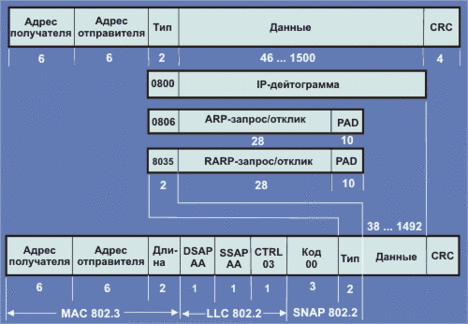
# Приложение 2

## Основные единицы обмена для различных уровней стека TCP/IP

### 2.1. Типы кадров технологии Ethernet (размеры полей указаны в байтах)



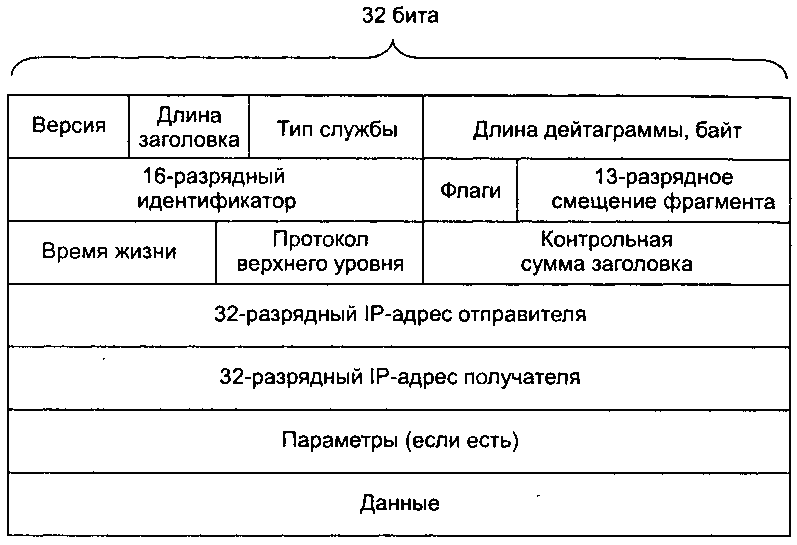
### 2.2. Ethernet инкапсуляция (RFC 894, размеры полей указаны в байтах.)



### 2.3. Алгоритм определения формата кадра

Отличить один формат кадра Ethernet от другого не представляет большого труда, и сделать это можно с помощью следующего простого алгоритма. Сначала программа должна проверить значение поля типа протокола/длины кадра (13-й и 14-й байты в заголовке). Если записанное там значение превышает 0x05DC (максимально возможная длина поля данных), то это кадр Ethernet\_II, иначе следует продолжить проверку. Если первые два байта поля данных равны 0xFFFF, то это формат Ethernet\_802.3 для NetWare 3.х. В противном случае это стандартный формат кадра 802.2, и остается только выяснить, какой из двух - обычный (Ethernet\_802.2) или расширенный (Ethernet\_SNAP). В случае Ethernet\_SNAP значение первого и второго байтов в поле данных равняется 0xAA.

2.4. Ключевые поля IPv4-дейтаграммы.



***□ Версия.*** Четыре бита в этом поле определяют номер версии протокола IP. По этому номеру маршрутизатор может определить, как интерпретировать остальные поля IP-дейтаграммы. В различных версиях протокола IP применяются различные форматы IP-дейтаграмм. На рисунке показан формат дейтаграммы текущей версии протокола IP **(IPv4)**.

***□ Длина заголовка.*** Поскольку IPv4-дейтаграмма может содержать разное количество необязательных полей параметров (включаемых в заголовок IPv4-дейтаграммы), эти четыре бита необходимы для того, чтобы определить, где заканчивается заголовок и начинаются данные. В большинстве IP-дейтаграмм не содержатся поля параметров, поэтому обычно заголовок IP-дейтаграммы 20-байтный.

***□ Тип службы.*** Поле типа службы (Type Of Service, TOS) было включено в заголовок IPv4-дейтаграммы, чтобы была возможность разделять IP-дейтаграммы на типы (например, выделять дейтаграммы, для которых требуется низкая задержка, или высокая пропускная способность, или высокая надежность). Так, может оказаться полезным отличать дейтаграммы реального времени (например, используемые в IP-телефонии) от прочего трафика (например, FTP).

***□ Длина дейтаграммы.*** Это полная длина IP-дейтаграммы (*заголовок плюс данные*) в байтах. Поскольку размер этого поля равен 16 бит, теоретически максимальный размер IP-дейтаграммы может составлять 65 535 байт. Однако размер дейтаграмм редко превосходит 1500 байт и обычно ограничивается значением 576 байт.

***□ Идентификатор, флаги, смещение фрагмента.*** Эти три поля имеют отношение к так называемой IP-фрагментации. Этот вопрос мы подробно рассмотрим чуть позже. Интересно отметить, что новая версия протокола IP (IPv6) запрещает фрагментацию в маршрутизаторах.

***□ Время жизни.*** Поле времени жизни (Time То Live, TTL) позволяет гарантировать, что дейтаграммы не будут вечно циркулировать в сети (например, из-за существующей в течение долгого времени маршрутной петли). Значение этого поля уменьшается на единицу на каждом маршрутизаторе. Когда значение поля TTL достигает нуля, маршрутизатор отбрасывает дейтаграмму.

***□ Протокол.*** Это поле используется только тогда, когда IP-дейтаграмма достигает конечного адресата. Значение поля определяет протокол транспортного уровня, которому следует передать данные из IP-дейтаграммы. Например, значение 6 означает, что порция данных должна быть передана протоколу TCP, а значение 17 — протоколу UDP. Список всех возможных номеров имеется в RFC 1700, RFC 3232. Обратите внимание, что роль номера протокола в IP-дейтаграмме полностью аналогична роли номера порта в сегменте транспортного уровня. Номер протокола представляет собой «клей», связывающий вместе сетевой и транспортный уровни, тогда как номер порта связывает транспортный уровень с прикладным ( в кадре канального уровня также есть специальное поле, связывающее канальный уровень с сетевым уровнем).

***□ Контрольная сумма заголовка.*** Контрольная сумма заголовка помогает маршрутизатору обнаруживать ошибки в полученных IP-дейтаграммах. Контрольная сумма заголовка вычисляется путем суммирования всех двухбайтовых слов заголовка в дополнительном коде. Маршрутизатор вычисляет контрольную сумму заголовка для каждой полученной дейтаграммы и таким образом проверяет ошибки в заголовке. Как правило, маршрутизаторы отбрасывают дейтаграммы, в которых обнаруживают ошибки. Обратите внимание, что контрольную сумму нужно вычислять заново и снова сохранять в поле заголовка на каждом маршрутизаторе, так как на единицу уменьшается поле времени жизни, могут также измениться поля параметров (описание быстрых алгоритмов для вычисления контрольной суммы заголовка IP-дейтаграммы содержится в RFC 1071).

***□ IP-адреса отправителя и получателя.*** Эти поля содержат 32-разрядные IP-адреса отправителя и конечного получателя IP-дейтаграммы.

***□ Параметры.*** Поле параметров позволяет расширить IP-заголовок. Параметры заголовка представляют собой редко используемые необязательные поля IP-дейтаграммы.

***□ Данные (полезная нагрузка).*** Наконец, мы добрались до последнего, самого важного поля, ради которого и существует дейтаграмма! В большинстве случаев поле данных IP-дейтаграммы содержит сегмент транспортного уровня (TCP или UDP), который необходимо доставить адресату. Однако поле данных может содержать и другие типы данных, например сообщения протокола ICMP.

### 2.5. Формат UDP-сообщений



Формат UDP-дейтаграммы.

***Основные поля:***

***Длина UDP-сообщения*** равна числу байт в UDP-дейтаграмме, включая заголовок.

***Контрольная сумма*** содержит код, полученный в результате контрольного суммирования UDP-заголовка и поля данные. (Не трудно видеть, что этот протокол использует заголовок минимального размера (***8 байт***)).

Номера портов от 0 до 1023 стандартизованы - использовать их в прикладных задачах не рекомендуется, поэтому прежде, чем использовать какой-то порт в своей программе, следует заглянуть в RFC-1700.

### 2.6. Формат TCP - сегмента

Протокол **TCP (transmission control protocol**, RFC-793, -1323, -1644[T/TCP], -2018, -2581, -2582[RENO], -2861, -2873, -2883[SACK], -2923[MTU], -2988[RTO], -3293[GSMP], -3448[TFRC], -3465, -3481) в отличии от UDP осуществляет доставку дейтограмм, называемых сегментами, в виде байтовых потоков с установлением соединения. Протокол TCP применяется в тех случаях, когда требуется гарантированная доставка сообщений. Он использует контрольные суммы пакетов для проверки их целостности, и освобождает прикладные процессы от необходимости таймаутов и повторных передач для обеспечения надежности. Для отслеживания подтверждения доставки в TCP реализуется алгоритм "скользящего" окна. Внутренняя структура модуля TCP гораздо сложнее структуры UDP. Подобно UDP прикладные процессы взаимодействуют с модулем TCP через порты. Под байтовыми потоками здесь подразумевается то, что один примитив, например, **read** или **write** может вызвать посылку адресату последовательности сегментов, которые образуют некоторый блок данных (сообщение).

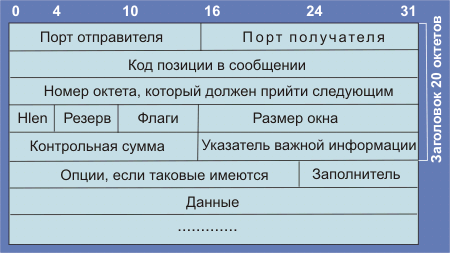
Хотя протоколы UDP и TCP могли бы для сходных задач использовать разные номера портов, обычно этого не происходит. Модули TCP и UDP выполняют функции мультиплексоров/демультиплексоров между прикладными процессами и IP-модулем. При поступлении пакета в модуль IP он будет передан в TCP- или UDP-модуль согласно коду, записанному в поле протокола данного IP-пакета.

Если IP-протокол работает с адресами, то TCP, также как и UDP, с портами. Именно с ***номеров портов*** отправителя и получателя начинается заголовок TCP-сегмента.

Поле ***код позиции в сообщении*** определяет порядковый номер первого октета в поле данных пользователя.

Поле ***Hlen*** – определяет длину заголовка сегмента, которая измеряется в 32-разрядных словах.

Далее следует поле ***резерв***, предназначенное для будущего использования, в настоящее время должно обнуляться.



Формат TCP сегмента

Поле ***размер окна*** сообщает, сколько октетов готов принять получатель. Окно имеет принципиальное значение, оно определяет число сегментов, которые могут быть посланы без получения подтверждения. Значение ширины окна может варьироваться во время сессии. Значение этого поля равное нулю также допустимо и указывает, что байты вплоть до указанного в поле *номер октета, который должен прийти следующим*, получены, но адресат временно не может принимать данные. Разрешение на посылку новой информации может быть дано с помощью посылки сегмента с тем же значением поля *номер октета, который должен прийти следующим*, но ненулевым значением поля ширины окна.

Поле ***контрольная сумма*** предназначено для обеспечения целостности сообщения.

Поле ***указатель важной информации*** представляет собой указатель последнего байта, содержащий информацию, которая требует немедленного реагирования.

Поле ***опции*** зарезервировано на будущее и в заголовке может отсутствовать, его размер переменен и дополняется до кратного 32-бит с помощью поля *заполнитель*.

Поле ***данные*** в TCP-сегменте может и отсутствовать, характер и формат передаваемой информации задается исключительно прикладной программой, максимальный размер этого поля составляет в отсутствии опций 65495 байт. TCP является протоколом, который ориентируется на согласованную работу хостов и программного обеспечения партнеров, участвующих в обмене информацией.