Network Working Group
Request for Comments: 893

Samuel J. Leffler
Michael J. Karels
University of California at Berkeley
April 1984

**Трейлерная инкапсуляция**

**Trailer Encapsulations**

## Статус документа

В этом RFC рассматриваются мотивы использования трейлерной инкапсуляции в локальных сетях и описана реализации такой инкапсуляции в разных средах. Документ служит лишь для информации и **не является** официальным стандартом сообщества ARPA Internet.

## *Введение*

Трейлерная инкапсуляция представляет собой формат пакетов канального уровня, используемый в 4.2BSD UNIX (среди прочих). Трейлерная инкапсуляция или «трейлер» может генерироваться системой при некоторых обстоятельствах для минимизации числа и размера операций копирования в памяти (memory-to-memory), выполняемых принимающим хостом при обработке пакета данных. Трейлер является форматом пакетов исключительно канального уровня и невидим (при подобающей реализации) при обработке на вышележащих уровнях. В этой заметке описаны мотивы использования трейлерной инкапсуляции и формат пакетов, используемый в сетях 3 Mb/s Experimental Ethernet, 10 Mb/s Ethernet и сетях 10 Mb/s V2LNI с кольцевой топологией [1].

Трейлерную инкапсуляцию предложил Greg Chesson, а описанную здесь инкапсуляцию разработал Bill Joy.

## *Мотивация*

Причиной использования трейлеров стали издержки, которые могут возникать при протокольной обработке, когда требуется одна или несколько операций копирования данных в памяти. Копирование может потребоваться на разных уровнях обработки, начиная от переноса данных из сетевой среды в память хоста и заканчивая передачей данных между операционной системой и пространством пользовательских адресов. От оптимальной реализации сети можно ожидать отсутствия операций копирования между доставкой пакета данных в память и представлением соответствующих данных принимающему процессу. Хотя многие пакеты не удается обработать без тех или иных операций копирования, при поддержке хост-компьютером подходящего управления памятью часто становится возможным предотвращение копирования за счет простых манипуляций, обеспечиваемых оборудованием виртуальной памяти.

В среде со страничным отображением виртуальной памяти обычно требуется выполнение двух условий, чтобы избежать операций копирования в процессе обработки пакетов. Доставленные для принимающего агента данные должны выравниваться по границе страницы памяти, а размер из должен быть кратным размеру аппаратной страницы памяти (или заполнять страницу до конца). Второе ограничение предполагает поддержку защиты вирутальной памяти на уровне страницы и в разных вариантах архитектуры это условие может меняться.

Данные, передаваемые через сеть, легко сегментировать до подходящего размера, но пока информация заголовка протокола инкапсуляции не имеет фиксированного размера, выравнивание по границам страниц практически невозможно. Информация протокольного заголовка может существенно меняться в результате применения разных протоколов (каждый со своим заголовком) или тех или иных соглашений (например, при включении в заголовок дополнительной информации). Для обеспечения выравнивания по краю страницы информация заголовка, предшествующего данных, предназначенным получателю, должна быть сведена к фиксированному размеру — это нормально для канального уровня сети. С учетом всей (возможной) информации переменного размера в заголовке ее размещение после сегмента данных передающим хостом, позволит улучшить ситуацию и позволит принимающему хосту возможность выровнять данные по границе страницы. Это перемещение данных на канальном уровне для размещения заголовков переменного размера «в хвосте» было названо трейлерной инкапсуляцией.

Приведенный выше аргумент содержит несколько неявных допущений.

1. Принимающий хост должен выразить согласие принимать трейлерную инкапсуляцию. Поскольку это инкапсуляция канального уровня, до согласования между хостами (предпочтительно на канальном уровне для сохранения структуры уровней) лишь некоторые хосты смогут обмениваться данными или коммуникации могут быть существенно нарушены за счет смешивания пакетов с обычной и трейлерной инкапсуляцией.
2. Издержки на прием данных с выравниванием по границе страниц должны быть совместимы с приемом без такого выравнивания. При слишком больших издержках на выравнивание экономия за счет копирования может сойти на нет.
3. Размер данных в заголовке переменной длины должен быть существенно меньше размера передаваемого сегмента данных. Можно переместить информацию трейлера без ее физического копирования, но зачастую ограничения реализации и характеристики базового сетевого оборудования исключают простое переотображение заголовков.
4. Издержки на копирование в памяти, которые предполагаются на принимающей стороне, должны быть достаточно велики, чтобы расходы, связанные с усложнением программ на приемной и передающей стороне не превысили их.

Первое допущение известно достаточно хорошо и послужило мотивом для этого документа.

Была идея согласования применения трейлеров на уровне хоста с использованием протокола преобразования адресов ARP[1] [2] (фактически дополняющего протокол), но в настоящее время все системы, использующие трейлеры, требуют от хостов разделяемой сетевой среды всегда воспринимать трейлеры или никогда их не передавать (последнее легко выполнимо во время загрузки 4.2BSD без изменения исходных кодов операционной системы).

Второй момент (на наш взгляд) не имеет существенного значения. Хотя хост может не получить преимуществ от выравнивания и размера трейлерных пакетов, препятствий они не создадут в любом случае.

По части третьего допущения предположим, что информация из трейлерного заголовка копируется без переотображения и рассмотрим в качестве примера издержки, связанные с заголовком в протоколах TCP/IP [3]. Если мы предположим, что заголовки протоколов TCP и IP являются частью данных заголовка переменного размера, минимальный трейлерный пакет (генерируемый системой VAX) будет иметь 512 байтов данных и более 40 байтов заголовков (в дополнение к трейлеру, описанному ниже). Хотя трейлерный заголовок может включать опции IP и/или TCP, это будет встречаться достаточно редко (можно ожидать, например, что большинство опции TCP будет включаться в начальный обмен соединения) и заголовок будет много меньше 512 байтов. Если сегмент данных больше, отношение уменьшается и ожидаемый эффект от снижения издержек копирования на приемной стороне будет расти. С учетом относительных издержек на операции копирования и манипуляции с отображением страниц памяти (включая аннулирование буфера трансляции) преимущество становится очевидным.

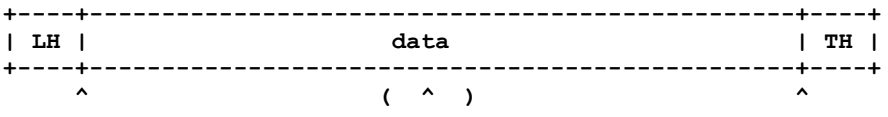
Четвертый вопрос на наш взгляд не является проблемой. В нашей реализации дополнительный код для поддержки трейлерной инкапсуляции составил около дюжины строк в каждом драйвере интерфейса канального уровня. Суммарное повышение производительности многократно превышает небольшие вложения в изменение программ.

Следует признать, что изменение формата пакетов на сетевом (и обычном канальном) уровне описываемым здесь способом заставляет принимающий хост буферизовать пакет целиком перед его обработкой. Умные реализации могут анализировать протокольные заголовки при получении пакета для определения реального размера (или типа пакета сетевого уровня) входящего сообщения. Это позволяет таким реализациям избежать выделения буферов максимального размера для входящих пакетов, которые потом могут быть признаны непригодными. Реализации, анализирующие формат сетевого уровня «на лету», нарушают принципы разделения по уровням, которые в течение некоторого времени провозглашаются для проектирования (но часто нарушаются в реализациях). Проблема отложенного распознавания типа канального уровня является разумным критицизмом. Однако в случае сетевого оборудования с поддержкой DMA[2] пакет всегда принимается целиком до начала его обработки.

## *Форматы пакетов с трейлерной инкапсуляцией*

В этом параграфе описывается формат канального уровня для пакетов, используемых в сетях 3 Mb/s Experimental Ethernet и 10 Mb/s Ethernet, а также кольцевых сетях 10 Mb/s V2LNI. Используемый в каждом из этих случаев формат отличается лишь значением поля типа (type) в заголовках локальной сети.

Формат трейлерного пакета показан на рисунке.




***LH***

Заголовок локальной сети (фиксированного размера). Для 10 Mb/s Ethernet это 16-байтовый заголовок Ethernet. Поле type в заголовке field показывает тип пакета (трейлер) и размер сегмента данных.

Для 10 Mb/s Ethernet поле type может принимать значения от 1001 до 1010 в шестнадцатеричном формате (от 4096 до 4112 в десятичном). Значение type рассчитывается как сумма шестнадцатеричного числа 1000 и числа 512-байтовых страниц данных. В одном трейлерном пакете может передаваться до 16 страниц данных (8192 байтов).

***data***

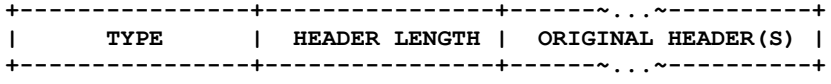
Данные пакета. Обычно это только данные, доставляемые принимающему процессу (т. е. они не содержат информации TCP и IP). Размер данных всегда кратен 512 байтам.

***TH***

Трейлер, представляющий собой комбинацию исходных протокольных заголовков и префикс трейлера с фиксированным размером, определяющий тип и размер данных. Формат трейлера показан ниже.

Символ ^ указывает границы страниц, на которых принимающий хост будет размещать свой входной буфер для оптимального выравнивания при получении трейлерного пакета. Функция приема канального уровня способна разместить трейлер, используя размер, указанный полем type в заголовке канального уровня. Предполагается, что эта функция отбросит заголовок канального уровня и префикс трейлера, а также отобразит сегмент данных на начало пакета для восстановления формата исходного заголовка сетевого уровня.

## *Формат трейлера*




***Type - 16 битов***

Поле type представляет исходный тип канального уровня для передаваемого пакета. Это значение обычно помещается в заголовке канального уровня, если трейлер не создается.

***Header length - 16 битов***

Поле header length трейлерного сегмента данных, указывающее размер последующих данных заголовка в байтах.

***Original headers - <переменный размер>***

Информация заголовка, которая логически размещается перед сегментом данных. Обычно это протокольные заголовки сетевого и транспортного уровня.

[1]Address Resolution Protocol.

[2]Direct memory access — прямой доступ к памяти. *Прим. перев.*

## *Заключение*

Описана инкапсуляция канального уровня, которая обеспечивает выравнивание, требуемое для эффективного использования виртуальной памяти. Этот формат инкапсуляции применяется во многих системах и является стандартным для 4.2BSD UNIX. Инкапсуляция обеспечивает эффективный механизм, с помощью которого взаимодействующие хосты локальной сети могут получить существенный рост производительности. Использование этой инкапсуляции в настоящее требует однородного взаимодействия всех хостов сети и мы надеемся на возможность добавления на каждом хосте механизмов согласования, которые позволят хостам согласовать использование инкапсуляции в неоднородной среде.

## *Литература*

[1] "The Ethernet - A Local Area Network"[1], Version 1.0, Digital Equipment Corporation, Intel Corporation, Xerox Corporation, September 1980.

[2] Plummer, David C., "An Ethernet Address Resolution Protocol", RFC-826, Symbolics Cambridge Research Center, November 1982.

[3] Postel, J., "Internet Protocol", RFC-791, USC/Information Sciences Institute, September 1981.

**Перевод на русский язык**

Николай Малых

nmalykh@gmail.com

[1]Доступна по ссылке http://decnet.ipv7.net/docs/dundas/aa-k759b-tk.pdf. *Прим. перев.*