маршрутизаторов в сети данного провайдера, который является апусast-адресом. При указании нечеткого адреса устройству не надо знать конкретный адрес маршрутизатора, так как он является членом группы маршрутизаторов с этим адресом. Положительным моментом здесь является то, что в случае изменения местоположения этого маршрутизатора адрес его менять не надо, так как дейтаграмма будет отправляться по-прежнему ближайшему члену этой нечеткой группы.

Для плавного перехода к протоколу IPv6 введен специальный тип адресов — IPv4-compatible (совместимые адреса). В этих адресах старшие 96 бит содержат нули, а младшие 32 бита — обычный адрес IPv4. Такие адреса позволяют решить проблему совместимости частей Internet, работающих с протоколами IP разных версий.

Для упрощения обработки заголовка дейтаграммы в протоколе IPv6 введены основной и дополнительный заголовки. Основной заголовок присутствует всегда. Дополнительный заголовок определяет некоторые необязательные параметры. Основной заголовок имеет длину 40 байт (рис. 7.15).

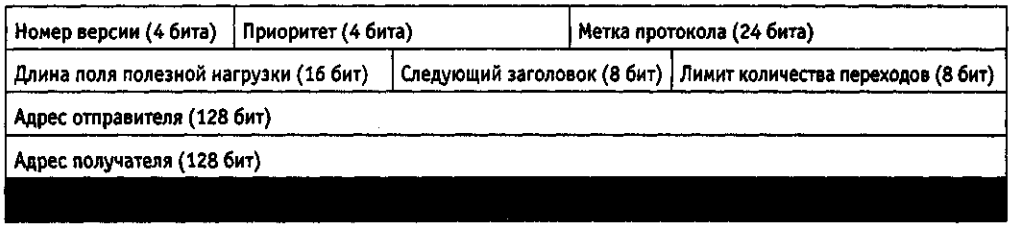

Рисунок 7.15. Формат основного заголовка IPv6

Поле «Следующий заголовок» по своему назначению соответствует полю «Протокол» в версии 4 и определяет тип заголовка, который следует за данными. Каждый следующий дополнительный заголовок содержит это поле. Если дейтаграмма не содержит дополнительных заголовков, то значение этого поля определяет протокол — TCP, UDP, RIP или OSPF. Поле «Лимит переходов» - аналог поля времени жизни дейтаграммы (TTL) в IPv4. Поле «Приоритет» позволяет отправителю задать приоритет своих дейтаграмм. Возможные 16 значений этого поля разделены на две категории: значения от 0 до 7 определяют трафик, которым маршрутизатор при необходимости может пренебречь, а значения от 8 до 15 указывают на трафик, к которому эти меры применяться не могут (аудио- и видеоинформация, передаваемая с постоянной скоростью в реальном времени). Используя поля «Приоритет» и «Метка» устройства могут идентифицировать дейтаграммы, которым требуется нестандартное обслуживание на маршрутизаторах. Для поддержки качества обслуживания протокол IPv6 работает с «меткой потока» (flow label). Метка потока — это признак, который размещается в поле заголовка «Метка протокола» дейтаграммы IPv6. Метка указывает на принадлежность данной дейтаграммы к последовательности дейтаграмм — потоку, который требует определенных параметров обслуживания. Маршрутизаторы обрабатывают потоки на основании значения метки и идентификатора отправителя дейтаграмм. Для предоставления нестандартного качества

обслуживания потоков разработан дополнительный протокол RSVP — протокол резервирования ресурсов.

В протоколе IPv6 определены следующие типы дополнительных заголовков:

- **Routing** — определяет полный маршрут при маршрутизации от источника. Данный заголовок позволяет отправителю указать список IP-адресов, которые диктуют путь передачи;
- **Fragmentation** — содержит сведения о проведении фрагментации на конечных узлах сети. В этом подзаголовке используются поля аналогичные тем, что отвечают за фрагментацию и сборку в основном заголовке IPv4. В протоколе IPv6 фрагментацию не разрешается выполнять на промежуточных узлах; это значительно повышает производительность при маршрутизации. В том случае, если распределенная сеть состоит из сегментов с различными значениями MTU, отправитель использует дополнительный заголовок Fragmentation для разделения дейтаграммы на произвольное число небольших фрагментов. В этом дополнительном заголовке содержатся поля, которые идентифицируют фрагменты исходной дейтаграммы по присвоенным им последовательным номерам. Так как промежуточные маршрутизаторы не выполняют фрагментацию, то вся ответственность за выбор правильного размера дейтаграммы возлагается на отправителя, которому необходимо определить значения MTU каждой промежуточной сети в пути до получателя, используя механизм MTU path discovery process (процесс выяснения значений MTU на пути), описанный в документе RFC 1191. При этом отправитель посылает дейтаграмму с длиной, равной значению MTU той сети, к которой он подключен. Если выбранный размер дейтаграммы слишком велик для некоторых промежуточных сетей, то отправителю будет послано сообщение протокола ICMP «Datagram Too Big» (Дейтаграмма слишком велика) с указанием рекомендованного значения MTU. После получения этого сообщения отправитель скорректирует размер дейтаграммы (например, с помощью фрагментации) и повторит процесс. Это будет продолжаться до тех пор, пока дейтаграмма не сможет пройти все промежуточные сети в пути до получателя;
- **Authentication** — служит для идентификации конечных узлов;
- **Encryption** — служит для шифрования и дешифровки передаваемых данных;
- **Hop-By-Hop** Option —Данный заголовок переносит дополнительные параметры, которые проверяются и промежуточными и конечными узлами. Заголовок должен следовать первым после основного заголовка. Так как заголовок проверяется всеми маршрутизаторами, его полезно использовать для передачи управляющей или отладочной информации. Например может использоваться параметр Router Alert, который информирует маршрутизаторы, что дейтаграмма должна быть обработана целиком до начала ее передачи следующему маршрутизатору в пути. Данный параметр применяется, например, при работе протокола RSVP;
- **Destination Option** — содержит дополнительную информацию для узла назначения.

Каждый дополнительный заголовок содержит тип следующего за ним заголовка, что позволяет создать цепочку заголовков. Основной заголовок является первым в цепочке и не содержит дополнительных заголовков. Поле «Следующий заголовок» указывает, какой дополнительный заголовок следует за основным. Поле «Следующий заголовок» первого дополнительного заголовка указывает на тип второго дополнительного заголовка и т. д. Это продолжается до тех пор, пока в поле «Следующий заголовок» очередного дополнительного заголовка не встретится запись о том, что далее, например, следует заголовок протокола TCP (рис. 7.16).

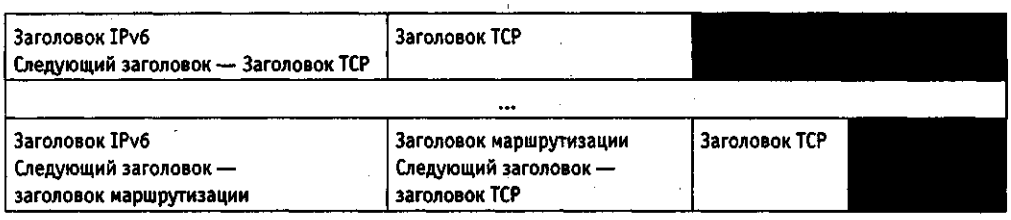

Рисунок 7.16. Формирование цепочки заголовков