

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<sup>1</sup> для протокола Internet (IP<sup>2</sup>). 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 распространённых протоколов сетевого уровня эталонной модели 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 породить 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*

### Оглавление

Предисловие.....	1
1. Введение.....	2
1.1. Мотивация.....	2
1.2. Сфера действия протокола.....	2
1.3. Интерфейсы.....	2
1.4. Работа протокола.....	3
2. Обзор.....	3
2.1. Связь с другими протоколами.....	3
2.2. Модель работы протокола.....	3
2.3. Функциональное описание.....	4
Адресация.....	4
Фрагментация.....	4
2.4. Шлюзы.....	5
3. Спецификация.....	5
3.1. Формат заголовка.....	5
Version - 4 бита.....	5
IHL - 4 бита.....	6
ToS - 8 битов.....	6
Total Length - 16 битов.....	6
Identification - 16 битов.....	6
Flags - 3 бита.....	6

<sup>1</sup>Министерство обороны США. Прим. перев.

<sup>2</sup>Internet Protocol.

Fragment Offset - 13 битов.....	7
TTL - 8 битов.....	7
Protocol - 8 битов.....	7
Header Checksum - 16 битов.....	7
Source Address - 32 бита.....	7
Destination Address - 32 бита.....	7
Options - переменный размер.....	7
Определения отдельных опций.....	8
End of Option List - конец списка опций.....	8
No Operation - нет операции.....	8
Security - безопасность.....	8
Security (S) - 16 битов.....	8
Compartments (C) - 16 битов.....	8
Handling Restrictions (H) - 16 битов.....	8
Transmission Control Code (поле TCC) - 24 бита.....	8
Loose Source and Record Route - нестрогое задание маршрута отправителем.....	8
Strict Source and Record Route - строгое задание маршрута отправителем.....	9
Record Route - запись маршрута.....	9
Stream Identifier - идентификатор потока.....	9
Internet Timestamp - временная метка.....	10
Padding - переменный размер.....	10
3.2. Обсуждение.....	10
Адресация.....	10
Формат адресов.....	11
Фрагментация и сборка.....	11
Пример процедуры фрагментации.....	11
Обозначения.....	12
Процедура.....	12
Пример процедуры сборки.....	12
Обозначения.....	12
Процедура.....	13
Идентификация.....	13
ToS.....	13
TTL.....	14
Опции.....	14
Контрольная сумма.....	14
Ошибки.....	14
3.3. Интерфейсы.....	14
Пример интерфейса с вышележащим уровнем.....	14
Приложение А: Примеры и сценарии.....	15
Пример 1.....	15
Пример 2.....	16
Пример 3.....	16
Приложение В: Порядок передачи данных.....	17
Глоссарий.....	17
Литература.....	18

## 1. Введение

### 1.1. Мотивация

Протокол IP предназначен для использования в соединённых 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 фиксированного размера. 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 соединённых между собой сетей. Протокол не поддерживает механизмов повышения надёжности 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»<sup>1</sup> и сам вызывает функции локальных сетевых протоколов<sup>2</sup> 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<sup>3</sup> для передачи дейтаграммы.

<sup>1</sup>Протоколами вышележащего уровня. *Прим. перев.*

<sup>2</sup>Протоколов нижележащего (канального) уровня. *Прим. перев.*

<sup>3</sup>Интерфейсу канального уровня. *Прим. перев.*

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 ещё 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 и вычисленная заново при обработке контрольная сумма не совпадёт 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.

## 2. Обзор

### 2.1. Связь с другими протоколами

На рисунке 1 показаны связи IP с другими протоколами.



Рисунок 1. Связь с другими протоколами.

Протокол IP взаимодействует с протоколом вышележащего уровня (протокол взаимодействия между хостами<sup>1</sup>) и с нижележащим протоколом локальной сети<sup>2</sup> (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.

<sup>1</sup>Протокол транспортного уровня эталонной модели OSI в современной терминологии. Прим. перев.

<sup>2</sup>Протокол канального уровня эталонной модели OSI в современной терминологии. Прим. перев.

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 передаёт дейтаграмму и локальный адрес локальному сетевому интерфейсу<sup>1</sup>.

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

Дейтаграмма приходит на хост-шлюз в кадре канального уровня. Интерфейс канального уровня удаляет заголовок канального уровня и передаёт 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 создаёт заголовок и, добавив к нему дейтаграмму, передаёт пакет хосту-адресату.

На хосте получателя дейтаграмма выделяется из пакета интерфейсом канального уровня и передаётся модулю IP.

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

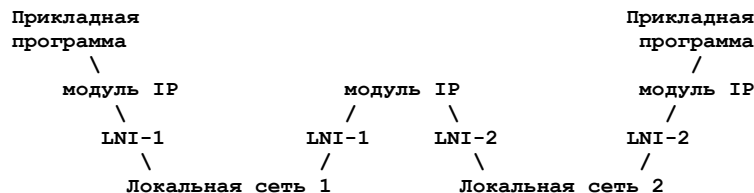


Рисунок 2. Путь передачи данных.

### 2.3. Функциональное описание

Задачей протокола IP является перемещение дейтаграмм через множество соединённых между собою сетей. Эта задача решается путём 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<sup>2</sup>). 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)<sup>3</sup>.

Адреса IP имеют фиксированную длину - 4 октета (32 бита). Адрес начинается с номера сети, за которым следует локальный адрес<sup>4</sup> (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 и класс C, в котором три старших бита имеют значения 110, следующие 21 - образуют номер сети, а последние 8 битов определяют номер хоста<sup>5</sup>.

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» [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.

<sup>1</sup>Интерфейсу канального уровня. Прим. перев.

<sup>2</sup>В современной терминологии. Прим. перев.

<sup>3</sup>В 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.

<sup>4</sup>В современной терминологии - номер хоста. Прим. перев.

<sup>5</sup>Внедрение концепции бесклассовой междоменной маршрутизации (CIDR, см. RFC 1812) привело к отказу от деления адресов IP на классы и внесению соответствующих изменений в интерпретацию адресов и парадигму маршрутизации. Прим. перев.

Допускается использование невидимой для модуля IP фрагментации<sup>1</sup>, 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 данном фрагменте. Сброшенный флаг наличия последующих фрагментов<sup>2</sup> 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) создаёт 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. Назовём число 8-октетных блоков в первой части дейтаграммы NFB<sup>3</sup>. Первая часть дейтаграммы помещается в первую из новых дейтаграмм IP и поле размера 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 размером новой дейтаграммы. Значение флага наличия дополнительных фрагментов копируется из исходной нефрагментированной дейтаграммы. Значение поля смещения увеличивается на величину NFB по сравнению с исходной дейтаграммой.

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

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 путём 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<sup>4</sup> выполняют пересылку дейтаграмм IP между сетями, обеспечивая также поддержку протокола GGP<sup>5</sup> [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.

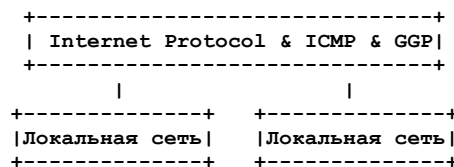


Рисунок 3. Протоколы шлюзов.

## 3. Спецификация

### 3.1. Формат заголовка

Формат заголовка дейтаграмм IP показан на рисунке 4.

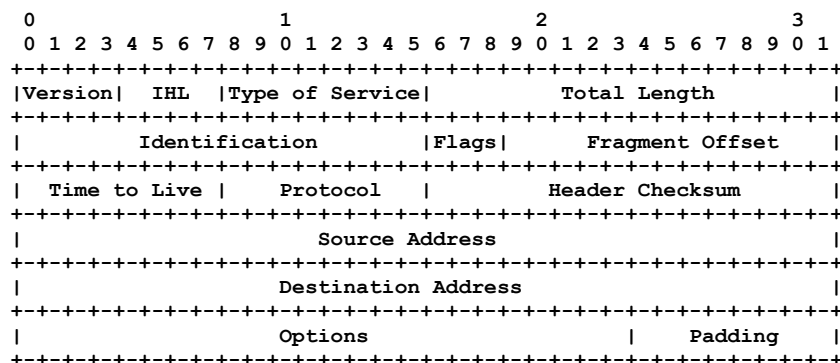


Рисунок 4. Формат заголовка дейтаграммы IP.

### Version - 4 бита

Указывает номер версии протокола и определяет формат заголовка. Данная спецификация описывает версию 4<sup>6</sup>.

<sup>1</sup>Ее называют intranet-фрагментацией.

<sup>2</sup>More-fragments - MF.

<sup>3</sup>Number of Fragment Blocks - число блоков фрагментации.

<sup>4</sup>Маршрутизаторы в соответствии с современной терминологией. *Прим. перев.*

<sup>5</sup>Gateway to Gateway Protocol - протокол обмена данными между шлюзами. В настоящее время для обмена информацией между маршрутизаторами используются более изощрённые протоколы маршрутизации, такие как RIP, OSPF, BGP. *Прим. перев.*

<sup>6</sup>В параграфе 3.2.1.1 [RFC 1122](#) указано, что дейтаграммы с некорректным номером версии должны отбрасываться без уведомления отправителя. *Прим. перев.*

**IHL<sup>1</sup> - 4 бита**

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

**ToS<sup>2</sup> - 8 битов**

Поле ToS<sup>3</sup> 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<sup>4</sup> (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 служит 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 или фрагмента<sup>5</sup> с учётом заголовка и данных. Размер этого поля позволяет создавать дейтаграммы размером 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 разумного размера. Например, такой размер позволяет передавать блок данных в 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.

<sup>1</sup>Internet Header Length - размер заголовка Internet. *Прим. перев.*

<sup>2</sup>Type of Service - тип обслуживания. *Прим. перев.*

<sup>3</sup>Предложенная 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 можно найти в [RFC 2474](https://www.rfc-editor.org/rfc/2474). *Прим. перев.*

<sup>4</sup>Задержка, пропускная способность и надёжность доставки, соответственно.

<sup>5</sup>В оригинале слова «или фрагмента» отсутствовали. См. <https://www.rfc-editor.org/errata/eid2519>. *Прим. перев.*

бит 2: (MF) 0 = последний фрагмент,

1 = имеются другие фрагменты.

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<sup>1</sup> - 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<sup>2</sup> уровня, содержащийся в поле данных дейтаграммы IP. Идентификаторы протоколов указаны в разделе ASSIGNED INTERNET PROTOCOL NUMBERS документа Assigned Numbers [9].

### Header Checksum - 16 битов

Контрольная сумма полей заголовка<sup>3</sup>. 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 её адекватность, но алгоритм вычисления контрольной суммы может быть заменён специальной процедурой CRC с учётом опыта использования<sup>4</sup>.

### Source Address - 32 бита

Адрес отправителя (см. параграф 3.2. Обсуждение).

### Destination Address - 32 бита

Адрес получателя (см. параграф 3.2. Обсуждение).

### Options - переменный размер

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 переменный размер. Существует два варианта форматирования опций:

однооктетные опции;

октет типа опции, октет размера опции и октеты собственно опции<sup>5</sup> (поле размера опции учитывает размеры полей 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.

Номера определенных опций перечислены в таблице.

Класс	Номер	Размер	Описание
0	0	-	End of Option list. Эта опция занимает только 1 октет и не использует поле размера.
0	1	-	No operation. Эта опция занимает только 1 октет и не использует поле размера.
0	2	11	Security. Используется для передачи опций Security (безопасность), Compartmentation (изоляция), User Group (TCC), Handling Restriction Codes (коды обработки ограничений), совместимых с требованиями DOD.
0	3	перем.	Loose Source Routing. Используется для маршрутизации дейтаграмм IP с учетом данных, указанных отправителем.

<sup>1</sup>Time to Live – время жизни. *Прим. перев.*

<sup>2</sup>Транспортного в современной терминологии. *Прим. перев.*

<sup>3</sup>В 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.*

<sup>4</sup>Упомянутая выше процедура вычисления контрольной суммы используется и сегодня. *Прим. перев.*

<sup>5</sup>В настоящее время для такого формата используется термин TLV (Type, Length, Value). *Прим. перев.*

0	9	перем.	Strict Source Routing. Используется для маршрутизации дейтаграмм IP на основе данных, указанных отправителем.
0	7	перем.	Record Route. Используется для трассировки пути дейтаграмм IP.
0	8	4	Stream ID. Используется для обозначения потоков дейтаграмм.
2	4	перем.	Internet Timestamp.

## Определения отдельных опций

### End of Option List - конец списка опций

```
+-----+
|00000000|
+-----+
Type=0
```

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 - нет операции

```
+-----+
|00000001|
+-----+
Type=1
```

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

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

### Security - безопасность

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

```
+-----+-----+---//---+---//---+---//---+---//---+
|10000010|00001011|sss sss|ccc ccc|nnn nnn| tcc |
+-----+-----+---//---+---//---+---//---+---//---+
Type=130 Length=11
```

### Security (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 - ЕФТО1
10111100 01001101 - ММММ
01011110 00100110 - ПРОГ
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 (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<sup>2</sup>.

### Handling Restrictions (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 ТСС) - 24 бита

Служит для разделения трафика и определения контролируемых «групп по интересам». Значения являются триграфами (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 - нестрогое задание маршрута отправителем

```
+-----+-----+-----+-----+-----+
|10000011| length | pointer| route data |
+-----+-----+-----+-----+-----+
Type=131
```

Опция LSRR обеспечивает отправителям дейтаграмм IP способ предоставления маршрутной информации, используемой маршрутизаторами при пересылке дейтаграмм адресату и записи маршрутной информации.

Опция начинается с кода типа (131), второй октет содержит значение размера опции с учётом полей типа и размера опции, а также октета указателя и собственно опции (маршрутных данных). Третий октет опции указывает октет

<sup>1</sup>Encrypt(ed) For Transmission Only - зашифровано только для передачи.

<sup>2</sup>Агентство военной разведки США. Прим. перев.



маршрутных данных, начиная с которого будет обрабатываться следующий адрес отправителя. Отсчёт смещения ведётся 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 размер 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 размер 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 - строгое задание маршрута отправителем

```
+-----+-----+-----+-----+-----+-----+-----+
|10001001| length | pointer|      route data   |
+-----+-----+-----+-----+-----+-----+
Type=137
```

Опция SSRR 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, размера, 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.

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 - запись маршрута

```
+-----+-----+-----+-----+-----+-----+-----+
|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 размера, 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 размер 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, заданного 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 - идентификатор потока

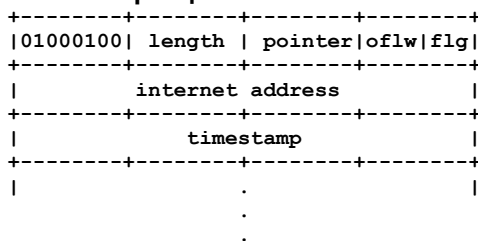
```
+-----+-----+-----+-----+-----+-----+-----+
|10001000|00000100|      Stream ID   |1
+-----+-----+-----+-----+-----+-----+
Type=136 Length=4
```

<sup>1</sup>В оригинале поле размера было ошибочно указано со значением 0000010. См. <https://www.rfc-editor.org/errata/eid716>. Прим. перев.

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

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

### Internet Timestamp - временная метка



Type = 68

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

Поле указателя (Pointer) показывает число октетов от начала этой опции конца временных меток + 1, т. е. указывает октет, начинающий пространство для следующей временной метки. 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 нестандартного значения 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 ещё остаётся 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 передаётся только в первом фрагменте. Опция появляется в дейтаграмме только один раз.

### Padding - переменный размер

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 взаимодействие с продукцией других разработчиков. Хотя целью данной спецификации является чёткое 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 далее в 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 множества сетей небольших и средних размеров используется специальная интерпретация полей адреса IP<sup>1</sup>. Такая интерпретация позволяет выделить небольшое число адресов для крупных сетей, большее число адресов для сетей среднего размера и многочисленные адреса для

<sup>1</sup>Опыт показал, что предложенная в спецификации схема распределения адресов недостаточно эффективна и ведёт к быстрому истощению адресного пространства. Кроме того, распределение адресов по классам не согласуется с междоменной маршрутизацией на основе протокола BGP. В документах [RFC 1517](#), [RFC 1518](#), [RFC 1519](#) предложена парадигма бесклассовой междоменной маршрутизации (CIDR). *Прим. перев.*

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 задаёт 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 определён. Обе эти возможности зарезервированы для использования в будущем.

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

Локальные адреса<sup>1</sup> 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<sup>2</sup>.

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

## Фрагментация и сборка<sup>3</sup>

Поле идентификации (ID) используется вместе с адресами отправителя/получателя и полем протокола для идентификации фрагментов дейтаграммы при сборке<sup>4</sup>.

Флаг наличия других фрагментов (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 общий размер дейтаграммы, а не её 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 приём дейтаграмм размером по меньшей 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 приведённом ниже псевдокоде означает «меньше или равно», # - «не равно», = - «равно», <- - «установить значение». Выражение «x - y» задаёт 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<sup>5</sup>.

Если общий размер дейтаграммы не превышает MTU, эта дейтаграмма передаётся на следующий этап обработки. В противном случае дейтаграмма делится на два фрагмента - первый фрагмент имеет максимальный размер, а во

<sup>1</sup>Номера подсетей и хостов в современной терминологии. Прим. перев.

<sup>2</sup>Транк, в современной терминологии. Прим. перев.

<sup>3</sup>В параграфе 3.2.1.4 RFC 1122 указано, что все хосты **должны** поддерживать фрагментацию и сборку дейтаграмм IP. Прим. перев.

<sup>4</sup>В параграфе 3.2.1.5 RFC 1122 указано, что при передаче идентичной копии ранее отправленной дейтаграммы хост **может** сохранять значение идентификационного поля. Прим. перев.

<sup>5</sup>Maximum transmission unit - максимальный передаваемый блок.

втором содержится оставшаяся часть дейтаграммы. Первый фрагмент передаётся 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	- размер 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**

IF TL ≤ MTU

THEN дейтаграмма передаётся на следующий этап обработки,

ELSE

IF DF = 1,

THEN дейтаграмма отбрасывается,

ELSE 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 передаётся на следующий этап обработки;
- переход ко второму фрагменту:
- (7) частичное копирование заголовка IP (некоторые опции не копируются; см. описания опций);
  - (8) добавляется оставшаяся часть данных;
  - (9) корректируется заголовок:  
IHL ← ((OIHL \* 4) - (размер опций не копируется)) / 4;  
TL ← OTL - NFB \* 8 - (OIHL - IHL) \* 4;  
FO ← OFO + NFB; MF ← OMF;  
заново вычисляется контрольная сумма;
  - (10) фрагмент передаётся процедуре проверки размера.

После выполнения п. 10 процедура завершается (если размер фрагмента не превышает допустимое значение) или повторяется. Эта процедура создаёт 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 путём 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 передаётся 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 устанавливаются 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 приём дейтаграммы (проверяется по таблице блоков), дейтаграмма передаётся 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 передаётся процедуре сборки фрагментов.

Если заданное таймером время истекло, освобождаются все ресурсы, выделенные для данного идентификатора буфера. Начальная установка таймера задаёт 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 учётом реального опыта 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	- размер 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 с октета FO*8 до (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-первого-фрагмента*4)1
(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 закончится время;
(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 и получателя, тип протокола и поле идентификации. Таким образом, отправитель должен выбирать значение идентификатора так, чтобы оно было уникальным для комбинации адресов отправителя и получателя, а также типа протокола в течение срока жизни дейтаграммы (или любого её 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**

Значение TOS<sup>2</sup> обеспечивает выбор уровня обслуживания с помощью абстрактных параметров precedence (предпочтение, приоритет), delay (задержка), throughput (пропускная способность), reliability (надёжность). 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 надёжность (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

```

<sup>1</sup>В оригинале ошибочно сказано THEN TL <- TDL+(IHL\*4). См. <https://www.rfc-editor.org/errata/eid6356>. Прим. перев.

<sup>2</sup>В параграфе 3.2.1.6 RFC 1122 указан ряд дополнительных требований к полю TTL. Прим. перев.

Reliability: 1

В этом случае отображение параметров сервиса на поддерживаемые в сети ARPANET параметры обслуживания приведёт 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

Значение TTL<sup>1</sup> устанавливается отправителем и задаёт 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 меньшей 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 даёт таким протоколам гарантию истинности этого предположения.

## Опции

Опции являются необязательными для дейтаграмм, но все реализации должны поддерживать опции<sup>2</sup>. Т. е., использование опций определяется источником, 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<sup>3</sup> (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 неприемлема 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 состояния практически не сохраняются и пользователь при каждом обращении к модулю IP передаёт 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;

<sup>1</sup>В параграфе 3.2.1.7 [RFC 1122](#) указан ряд дополнительных требований к полю TTL. *Прим. перев.*

<sup>2</sup>В параграфе 3.2.1.8 [RFC 1122](#) указан ряд дополнительных требований к полю опций. *Прим. перев.*

<sup>3</sup>Опции защиты, определённые здесь, утратили силу (см. [RFC 1122](#)). *Прим. перев.*

len = размер буфера;  
 id = идентификатор;  
 DF = запрет фрагментирования;  
 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 пользователя с указанным адресом не существует, отправителю возвращается сообщение ICMP об ошибке и дейтаграмма уничтожается.

Для уведомления пользователя могут применяться псевдопрерывания 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. Используемые обозначения похожи на нотацию большинства языков высокого уровня, однако такое применение не требует «функций-ловушек» (например, SVC, UUC, EMT) или иных форм взаимодействия между процессами.

## Приложение А: Примеры и сценарии

### Пример 1

На рисунке 5 приведен пример минимальной дейтаграммы IP, содержащей данные<sup>1</sup>.

```

      0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Ver= 4 |IHL= 5 |Type of Service|           Total Length = 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Identification = 111   |Flg=0|   Fragment Offset = 0   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Time = 123 | Protocol = 1 |           header checksum   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     source address           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     destination address      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           data           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Рисунок 5. Пример дейтаграммы IP.

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).

<sup>1</sup> Каждая цифра второй строки соответствует одному биту.

**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.

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|Ver= 4 |IHL= 5 |Type of Service|       Total Length = 472 |
+-----+-----+-----+-----+
|      Identification = 111      |Flg=0|      Fragment Offset = 0 |
+-----+-----+-----+-----+
|   Time = 123 | Protocol = 6 |       header checksum |
+-----+-----+-----+-----+
|                                     source address |
+-----+-----+-----+-----+
|                                     destination address |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+
| \                                     \ |
| \                                     \ |
|                                     data |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+

```

Рисунок 6. Пример большой дейтаграммы IP.

Рисунок 7 показывает первый фрагмент, после выделения из дейтаграммы первых 256 октетов данных.

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|Ver= 4 |IHL= 5 |Type of Service|       Total Length = 276 |
+-----+-----+-----+-----+
|      Identification = 111      |Flg=1|      Fragment Offset = 0 |
+-----+-----+-----+-----+
|   Time = 119 | Protocol = 6 |       Header Checksum |
+-----+-----+-----+-----+
|                                     source address |
+-----+-----+-----+-----+
|                                     destination address |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+
| \                                     \ |
| \                                     \ |
|                                     data |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+

```

Рисунок 7. Пример первого фрагмента дейтаграммы IP.

Второй фрагмент будет иметь форму, показанную на рисунке 8.

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|Ver= 4 |IHL= 5 |Type of Service|       Total Length = 216 |
+-----+-----+-----+-----+
|      Identification = 111      |Flg=0|      Fragment Offset = 32 |
+-----+-----+-----+-----+
|   Time = 119 | Protocol = 6 |       Header Checksum |
+-----+-----+-----+-----+
|                                     source address |
+-----+-----+-----+-----+
|                                     destination address |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+
| \                                     \ |
| \                                     \ |
|                                     data |
+-----+-----+-----+-----+
|                                     data |
+-----+-----+-----+-----+

```

Рисунок 8. Пример фрагмента дейтаграммы IP.

**Пример 3**

Рисунок 9 показывает пример дейтаграммы, содержащей опции.



```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Ver= 4 |IHL= 8 |Type of Service|          Total Length = 576 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Identification = 111 |Flg=0|          Fragment Offset = 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Time = 123 | Protocol = 6 |          Header Checksum |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          source address |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          destination address |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Opt. Code = x | Opt. Len.= 3 | option value | Opt. Code = x |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Opt. Len. = 4 |          option value | Opt. Code = 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Opt. Code = y | Opt. Len. = 3 | option value | Opt. Code = 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          data |
\          \
\          \
|          data |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          data |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Рисунок 9. Пример дейтаграммы IP с опциями.

## Приложение В: Порядок передачи данных

Порядок передачи заголовков и данных, описываемых в этой спецификации, задаётся 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.

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          1 |          2 |          3 |          4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          5 |          6 |          7 |          8 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          9 |         10 |         11 |         12 |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Рисунок 10. Порядок передачи байтов.

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.

```

0 1 2 3 4 5 6 7
+-----+-----+
|1 0 1 0 1 0 1 0|
+-----+-----+

```

Рисунок 11. Значимость битов.

Подобно этому для многооктетных полей, представляющих числа, указанный слева бит является старшим. При передаче многооктетных значений старший октет передаётся 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 сети является 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 - общий размер**

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**

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](#)<sup>1</sup>, USC/Information Sciences Institute, September 1981.
- [4] Shoch, J., "Inter-Network Naming, Addressing, and Routing," COMPCON, IEEE Computer Society, Fall 1978.

<sup>1</sup>Частично обновлён [RFC 950](#). Прим. перев.

- [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<sup>1</sup>, 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<sup>2</sup>, USC/Information Sciences Institute, September 1981.

**Перевод на русский язык****Николай Малых**[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)

---

<sup>1</sup>Этот документ частично обновлён в RFC 823. *Прим. перев.*

<sup>2</sup>Этот документ регулярно обновляется (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](http://www.iana.org/numbers.html). *Прим. перев.*