Network Working Group R. Woodburn
Request for Comments: 1241 SAIC
D. Mills
University of Delaware
July 1991

Схема для протокола инкапсуляции Internet (версия 1)

A Scheme for an Internet Encapsulation Protocol:

Version 1

1. Статус документа

Этот документ определяет экспериментальный протокол для сообщества Internet и служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации и статус протокола можно узнать из документа IAB Official Protocol Standards. Документ может распространяться без ограничений.

2. Глоссарий

Clear Datagram - исходная (чистая) дейтаграмма
Неизмененная дейтаграмма IP в Пользовательском Пространстве (User Space) до Инкапсуляции.

Clear Header - исходный заголовок
Заголовочная часть Исходной Дейтаграммы (Clear Datagram) до Инкапсуляции. Этот заголовок включает заголовок IP, а также может включать часть или все заголовки протокола следующего уровня (например, заголовок TCP).

Decapsulation - декапсуляция
Отбрасывание Заголовка Инкапсуляции (Encapsulation Header) и пересылка чистой дейтаграммы Декапсулятором.

Decapsulator - декапсулятор
Элемент (объект), отвечающий за получение Инкапсулированной Дейтаграммы, ее Декапсуляцию и доставку в пользовательское пространство получателя. Доставка может быть прямой или с использованием инкапсуляции. Декапсулятор может быть хостом или шлюзом (маршрутизатором).

Encapsulated Datagram - инкапсулированная дейтаграмма
Дейтаграмма, содержащая Исходную Дейтаграмму (Clear Datagram) и добавленный перед ней Заголовок Инкапсуляции.

Encapsulation - инкапсуляция
Процесс отображения Исходной Дейтаграммы в Пространство Инкапсуляции (Encapsulation Space) добавление перед дейтаграммой Заголовка Инкапсуляции и маршрутизация инкапсулированной дейтаграммы Декапсулятору.

Encapsulation Header - заголовок инкапсуляции
Заголовок для Протокола Инкапсуляции, помещаемый перед Исходной Дейтаграммой в процессе Инкапсуляции. Этот заголовок включает заголовок IP, за которым следует Заголовок Протокола Инкапсуляции .

Encapsulation Protocol Header - заголовок протокола инкапсуляции
Определяемая Протоколом Инкапсуляции часть Заголовка Инкапсуляции.

Encapsulation Space - пространство инкапсуляции
Адрес и пространство маршрутизации, в котором размещаются Инкапсуляторы и Декапсуляторы. Маршрутизация в этом пространстве осуществляется через Потоки (Flow). Пространства Инкапсуляции не перекрываются, т. е., адреса любого Инкапсулятора и Декапсулятора уникальны для всех Пространств Инкапсуляции.

Encapsulator - инкапсулятор
Объект, отвечающий за отображение данной дейтаграммы из Пользовательского Пространства в Пространство Инкапсуляции, инкапсуляцию дейтаграммы и пересылку Инкапсулированной Дейтаграммы Декапсулятору. Инкапсулятор может быть хостом или шлюзом (маршрутизатором).

Flow - поток
Поток или туннель (tunnel) представляет собой сквозной (end-to-end) путь в Пространстве Инкапсуляции, по которому передаются Инкапсулированные Дейтаграммы. Вдоль одного туннеля (потока) может размещаться несколько пар Инкапсулятор-Декапсулятор. Отметим, что поток не обязан включать шлюзы (маршрутизаторы) Пользовательского Пространства.

Flow ID - идентификатор потока
32-битовое значение, позволяющее уникально идентифицировать поток (туннель) в данном Инкапсуляторе/Декапсуляторе. Идентификаторы потоков специфичны для Инкапсулятора/Декапсулятора и не являются глобальными.

Mapping Function - функция отображения
Это функция, отображающая Исходную Дейтаграмму на конкретный Поток (туннель). Все Инкапсуляторы одного Потока должны отображать данную Исходную Дейтаграмму на один Поток.

User Address - адрес пользователя
Адрес или идентификатор, позволяющий уникально определить объект в Пользовательском Пространстве.

Source Route - маршрут, заданный отправителем
Полный сквозной маршрут, рассчитанный на стороне отправителя и включающий все транзитные маршрутизаторы.

User Space - пользовательское пространство

Пространство адресов и маршрутизации, в котором размещаются пользователи. Маршрутизация в рамках этого пространства обеспечивает доступность любого из адресов в данном пространстве. Пользовательские Пространства не пересекаются, т. е., данный Адрес Пользователя является уникальным для всех Пользовательских Пространств.

3. Предпосылки

В течении нескольких лет исследователи из сообщества Internet показали необходимость «туннелирования» между сетями. Туннель, по сути, представляет собой задаваемый отправителем маршрут (Source Route) в обход обычных механизмов маршрутизации. Туннели обеспечивают способ обхода нарушений (отказов) в маршрутизации, неработающих шлюзов и доменов маршрутизации или организовать предопределенные пути для экспериментов.

Существует несколько способов организации туннелирования. В прошлом для туннелей использовались опции маршрутизации отправителем (source routing) в заголовке IP, которые позволяют перечислить все маршрутизаторы на данном пути. Недостатком такого подхода является то, что отправителю необходимо иметь информацию о сетях на пути к получателю. Отправитель должен изменять исходящие пакеты с учетом задаваемого им маршрута. Современные реализации маршрутизаторов в общем случае не поддерживают задаваемые отправителями маршруты в своих таблицах маршрутизации, не учитывают таких маршрутов и современные протоколы маршрутизации.

Другим способом организации туннелирования будет разработка новой опции IP. Поле этой опции будет частью отдельного заголовка IP, который будет добавляться перед дейтаграммой IP. Опция IP будет содержать информацию об исходной дейтаграмме. Такая опция туннелирования обладает существенным недостатком в связи с необходимостью вносить изменения в существующие реализации IP для обработки новой опции IP. Она также не обеспечит достаточной гибкости при туннелировании других протоколов (например, ISO) через среды IP. Еще менее подходящим вариантом была бы замена IP новым сетевым протоколом или разработка новой версии IP со встроенными функциями туннелирования.

Рисунок 1 Модель архитектуры инкапсуляции

Другим вариантом является разработка нового протокола инкапсуляции IP, использующего текущий формат заголовков IP. При использовании инкапсуляции получателя можно задать без информации о топологии сети на стороне отправителя. Виртуальные сети можно создавать путем связывания не соединенных непосредственно машин через потоки (туннели) в пространстве инкапсуляции.

До настоящего времени не существовало стандарта для протокола инкапсуляции. Этот документ предлагает механизм инкапсуляции в среде Internet.

4. Архитектура и модель

Архитектура инкапсуляции основана на двух базовых элементах - Инкапсуляторах и Декапсуляторах. Эти элементы и связанные с ними пространства показаны на рисунке 1.

Инкапсуляторы и декапсуляторы имеют адреса с Пользовательских Пространствах, к которым они относятся, а также адреса в Пространствах Инкапсуляции, к которым они принадлежат. Инкапсулятор будет получать Исходную Дейтаграмму (Clear Datagram) из своего Пользовательского Пространства и, после определения необходимости использования инкапсуляции, выполнять функцию отображения, которая транслирует информацию Пользовательского Пространства из Исходного Заголовка в Заголовок Инкапсуляции. Созданный Заголовок Инкапсуляции помещается перед Исходной Дейтаграммой для формирования Инкапсулированной Дейтаграммы, как показано на рисунке 2. Желательно обеспечить прозрачность процесса инкапсуляции для объектов Пользовательского Пространства. Знать об инкапсуляции должны только Инкапсуляторы.

Инкапсулятор пересылает дейтаграмму Декапсулятору, который был идентифицирован во время инкапсуляции. Декапсулятор получает Инкапсулированную Дейтаграмму, удаляет Заголовок Инкапсуляции и трактует Исходную Дейтаграмму, как будто она

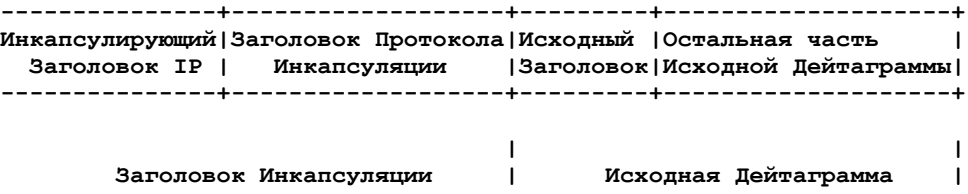

Рисунок 2 Пример Инкапсулированной Дейтаграммы

была получена от локального отправителя. К адресу Декапсулятора предъявляется требование его достижимости из Пространства Инкапсуляции на стороне Инкапсулятора.

5. Генерация Заголовка Инкапсуляции

Содержимое Заголовка Инкапсуляции генерируется путем отображения Исходного Заголовка. Функция отображения может быть реализована по разному, но результат должен быть один и тот же. Процесс генерации показан на рисунке 3.

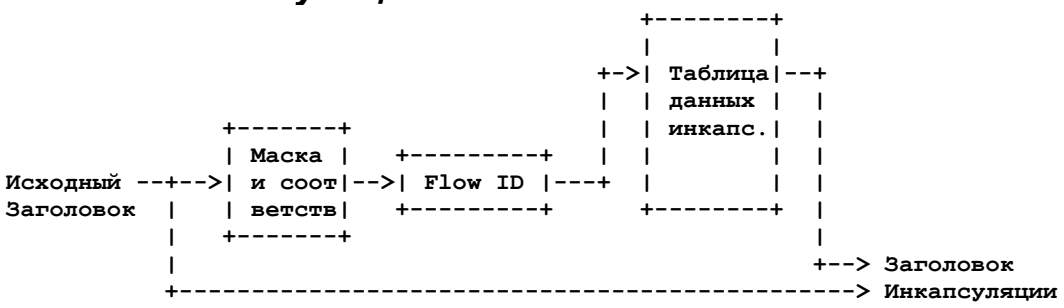

Рисунок 3 Генерация Заголовка Инкапсуляции

На первом этапе отображения функция сравнивает Исходный Заголовок с сохраненными заголовками и масками для определения идентификатора потока Flow ID. По сути, этот этап заключается в просмотре таблицы «масок и соответствий» (mask-and-match), содержащей записи из трех элементов - Исходный Заголовок, маска заголовка и соответствующий идентификатор туннеля (Flow ID). Маска может использоваться для определения диапазонов адресов отправителей и получателей, чьи пакеты будут отображаться в данный поток (туннель). Для разделения потоков могут использоваться и другие поля типа битов IP TOS или номеров портов TCP для отправителя и получателя. Такая гибкость обеспечивает многочисленные возможности применения функции отображения. С туннелем можно связать не только заданную сеть, но конкретный протокол TCP или соединение.

Устройство таблицы и методы поиска в ней не задаются данным протоколом. Предполагается, что работа с таблицей поддерживается на вышележащем уровне. При необходимости таблицы можно просто хранить в текстовых файлах ASCII.

Идентификаторы туннелей (Flow ID) также обрабатываются на вышележащем уровне. Пример обработки Flow ID можно найти в протоколе Setup протокола IDPR[1] [4]. Протокол вышележащего уровня будет отвечать за поддержку информации, которая не передается в протоколе инкапсуляции, связанном с туннелем. Эта информация может включать данные, требуемые для создания Заголовка Инкапсуляции (см. ниже), а также сведения о типе инкапсулируемых данных (в настоящее время только IP) и типе используемой аутентификации. Отметим, что протокол IDPR Setup требует использования более длинных идентификаторов Flow ID для обеспечения уникальности в масштабе всего множества Инкапсуляторов.

Значение Flow ID, получаемое в результате отображения Clear Header, представляет собой 32-битовое число, которое идентифицирует туннель (Flow) с точки зрения Инкапсулятора. Если Исходную Дейтаграмму в процессе доставки адресату требуется инкапсулировать и декапсулировать неоднократно, значения Flow ID могут различаться на каждом Инкапсуляторе, но это не обязательно. Значение Flow ID служит индексом для таблицы с информацией о Заголовках Инкапсуляции, которая используется для генерации этих заголовков. Отметим, что решение о локальной значимости значений Flow ID для каждого Инкапсулятора было принято с учетом сложностей, возникающих при выборе и поддержке идентификаторов глобальной значимости.

Промежуточный этап использования Flow ID является сугубо опциональным. Важным требованием является использование всеми Инкапсуляторами на пути доставки одного Исходного Заголовка (Clear Header) в рамках одного туннеля Flow (сам туннель вдоль пути может иметь разные идентификаторы). Тем не менее, учет значения Flow ID в протоколе позволяет более эффективно реализовать функцию отображения. Этот вопрос более подробно рассматривается ниже при описании Декапсулятора.

Для создания Заголовка Инкапсуляции требуется следующая информация:

Flow ID - идентификатор туннеля
: Это ключ к таблице информации, представляющий туннель в рамках текущего Инкапсулятора.

Decapsulator Address - адрес получателя
: IP-адрес Декапсулятора и Пространстве Инкапсуляции, требуемый для построения IP-части Заголовка Инкапсуляции.

Decapsulator's Flow ID - идентификатор потока на стороне декапсулятора
: Значение Flow ID (при его наличии) для потока Flow с точки зрения Декапсулятора.

Previous Encapsulator's Address - адрес предыдущего инкапсулятора
: Если данный Инкапсулятор не является первым в туннеле Flow, адрес предыдущего Инкапсулятора требуется для передачи сообщений об ошибках.

Previous Encapsulator's Flow ID - идентификатор туннеля на предыдущем инкапсуляторе
: В дополнение к адресу предыдущего Инкапсулятора должно быть известно значение Flow ID для туннеля Flow относительно предыдущего Инкапсулятора.

Заголовок Инкапсуляции состоит из заголовка IP (IP Header) и заголовка протокола инкапсуляции (Encapsulation Protocol Header). Во время инкапсуляции должны быть определены две части Заголовка Протокола Инкапсуляции — инкапсулируемый протокол и значение Flow ID для передачи Декапсулятору. Генерация заголовка IP более сложна.

Для каждого поля Исходного Заголовка возможны две операции при создании нового заголовка IP:

Копировать
: Существующее поле Исходного Заголовка копируется в заголовок IP Заголовка Инкапсуляции.

Игнорировать
: Это поле может присутствовать в Исходном Заголовке, но оно не применимо для нового заголовка IP.

[1] Inter-Domain Policy Sensitive Routing Protocol - протокол междоменной маршрутизации с учетом политики.

Заголовок IP имеет фиксированную и переменную (список опций) часть. Список всех полей заголовка IP и связь их с полями Исходного Заголовка приведены в таблице 1 [2].

Таблица 1 Отображение полей заголовка IP

Поле	Операция
Version	Игнорировать
Header Length	Игнорировать
Precedence	Копировать
Биты QoS	Копировать
Total Length	Игнорировать
Identification	Игнорировать
Флаг Don't Fragment	Игнорировать
Флаг More Fragments	Игнорировать
Fragment Offset	Игнорировать
Time to Live	Игнорировать
Protocol	Игнорировать
Header Checksum	Игнорировать
Source Address	Игнорировать
Destination Address	Игнорировать
End of Option List	Игнорировать
Опция NOP	Игнорировать
Опция Security	Копировать
Опция LSR	Игнорировать
Опция SSR	Игнорировать
Опция RR	Игнорировать
Опция Stream ID	Игнорировать
Опция Timestamp	Игнорировать

Отметим, что большинство полей Исходного Заголовка просто игнорируется. Например, поле Header Length в Исходном Заголовке не определяет значение поля Header Length в новом заголовке IP. Далее будут рассматриваться поля, которые более интересны и могут копироваться в новый заголовок.

Биты качества обслуживания (QoS[1]) следует копировать из Исходного Заголовка в новый заголовок IP. Это делается из соображений прозрачности, чтобы услуги, предоставляемые в Пользовательском Пространстве, обеспечивались и в Пространстве Инкапсуляции.

Поля More Fragments и Fragment Offset не следует копировать, поскольку дейтаграмма с инкапсуляцией будет строиться, как полная (не фрагмент), независимо от состояния инкапсулируемой дейтаграммы. Если полученная в результате дейтаграмма окажется слишком большой, на пути к декапсулятору будет использоваться обычный механизм фрагментации IP.

Флаг Don't Fragment не следует копировать в Заголовок Инкапсуляции. Здесь снова будет нарушаться принцип прозрачности. Решение вопроса о запрете фрагментации в Пространстве Инкапсуляции следует предоставить Инкапсулятору. Если Инкапсулятор решит установить флаг DF, при возникновении необходимости фрагментации Инкапсулированной Дейтаграммы в Пространстве Инкапсуляции ему должно быть возвращено сообщение ICMP. Однако в этом случае будет меняться механизм возврата сообщения ICMP отправителю в Пользовательское Пространство, как это описано в Приложении B.

Относительно поля времени жизни (TTL) самым простым решением будет игнорирование значения этого поля в Исходном Заголовке. Если это поле копировать из Исходного Заголовка в новый заголовок IP, жизнь пакета может завершиться преждевременно в процессе передачи через Пространство Инкапсуляции. Это нарушит принцип прозрачности инкапсуляции с точки зрения Пользовательского Пространства. Значение TTL из Исходного Заголовка будет уменьшено перед инкапсуляцией функцией пересылки IP, следовательно не возникает предпосылок зацикливания пакетов в случае замыкания туннеля Flow в петлю.

В поле протокола для нового заголовка IP следует указывать номер протокола инкапсуляции.

В качестве адреса отправителя в новом заголовке IP используется IP-адрес Инкапсулятора в Домене Инкапсуляции. В качестве адреса получателя указывается IP-адрес Декапсулятора из таблицы инкапсуляции.

Опции IP в общем случае не копируются, поскольку они не имеют смысла в контексте Пространства Инкапсуляции. Опция безопасности (Security) является, возможно, единственным исключением из этого правила и ее следует копировать по тем же причинам, которые требуют копирования полей QOS и Precedence (Пространство Инкапсуляции должно обеспечивать ожидаемый сервис). Опции Timestamp, Loose Source Route, Strict Source Route и Record Route не копируются при инкапсуляции.

6. Декапсуляция

В идеальном варианте Декапсулятор принимает Инкапсулированные Дейтаграммы, вырезает Заголовок Инкапсуляции и передает Исходные Дейтаграммы обратно в IP, как будто он просто пересылает их. Однако, если Исходная Дейтаграмма не дошла до Пользовательского Пространства получателя, ее требуется инкапсулировать снова для переноса ближе к Пользовательскому Пространству получателя. В этом случае Декапсулятор становится также Инкапсулятором и будет выполнять те же операции по созданию Заголовка Инкапсуляции, которые были описаны ранее. Для эффективного выполнения этого процесса в протокол было включено использование идентификаторов туннелей Flow ID.

При использовании идентификаторов туннелей значение Flow ID, полученное в Заголовке Инкапсуляции, соответствует значению Flow ID, сохраненному Декапсулятором. В этом случае Декапсулятор может пропустить операции маскирования и соответствия для Исходного Заголовка. Принятое значение Flow ID можно непосредственно включать в локальные таблицы Инкапсулятора для создания следующего Заголовка Инкапсуляции. Если значение Flow ID неизвестно, предыдущему Инкапсулятору возвращается сообщение об ошибке для передачи информации вышележащему уровню, управляющему таблицами инкапсуляции.

Поскольку при использовании Flow ID нормальный механизм пересылки IP обходится, некоторые операции, обычно выполняемые IP, должны быть реализованы Декапсулятором перед инкапсуляцией. Декапсулятор должен уменьшить значение поля TTL перед выполнением следующей инкапсуляции. При возникновении ошибки Time Exceeded[2] отправителю, указанному в Исходном Заголовке должно быть передано сообщение ICMP.

7. Сообщения об ошибках

В протокол инкапсуляции включено два типа сообщений об ошибках. Первый тип используется для передачи Декапсулятором информации о неизвестных идентификаторах потоков, второй служит для пересылки сообщений ICMP.

[1]Quality of Service.
[2]Время жизни истекло.

Когда Декапсулятор использует полученное значение Flow ID в Заголовке Инкапсуляции для пересылки дейтаграммы следующему Декапсулятору в туннеле Flow, значение Flow ID может оказаться неизвестным. В таких случаях Декапсулятор будет уведомлять предыдущий Инкапсулятор о том, что туннель ему не известен, чтобы информация об этом могла быть переправлена уровню, отвечающему за таблицы туннелей. Для этого используются сообщения об ошибках инкапсуляции.

Если Инкапсулятор получает сообщение ICMP, относящееся к данному туннелю, это сообщение следует передать в направлении исходного Инкапсулятора. Для этих целей используется второй тип сообщений об ошибках. Сообщение ICMP будет включать значение Flow ID сообщения, вызвавшего ошибку. Это значение Flow ID должно быть оттранслировано в идентификатор туннеля (Flow ID) на Инкапсуляторе, которому передается сообщение об ошибке.

Если ошибка возникает при передаче сообщения об ошибке, дополнительных сообщений об ошибках не создается.

8. Литература

[1] J. Postel, Internet Control Message Protocol, RFC 792[1], September 1981.

[2] J. Postel, Internet Protocol, RFC 791[1], September 1981.

[3] J. Postel, Transmission Control Protocol, RFC 793[1], September 1981.

[4] ORWG, Inter-Domain Policy Routing Protocol Specification and Usage, Draft[2], August 1990

A. Формат пакетов

В этом приложении описан формат пакетов протокола инкапсуляции.

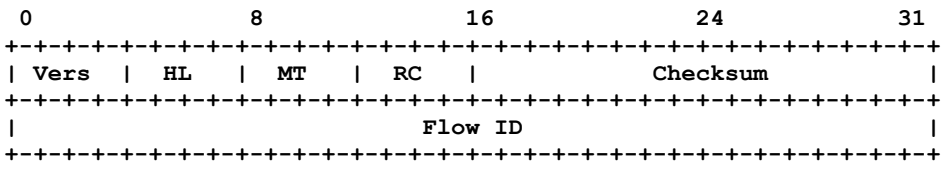

Рисунок А.1. Пример Заголовка Инкапсуляции

Vers 4 бита
Номер версии протокола инкапсуляции. Для описываемого в данном документе протокола используется значение 1.

HL 4 бита
Размер Заголовка Протокола Инкапсуляции (Encapsulation Protocol Header) в октетах.

MT 4 бита
Тип сообщения Протокола Инкапсуляции. Для сообщений с данными (Data Message) указывается тип 1, для сообщений об ошибках - тип 2.

RC 4 бита
Код причины. Это поле не используется в сообщениях с данными Data Message и должно иметь значение 0. В сообщениях об ошибках код указывает причину генерации сообщения:

1 Unknown Flow ID - неизвестный идентификатор туннеля;

2 ICMP returned - возврат сообщения ICMP.

Checksum 16 битов
Контрольная сумма (дополнение до 1) для Заголовка Протокола Инкапсуляции. При расчете для поля устанавливается значение 0, а рассчитанное значение помещается в это поле до передачи сообщения.

Flow ID 32 бита
Идентификатор туннеля Flow ID с точки зрения Декапсулятора или Инкапсулятора, которому будет отправлено сообщение. При передаче сообщений об ошибках Unknown Flow ID используется значение идентификатора, вызвавшего ошибку.

Для сообщений с данными после Заголовка Протокола Инкапсуляции следует Исходная дейтаграмма (Clear Datagram). Для сообщений об ошибках после заголовка следует сообщение ICMP, передаваемое по туннелю.

B. Инкапсуляция и существующие механизмы IP

В этом разделе подробно рассматривается влияние протокола инкапсуляции на существующие механизмы, доступные в IP, и некоторые возможные воздействия механизмов IP на данный протокол. В частности, рассмотрена фрагментация и сообщения ICMP.

B.1 Фрагментация и MTU

Прямым следствием использования механизмов инкапсуляции являются ограничения, связанные с размером MTU. Отправитель Исходных дейтаграмм генерирует пакеты в соответствии со значением MTU для интерфейса, через который дейтаграмма передается. Если такие пакеты пришли на Инкапсулятор и были инкапсулированы, они могут быть фрагментированы в случаях, когда размер пакета превышает значение MTU для Инкапсулятора, несмотря на то, что физические интерфейсы источника и Инкапсулятора могут иметь одинаковое значение MTU. Поскольку Инкапсулированная Дейтаграмма передается декапсулятору с использованием IP, не возникает проблем с разрешением протоколу IP выполнять фрагментацию и сборку. Однако фрагментация снижает эффективность и в общем случае ее стараются предотвратить. Поскольку к Исходной Дейтаграмме процесс инкапсуляции добавляет новый заголовок, вероятность фрагментирования возрастает. Если Инкапсулятор принимает решение о запрете фрагментирования дейтаграмм в Пространстве Инкапсуляции, он должен передать сообщение ICMP источнику дейтаграмм. Это означает, что значение MTU на интерфейсе в пространство инкапсуляции эффективно меньше значения MTU на физическом интерфейсе.

Фрагментация на промежуточных шлюзах (маршрутизаторах) Пользовательского Пространства создает другую проблему. Фрагментация происходит на уровне IP. Если используется протокол TCP и возникает фрагментация, заголовок TCP будет присутствовать только в первом фрагменте [3]. Если эти фрагменты пересылаются Инкапсулятором, распознавание Исходного Заголовка (Clear Header) для данного туннеля будет возможно только для

[1]Перевод этого документа доступен на сайте www.protocols.ru. *Прим. перев.*
[2]Работа была завершена и опубликована в RFC 1477, RFC 1478 и RFC 1479. *Прим. перев.*

IP-части. При попыке распознать TCP-часть заголовка, соответствие будет найдено только для первого фрагмента, а остальные фрагменты не будут отнесены к данному туннелю.

B.2 Сообщения ICMP

Наиболее спорным аспектом инкапсуляции является обработка сообщений ICMP [1]. Поскольку Заголовок Инкапсуляции содержит адрес отправителя (Инкапсулятор) в Пространстве Инкапсуляции, сообщения ICMP, появившиеся в Пространстве Инкапсуляции, будут передаваться Инкапсулятору. При получении Инкапсулятором сообщения ICMP он должен принять решение о дальнейших действиях с этим сообщением. Поскольку отправитель Исходной Дейтаграммы ничего не знает о Пространстве Инкапсуляции, нет смысла пересылать ему полученное сообщение ICMP, а сообщения ICMP не должны вызывать генерацию новых сообщений ICMP. Если же не передавать некоторую информацию источнику исходных дейтаграмм, работа ряда важных механизмов может нарушиться.

В дополнение к решению вопроса о пересылке сообщений отправителю Исходной Дейтаграммы возникает проблема отсутствия информации, требуемой для передачи сообщения исходному отправителю. Сообщение ICMP возвращает заголовок вызвавшего ошибку пакета и первые восемь октетов данных после заголовка. При использовании протокола инкапсуляции эта информация транслируется в IP-часть Заголовка Инкапсуляции и первые восемь октетов Заголовка Протокола Инкапсуляции. Содержимое Исходной Дейтаграммы полностью теряется. Следовательно Инкапсулятор для пересылки сообщения ICMP исходному отправителю должен реконструировать Исходный Заголовок. Однако на практике точное восстановление этого заголовка невозможно.

В данной спецификации идентификатор туннеля Flow ID рассматривается, как уникальный способ одностороннего отображения Исходного Заголовка. Нет гарантии возможности использования Flow ID для обратного отображения на Исходный Заголовок, поскольку на один поток может быть отображено множество разных заголовков. Поскольку эффективного способа восстановления исходной дейтаграммы не существует, требуется проверить несколько компромиссных вариантов.

Для каждого из возможных сообщений ICMP будут оценены варианты и последствия. Рассматриваются три категории сообщений ICMP. К первой категории отнесены сообщения ICMP, не применимые в контексте Инкапсуляции (Echo/Echo Reply и Timestamp/Timestamp Reply).

Вторая категория включает сообщения ICMP, относящиеся к локальным механизмам области инкапсуляции. Есть сообщения, которые не будут иметь никакого смысла для исходного отправителя, даже если он их получит. В таких случаях инкапсулятор принимает решение самостоятельно, но передавать какие-либо сообщения ICMP исходному отправителю не требуется. Дейтаграмма просто теряется - протокол IP не гарантирует доставки. Последующие сообщения, полученные для инкапсуляции, могут заставить инкапсулятор генерировать сообщения ICMP Destination Unreachable для исходного отправителя, если инкапсулятор больше не может передавать сообщения декапсулятору адресата. Для этого нужно, чтобы сообщения ICMP в области инкапсуляции оказывали влияние на отображение из Flow ID. К сообщениям ICMP второй категории относятся: Parameter Problem, Redirect, Destination Unreachable, Time Exceeded.

К третьей категории относится сообщение ICMP, которое оказывает непосредственное воздействие на работу исходного отправителя, - ICMP Source Quench. Для Инкапсулятора единственным возможным механизмом обработки таких сообщений является установка для соответствующего Flow ID такого флага, который потребует генерации сообщений ICMP Source Quench исходному отправителю до инкапсуляции его дейтаграмм.

Этот механизм может послужить решением более общей проблемы. Правило заключается в том, что при получении сообщения ICMP для данного туннеля, он помечается таким образом, инкапсуляция следующего пакета приводит к тому, что инкапсуляция еще одного пакета в данный туннель будет приводить к генерации сообщения ICMP для исходного отправителя. После отправки сообщения ICMP источнику механизм может быть сброшен. Это будет приводить к получению сообщения ICMP для каждого второго пакета при условии, что проблема сохраняется. Этот механизм может оказаться единственным безопасным способом доставки сообщений Unreachable и Source Quench.

C. Прием Исходных Дейтаграмм

Для использования протокола инкапсуляции нужно изменить пересылку IP. Изменения включают обеспечение модулю IP в системе способа передачи Исходных дейтаграмм протоколу инкапсуляции. Предлагаемый способ заключается в добавлении к структурам маршрутизации системы флага для маршрутов, который будет говорить функции пересылки о необходимости инкапсуляции. Отметим, что для инкапсуляции может использоваться свой маршрут по умолчанию.

При использовании этого метода системный механизм пересылки IP будет просматривать свои таблицы маршрутизации на предмет соответствия адресата IP конкретному маршруту. Если маршрут найден, механизм пересылки будет проверять необходимость инкапсуляции пакетов для него. Если инкапсуляция не нужна, пакет будет обрабатываться обычным способом. Если же для маршрута задана инкапсуляция, дейтаграмма будет передана на инкапсуляцию для пересылки.

На последнем Декапсуляторе пакета требуется способ корректной передачи декапсулированных дейтаграмм модулю IP для доставки. Поскольку пакеты инкапсулируются непосредственно перед пересылкой, для декапсулированных дейтаграмм нужен простой метод вставки дейтаграм на выход IP. Однако адрес отправителя в Исходном Заголовке менять недопустимо. Адрес должен указывать на отправителя в Пользовательском Пространстве и не переписывается на адрес Декапсулятора.

D. Создание виртуальных сетей на базе инкапсуляции

По причине изменения таблицы маршрутизации для поддержки инкапсуляции пакетов можно задать виртуальный интерфейс, единственной задачей которого будет инкапсуляция. При использовании этого механизма можно связать топологически разнесенные объекты с помощью туннеля (Flow). Это позволяет создавать Виртуальные Сети (Virtual Network) которые будут работать «поверх» обычной топологии маршрутов. Пример такой виртуальной сети показан на рисунке 4.

Каждый Инкапсулятор имеет виртуальный интерфейс в одну из виртуальных сетей. Линии на рисунке представляют отдельные каналы в туннелях, соединяющие узлы виртуальной сети. Отметим, что новые каналы могут быть созданы между любой парой объектов, которые «видят» друг друга в общем Пространстве Инкапсуляции. Маршрутизация внутри виртуальной сети будет обеспечиваться механизмом инкапсуляции. Поддержка таблиц маршрутизации может обеспечиваться любым из имеющихся протоколов маршрутизации (например, инкапсулированным OSPF).

С учетом сказанного, можно создать специальные шлюзы инкапсуляции с виртуальными интерфейсами в две виртуальных сети для формирования виртуальной «сети сетей» (virtual internet). Эту роль на рисунке играют Инкапсуляторы, соединяющие виртуальные сети A и B.

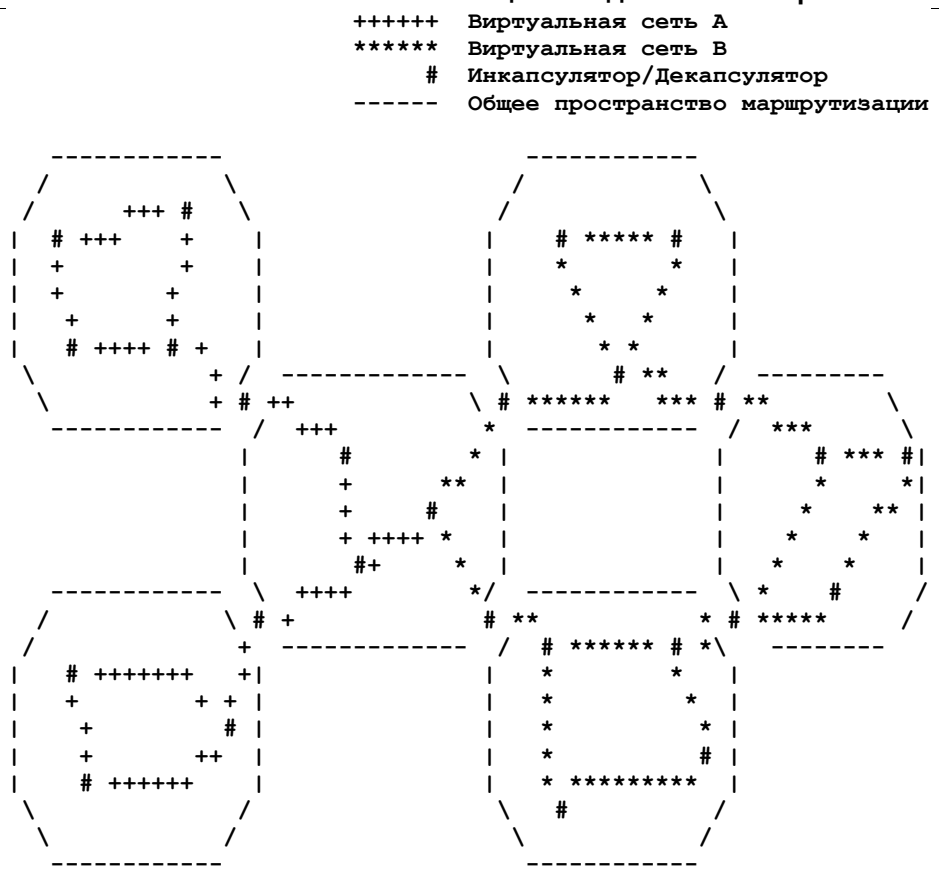

Рисунок 4 Пример виртуальных сетей

E. Инкапсуляция и OSI

Предполагается, что описанный в этом документе механизм инкапсуляции, может быть распространен и на другие среды, не относящиеся к Internet. Следует обеспечивать возможность инкапсуляции множества разных протоколов в IP и инкапсуляции IP во множество других протоколов.

Ключевой концепцией, определенной в этом документе, является отображение заголовка на идентификатор туннеля Flow ID и отображение полей исходного заголовка в поля заголовка инкапсуляции. Могут потребоваться специальные варианты отображения (например, для битов QoS), а некоторые типы трансляции потребуют особой осторожности, но в этом нет ничего невозможного.

F. Вопросы безопасности

Для этого протокола не определено никаких специальных средств аутентификации или контроля целостности сверх контрольных сумм в заголовке. Однако для аутентификации и контроля целостности в рамках данного протокола предлагается добавлять аутентификационные данные в конец Инкапсулированной Дейтаграммы. Информация о типе используемой аутентификации или контроля целостности будет включена в протокол управления потоком (туннелем) который используется для распространения информации о потоке.

G. Адреса авторов

Robert A. Woodburn

SAIC

8619 Westwood Center Drive

Vienna, VA 22182

Phone: (703) 734-9000 or (703) 448-0210

EMail: woody@cseic.saic.com

David L. Mills

Electrical Engineering Department

University of Delaware

Newark, DE 19716

Phone: (302) 451-8247

EMail: mills@udel.edu

Перевод на русский язык

Nikolai Malykh

nmalykh@gmail.com