RFC 791 INTERNET PROTOCOL (IP)

RFC: 791
Replaces: RFC 760
IENs 128, 123, 111,
80, 54, 44, 41, 28, 26

Спецификация протокола IP

PROTOCOL SPECIFICATION

INTERNET PROTOCOL

DARPA INTERNET PROGRAM

PROTOCOL SPECIFICATION

September 1981

prepared for

Defense Advanced Research Projects Agency

Information Processing Techniques Office

1400 Wilson Boulevard

Arlington, Virginia 22209

by

Information Sciences Institute

University of Southern California

4676 Admiralty Way

Marina del Rey, California 90291

Предисловие

Этот документ содержит спецификацию стандарта DoD[1] для протокола Internet (IP[2]). Документ основан на 6 предварительных вариантах спецификации протокола ARPA Internet и содержит фрагменты этих спецификаций. В разработке используемых в документе концепций и терминологии принимало участие множество людей. В данной редакции пересмотрены вопросы адресации, обработки ошибок, кодирования опций, приоритетов, изоляции (compartments) и ограничений протокола Internet.

Jon Postel,

редактор

За время, прошедшее с момента завершения данного документа, протокол IP стал одним из самых распространенных протоколов сетевого уровня OSI/ISO и сегодня этот протокол используется практически на каждом компьютере. Вместе с протоколом за прошедшие годы сильно изменилось толкование некоторых используемых в документе терминов и в переводе используются термины в их современном толковании, дабы не порождать путаницы.

Термин gateway (шлюз) в исходном документе использовался для обозначения устройств, которые сегодня называют маршрутизаторами (router), а термином шлюз сегодня обозначают обычно устройства (программы), обеспечивающие преобразование протоколов на более

высоких уровнях модели ISO/OSI (например, почтовые шлюзы).

Часто встречающийся в исходном документе термин local network interface переводится как «интерфейс канального уровня», в соответствии с современной терминологией.

Николай Малых,

переводчик

(PDF)

1. Введение

1.1. Мотивация

Протокол IP предназначен для использования в соединенных между собой компьютерных сетях обмена данными на основе коммутации пакетов. Такие системы получили название catenet [1]. Протокол обеспечивает передачу блоков данных, называемых дейтаграммами, между отправителем и получателем, хосты которых идентифицируются адресами фиксированной длины. Протокол также обеспечивает фрагментацию и сборку дейтаграмм большого размера, если сеть не позволяет передать дейтаграмму целиком.

1.2. Сфера действия протокола

Протокол IP ограничивается доставкой битовых пакетов (дейтаграмм) от отправителя к получателю через систему соединенных между собой сетей. Протокол не поддерживает механизмов повышения надежности сквозной доставки, управления потоком данных, сохранения порядка и других функций, общепринятых для протоколов прямого взаимодействия между хостами. Протокол IP использует услуги поддерживающих этот протокол сетей для предоставления услуг различного типа и с разным качеством.

1.3. Интерфейсы

Этот протокол вызывается протоколами взаимодействия “хост-хост”[3] и сам вызывает функции локальных сетевых протоколов[4] для передачи дейтаграмм следующему маршрутизатору или хосту-получателю.

Например, модуль TCP будет вызывать модуль IP для размещения сегмента TCP (заголовок TCP и пользовательские данные) в качестве объекта данных дейтаграммы IP. Модуль TCP будет указывать адреса и другие параметры заголовка IP в качестве аргументов при вызове функции IP. Модуль IP будет создавать дейтаграмму IP и обращаться к локальному сетевому интерфейсу[5] для передачи дейтаграммы.

Для случая ARPANET, например, модуль IP будет вызывать модуль локальной сети, который добавит заголовок типа 1822 [2] к дейтаграмме, создавая сообщение ARPANET для передачи IMP. Адрес ARPANET определяется из адреса IP интерфейсом с локальной сетью и будет принимать значение адреса какого-либо из хостов ARPANET, который может быть шлюзом в другую сеть.

1.4. Работа протокола

Протокол IP выполняет две основных функции – адресацию и фрагментацию/сборку дейтаграмм.

Модули IP используют адреса из заголовков IP для передачи дейтаграмм в направлении

получателя. Процесс выбора пути к адресату называется маршрутизацией.

Модули IP используют поля заголовков IP для фрагментации и сборки дейтаграмм IP при необходимости передачи через сети с малым размером пакетов.

Модули IP используются на каждом хосте, участвующем в сети, и на каждом маршрутизаторе, соединяющем сети. Эти модули используют общие правила интерпретации полей адреса и фрагментации/сборки дейтаграмм IP. Кроме того, эти модули (особенно в маршрутизаторах) выполняют процедуры принятия решения о пересылке дейтаграмм и еще ряд функций.

Протокол IP трактует каждую дейтаграмму как независимый элемент, не связанный с другими дейтаграммами IP. Протокол не использует (явных) соединений или логических устройств (виртуальных или иных).

Для обеспечения сервиса протокол IP использует 4 ключевых механизма – ToS (тип обслуживания), TTL (время жизни), Options (опции) и Header Checksum (контрольная сумма заголовка).

Тип обслуживания (ToS) используется для индикации желаемого качества сервиса. ToS представляет собой абстрактный или обобщенный набор параметров, характеризующих выбранный сервис, который обеспечивается в сетях, образующих Internet. Индикация ToS используется маршрутизаторами для выбора реальных параметров передачи применительно к конкретной сети, следующего интервала или следующего маршрутизатора при доставке дейтаграмм IP.

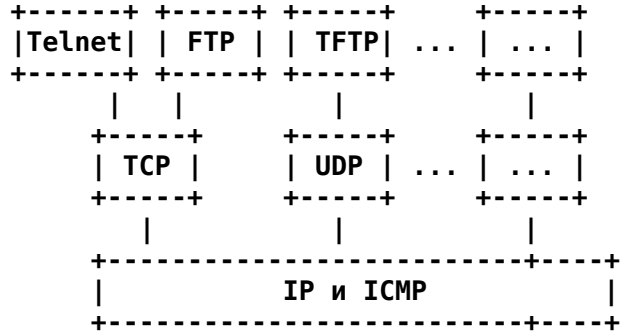
Время жизни TTL определяет максимальный срок существования дейтаграмм IP. Это значение устанавливается отправителем и уменьшается в каждой точке на пути доставки, где дейтаграмма подвергается обработке. Если значение TTL становится нулевым до того, как дейтаграмма будет доставлена адресату, такая дейтаграмма просто уничтожается. Можно рассматривать TTL как время саморазрушения дейтаграмм.

Опции обеспечивают функции контроля, требуемые или полезные в некоторых ситуациях, но не используемые для большинства рутинных задач. Опции включают временные метки, параметры безопасности и специальные средства маршрутизации.

Контрольная сумма заголовка обеспечивает возможность проверки корректности передачи дейтаграмм IP. Если при передаче дейтаграмма была повреждена, вычисленная заново при обработке контрольная сумма не совпадет с содержащимся в дейтаграмме значением контрольной суммы, такая дейтаграмма отбрасывается как ошибочная.

Протокол IP не обеспечивает механизма гарантированной доставки. В протоколе не используется подтверждений (сквозных или поэтапных) доставки или средств контроля ошибок (за исключением контрольных сумм заголовка). Протокол также не поддерживает средств повтора передачи и управления потоком данных.

При обнаружении ошибок информация о них может передаваться с помощью протокола ICMP (Internet Control Message Protocol) [3], реализуемого в модуле IP.



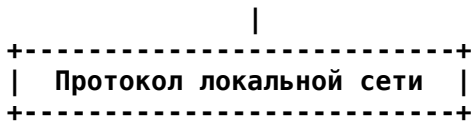


Рисунок 1 Связь с другими протоколами

2. Обзор

2.1. Связь с другими протоколами

На рисунке 1 показаны связи IP с другими протоколами.

Протокол IP взаимодействует с протоколом вышележащего уровня (протоколы взаимодействия между хостами – host-to-host[6]) и с нижележащим протоколом локальной сети[7] (в этом контексте локальной сетью может считаться небольшая сеть в одном здании или распределенная сеть типа ARPANET).

2.2. Модель работы протокола

Модель передачи дейтаграмм от одной прикладной программы к другой можно проиллюстрировать описанным ниже сценарием.

Будем предполагать, что передача включает лишь один промежуточный шлюз.

Передающая программа готовит свои данные и вызывает локальный модуль IP для передачи этих данных, как дейтаграммы, указывая адрес получателя и другие параметры в качестве аргументов.

Модуль IP готовит заголовок дейтаграммы и присоединяет к нему данные. После этого модуль IP определяет локальный сетевой адрес для указанного получателя (в данном случае это адрес шлюза).

Модуль передает дейтаграмму и локальный адрес локальному сетевому интерфейсу[8].

Интерфейс канального уровня создает заголовок и присоединяет к нему дейтаграмму IP, после чего пакет передается в локальную сеть.

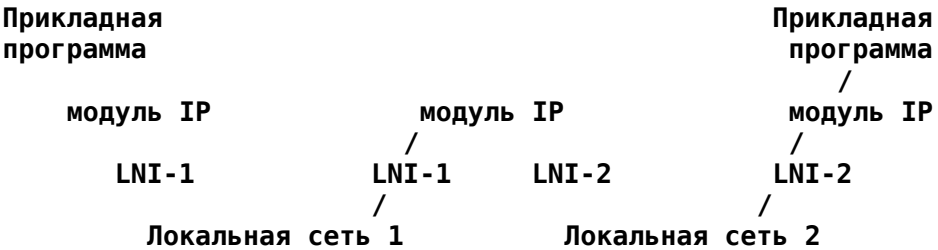


Рисунок 2 Путь передачи данных

Дейтаграмма приходит на хост-шлюз в кадре канального уровня. Интерфейс канального уровня удаляет заголовок канального уровня и передает дейтаграмму модулю IP, который определяет по адресу IP, что дейтаграмму следует переслать хосту другой сети. Тогда модуль IP определяет адрес канального уровня для пересылки дейтаграммы получателю и вызывает интерфейс канального уровня той сети, куда будет передаваться дейтаграмма.

Интерфейс канального уровня создает заголовок и, добавив к нему дейтаграмму, передает пакет хосту-адресату.

На хосте получателя дейтаграмма выделяется из пакета интерфейсом канального уровня и

передается модулю IP.

Модуль IP определяет по заголовку, что дейтаграмма адресована приложению на данном хосте и передает программе данные из дейтаграммы вместе с адресом отправителя и другими параметрами в ответ на системный вызов.

2.3. Функциональное описание

Задачей протокола IP является перемещение дейтаграмм через множество соединенных между собою сетей. Эта задача решается путем передачи дейтаграмм от одного модуля IP к другому, пока дейтаграмма не будет доставлена адресату. Модули IP размещаются на хостах и шлюзах (маршрутизаторах) Internet. Дейтаграммы маршрутизируются от одного модуля IP к другому через промежуточные сети на основе интерпретации адресов IP. Таким образом, одним из важнейших механизмов IP является адресация.

При маршрутизации сообщений от одного модуля IP к другому может потребоваться передача дейтаграмм через сети, для которых максимальный размер пакета меньше размера дейтаграммы. Для решения этой проблемы протокол IP обеспечивает механизмы фрагментации и сборки дейтаграмм.

Адресация

Следует различать имена, адреса и маршруты [4]. Имя указывает объект, адрес показывает местонахождение объекта, а маршрут говорит, как до него добраться. Протокол IP имеет дело преимущественно с адресами. Отображение адресов на имена и обратно (преобразование) является задачей протоколов более высоких уровней (т. е., транспортного и сеансового[9]). Модуль IP преобразует адреса IP в адреса локальной сети. Отображение адресов локальной сети на маршруты является задачей процедур нижележащего уровня (т. е.. локальной сети или шлюзов)[10].

Адреса IP имеют фиксированную длину – 4 октета (32 бита). Адрес начинается с номера сети, за которым следует локальный адрес[11] (его называют полем rest – остаток). Существует три класса адресов IP – класс A, в котором старший бит имеет значение 0, остальные 7 битов старшего октета задают номер сети, а 24 младших бита – номер хоста; класс B, в котором два старших бита имеют значения 10, следующие 14 битов определяют номер сети, а последние 16 битов – номер хоста; класс C, в котором три старших бита имеют значения 110, следующие 21 – образуют номер сети, а последние 8 битов определяют номер хоста[12].

Следует с осторожностью относиться к преобразованию адресов IP в адреса локальной сети, поскольку один физический хост может функционировать, как несколько различных хостов, использующих разные адреса IP. Некоторые хосты могут использовать множество физических интерфейсов (многодомные хосты – multi-homing).

Таким образом, следует обеспечить возможность присутствия на хосте множества физических интерфейсов в сеть, каждый из которых может иметь несколько логических адресов IP.

Примеры отображения адресов приводятся в работе «Address Mappings[13]» [5].

Фрагментация

Фрагментация дейтаграмм IP требуется в тех случаях, когда дейтаграмма происходит из сети, которая поддерживает больший размер пакетов, нежели промежуточные сети на пути к адресату.

Дейтаграмма IP может быть помечена флагом «don’t fragment» (не фрагментировать). Такие

дейтаграммы не будут фрагментироваться ни при каких обстоятельствах. Если нефрагментируемая дейтаграмма IP не может быть доставлена адресату без фрагментации, она просто отбрасывается.

Допускается использование невидимой для модуля IP фрагментации[14], передачи и сборки дейтаграмм в локальной сети [6].

Процедуры фрагментации и сборки дейтаграмм должны обеспечивать возможность разбиения дейтаграмм на почти произвольное число частей, которые впоследствии могут быть собраны воедино. Получатель фрагментов использует поле идентификации для того, чтобы фрагменты разных дейтаграмм не перемешивались. Поле смещения дейтаграммы говорит получателю о положении фрагмента в исходной дейтаграмме. Поля смещения и размера фрагмента определяют часть исходной дейтаграммы, содержащуюся в lfyyjv фрагменте. Сброшенный флаг наличия последующих фрагментов[15] говорит о том, что данный фрагмент является последним в дейтаграмме.

Поле идентификации позволяет различать фрагменты разных дейтаграмм. Отправляющий дейтаграмму модуль протокола устанавливает значение поля идентификации в каждой дейтаграмме так, чтобы оно было уникальным для данной пары отправитель-получатель и протокола в течение времени присутствия дейтаграммы в сети Internet. Этот модуль также устанавливает нулевые значения смещения фрагмента и флага наличия других фрагментов.

Для фрагментирования длинной дейтаграммы модуль IP (например, на маршрутизаторе) создает две новых дейтаграммы IP и копирует содержимое полей заголовка из длинной дейтаграммы в заголовки обеих новых дейтаграмм. Данные исходной дейтаграммы делятся на две части по 64-битовой (8 октетов) границе. Вторая часть дейтаграммы может иметь размер, не кратный 8 октетам (64 битам), но первая часть должна содержать целое число 8-октетных блоков. Назовем число 8-октетных блоков в первой части дейтаграммы NFB[16]. Первая часть дейтаграммы помещается в первую из новых дейтаграмм IP и поле длины устанавливается в соответствии с размером первой дейтаграммы. Для первой дейтаграммы устанавливается флаг наличия дополнительных фрагментов. Вторая часть данных помещается во вторую из созданных заново дейтаграмм и поле размера устанавливается в соответствии с длиной новой дейтаграммы. Значение поля смещения увеличивается на величину NFB. Значение флага наличия дополнительных фрагментов сохраняется в соответствии с флагом исходной нефрагментированной дейтаграммы.

Эту процедуру легко обобщить на случай разбиения дейтаграммы на n фрагментов, где n > 2.

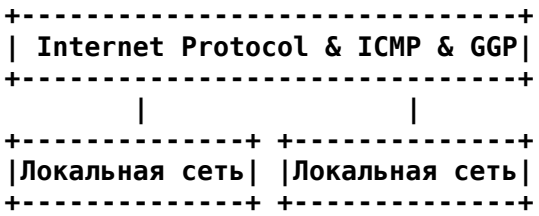


Рисунок 3: Протоколы шлюзов

Для сборки фрагментов дейтаграммы модуль IP (например, на хосте адресата) объединяет дейтаграммы IP с совпадающими значениями полей идентификации, адресов отправителя и получателя, а также протокола. Объединение осуществляется путем размещения данных из каждой дейтаграммы в позицию буфера, указанную полем смещения фрагмента в заголовке IP. Первый фрагмент будет иметь нулевое смещение, а для последнего фрагмента флаг more-fragments будет иметь нулевое значение.

2.4. Шлюзы

Шлюзы[17] обеспечивают пересылку дейтаграмм IP между сетями, обеспечивая также поддержку протокола GGP[18] [7] для обмена данными маршрутизации и другой управляющей информацией.

В шлюзах реализация протоколов вышележащих уровней не обязательна и функции GGP могут быть реализованы в модуле IP.

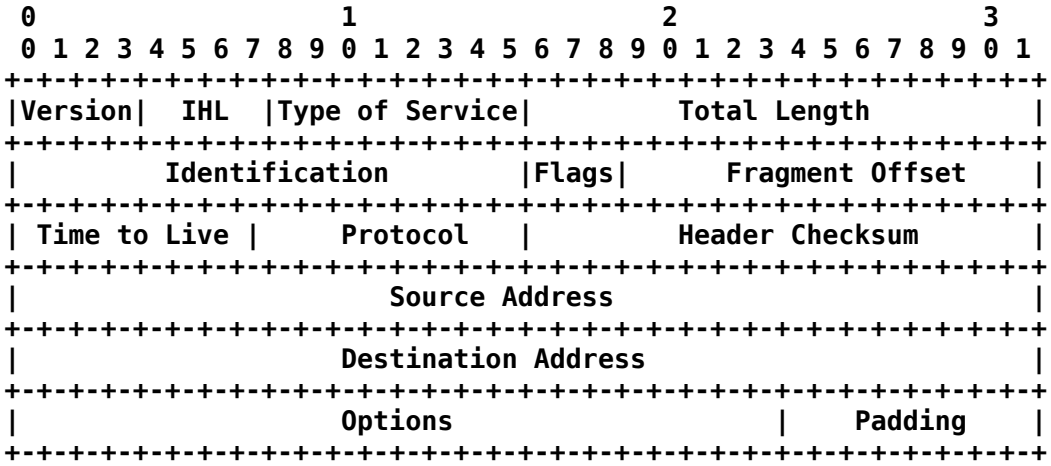


Рисунок 4: Формат заголовка дейтаграммы IP.

3. Спецификация

3.1. Формат заголовка

Формат заголовка дейтаграмм IP показан на рисунке 4.

Version – 4 бита

Это поле указывает номер версии протокола и определяет формат заголовка. Данная спецификация описывает версию 4[19].

IHL[20] – 4 бита

Это поле содержит размер заголовка IP в 32-битовых словах и указывает на начало данных. Отметим, что минимальное значение этого поля для корректного заголовка составляет 5.

ToS[21] – 8 битов

Поле ToS[22] обеспечивает индикацию абстрактных параметров желаемого качества обслуживания. Это значение используется при выборе реальных параметров обслуживания в процессе передачи дейтаграммы через отдельную сеть. Некоторые сети предлагают приоритетный сервис, который тем или иным способом трактует трафик с большим уровнем предпочтения, как более важный, нежели трафик другого типа (обычно при высокой загрузке просто воспринимается только трафик с уровнем предпочтения выше некоторого порога). Основной выбор осуществляется между тремя вариантами – малая задержка, высокая надежность, высокая пропускная способность.

биты 0-2: предпочтения.

бит 3: 0 = обычная задержка, 1 = малая задержка.

бит 4: 0 = обычная пропускная способность, 1 = высокая пропускная способность.

бит 5: 0 = обычная надежность, 1 = высокая надежность.

Биты 6-7: зарезервированы для использования в будущем.

0	1	2	3	4	5	6	7
Precedence			**D**	**T**	**R**	**0**	**0**

Precedence (предпочтения)

111 – управление сетью

110 – межсетевое управление

101 – CRITIC/ECP

100 – сверхсрочно

011 – срочно

010 – незамедлительно

001 – приоритетный

000 – обычный

Использование флагов Delay (D), Throughput (T), Reliability[23] (R) может увеличивать стоимость обслуживания (в том или ином смысле). Во многих сетях предпочтение по одному из этих параметров может быть связано с потерями по другому. За исключением специальных случаев следует использовать не более двух флагов из трех возможных.

Значение ToS используется для задания способа обработки дейтаграмм в процессе их передачи через internet. Например, отображение значений ToS на реальные параметры обслуживания в сетях AUTODIN II, ARPANET, SATNET, PRNET описано в работе Service Mappings [8].

Уровень предпочтения Network Control (управление сетью) означает, что дейтаграмма предназначена для использования внутри сети. Реальная трактовка этого обозначения определяется местными условиями сети. Значение Internetwork Control (межсетевое управление) показывает дейтаграммы, предназначенные только для управления шлюзами. Если та или иная сеть использует значение уровня предпочтения, она берет на себя ответственность за доступ к этому полю и его использование.

Total Length – 16 битов

Это поле указывает общий размер (в октетах) дейтаграммы с учетом заголовка и данных. Размер этого поля позволяет создавать дейтаграммы длиной до 65 535 октетов. Столь большие дейтаграммы неприемлемы для большинства хостов и сетей. Все хосты должны быть готовы к восприятию дейтаграмм размером до 576 октетов (целиком или в виде фрагментов). Хостам рекомендуется передавать дейтаграммы, размер которых превышает 576 октетов, только в тех случаях, когда есть уверенность, что адресат может принимать такие дейтаграммы.

Значение 576 выбрано для того, чтобы дейтаграммы могли кроме заголовка содержать блок данных разумных размеров. Например, такой размер позволяет передавать блок данных размером 512 октетов с 64-октетным заголовком. Максимальный размер заголовка IP составляет 60 октетов, а размер типичного заголовка IP – 20 октетов, что оставляет достаточно места для заголовков вышележащих уровней.

Identification – 16 битов

Значение поля идентификации присваивается отправителем для обеспечения корректной

сборки фрагментов дейтаграммы.

Flags – 3 бита

Набор флагов управления.

бит 0: зарезервирован (должен иметь значение 0)

бит 1: (DF) 0 = фрагментация возможна, 1 = фрагментация недопустима.

бит 2: (MF) 0 = последний фрагмент, 1 = фрагмент не является последним.

0	**1**	**2**
0	**DF**	**MF**

Fragment Offset – 13 битов

Это поле показывает положение данного фрагмента в исходной дейтаграмме. Смещение измеряется в единицах, кратных 8 октетам (64 бита). Смещение первого фрагмента равно нулю.

TTL[24] – 8 битов

Это поле определяет максимальный срок существования дейтаграммы в системе internet. Дейтаграммы с нулевым значением времени жизни должны уничтожаться. Значение этого поля изменяется при обработке заголовков IP. Время измеряется в секундах, но, поскольку каждый обрабатывающий дейтаграмму модуль должен уменьшать значение TTL по крайней мере на 1 (даже если обработка длилась меньше секунды), значение TTL следует рассматривать как верхний предел срока жизни дейтаграммы в систем. Это поле введено для того, чтобы можно было избавиться от недоставленных дейтаграмм.

Protocol – 8 битов

Это поле указывает протокол следующего[25] уровня, содержащийся в поле данных дейтаграммы IP. Идентификаторы протоколов указаны в Assigned Numbers [9].

Header Checksum – 16 битов

Контрольная сумма полей заголовка[26]. Поскольку некоторые поля заголовка (например, TTL) изменяются в процессе доставки, значение контрольной суммы проверяется и вычисляется заново в каждой точке обработки заголовков IP.

Контрольная сумма заголовка представляет собой 16-битовое поразрядное дополнение до единицы суммы поразрядных дополнений до единицы всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля принимается нулевым.

Контрольную сумму легко посчитать и можно показать ее адекватность, но алгоритм вычисления контрольной суммы может быть заменен специальной процедурой CRC с учетом опыта использования[27].

Source Address – 32 бита

Адрес отправителя (см. параграф 3.2. Обсуждение).

Destination Address – 32 бита

Адрес получателя (см. параграф 3.2. Обсуждение).

Options – переменная длина

Поле опций является необязательным. Поддержка опций должна реализоваться во всех модулях IP (на хостах и шлюзах). Использование опций определяется для отдельной дейтаграммы, а не для реализации модуля.

В некоторых средах использование опций безопасности является обязательным.

Поле опций имеет переменную длину. Существует два варианта форматирования опций:

одноктетные опции;

октет типа опции, октет размера опции и октеты собственно опций[28] (поле размера опции учитывает поля типа, размера и данных опции).

Октет типа опции содержит три поля:

флаг копирования (1 бит);

класс опции (2 бита);

номер опции (5 битов).

Флаг копирования показывает, копируется ли данная опция во все фрагменты дейтаграммы:

0 – опция не копируется;

1 – опция копируется во фрагменты.

Поле класса опций может принимать 4 значения:

0 – управление;

1 – зарезервирован;

2 – отладка и измерения;

3 – зарезервирован.

Определены следующие номера опций IP:

Класс	Номер	Длина	Описание
0	0	–	Конец списка опций (End of Option list). Эта опция занимает только 1 октет и не использует поле длины.
0	1	–	Нет операции (No operation). Эта опция занимает только 1 октет и не использует поле длины.
0	2	11	Безопасность (Security). Используется для передачи опций Security (безопасность), Compartmentation (изоляция), User Group (TCC), Handling Restriction Codes (коды обработки ограничений), совместимых с требованиями DOD[29].
0	3	перем.	Loose Source Routing (нестрогое задание маршрута отправителем). Используется для маршрутизации дейтаграмм IP с учетом данных, указанных отправителем.
0	9	перем.	Strict Source Routing (строгое задание маршрута отправителем). Используется для маршрутизации дейтаграмм IP на основе данных, указанных отправителем.

0	7	перем.	Record Route (запись маршрута). Используется для трассировки пути дейтаграмм IP.
0	8	4	Stream ID (идентификатор потока). Используется для обозначения потоков дейтаграмм.
2	4	перем.	Internet Timestamp (временная метка).

Определения отдельных опций

00000000

Type=0

End of Option List

Эта опция указывает на завершение списка опций. Граница этой опции может не совпадать с окончанием заголовка IP, указанным полем размера заголовка. Данная опция служит для того, чтобы показать завершение списка использованных для дейтаграммы опций (а не отдельной опции). Использование опции требуется только в тех случаях, когда окончание списка опций не совпадает с завершением заголовка IP.

Эту опцию можно копировать, добавлять и удалять при фрагментации или по иным причинам.

No Operation

00000001 Такая опция может использоваться между другими опциями (например, для выравнивания начала следующей опции по 32-битовой границе).

Type=1 Эту опцию можно копировать, добавлять и удалять при фрагментации или по иным причинам.

Security

Эта опция обеспечивает хостам способ передачи информации о безопасности, изоляции (compartmentation), ограничениях обслуживания и TCC (закрытые группы пользователей). Опция использует следующий формат:

Security (поле S) – 16 битов

Опция позволяет задать 16 уровней безопасности (8 уровней зарезервированы для использования в будущем).

00000000 00000000 – Unclassified (несекретно)

11110001 00110101 – Confidential (конфиденциально)

01111000 10011010 – EFTO[30]

10111100 01001101 – MMMM

01011110 00100110 – PROG

10101111 00010011 – Restricted (ограниченный доступ)

11010111 10001000 – Secret (секретные данные)

01101011 11000101 – Top Secret (совершенно секретные данные)

00110101 11100010 – (зарезервировано)

10011010 11110001 – (зарезервировано)

01001101 01111000 – (зарезервировано)

00100100 10111101 – (зарезервировано)

00010011 01011110 – (зарезервировано)

10001001 10101111 – (зарезервировано)

11000100 11010110 – (зарезервировано)

11100010 01101011 – (зарезервировано)

Compartments (поле C) – 16 битов

Значение этого поля, содержащее только нули, означает, что передаваемые данные не требуют изоляции. Другие значения могут использоваться с разрешения Defense Intelligence Agency.

Handling Restrictions (поле H) – 16 битов

Эти значения служат для управления разметкой и являются алфавитно-цифровыми направленными графами (определены в документе Defense Intelligence Agency Manual DIAM 65-19, “Standard Security Markings”).

Transmission Control Code (поле TCC) – 24 бита

Служит для разделения трафика и определения контролируемых «групп по интересам». Значения являются графами типа trigraph и доступны в HQ DCA Code 530.

Эта опция должна копироваться при фрагментации и появляется в дейтаграмме не более одного раза.

Loose Source and Record Route

Опция LSRR (нежестко заданный отправителем маршрут и запись пути) обеспечивает отправителям дейтаграмм IP способ предоставления маршрутной информации, используемой маршрутизаторами при пересылке дейтаграмм адресату и записи маршрутной информации.

10000011	**length**	**pointer**	**route data**

Type=131

Опция начинается с кода типа (131), второй октет содержит значение размера опции с учетом полей типа и размера опции, а также октета указателя и собственно опции (маршрутных данных). Третий октет опции показывает октет маршрутных данных, начиная с которого будет обрабатываться следующий адрес отправителя. Отсчет смещения ведется от начала опции и значение указателя не может быть меньше 4.

Маршрутные данные состоят из цепочки адресов IP. Каждый адрес занимает 32 бита (4 октета). Если значение указателя превышает длину опции, это говорит о пустом маршруте source route (завершении записи пути) и маршрутизации на основе поля адреса получателя.

Если достигнут адрес из поля адреса получателя и указатель не превышает длину опции, следующий адрес из маршрута source route заменяет значение поля адреса получателя и адрес в записываемом маршруте заменяет используемый адрес отправителя, а значение указателя увеличивается на 4.

Адресом записываемого маршрута является собственный адрес модуля IP, известный в той среде, куда пересылается дейтаграмма.

Такая процедура замены source route записанным маршрутом (хотя в них и используется обратный по отношению друг к другу порядок) означает, что опция (и весь заголовок IP) сохраняет постоянный размер при передаче дейтаграммы через internet.

Эта опция не задает маршрут жестко, поскольку шлюзам и хостам IP разрешается использовать любой маршрут через любые промежуточные шлюзы для доставки дейтаграммы по адресу следующей точки маршрута.

Опция должна копироваться при фрагментации и появляется в дейтаграмме не более одного раза.

Strict Source and Record Route

10001001	**length**	**pointer**	**route data**

Type=137

Опция SSRR (жестко заданный отправителем маршрут и запись пути) обеспечивает отправителю дейтаграммы IP способ задания маршрутной информации, которая должна использоваться промежуточными маршрутизаторами для доставки дейтаграммы адресату и записи пути.

Опция начинается с идентификатора типа (137), за которым следует октет размера опции с учетом полей типа, длины, смещения и маршрутных данных. Третий октет опции содержит указатель (смещение от начала данной опции) на октет, с которого начинается следующий обрабатываемый адрес отправителя. Минимальное значение указателя составляет 4 (см. выше).

Маршрутные данные представляют собой цепочку адресов IP, каждый из которых занимает 32 бита (4 октета). Если значение указателя превышает размер опции, это говорит о пустом маршруте source route (завершении записи пути) и маршрутизации на основе адреса получателя.

Если достигнут адрес из поля адреса получателя и указатель не превышает длину опции, следующий адрес из маршрута source route заменяет значение поля адреса получателя и адрес в записываемом маршруте заменяет используемый адрес отправителя, а значение указателя увеличивается на 4.

Адресом записываемого маршрута является собственный адрес модуля IP, известный в той среде, куда пересылается дейтаграмма.

Такая процедура замены source route записанным маршрутом (хотя в них и используется обратный по отношению друг к другу порядок) означает, что опция (и весь заголовок IP) сохраняет постоянный размер при передаче дейтаграммы через internet.

Эта опция жестко задает маршрут, поскольку шлюзы и хосты IP должны передавать дейтаграмму непосредственно по следующему адресу, указанному в заданном отправителем маршруте, по всему пути доставки дейтаграммы адресату.

Опция должна копироваться при фрагментации и появляется в дейтаграмме не более одного раза.

Record Route

00000111	**length**	**pointer**	**route data**

Type=7

Опция записи маршрута обеспечивает способ записи пути передачи дейтаграммы IP.

Опция начинается с поля типа (7), за которым

следует поле длины, учитывающее полный размер опции (тип, размер, смещение, маршрутные данные). Третий октет содержит указатель на октет, с которого начинается следующая область записи маршрута. Смещение отсчитывается от начала опции, поэтому значение указателя не может быть меньше 4.

Записываемый маршрут представляет собой последовательность адресов IP, каждый из которых имеет длину 32 бита (4 октета). Если значение указателя превышает размер опции, это говорит о завершении записи маршрута. Отправляющий дейтаграмму хост должен обеспечить достаточное пространство (размер опции) для записи адресов на пути к получателю. В исходной дейтаграмме поля адресов должны иметь нулевые значения.

Когда модуль IP маршрутизирует дейтаграмму, он проверяет в ней наличие маршрутной записи. При наличии такой записи модуль помещает в нее свой адрес, известный в той среде, куда пересылается дейтаграмма, начиная со смещения, которое задано указателем, и увеличивает значение указателя на 4.

Если поле маршрутных данных уже заполнено (значение указателя превышает размер опции), дейтаграмма пересылается без дальнейшей записи маршрута. Если оставшегося пространства для записи маршрутных данных недостаточно для включения адреса, дейтаграмма рассматривается как ошибочная и отбрасывается. В таких случаях отправителю дейтаграммы может быть передано сообщение ICMP об ошибке в параметрах [3].

Эта опция не копируется во фрагменты и не должна включаться в дейтаграмму более одного раза.

Stream Identifier

Эта опция позволяет передавать 16-битовые идентификаторы потоков SATNET через сети, которые не поддерживают концепцию потоков.

10001000	**00000010**	**Stream ID**

Type=136, Length=4

Опция должна копироваться при фрагментации и появляется в дейтаграмме не более одного раза.

Internet Timestamp

Поле длины опции (Length) содержит значение, указывающее число октетов в полях типа опции, ее длины, указателя, Overflow (переполнение), флагов и временных меток (всего до 40 октетов).

01000100	**length**	**pointer**	**oflw**	**flg**
IP address				
Timestamp				
.				
.				
.				

Type = 68

Поле указателя (Pointer) показывает смещение (в октетах) от начала опции до начала следующей временной метки. Таким образом, минимальное значение указателя равно 5. Область временных меток считается заполненной, когда значение указателя превышает размер опции.

Поле Overflow (oflw, переполнение – 4 бита) показывает число модулей IP, которые не смогли включить свои временные метки в результате нехватки места в опции.

Поле флагов (flg, 4 бита) может принимать следующие значения:

0 – только временные метки, сохраняемые в последовательности 32-битовых слов;

1 – перед каждой меткой помещается IP-адрес регистрирующего метку модуля;

3 – поля адресов IP указываются заранее и модуль IP помещает временную метку только в том случае, когда адрес этого модуля указан следующим в списке адресов опции.

Поля Timestamp (временная метка) выравниваются по правому краю и содержат число миллисекунд после полуночи по универсальному времени (UT). Если невозможно указать время в миллисекундах или нет привязки к универсальному времени, в поле метки может помещаться любое значение времени, а старший бит метки должен быть установлен (1), для индикации некорректности данной метки.

Отправляющий дейтаграмму хост должен предусмотреть достаточно места для записи временных меток по пути доставки дейтаграммы. Размер опции не может увеличиваться по мере добавления меток. Исходная дейтаграмма содержит нулевые значения всех предусмотренных в опции полей временных меток (кроме заранее указанных адресов IP).

Если область записи временных меток уже заполнена (значение указателя превышает размер опции), дейтаграмма пересылается без дальнейшей записи временных меток, но значение поля Overflow должно увеличиваться при каждой пересылке дейтаграммы.

Если в поле записи временных меток еще остается свободное место, но полная метка не помещается, дейтаграмма трактуется как ошибочная и отбрасывается. Отправителю такой дейтаграммы может быть передано сообщение ICMP о некорректности параметров дейтаграммы [3].

Опция не копируется во фрагменты и передается только в первом фрагменте. Опция появляется в дейтаграмме только один раз.

Padding – переменная длина

Поле заполнения служит для выравнивания размера заголовка IP по 32-битовой границе. Для заполнения используется значение 0.

3.2. Обсуждение

Реализация протокола должна быть гибкой и разумной. Каждая реализация должна предполагать интероперабельность с продукцией других разработчиков. Хотя целью данной спецификации является четкое и строгое описание протокола, существует вероятность различных интерпретаций стандарта. При передаче дейтаграмм следует строго следовать спецификации, сохраняя в то же время готовность к восприятию любых дейтаграмм, которые можно интерпретировать (например, не содержащих технических ошибок).

Базовый сервис internet ориентирован на обработку дейтаграмм и обеспечивает фрагментацию дейтаграмм на шлюзах и сборку фрагментов модулем IP хоста-получателя. Фрагментация и сборка дейтаграмм внутри отдельной сети или на основе частного соглашения также допустимы, поскольку процесс фрагментации и сборки совершенно прозрачен для IP и протоколов вышележащих уровней. Такая прозрачная фрагментация/сборка называется внутрисетевой и не рассматривается в данной спецификации.

Адреса IP позволяют различить отправителя и получателя на уровне хоста. Дополнительные сведения содержатся в поле протокола. Предполагается, что каждый протокол обеспечивает мультиплексирование (если оно требуется) на хосте.

Адресация

Для обеспечения гибкого распределения адресов и поддержки большого числа сетей небольших и средних размеров используется специальная интерпретация полей адреса IP[31]. Такая интерпретация позволяет выделить небольшое число адресов для крупных сетей,

большее число адресов для сетей среднего размера и многочисленные адреса для небольших сетей. В дополнение к этому выделяется блок адресов для использования расширенного режима адресации.

Формат адресов

Старшие биты	Формат	Класс
0	7 битов задают номер сети, остальные 24 – номер хоста	A
10	14 битов задают номер сети, остальные 16 – номер хоста	B
110	21 бит задает номер сети, остальные 8 – номер хоста	C
111	Используется для расширенной адресации	

Нулевое значение поля номера сети означает данную сеть. Такая адресация используется только для некоторых сообщений ICMP. Расширенный режим адресации не определен. Обе эти возможности зарезервированы для использования в будущем.

Реальные значения, выделенные для разных групп сетевых адресов, указаны в работе Assigned Numbers [9].

Локальные адреса[32] распределяются на уровне локальной сети и должны позволять одному физическому хосту действовать как множество различных хостов internet. Т. е., должно поддерживаться отображение между адресами IP и физическим интерфейсом хоста, позволяющее связать несколько IP-адресов с одним физическим интерфейсом хоста. Должна также поддерживаться и обратная возможность – связывание нескольких физических интерфейсов с одним адресом IP[33].

Преобразование адресов IP в адреса сетей ARPANET, SATNET, PRNET и т. п. описано в работе Address Mappings [5].

Фрагментация и сборка[34]

Поле идентификации (ID) используется вместе с адресами отправителя/получателя и полем протокола для идентификации фрагментов дейтаграммы при сборке[35].

Флаг наличия других фрагментов (More Fragments или MF) устанавливается для всех фрагментов дейтаграммы, кроме последнего. Поле Fragment Offset показывает положение фрагмента относительно начала исходной (нефрагментированной) дейтаграммы. Смещение учитывается в блоках размером 8 октетов. Стратегия фрагментации разработана таким образом, чтобы в нефрагментированной дейтаграмме вся поля, связанные с фрагментацией, имели нулевые значения (MF = 0, fragment offset = 0). Если дейтаграмма IP фрагментируется, ее поле данных должно делиться на части по 8-октетным границам.

Таким образом, используемый формат поддерживает до 2^{13} = 8192 фрагментов по 8 октетов (т. е., до 65 536 октетов). Отметим, что это значение соответствует возможным значениям поля размера дейтаграммы в ее заголовке (естественно, заголовок показывает общую длину дейтаграммы, а не ее фрагментов).

При фрагментации дейтаграммы некоторые опции копируются из оригинальной дейтаграммы в каждый фрагмент, а часть опций сохраняется только в первом фрагменте.

Каждый модуль IP должен поддерживать пересылку без фрагментации дейтаграмм размером 68 октетов. Это значение выбрано потому, что заголовок дейтаграммы может достигать 60 октетов и поле данных должно содержать, по крайней мере, 8 октетов.

Каждый получатель должен поддерживать прием дейтаграмм размером, по крайней мере, 576

октетов целиком или с использованием фрагментации и сборки.

При фрагментации могут изменяться следующие поля:

(1) опции

(2) флаг MF

(3) смещение фрагмента

(4) размер заголовка дейтаграммы

(5) общий размер

(6) контрольная сумма заголовка.

Если установлен флаг запрета фрагментации (DF), дейтаграммы, которые невозможно передать целиком, отбрасываются. Этот вариант используется в тех случаях, когда принимающий хост не может собирать фрагменты дейтаграммы по причине нехватки ресурсов. Примером запрета фрагментации может служить ситуация с линией к небольшому хосту. Такой хост может иметь программу самозагрузки (boot strap), которая воспринимает дейтаграмму, хранящуюся в памяти и потом выполняет содержащийся в ней код.

Процедуры фрагментации и сборки проще описать на примерах. Описанные ниже процедуры содержат примеры реализации.

Знак =< в приведенном ниже псевдокоде означает «меньше или равно», # – «не равно», = – «равно», <- – «установить значение». Выражение “x – y” задает диапазон значений, включающий x, но не включающий y (например диапазон “4 – 7” будет включать числа 4, 5 и 6, но не будет включать 7).

Пример процедуры фрагментации

Размер максимальной дейтаграммы, которая может быть передана через следующую сеть, называется MTU[36].

Если общий размер дейтаграммы не превышает MTU, эта дейтаграмма обрабатывается дальше. В противном случае дейтаграмма делится на два фрагмента – первый фрагмент имеет максимальный размер, а во втором содержится оставшаяся часть дейтаграммы. Первый фрагмент передается на следующий этап обработки, а для второго заново выполняется процедура проверки размера и при необходимости выполняется дополнительная фрагментация (это продолжается до тех пор, пока размер всех фрагментов перестанет превышать максимальное значение).

Обозначения:

FO – смещение фрагмента

IHL – длина заголовка Internet

DF – флаг запрета фрагментирования

MF – флаг наличия других фрагментов

TL – общий размер

OFO – смещение старого фрагмента

OIHL – размер заголовка старого фрагмента

OMF – старый флаг наличия других фрагментов

OTL – старое значение общей длины

NFB – число фрагментов

MTU – максимальный передаваемый блок.

Процедура:

Если TL =< MTU, **то** дейтаграмма передается на следующий этап обработки, а **в противном случае** проверяется флаг DF. **Если** DF = 1, **то** дейтаграмма отбрасывается; **в противном случае** выполняются следующие операции:

1. **копируется исходный заголовок IP;**
2. **OIHL <- IHL; OTL <- TL; OFO <- FO; OMF <- MF;**
3. **NFB <- (MTU-IHL*4)/8;**
4. **присоединяются первые NFB*8 октетов данных;**
5. **корректируется заголовок:**
   MF <- 1; TL <- (IHL*4)+(NFB*8);
   заново вычисляется контрольная сумма;
6. **фрагмент передается на следующий этап обработки;**
   переход ко второму фрагменту:
7. **частичное копирование заголовка IP (некоторые опции не копируются; см. приведенные выше описания опций);**
8. **добавляется оставшаяся часть данных;**
9. **корректируется заголовок:**
   IHL <- (((OIHL*4)-(размер опций не копируется))+3)/4;
   TL <- OTL – NFB*8 – (OIHL-IHL)*4);
   FO <- OFO + NFB; MF <- OMF;
   заново вычисляется контрольная сумма;
10. **фрагмент передается процедуре проверки размера.**

После выполнения п. 10 процедура завершается (если размер фрагмента не превышает допустимое значение) или повторяется. Эта процедура создает фрагменты одинакового (максимального) размера (за исключением последнего). Могут использоваться и другие процедуры, которые создают фрагменты с размером меньше максимального. Например, процедура фрагментации может использовать повторяющиеся операции деления данных дейтаграммы пополам, пока оно не достигнет приемлемого для передачи размера.

Пример процедуры сборки

Для каждой дейтаграммы идентификатор буфера определяется путем объединения (конкатенации) полей адресов отправителя и получателя, протокола и идентификации. Если дейтаграмма не является фрагментом (поля смещения фрагмента и флага дополнительных фрагментов имеют нулевые значения), все ресурсы сборки, связанные с этим идентификатором буфера, освобождаются и дейтаграмма передается на следующий этап обработки.

Если больше нет фрагментов с таким же идентификатором буфера, происходит выделение ресурсов для сборки. Эти ресурсы включают буфер данных, буфер заголовков, таблицу фрагментов (блоков), поле общего размера и таймер. Данные из фрагментов помещаются в буфер данных в соответствии с размером и смещением фрагментов; флаги в таблице фрагментов устанавливается при получении соответствующего фрагмента.

Заголовок первого фрагмента (fragment offset=0) помещается в буфер заголовков. Если фрагмент является последним (MF=0), определяется общий размер дейтаграммы. Если фрагмент завершает прием дейтаграммы (проверяется по таблице блоков), дейтаграмма

передается на следующий этап обработки; в противном случае для таймера устанавливается наибольшее из двух значений – текущие показания таймера и значение поля TTL из данного фрагмента; управление передается процедуре сборки фрагментов.

Если заданное таймером время истекло, освобождаются все ресурсы, выделенные для данного идентификатора буфера. Начальная установка таймера задает нижнюю границу времени ожидания при сборке. При получении фрагментов время ожидания может быть увеличено в соответствии с TTL, а уменьшение времени ожидания не предусмотрено. Максимальное значение таймера может достигать максимального значения TTL (около 4,25 мин.). Рекомендуется устанавливать начальное значение таймера равным 15 сек. Эта рекомендация может быть изменена с учетом реального опыт использования протокола. Отметим, что выбор этого значения связан с доступным размером буфера и скоростью среды передачи, т. е., при скорости 10 кбит/с и времени ожидания 15 сек. может потребоваться буфер размером 150 кбит.

Обозначения:

FO – смещение фрагмента

IHL – длина заголовка Internet

DF – флаг запрета фрагментирования

MF – флаг наличия других фрагментов

TTL – время жизни

NFB – число фрагментов

TL – общий размер

TDL – общий размер данных

BUFID – идентификатор буфера

RCVBT – таблица полученных фрагментов

TLB – нижняя граница таймера.

Процедура:

(1) BUFID <- source|destination|protocol|identification;

(2) IF FO = 0 AND MF = 0

(3) THEN IF выделен буфер с BUFID

(4) THEN удалить все ресурсы сборки для этого BUFID;

(5) Передать дейтаграмму на следующий этап обработки; DONE.

(6) ELSE IF буфер с BUFID не выделен

(7) THEN выделить ресурсы сборки для BUFID;

TIMER <- TLB; TDL <- 0;

(8) Поместить данные из фрагмента в буфер BUFID, начиная с октета FO*8 и до октета (TL-(IHL*4))+FO*8;

(9) Установить биты RCVBT для октетов с FO до FO+((TL-(IHL*4)+7)/8);

(10) IF MF = 0 THEN TDL <- TL-(IHL*4)+(FO*8)

(11) IF FO = 0 THEN поместить заголовок в буфер заголовков

(12) IF TDL # 0

(13) AND все биты RCVBT от 0 до (TDL+7)/8 установлены

(14) THEN TL <- TDL+(IHL*4)

(15) Передать дейтаграмму на следующий этап обработки;

(16) Освободить все ресурсы сборки, выделенные для BUFID; DONE.

(17) TIMER <- MAX(TIMER,TTL);

(18) повторять, пока не будут получены все фрагменты или не истечет время, заданное таймером;

(19) истекло время ожидания: освобождаются все ресурсы, выделенные для BUFID; DONE.

В случаях, когда два фрагмента идентичны или перекрываются, эта процедура будет использовать более позднюю копию.

Идентификация

Выбор идентификатора для дейтаграммы базируется на обеспечении уникальности обозначения фрагментов отдельно взятой дейтаграммы. Модуль протокола, собирающий дейтаграмму из фрагментов, считает фрагменты относящимися к одной дейтаграмме, если у них совпадают адреса отправителя/получателя, тип протокола и поле идентификации. Таким образом, отправитель должен выбирать значение идентификатора так, чтобы оно было уникальным для комбинации адресов отправителя/получателя и типа протокола в течение срока жизни дейтаграммы (или любого ее фрагмента).

Представляется целесообразным для передающего хоста сохранение таблицы использованных идентификаторов – по одной записи для каждого получателя, с которым хост взаимодействовал в течение последнего промежутка времени, равного максимальному сроку существования пакета в internet.

Однако поле идентификации позволяет выбрать любое из 65536 значений идентификатора, поэтому некоторые хосты могут использовать уникальные идентификаторы, не связанные с получателем.

Для некоторых ситуаций разумно выбирать значения идентификаторов с помощью протокола вышележащего уровня. Например, модули протокола TCP могут повторно передавать идентичные сегменты TCP и вероятность корректной доставки будет повышаться, если при повторе использовать такой же идентификатор, как при первой передаче, поскольку фрагменты любой из дейтаграмм (первоначальной или повторных) могут использоваться для корректного восстановления сегмента TCP.

ToS[37]

Значение TOS обеспечивает выбор уровня обслуживания с помощью абстрактных параметров precedence (предпочтение, приоритет), delay (задержка), throughput (пропускная способность), reliability (надежность). Эти абстрактные параметры преобразуются в реальные параметры обработки дейтаграмм в сетях на пути доставки.

Precedence – степень важности дейтаграммы.

Delay – для таких дейтаграмм большое значение имеет время доставки.

Throughput – для дейтаграмм имеет важное значение скорость передачи данных.

Reliability – для дейтаграмм важное значение имеет надежность (гарантия) доставки.

Например, ARPANET использует бит приоритета и разделяет «стандартные» (тип 0) и

неконтролируемые (тип 3) сообщения (выбор между одно- и многопакетными сообщениями также может рассматриваться как параметр обслуживания). Неконтролируемые сообщения доставляются с меньшими гарантиями, но имеют более низкую задержку. Предположим, что дейтаграмма IP доставляется через сеть ARPANET с параметрами ToS:

Precedence: 5

Delay: 0

Throughput: 1

Reliability: 1

В этом случае отображение параметров сервиса на поддерживаемые в сети ARPANET параметры обслуживания приведет к установке бита приоритета ARPANET, поскольку значение precedence находится в старшей половине возможного диапазона, и выбора стандартных сообщений, поскольку заданы параметры throughput и reliability, а бит delay не установлен. Более детальную информацию по этому вопросу можно найти в работе “Service Mappings” [8].

TTL[38]

Значение TTL устанавливается отправителем и задает максимальное время существования дейтаграммы в internet. Если время, заданное полем TTL, истекло, дейтаграмма уничтожается.

Значение этого поля должно уменьшаться в каждой точке обработки заголовка дейтаграммы для того, чтобы учесть затраты времени на такую обработку. Даже в тех случаях, когда нет информации о затратах времени на обработку, значение поля должно уменьшаться на 1. Время жизни измеряется в секундах и максимальное значение TTL (255) соответствует 255 секундам или 4,25 мин. Поскольку каждый модуль, обрабатывающий дейтаграмму, должен уменьшать значение TTL, по крайней мере, на 1, значение TTL должно трактоваться как верхний предел срока существования дейтаграммы. Назначение поля TTL состоит в том, чтобы уничтожать недоставленные дейтаграммы и ограничить срок существования дейтаграмм в системе.

Некоторые протоколы вышележащих уровней, обеспечивающие гарантированную доставку, базируются на предположении, что старые копии дейтаграмм не могут приходить по истечении некоторого времени. Поле TTL дает таким протоколам гарантию истинности этого предположения.

Опции

Опции являются необязательными для дейтаграмм, но все реализации должны поддерживать опции[39]. Т. е., использование опций определяется отправителем, но каждый модуль IP должен быть способен разобрать любые опции. Поле опций может содержать несколько значений опций.

Опции не обязательно заканчиваются на 32-битовой границе. Заголовок IP в целях выравнивания по 32-битовой границе дополняется октетами нулей. Первый из таких октетов будет интерпретироваться как завершение поля опций, а остальная часть – как обычное заполнение.

Каждый модуль IP должен уметь обрабатывать любые опции. Опции безопасности[40] (Security) требуются для классифицированного и изолированного трафика, а также трафика с ограничением доступа.

Контрольная сумма

Контрольная сумма заголовка заново вычисляется каждый раз при изменении заголовка (например, при уменьшении TTL, добавлении или изменении опций, фрагментации дейтаграммы). Контрольная сумма на уровне IP предназначена для защиты полей заголовка от ошибок при передаче.

Для некоторых приложений допустимо небольшое число ошибок, но недопустима задержка передачи. Если протокол IP будет заниматься исправлением ошибок, такие приложения не смогут работать.

Ошибки

Сообщения об ошибках протокола IP могут передаваться с использованием протокола ICMP [3].

3.3. Интерфейсы

Функциональное описание пользовательского интерфейса IP является, в лучшем случае, умозрительным, поскольку каждая операционная система использует свои функции. Следовательно, мы должны предупредить читателя, что различные реализации IP могут иметь совершенно разные пользовательские интерфейсы. Однако любой модуль IP должен обеспечивать, по крайней мере, минимальный набор сервиса для обеспечения возможности совместного использования разных реализаций IP. В этом параграфе рассматриваются функциональные интерфейсы, требуемые от каждой реализации IP.

Протокол IP взаимодействует, с одной стороны, с локальной сетью, а с другой – с протоколом вышележащего уровня или прикладной программой. Далее протокол вышележащего уровня и прикладные программы (или даже программы шлюзов) будем называть для краткости «пользователь», поскольку они используют услуги модуля IP. Поскольку IP имеет дело с дейтаграммами, между передачей отдельных дейтаграмм нет почти никакой связи и пользователь при каждом обращении к модулю IP передает ему все требуемые параметры для выполнения запрашиваемой операции.

Пример интерфейса с вышележащим уровнем

Ниже приведены два примера вызовов, удовлетворяющих требованиям к пользовательским вызовам IP (“=>” означает возврат):

SEND (src, dst, prot, TOS, TTL, BufPTR, len, Id, DF, opt => result)

где:

src = адрес отправителя

dst = адрес получателя

prot = протокол

TOS = тип обслуживания

TTL = время жизни

BufPTR = указатель на буфер

len = размер буфера

Id = идентификатор

DF = не фрагментировать

opt = необязательные данные (опции)

result = отклик

OK = дейтаграмма успешно отправлена

Error = ошибка в аргументах или сетевая ошибка (локальная сеть)

Отметим, что предпочтения включены в TOS а параметры безопасности/изоляции передаются как опция.

RECV (BufPTR, prot, => result, src, dst, TOS, len, opt)

где:

BufPTR = указатель на буфер

prot = протокол

result = отклик

OK = дейтаграмма успешно принята

Error = ошибка в аргументах

len = размер буфера

src = адрес отправителя

dst = адрес получателя

TOS = тип обслуживания

opt = опции

Для передачи дейтаграммы пользователь применяет вызов SEND, передавая все требуемые аргументы. Модуль IP, принявший вызов, проверяет аргументы, готовит и передает сообщение. Если параметры указаны корректно и дейтаграмма воспринята локальной сетью, модуль возвращает сообщение об успешной передаче. Если какой-то из параметров указан некорректно или дейтаграмма не принята локальной сетью, модуль возвращает сообщение об ошибке. В таких случаях модуль должен также возвращать соответствующий отклик, указывающий причину ошибки. Уровень детализации таких откликов зависит от реализации.

Получение модулем IP дейтаграммы из локальной сети может быть связано с ожидающим пользовательским вызовом RECV. Если такой вызов имеется, информация из дейтаграммы передается пользователю. Если же вызова нет, пользователю передается уведомление о прибывшей дейтаграмме. Если пользователь с указанным адресом не существует, отправителю возвращается сообщение ICMP об ошибке и дейтаграмма уничтожается.

Для уведомления пользователя могут применяться псевдо-прерывания или аналогичный механизм, приемлемый в используемой реализацией среде.

Пользовательский вызов RECV может быть исполнен незамедлительно при наличии ожидающей дейтаграммы или помещен в состояние ожидания прихода дейтаграммы.

Адрес отправителя включается в вызов SEND, если передающий хост имеет несколько адресов (для логических или физических устройств). Модуль IP должен убедиться, что полученный адрес корректен и связан с данным хостом.

Реализация также может разрешать или требовать вызов модуля IP для индикации заинтересованности в получении или предоставления исключительного использования для класса дейтаграмм (например, все дейтаграммы с определенным значением поля протокола).

В этом параграфе было дано функциональное описание интерфейса пользователь – IP. Использованные обозначения похожи на нотацию большинства языков высокого уровня,

однако такое применение не требует «функций-ловушек» (например, SVC, UUO, EMT) или иных форм взаимодействия между процессами.

Приложение A: Примеры и сценарии

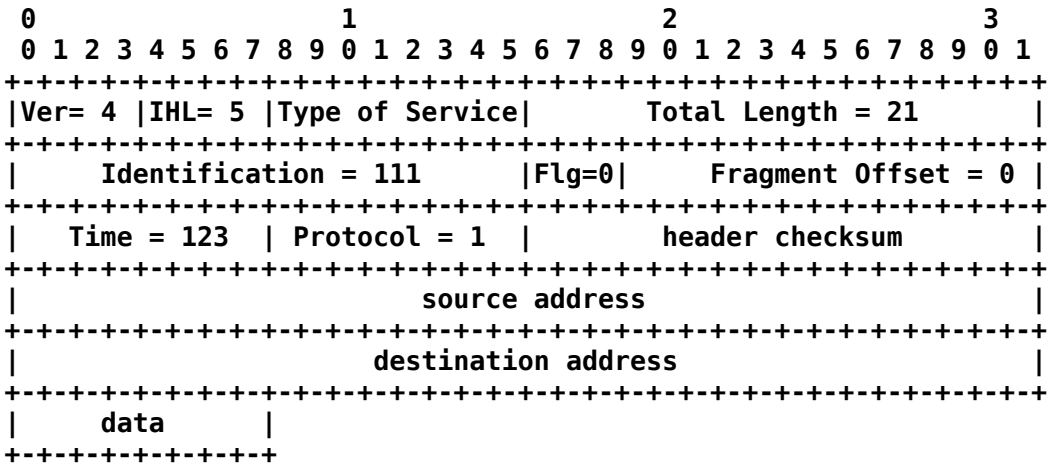


Рисунок 5: Пример дейтаграммы IP

Пример 1

На рисунке 5 приведен пример минимальной дейтаграммы IP, содержащей данные[41].

Показанная на рисунке дейтаграмма соответствует протоколу IP версии 4; заголовок содержит пять 32-битовых слов, а полный размер дейтаграммы составляет 21 октет. Показанная дейтаграмма является полной (не фрагментом).

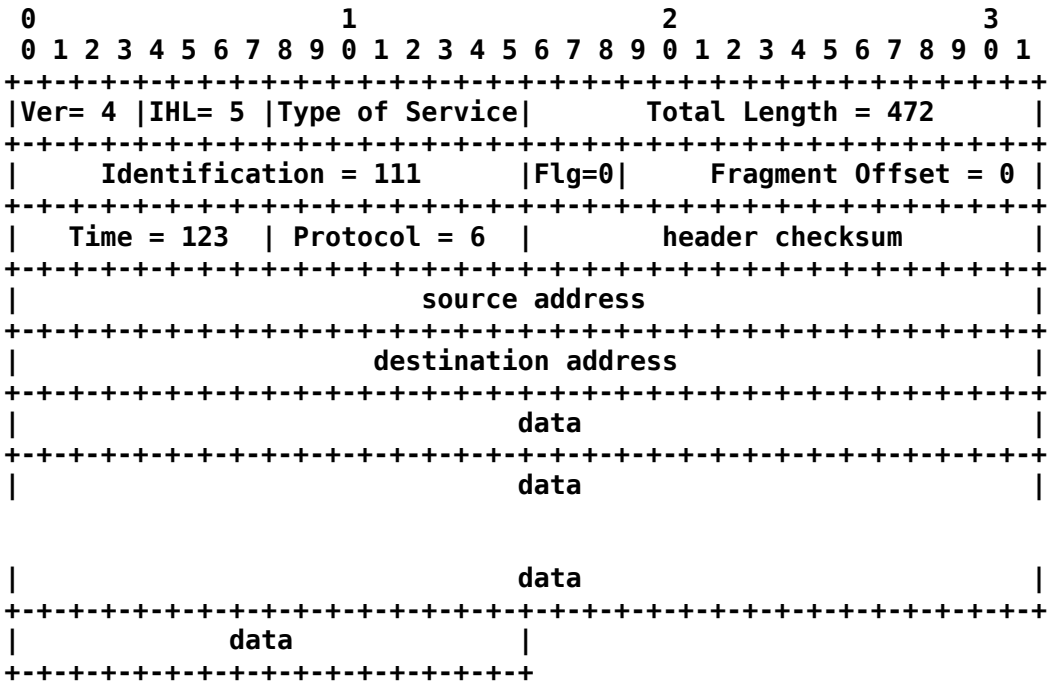


Рисунок 6: Пример дейтаграммы IP

Пример 2

В этом примере (Рисунок 6) показана дейтаграмма среднего размера (452 октета данных),

которая разбивается на два фрагмента, поскольку сеть не поддерживает передачу дейтаграмм, размер которых превышает 280 октетов.

Ниже (Рисунок 7) показан первый фрагмент, после выделения из дейтаграммы первых 256 октетов данных.

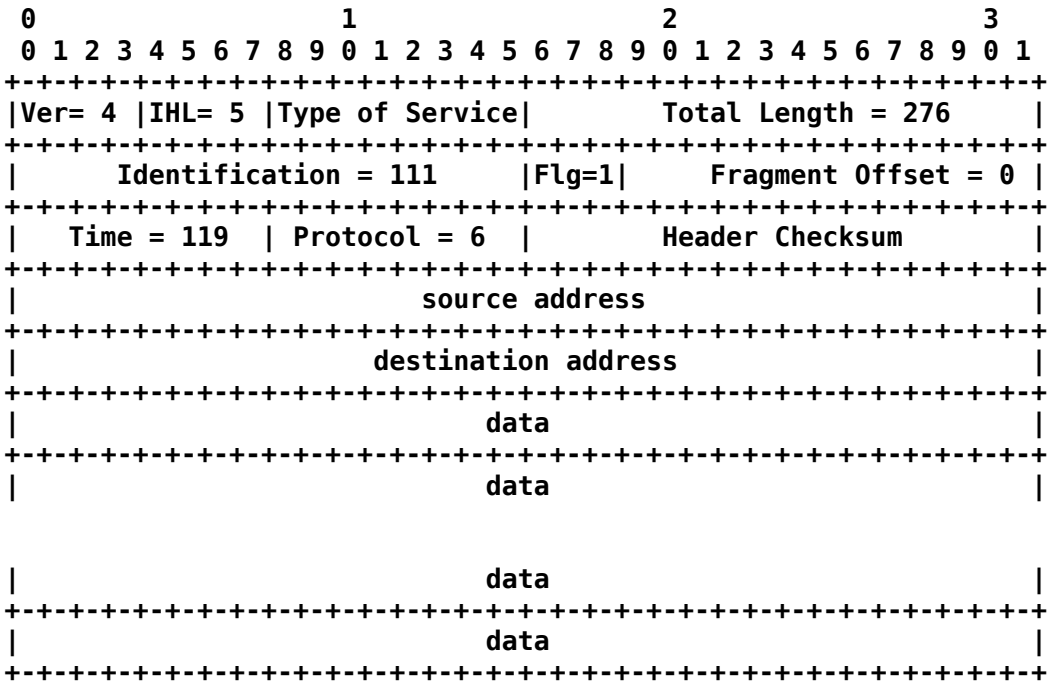


Рисунок 7: Пример фрагмента дейтаграммы IP

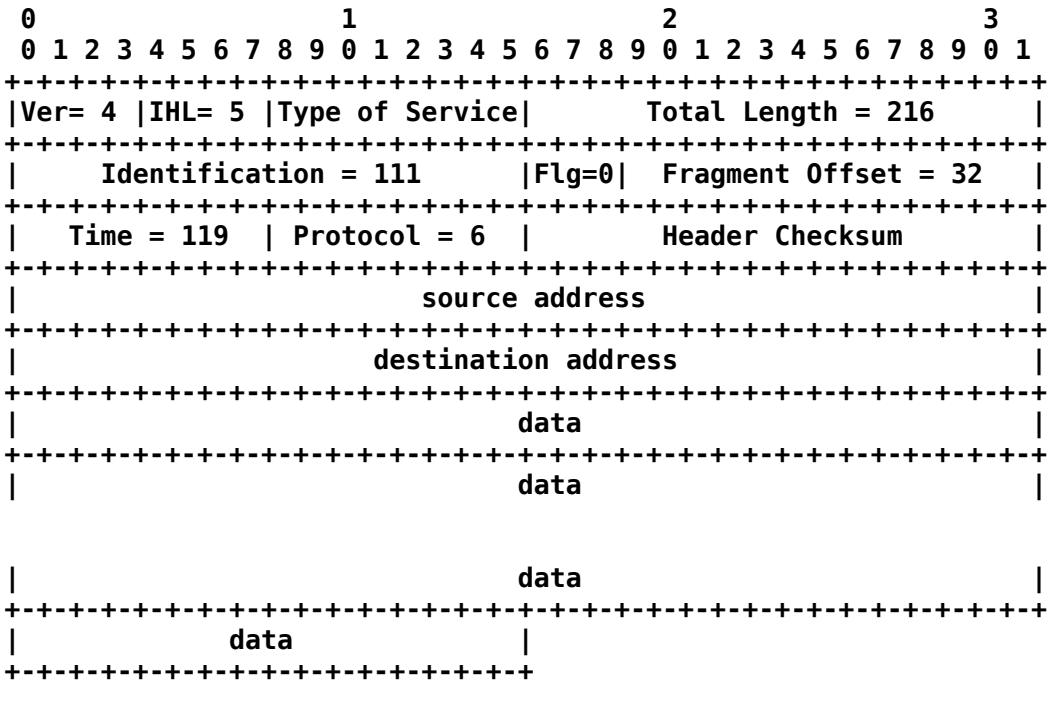
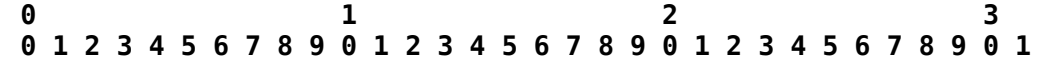


Рисунок 8: Пример фрагмента дейтаграммы IP

Второй фрагмент будет иметь форму, показанную на рисунке 8.

Пример 3



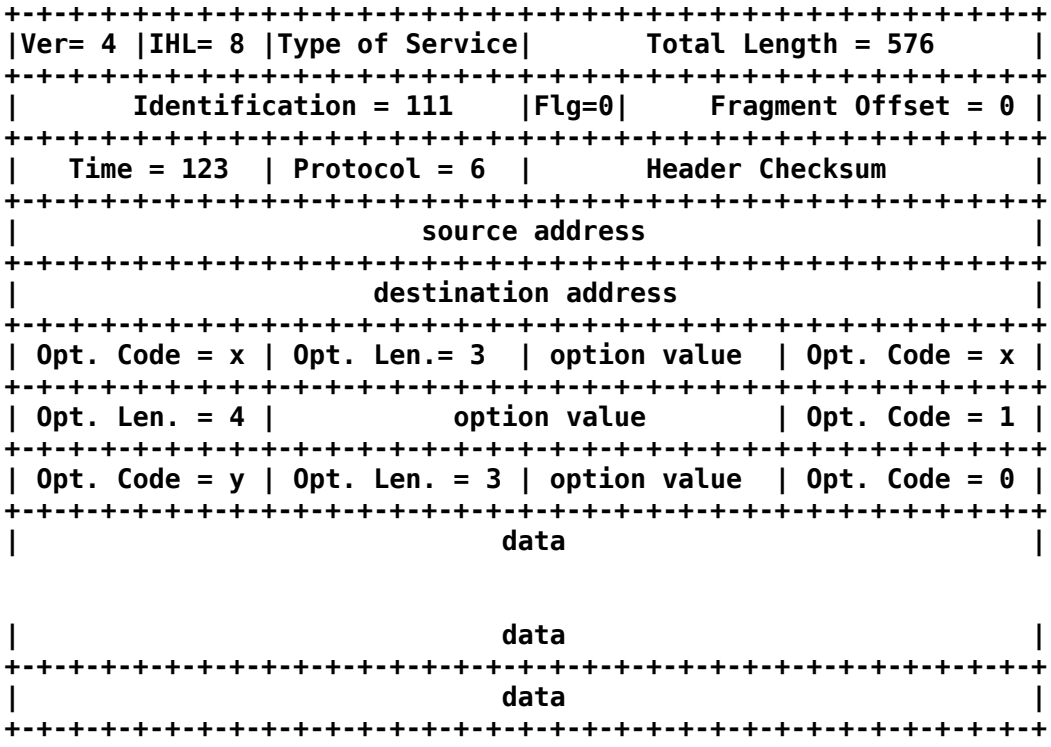


Рисунок 9: Пример дейтаграммы IP с опциями

Выше (Рисунок 9) показан пример дейтаграммы, содержащей опции.

Приложение B: Порядок передачи данных

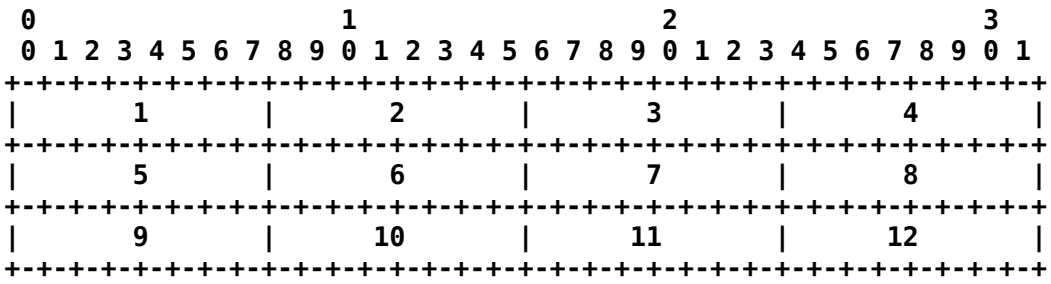


Рисунок 10: Порядок передачи байтов.

Порядок передачи заголовков и данных, описываемых в этой спецификации, задается на уровне октетов. Дейтаграмма представляет собой группу октетов, которые передаются в том же порядке, в котором мы читаем. Например, в показанной на рисунке 10 дейтаграмме октеты передаются в порядке возрастания номеров на рисунке.

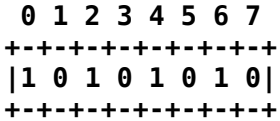


Рисунок 11: Значимость битов

Когда октет представляет числовое значение, показанный слева бит является старшим или наиболее значимым. На приведенных в спецификации рисунках это бит 0. Например, показанная на рисунке 11 последовательность битов задает десятичное число 170.

Подобно этому для многооктетных полей, представляющих числа, указанный слева бит является старшим. При передаче многооктетных значений старший октет передается первым.

Глоссарий

1822

BBN Report 1822, “The Specification of the Interconnection of a Host and an IMP” – спецификация интерфейса между хостом и сетью ARPANET.

ARPANET leader

Управляющая информация сообщения ARPANET на интерфейсе хост-IMP.

ARPANET message – сообщение ARPANET

Элемент передачи между хостом и IMP в сети ARPANET. Максимальный размер сообщения около 1012 октетов (8096 битов).

ARPANET packet – пакет ARPANET

Единица передачи данных, используемая в сети ARPANET при обмене данными между IMP. Максимальный размер пакета составляет 126 октетов (1008 битов).

Destination – получатель

Адрес получателя – поле заголовка IP.

DF

Флаг запрета фрагментирования (Don’t Fragment).

Flags – флаги

Поле заголовка IP, содержащее различные флаги управления.

Fragment Offset – смещение фрагмента

Поле заголовка IP, указывающее смещение данного фрагмента в исходной дейтаграмме.

GGP

Gateway to Gateway Protocol – протокол, используемый шлюзами для управления маршрутизацией и других целей.

Header – заголовок

Служебная информация в начале сообщения, сегмента, дейтаграммы, пакета или блока данных.

ICMP

Internet Control Message Protocol – протокол, реализованный в модуле IP и используемый шлюзами и хостами для передачи информации об ошибках и проверки маршрутов.

Identification – идентификация

Поле заголовка IP, содержащее идентификатор, присваиваемый отправителем для правильной сборки фрагментов.

IHL

Internet Header Length – поле заголовка IP, указывающее размер заголовка в 32-битовых словах.

IMP

Interface Message Processor – коммутатор пакетов в сети ARPANET.

Internet Address – адрес IP

4-октетный (32 бита) адрес отправителя или получателя, состоящий из полей Network (номер

сети) и Local Address (номер хоста).

Internet datagram – дейтаграмма IP

Единица информации, используемая при обмене данными между парой модулей IP (включает заголовок IP).

internet fragment – фрагмент IP

Часть дейтаграммы IP с заголовком IP.

Local Address – локальный адрес (номер хоста)

Адрес хоста в сети. Реальное отображение локальной части адреса internet на адрес хоста в сетиявляется достаточно общим и позволяет использовать отображения «множества на один».

MF

Флаг наличия других фрагментов (More-Fragments), передаваемый в поле флагов заголовка IP.

Module – модуль

Реализация (обычно программная) протокола или другой процедуры.

more-fragments flag

Флаг наличия других фрагментов (MF), передаваемый в поле флагов заголовка IP.

NFB

Number of Fragment Blocks – число блоков (по 8 октетов) данных во фрагменте дейтаграммы.

octet

Восьмибитовый байт.

Options – опции

Поле Options в заголовке IP может содержать тот или иной набор опций. Размер опции может быть переменным.

Padding – заполнение

Поле Padding используется для выравнивания заголовка IP по 32-битовой границе. Для заполнения используется значение 0.

Protocol – протокол

Поле Protocol заголовка IP, содержащее идентификатор протокола вышележащего уровня.

Rest – остаток

Локальная часть адреса IP (номер хоста).

Source – отправитель

Адрес отправителя в заголовке IP.

TCP

Transmission Control Protocol – протокол обмена данными между хостами, обеспечивающий гарантированную доставку в среде internet.

TCP Segment – сегмент TCP

Единица данных при обмене информацией между модулями TCP (включает заголовок TCP).

TFTP

Trivial File Transfer Protocol – простой протокол обмена файлами на основе транспортного протокола UDP.

Time to Live

Поле заголовка, определяющее верхнюю границу срока существования дейтаграммы IP.

TOS

ToS – тип обслуживания.

Total Length – общая длина

Поле заголовка Total Length показывает полный размер дейтаграммы в октетах с учетом заголовка и данных.

TTL

Время жизни (Time to Live).

Type of Service – тип обслуживания

Поле заголовка, определяющее тип (или качество) обслуживания дейтаграммы IP.

UDP

Протокол пользовательских дейтаграмм (User Datagram Protocol) – протокол пользовательского (транспортного в современной терминологии. *прим. перев.*) уровня для приложений на базе транзакций.

User

Пользователь IP – протокол вышележащего уровня, прикладная программа, программа шлюза.

Version

Поле версии определяет формат заголовка internet.

Литература

[1] Cerf, V., “The Catenet Model for Internetworking,” Information Processing Techniques Office, Defense Advanced Research Projects Agency, IEN 48, July 1978.

[2] Bolt Beranek and Newman, “Specification for the Interconnection of a Host and an IMP,” BBN Technical Report 1822, Revised May 1978.

[3] Postel, J., “Internet Control Message Protocol – DARPA Internet Program Protocol Specification,” RFC 792, USC/Information Sciences Institute, September 1981.

[4] Shoch, J., “Inter-Network Naming, Addressing, and Routing,” COMPCON, IEEE Computer Society, Fall 1978.

[5] Postel, J., “Address Mappings,” RFC 796, USC/Information Sciences Institute, September 1981.

[6] Shoch, J., “Packet Fragmentation in Inter-Network Protocols,” Computer Networks, v. 3, n. 1, February 1979.

[7] Strazisar, V., “How to Build a Gateway”, IEN 109[43], Bolt Beranek and Newman, August 1979.

[8] Postel, J., “Service Mappings,” RFC 795, USC/Information Sciences Institute, September 1981.

[9] Postel, J., “Assigned Numbers,” RFC 790, USC/Information Sciences Institute, September 1981.

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

1Министерство обороны США. *Прим. перев.*

2Internet Protocol.

3Протоколами вышележащего уровня. *Прим. перев.*

4Протоколов нижележащего (канального) уровня. *Прим. перев.*

5Интерфейсу канального уровня. *Прим. перев.*

6Протоколы транспортного уровня эталонной модели OSI в современной терминологии. *Прим. перев.*

7Протоколы канального уровня эталонной модели OSI в современной терминологии. *Прим. перев.*

8Интерфейсу канального уровня. *Прим. перев.*

9В современной терминологии. *Прим. перев.*

10В современном понимании терминов это выражение не совсем корректно, поскольку отображение адресов на маршруты (маршрутизация) осуществляется как раз на уровне IP, а не на канальном уровне. Преобразование адресов IP в адреса канального уровня (MAC-адреса) и обратно обеспечивается с помощью протоколов типа ARP/RARP. Пользуясь современной терминологией, лучше было бы сказать “отображение адресов локальной сети на каналы передачи”. *Прим. перев.*

11В современной терминологии – номер хоста. *Прим. перев.*

12Внедрение концепции бесклассовой междоменной маршрутизации (CIDR, см. RFC 1518) привело к отказу от деления адресов IP на классы и внесению соответствующих изменений в интерпретацию адресов и парадигму маршрутизации. *Прим. перев.*

13Отображение адресов.

14Ее называют intranet-фрагментацией.

15More-fragments

16Number of Fragment Blocks – число блоков фрагментации.

17Маршрутизаторы в соответствии с современной терминологией. *Прим. перев.*

18Gateway to Gateway Protocol – протокол обмена данными между шлюзами. В настоящее время для обмена информацией между маршрутизаторами используются более изощренные протоколы типа RIP, OSPF, BGP. *Прим. перев.*

19В параграфе 3.2.1.1 RFC 1122 указано, что дейтаграммы с некорректным номером версии должны отбрасываться без уведомления. *Прим. перев.*

20Internet Header Length – размер заголовка Internet. *Прим. перев.*

21Type of Service – тип обслуживания. *Прим. перев.*

22Предложенная в спецификации протокола IP концепция типа обслуживания практически не нашла своего применения и в настоящее время для задания типа обслуживания используется значение DSCP, задаваемое в полях TOS и Identification заголовка IP. Более подробную информацию вы сможете найти в RFC 2474. *Прим. перев.*

23Задержка, пропускная способность и надежность доставки, соответственно.

24Time to Live. *Прим. перев.*

25Транспортного (*прим. перев.*)

26В параграфе 3.2.1.2 RFC 1122 Указано, что принимающий хост должен проверять значение контрольной суммы и отбрасывать без уведомления пакеты с ошибочным значением. *Прим. перев.*

27Эта процедура вычисления контрольной суммы используется и сегодня. *Прим. перев.*

28В настоящее время для такого формата используется термин TLV (Type, Length, Value). *Прим. перв.*

29Department of Defense – Министерство обороны США. *Прим. перев.*

30 Encrypt(ed) For Transmission Only — зашифровано только для передачи.

31Опыт показал, что предложенная в спецификации схема распределения адресов недостаточно эффективна и ведет к быстрому истощению адресного пространства. Кроме того, распределение адресов по классам не согласуется с междоменной маршрутизации на основе протокола BGP. В документах RFC 1517, RFC 1518, RFC 1519 предложена парадигма бесклассовой междоменной маршрутизации (CIDR). *Прим. перев.*

32Номера подсетей и хостов в современной терминологии. *Прим. перев.*

33Транк, в современной терминологии. *Прим. перев.*

34В параграфе 3.2.1.4 RFC 1122 указано, что все хосты **должны** поддерживать фрагментацию и сборку дейтаграмм IP. *Прим. перев.*

35В параграфе 3.2.1.5 RFC 1122 указано, что при передаче идентичной копии ранее отправленной дейтаграммы хост **может** сохранять значение идентификационного поля. *Прим. перев.*

36Maximum transmission unit — максимальный передаваемый блок.

37В параграфе 3.2.1.6 RFC 1122 указан ряд дополнительных требований к полю TTL. *Прим. перев.*

38В параграфе 3.2.1.7 RFC 1122 указан ряд дополнительных требований к полю TTL. *Прим. перев.*

39В параграфе 3.2.1.8 RFC 1122 указан ряд дополнительных требований к полю опций. *Прим. перев.*

40Опции защиты, определенные здесь, утратили силу (см. RFC 1122). *Прим. перев.*

41 Каждая цифра второй строки соответствует одному биту.

42Перевод RFC 792 и частично обновляющего его RFC 950 доступен на сайте www.protocols.ru. *Прим. перев.*

43Этот документ частично обновлен в RFC 823. *Прим. перев.*

44Этот документ регулярно обновлялся (последнюю версию можно найти в RFC 1700), но в соответствии с RFC 3232 документ STD 2 утратил силу. Значения Assigned Numbers следует искать в базе данных, доступной на сайте www.iana.org/numbers.html. *Прим. перев.*