Network Working Group
Request for Comments: 3032
Category: Standards Track

E. Rosen
D. Tappan
G. Fedorkow
Cisco Systems, Inc.
Y. Rekhter
Juniper Networks
D. Farinacci
T. Li
Procket Networks, Inc.
A. Conta
TranSwitch Corporation

January 2001

Представление стека меток MPLS

MPLS Label Stack Encoding

Статус документа

Данный документ содержит спецификацию протокола, предложенного сообществу Internet, и служит запросом к дискуссии в целях развития протокола. Информацию о статусе данного протокола можно найти в текущей редакции документа «Internet Official Protocol Standards» (STD 1). Документ может распространяться свободно.

Тезисы

Для многопротокольной коммутации по меткам (MPLS[1]) [1] требуется набор процедур добавления в пакеты сетевого уровня «стека меток», превращающего такие пакеты в «помеченные». Маршрутизаторы, поддерживающие MPLS, называют LSR[2]. Для передачи помеченных пакетов через определенный канал маршрутизатор LSR должен поддерживать метод кодирования меток, позволяющий из данного стека меток и пакета сетевого уровня создавать помеченный пакет. В этом документе описаны процедуры кодирования, используемые LSR для передачи пакетов в каналы PPP[3], ЛВС и, возможно, в каналы других типов. На некоторых типах каналов передачи данных верхняя метка стека[4] может кодироваться по-иному, но для остальной части стека меток **должны** использоваться методы кодирования, описанные в настоящем документе. В документе также приведены правила и процедуры обработки различных полей стека меток.

Оглавление

[1]Multi-Protocol Label Switching
[2]Label Switching Router
[3]Point-to-Point Protocol
[4]Иное кодирование может использоваться для двух верхних меток стека. *Прим. перев.*

1. Введение

Для многопротокольной коммутации по меткам (MPLS) [1] требуется набор процедур добавления в пакеты сетевого уровня «стека меток», превращающего такие пакеты в «помеченные». Маршрутизаторы, поддерживающие MPLS, называют LSR. Для передачи помеченных пакетов через определенный канал маршрутизатор LSR должен поддерживать метод кодирования меток, позволяющий из данного стека меток и пакета сетевого уровня создавать помеченный пакет.

В этом документе описаны процедуры кодирования, используемые LSR для передачи пакетов в каналы PPP и ЛВС. Описанные здесь методы кодирования могут быть полезны и для других типов каналов.

В документе также приведены правила и процедуры обработки различных полей стека меток. Поскольку MPLS не зависит от протокола сетевого уровня, основная часть таких процедур также независима от протокола. Однако часть процедур различается для разных протоколов. В этом документе рассматриваются протокольно-независимые процедуры и зависимые от протокола процедуры для протоколов IPv4 и IPv6.

LSR, реализованные в виде коммутаторов (например, коммутаторов ATM), могут использовать разные методы кодирования для одной или двух верхних меток и остальной части стека. При наличии в стеке дополнительных меток их кодирование **должно** осуществляться с использованием описанных в данном документе методов.

1.1. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с RFC 2119.

2. Стек меток

2.1. Представление стека меток

Стек меток кодируется, как последовательность элементов стека, каждый из которых представляется 4 октетами, как показано на рисунке 1.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Элемент
|                Label                  | Exp |S|       TTL     | стека
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ меток
```

Label - значение метки (20 битов)

Exp - для экспериментов (3 бита)

S - дно стека (1 бит)

TTL - время жизни (8 битов)

Рисунок 1: Элемент стека меток

Элементы стека меток размещаются в заголовке **после** заголовка канального уровня, но **перед** заголовком сетевого уровня. Вершина стека размещается ближе к началу пакета, а дно стека - дальше. Заголовок сетевого уровня следует непосредственно за элементом стека меток, в котором установлен флаг S.

Каждый элемент стека состоит из 4 полей:

1. *Дно стека* (S)

   Этот бит устанавливается для последней записи в стеке меток (дно стека) и имеет нулевое значение для остальных элементов стека.

2. *Время жизни* (TTL)

   Это 8-битовое поле используется для указания времени жизни. Обработка поля описана в параграфе 2.4.

3. *Для экспериментов*

   Это 3-битовое поле зарезервировано для экспериментального использования.

4. *Значение метки*

   Это 20-битовое поле содержит собственно метку.

При получении помеченного пакета просматривается значение верхней метки в стеке. В результате просмотра определяется:

а) следующий интервал пересылки пакета;

b) операция, выполняемая над стеком меток до пересылки (замена верхней метки в стеке, выталкивание верхней метки из стека или замена верхней метки с вталкиванием в стек одной или множества дополнительных меток).

Кроме того, в результате просмотра верхней метки стека может быть определена инкапсуляция для выходного канала, а также иные данные, которые требуются для корректной пересылки пакета.

Для меток имеется несколько зарезервированных значений:

i. Нулевое значение представляет IPv4 Explicit NULL Label[1]. Это значение допустимо только на дне стека меток и показывает, что метка должна быть вытолкнута из стека, а пересылка должна осуществляться на основе заголовка IPv4.

ii. Значение 1 представляет Router Alert Label. Такая метка может находиться в любом месте стека, за исключением дна. При получении пакета с такой меткой наверху стека, пакет передается для обработки локальным программам. Реальная пересылка такого пакета определяется по следующей метке в стеке. Однако, если пакет пересылается дальше, метка Router Alert должна быть помещена на вершину стека до пересылки пакета. Использование этой метки осуществляется аналогично опции Router Alert в заголовках пакетов IP [5]. Поскольку такая метка не может находиться на дне стека, она не связывается с каким-либо протоколом сетевого уровня.

iii. Значение 2 представляет IPv6 Explicit NULL Label. Это значение допустимо только на дне стека меток и показывает, что метка должна быть вытолкнута из стека, а пересылка должна осуществляться на основе заголовка IPv6.

iv. Значение 3 представляет неявную пустую метку (Implicit NULL Label). Такую метку LSR может выделить и распространять, но эта метка никогда не появляется в инкапсуляции. Когда LSR выполняет операцию замены метки и новым значением является Implicit NULL, маршрутизатор LSR будет просто выталкивать метку из стека вместо ее замены пустой меткой. Хотя это значение никогда не используется при инкапсуляции, оно требуется для протокола распространения меток и по этой причине зарезервировано.

v. Значения 4-15 являются резервными.

2.2. Определение протокола сетевого уровня

После выталкивания из стека последней метки (стек становится пустым) дальнейшая обработка такого пакета осуществляется на основе данных из заголовка сетевого уровня. LSR, выталкивающий из стека последнюю метку, должен, следовательно, иметь возможность идентификации протокола сетевого уровня. Однако стек меток не содержит какой-либо информации, явно идентифицирующей протокол сетевого уровня. Это означает, что такая идентификация должна выводиться из значения метки, выталкиваемой со дна стека, и, возможно, содержимого самого заголовка сетевого уровня.

Следовательно, при вталкивании в стек пакета сетевого уровня первой[2] метки, такая метка должна быть **однозначно** связана с конкретным протоколом сетевого уровня или неким набором протоколов сетевого уровня, когда пакеты могут быть более точно идентифицированы путем просмотра заголовка сетевого уровня. Более того, при замене такой метки в процессе передачи пакета другой меткой, новая метка должна удовлетворять тем же критериям. Если эти условия не выполняются, маршрутизатор LSR, вытолкнувший из стека последнюю метку, не сможет идентифицировать протокол сетевого уровня.

Точное выполнение этих условий не требует от промежуточных узлов идентификации протокола сетевого уровня. При нормальных условиях такого требования не возникает, однако при возникновении некоторых ошибок идентификация протокола сетевого уровня может оказаться желательной. Например, если промежуточный маршрутизатор LSR определяет, что помеченный пакет не может быть доставлен, для этого LSR может оказаться желательной генерация сообщения об ошибке в соответствии с используемым на сетевом уровне протоколом. Единственным способом определения протокола сетевого уровня на промежуточном LSR является просмотр верхней метки стека и заголовка сетевого уровня. Поэтому, если от промежуточных узлов требуется генерация специфических для протокола сетевого уровня сообщений об ошибках, все метки в стеке должны соответствовать требованиям, приведенным выше для последней (нижней) метки стека.

Если пакет по какой-либо причине (например, превышение MTU для канального уровня) не может быть переслан и протокол сетевого уровня невозможно идентифицировать или для данного протокола не существует правил обработки ошибок, такой пакет **должен** отбрасываться без уведомления.

2.3. Генерация сообщений ICMP для помеченных пакетов IP

В параграфах 2.4 и 3 обсуждаются ситуации, когда желательна генерация сообщений ICMP для помеченных пакетов IP. Чтобы маршрутизатор LSR мог сгенерировать пакет ICMP и передать его отправителю исходного пакета IP, нужно выполнить два условия:

1. LSR должен иметь возможность определения принадлежности конкретного помеченного пакета к протоколу IP;
2. LSR должен иметь возможность маршрутизации пакетов по адресу отправителя исходного пакета IP.

Условие 1 рассматривалось в параграфе 2.2. В двух следующих параграфах обсуждается условие 2. Однако следует отметить, что в некоторых ситуациях условие 2 не выполняется и генерация сообщения ICMP становится невозможной.

2.3.1. Туннелирование через транзитный домен маршрутизации

Предположим, что MPLS используется для туннелирования данных через транзитный маршрутный домен, где информация о внешних маршрутах не передается внутренним маршрутизаторам домена. Например, внутренние маршрутизаторы могут работать на основе протокола OSPF и знать лишь о доступности адресатов в своем домене

[1]Явная пустая (нулевая) метка IPv4.
[2]Эта метка в дальнейшем будет в стеке последней (нижней). *Прим. перев.*

OSPF. Домен может включать несколько граничных маршрутизаторов AC (ASBR[1]), которые связаны между собой по протоколу BGP. Однако в этом примере маршруты от BGP не передаются в OSPF, а на маршрутизаторах LSR, которые не относятся к числу ASBR, не используется протокол BGP.

В приведенном примере только маршрутизаторы ASBR будут знать путь к источнику того или иного пакета. Если внутреннему маршрутизатору нужно отправить сообщение ICMP отправителю пакета IP, он не будет знать, как следует маршрутизировать сообщение ICMP.

Одним из решений этой проблемы будет наличие одного или нескольких ASBR, которые передают в IGP маршрут по умолчанию (важно отметить, что это **не** требует передачи маршрута по умолчанию через BGP). В этом случае любой непомеченный пакет, который должен уйти из домена (например, пакет ICMP), будет передаваться маршрутизатору, имеющему полную маршрутную информацию. Такие маршрутизаторы будут помечать пакеты до их передачи назад через транзитный домен, поэтому использование маршрута по умолчанию внутри транзитного домена не будет порождать маршрутных петель.

Это решение будет работать только для пакетов, которые используют уникальные в глобальном масштабе адреса, и сетей, где все ASBR имеют полную маршрутную информацию. В следующем параграфе описано решение, которое будет работать без этих условий.

2.3.2. Туннелирование приватных адресов через публичные сети

В некоторых случаях при использовании MPLS для туннелирования через маршрутные домены маршрутизация по адресу отправителя фрагментов может оказаться невозможной совсем. Такие ситуации могут возникать, например, при туннелировании пакетов с приватными (не уникальными в глобальном масштабе) адресами IP через публичные сети на основе MPLS. В такой ситуации принятая по умолчанию маршрутизация в ASBR не решает проблемы.

В таких средах для передачи сообщения ICMP отправителю пакета можно копировать стек меток из исходного пакета в сообщение ICMP и использовать для этого сообщения коммутацию по меткам. Это приведет к передаче сообщения по адресу получателя исходного пакета, а не его отправителя. Если сообщение не передается на основе коммутации меток по всему пути, оно утратит последнюю метку на маршрутизаторе, который будет знать отправителя исходного пакета и сможет передать ему это сообщение.

Этот метод очень полезен для доставки ICMP-сообщений «Time Exceeded» или «Destination Unreachable because fragmentation needed and DF set».

При копировании стека из исходного пакета в сообщение ICMP значения меток должны копироваться в точности, за исключением значений TTL, для которых следует устанавливать значение поля TTL из IP-заголовка сообщения ICMP. Это значение TTL должно быть достаточно велико для того, сообщение ICMP могло дойти до адресата кружным путем.

Отметим, что при истечении TTL в результате возникновения маршрутной петли сообщение ICMP, отправленное описанным выше способом, также может зациклиться. Поскольку сообщения ICMP никогда не передаются в результате приема сообщения ICMP[2], а многие реализации ограничивают скорость генерации сообщений ICMP, такое поведение не должно вызывать проблем.

2.4. Обработка поля TTL

2.4.1. Определения

Входящее значение TTL для помеченного пакета определяется, как значение поля TTL в верхней метке стека при получении пакета.

Исходящим значением TTL для помеченного пакета является большее из двух значений:

a) уменьшенное на 1 входящее значение TTL;

b) 0.

2.4.2. Независимые от протокола правила

Если исходящее значение TTL для помеченного пакета равно 0, дальнейшая пересылка такого помеченного пакета **недопустима**, равно как и вырезание метки с дальнейшей пересылкой непомеченного пакета. Время жизни пакета считается завершенным.

В зависимости от значения метки в записи стека меток пакет **можно** просто отбросить или передать соответствующему (обычному) сетевому уровню для обработки ошибок (например, для генерации сообщения ICMP, как описано в параграфе 2.3).

При пересылке помеченного пакета в поле TTL верхнего элемента стека меток **должно** устанавливаться исходящее значение TTL.

Отметим, что исходящее значение TTL полностью определяется входящим значением и не зависит от вталкивания или выталкивания меток из стека перед пересылкой пакета. Значения поля TTL из элементов стека меток, не являющихся верхними, не играют никакой роли.

2.4.3. Правила для протокола IP

Определим значение поля IP TTL, как значение поля IPv4 TTL или поля IPv6 Hop Limit в зависимости от версии IP.

[1] Autonomous System Border Router.

[2] Это заявление не вполне соответствует требованиям параграфа 3.2.2 RFC 1122, где сказано, что сообщения ICMP **недопустимо** передавать в ответ на сообщения ICMP *об ошибках*. Однако в контексте этого документа проблем не возникает, поскольку речь здесь идет именно о сообщениях ICMP об ошибках. *Прим. перев.*

При создании первой метки для пакета IP значение поля TTL в элементе стека меток **должно** устанавливаться равным значению поля IP TTL (если IP TTL требуется декрементировать в процессе обработки IP, предполагается, что это уже сделано).

При выталкивании метки с опустошением стека значение поля IP TTL **следует** заменять исходящим значением TTL, как описано выше. В IPv4 при этом потребуется также пересчитать контрольную сумму заголовка IP.

Известно, что во многих случаях сетевые администраторы предпочитают декрементировать значение IPv4 TTL один раз в процессе прохождения через домен MPLS вместо декрементирования IPv4 TTL на число интервалов LSP в данном домене.

2.4.4. Преобразование типа инкапсуляции

Иногда LSR может получать пакет через интерфейс, управляемый коммутацией по меткам (например, LC-ATM[1] [9]), и этот пакет нужно будет передать через канал PPP или ЛВС. В этом случае входящий пакет не будет инкапсулирован в соответствии с данной спецификацией, но исходящий должен инкапсулироваться в соответствии с этой спецификацией.

В таких случаях значение «входящего TTL» определяется процедурами, используемыми для передачи помеченных пакетов (например, для интерфейсов LC-ATM) и обработка TTL выполняется в соответствии с приведенным выше описанием.

Иногда LSR может получать пакет через канал PPP или ЛВС и этот пакет нужно будет передать через интерфейс типа LC-ATM. В таких случаях входящий пакет использует инкапсуляцию, описанную в данном документе, но исходящий пакет должен инкапсулироваться иначе. Процедура передачи «исходящего TTL» будет определяться процедурами передачи помеченных пакетов через соответствующий интерфейс (например, LC-ATM).

3. Фрагментация и определение Path MTU

Как и для дейтаграмм IP без меток возможны ситуации, когда полученный пакет с меткой слишком велик для его передачи в выходной канал.

Возможны также случаи, когда принятый пакет (с меткой или без нее) приемлемого размера становится слишком большим в результате вталкивания в стек одной или множества меток. Например, при получении помеченного пакета с 1500 байтами информации и добавлении в стек дополнительной метки размер поля данных кадра канального уровня достигает 1504 байтов.

В этом разделе рассматриваются правила обработки помеченных пакетов избыточного размера. В частности, приведены правила, при выполнении которых хосты, реализующие процедуры Path MTU Discovery [4], и хосты IPv6 [7,8] смогут генерировать дейтаграммы IP, не требующие фрагментации даже в случае добавления к ним меток при передаче через сеть.

В общем случае хосты IPv4, не использующие Path MTU Discovery [4], передают дейтаграммы IP, которые содержат не более 576 байтов. Однако на большинстве современных каналов передачи данных значение MTU достигает 1500 и более байтов, поэтому вероятность фрагментирования таких дейтаграмм даже при добавлении к ним меток очень мала.

Некоторые хосты, не поддерживающие Path MTU Discovery [4], будут генерировать дейтаграммы IP, содержащие 1500 байтов, в которых IP-адреса отправителя и получателя относятся к одной подсети. Такие дейтаграммы не будут проходить через маршрутизаторы и, следовательно, не фрагментируются.

К сожалению некоторые хосты генерируют дейтаграммы IP, содержащие 1500 байтов, в тех случаях, когда IP-адреса отправителя и получателя имеют одинаковый (бесклассовый) номер сети. Это один из случаев, когда в результате добавления меток может потребоваться фрагментация дейтаграмм (тем не менее, необходимость фрагментации не очевидна, если пакет между моментами добавления и удаления меток не проходит через сеть Ethernet того или иного типа).

В этом документе описаны процедуры, которые позволяют настроить сеть так, что большие дейтаграммы от хостов, не поддерживающих Path MTU Discovery, будут фрагментироваться только один раз (при первом добавлении метки). Эти процедуры избавляют от необходимости фрагментировать пакеты, которые уже имеют метки (в предположении корректной настройки конфигурации).

3.1. Терминология

Применительно к конкретному каналу данных могут использоваться те или иные перечисленные ниже термины.

Frame Payload - данные кадра

Содержимое кадра канального уровня без заголовков и трейлеров канального уровня (например, заголовков MAC, LLC, 802.1Q, PPP, контрольных сумм кадра и т. п.).

Когда кадр передает непомеченную IP-дейтаграмму, Frame Payload представляет собой просто саму дейтаграмму IP. Если в кадре содержится помеченная дейтаграмма, Frame Payload будет включать стек меток и саму дейтаграмму IP.

Conventional Maximum Frame Payload Size - согласованный максимальный размер данных кадра

Максимальный размер Frame Payload, допускаемый стандартами канального уровня. Например, Conventional Maximum Frame Payload Size для сетей Ethernet составляет 1500 байтов.

True Maximum Frame Payload Size - истинный максимальный размер данных кадра

Максимальный размер данных кадра, которые могут быть корректно переданы и приняты интерфейсным оборудованием, подключенным к каналу данных.

Для сетей Ethernet и 802.3 предполагается, что True Maximum Frame Payload Size на 4-8 байтов превышает Conventional Maximum Frame Payload Size (пока заголовки 802.1Q и 802.1p отсутствуют и ни один из таких заголовков не может быть добавлен коммутатором или мостом, через который пакет передается на следующий интервал). Например, предполагается, что большая часть оборудования Ethernet может корректно передавать и

[1] Label switching controlled ATM.

принимать кадры с размером данных 1504, а возможно и 1508 байтов, пока заголовок кадра Ethernet не включает полей 802.1Q или 802.1p.

На каналах PPP значение True Maximum Frame Payload Size виртуально может быть неограниченным.

Effective Maximum Frame Payload Size for Labeled Packets - эффективный максимальный размер данных кадра для помеченных пакетов

Значение Conventional Maximum Frame Payload Size или True Maximum Frame Payload Size, в зависимости от возможностей оборудования передачи данных и используемого размера заголовков канального уровня.

Initially Labeled IP Datagram - первоначально помечаемая дейтаграмма IP

Предположим, что непомеченная дейтаграмма IP получена неким LSR и этот маршрутизатор LSR втолкнул метку в стек до пересылки дейтаграммы. Такая дейтаграмма на данном LSR будет называться первоначально помечаемой.

Previously Labeled IP Datagram - ранее помеченная дейтаграмма IP

Дейтаграмма IP, которая уже была помечена до ее получения LSR.

3.2. Максимальный размер первоначально помечаемых дейтаграмм IP

Каждому LSR, который способен

a) принимать непомеченные дейтаграммы IP;

b) добавлять стек меток к дейтаграмме;

c) пересылать полученный в результате пакет с меткой,

следует поддерживать конфигурационный параметр Maximum Initially Labeled IP Datagram Size, который может принимать неотрицательные значения.

Если этот параметр имеет нулевое значение, он не оказывает никакого влияния.

Если этот параметр имеет положительное значение, оно используется, как описано ниже. Если выполняются все приведенные условия:

a) получена дейтаграмма IP без метки;

b) бит DF[1] в заголовке дейтаграммы IP не установлен;

c) дейтаграмму нужно пометить до ее пересылки;

d) размер дейтаграммы (без добавляемой метки) превосходит значение параметра,

то выполняются следующие действия:

a) дейтаграмма делится на фрагменты, размеры которых не превышают значение параметра;

b) каждый фрагмент помечается и пересылается.

Например, при установке для параметра значения 1488, все непомеченные дейтаграммы IP, содержащие более 1488 байтов будут фрагментироваться перед добавлением метки. Каждый фрагмент в этом случае может без дополнительной фрагментации передавать 1500 байтов данных на канальном уровне даже при размещении в стеке 3 меток.

Иными словами, установка ненулевого значения параметра позволяет избежать дополнительной фрагментации ранее помеченных дейтаграмм IP, но может привести к ненужной фрагментации первоначально помечаемых дейтаграмм IP.

Отметим, что установка этого параметра не оказывает влияния на обработку дейтаграмм IP с установленным флагом DF, следовательно, этот параметр не оказывает влияния на параметры фрагментации, установленные с помощью Path MTU discovery.

3.3. Когда помеченная дейтаграмма IP слишком велика?

Помеченная дейтаграмма IP, размер которой превышает Conventional Maximum Frame Payload Size для канального уровня, в который пакет будет пересылаться, **может** трактоваться, как «слишком большая».

Помеченная дейтаграмма IP, размер которой превышает True Maximum Frame Payload Size для канального уровня, через который она будет пересылаться, **должна** трактоваться, как «слишком большая».

Помеченные дейтаграммы IP, которые не являются «слишком большими» **должны** пересылаться без фрагментации.

3.4. Обработка помеченных дейтаграмм IPv4 избыточного размера

Если помеченная дейтаграмма IPv4 «слишком велика» и в заголовке IP не установлен флаг DF, LSR **может** отбросить дейтаграмму без уведомления.

Отметим, что отбрасывание таких дейтаграмм является осмысленной процедурой лишь в том случае, когда установлено ненулевое значение параметра Maximum Initially Labeled IP Datagram Size на каждом LSR в сети, который способен добавлять стек меток в непомеченные дейтаграммы IP.

Если LSR принимает решение не отбрасывать помеченную дейтаграмму IPv4 избыточного размера или в этой дейтаграмме установлен флаг DF, маршрутизатор **должен** использовать приведенный ниже алгоритм:

1. «Вырезать» элементы стека меток для получения дейтаграммы IP.
2. Пусть N - число байтов в стеке меток (число меток в стеке, умноженное на 4).
3. Если в дейтаграмме IP **не** установлен флаг запрета фрагментирования в заголовке IP, нужно:

[1] Флаг запрета фрагментирования. *Прим. перев.*

a. разбить дейтаграмму на фрагменты, каждый из которых **должен** быть по крайней мере на N байтов меньше эффективного максимума размера данных кадра (Effective Maximum Frame Payload Size);

b. присоединить к каждому фрагменту спереди тот же заголовок с метками, который имела исходная дейтаграмма до фрагментирования;

c. переслать все фрагменты.

4. Если в заголовке дейтаграммы IP установлен флаг запрета фрагментирования:

   a. **недопустимо** пересылать дейтаграмму;

   b. нужно создать сообщение ICMP Destination Unreachable:

      i. установить в поле Code [3] этого сообщения значение Fragmentation Required and DF Set;

      ii. установить в качестве значения поля Next-Hop MTU [4] разницу между значением эффективного максимума размера данных в кадре и N;

   c. по возможности, передать сообщение ICMP Destination Unreachable отправителю отброшенной дейтаграммы.

3.5. Обработка помеченных дейтаграмм IPv6 избыточного размера

Для обработки дейтаграмм IPv6 избыточного размера LSR **должен** использовать описанный ниже алгоритм.

1. «Вырезать» элементы стека меток для получения дейтаграммы IP.
2. Пусть N - число байтов в стеке меток (число меток в стеке, умноженное на 4).
3. Если дейтаграмма IP содержит более 1280 байтов (без учета стека меток) или не содержит заголовка фрагмента, нужно:

   a. создать сообщение ICMP Packet Too Big и установить в нем для поля Next-Hop MTU значение разницы между эффективным максимумом размера данных в кадре и N;

   b. по возможности, передать сообщение ICMP Packet Too Big отправителю отброшенной дейтаграммы;

   c. отбросить помеченную дейтаграмму IPv6.

4. Если размер дейтаграммы IP не превышает 1280 октетов и она имеет заголовок фрагмента, нужно:

   a. разбить дейтаграмму на фрагменты, каждый из которых **должен** быть по крайней мере на N байтов меньше эффективного максимума размера данных в кадре;

   b. присоединить к каждому фрагменту спереди тот же заголовок с метками, который имела исходная дейтаграмма до фрагментирования;

   c. переслать все фрагменты.

   Сборка фрагментов будет осуществляться получателем.

3.6. Взаимодействие с Path MTU Discovery

Описанные выше процедуры обработки дейтаграмм избыточного размера с установленным флагом запрета фрагментирования DF оказывают влияние на процедуры Path MTU Discovery, описанные в RFC 1191 [4]. Хосты, использующие эти процедуры, будут получать значение MTU, которое достаточно мало и позволяет поместить в дейтаграмму n[1] меток без возникновения необходимости фрагментирования дейтаграммы.

Иными словами, дейтаграммы от хостов, использующих Path MTU Discovery никогда не потребуется фрагментировать в результате необходимости добавления заголовка меток или добавления меток в существующий заголовок (обычно дейтаграммы от хостов, использующих Path MTU Discovery имеют флаг DF и, следовательно, не могут фрагментироваться).

Отметим, что Path MTU Discovery будет корректно работать только при наличии в точке, где возникает необходимость фрагментирования, помеченной дейтаграммы IP, есть возможность генерации и передачи отправителю исходной дейтаграммы сообщения ICMP Destination Unreachable (см. 2.3. Генерация сообщений ICMP для помеченных пакетов IP).

Если нет возможности пересылки сообщения ICMP в «туннеле» MPLS по адресу отправителя пакета, а конфигурация сети позволяет LSR на передающей стороне туннеля принимать пакеты, которые должны пройти сквозь туннель, но слишком велики для туннелирования без фрагментации:

- LSR на передающей стороне туннеля **должен** иметь возможность определения MTU для туннеля в целом. Это **можно** сделать путем передачи пакетов через туннель на приемную сторону и выполнения для этих пакетов процедуры Path MTU Discovery.
- Когда передающей стороне требуется передать в туннель пакет с установленным флагом DF и размером, превышающим значение MTU для туннеля, передающая сторона туннеля **должна** передать отправителю пакета сообщение ICMP Destination Unreachable с кодом Fragmentation Required and DF Set и значением поля Next-Hop MTU, установленным, как сказано выше.

[1] Число меток, которые могут быть помещены в дейтаграмму на пути доставки.

4. Передача помеченных пакетов по каналам PPP

Протокол PPP[1] [6] обеспечивает стандартный метод транспортировки дейтаграмм разных протоколов по каналам «точка-точка». PPP определяет расширяемый протокол управления каналом LCP[2] и предлагает семейство протоколов управления сетью NCP[3] для организации и настройки конфигурации различных протоколов сетевого уровня.

В этом параграфе определен протокол управления сетью для организации и настройки коммутации по меткам через PPP.

4.1. Введение

PPP включает три основных компоненты:

1. метод инкапсуляции дейтаграмм различных протоколов;
2. протокол LCP для организации, настройки параметров и тестирования соединений канального уровня;
3. семейство протоколов NCP для организации и настройки различных протоколов сетевого уровня.

Для организации соединения через канал «точка-точка» каждая сторона канала PPP должна сначала передать пакеты LCP для настройки и тестирования канала данных. После организации канала и согласования опций, как требует LCP, протокол PPP должен передать пакеты MPLS Control Protocol для того, чтобы разрешить передачу помеченных пакетов. После перехода протокола MPLS Control в состояние Opened можно передавать через канал пакеты с метками.

Канал будет сохранять коммуникационные настройки, пока не будет явно закрыт с помощью пакета протокола управления LCP или MPLS Control или в результате некого внешнего события (например, по тайм-ауту или в результате действий сетевого администратора).

4.2. Протокол PPP NCP для MPLS

Протокол MPLSCP[4] отвечает за разрешение и запрет коммутации по меткам на канале PPP. Он использует такой же механизм обмена сообщениями, как в протоколе LCP. Обмен пакетами MPLSCP не допускается, пока PPP не достигнет фазы Network-Layer Protocol. Полученные до этого момента пакеты MPLSCP следует отбрасывать без уведомления.

Ниже перечислены отличия протокола MPLSCP от LCP [6].

1. Изменение кадров

   Пакет может использовать любые изменения базового формата кадров, которые были согласованы в фазе Link Establishment.

2. Поле протокола канального уровня

   В поле PPP Information может инкапсулироваться единственный пакет MPLSCP. Поле PPP Protocol в этом случае содержит шестнадцатеричное значение 8281 (MPLSCP[5]).

3. Поле Code

   Используются только значения кодов от 1 до 7 (Configure-Request, Configure-Ack, Configure-Nak, Configure-Reject, Terminate-Request, Terminate-Ack и Code-Reject). Остальные коды следует трактовать, как нераспознанные, и передавать в ответ Code-Reject.

4. Тайм-ауты

   Обмен пакетами MPLSCP не допускается, пока PPP не перейдет в фазу Network-Layer Protocol. Реализациям следует быть готовыми к ожиданию завершения фаз Authentication и Link Quality Determination прежде, чем прервать ожидание Configure-Ack или другого отклика. Предполагается, что реализация будет прерывать ожидание только после вмешательства пользователя или по истечении заданного в конфигурации времени.

5. Типы конфигурационных опций

   Нет.

4.3. Передача помеченных пакетов

До начала обмена помеченными пакетами PPP должен достичь фазы Network-Layer Protocol, а MPLSCP - состояния Opened.

В поле PPP Information инкапсулируется единственный помеченный пакет и поле PPP Protocol содержит шестнадцатеричное значение 0281 (MPLS Unicast) или 0283 (MPLS Multicast). Максимальный размер пакета с меткой, передаваемого по каналу PPP, совпадает с максимальным размером поля Information для пакетов, инкапсулируемых в PPP.

Формат самого поля Information для пакетов с метками определен в разделе 2.

Отметим, что для пакетов с метками выделены два кода - один для индивидуальной адресации, другой - для групповой. Как только протокол MPLSCP достигнет состояния Opened, через канал PPP можно будет передавать помеченные пакеты как с индивидуальной, так и с групповой адресацией.

---

[1] Point-to-Point Protocol.
[2] Link Control Protocol.
[3] Network Control Protocol.
[4] MPLS Control Protocol.
[5] В исходном документе ошибочно указано «MPLS». *Прим. перев.*

4.4. Конфигурационные опции MPLSCP

Протокол не имеет конфигурационных опций.

5. Передача помеченных пакетов через ЛВС

В каждом кадре передается единственный пакет с меткой.

Элементы стека меток непосредственно предшествуют заголовку сетевого уровня, следуя сразу после всех заголовков канального уровня (включая любые заголовки 802.1Q, которые могут существовать).

Шестнадцатеричное значение ethertype = 8847 используется для индикации кадров с пакетами MPLS с индивидуальной адресацией.

Шестнадцатеричное значение ethertype = 8848 используется для индикации кадров с пакетами MPLS с групповой адресацией.

Эти значение ethertype могут использоваться с инкапсуляцией Ethernet или 802.3 LLC/SNAP для передачи помеченных пакетов. Процедура выбора конкретного типа инкапсуляции выходит на пределы этого документа.

6. Согласование с IANA

Значение меток от 0 до 15, включительно, используются для специальных целей, указанных в этом документе или заданных позднее агентством IANA.

В этом документе значения меток от 0 до 3 описаны в параграфе 2.1.

Значения меток от 4 до 15 выделяются агентством IANA по согласованию с IETF (IETF Consensus).

7. Вопросы безопасности

Описанная здесь инкапсуляция MPLS не вызывает новых проблем в плане безопасности, которых уже не было отмечено в архитектуре MPLS [1] или протокола сетевого уровня, используемого для инкапсуляции.

Существует две унаследованных от архитектуры MPLS проблемы безопасности, которые следует отметить здесь:

- Некоторые маршрутизаторы могут поддерживать процедуры защиты, зависящие от положения заголовка сетевого уровня относительно заголовка канального уровня. Такие процедуры не будут работать при использовании инкапсуляции MPLS, поскольку при этом применяются заголовки переменного размера.
- Значение метки MPLS определяется соглашением между LSR, помещающим метку в стек (label writer), и LSR, который интерпретирует эту метку (label reader). Однако в стеке меток не содержится информации, позволяющей идентифицировать создателя конкретной метки. Если помеченные пакеты принимаются от недоверенного источника, это может привести к нарушению картины маршрутизации.

8. Интеллектуальная собственность

IETF уведомлен о правах интеллектуальной собственности, связанными со всеми или некоторыми спецификациями, содержащимися в документе. Дополнительную информацию можно получить в online-списке заявленных прав.

9. Адреса авторов

Eric C. Rosen

Cisco Systems, Inc.

250 Apollo Drive

Chelmsford, MA, 01824

EMail: erosen@cisco.com

Dan Tappan

Cisco Systems, Inc.

250 Apollo Drive

Chelmsford, MA, 01824

EMail: tappan@cisco.com

Yakov Rekhter

Juniper Networks

1194 N. Mathilda Avenue

Sunnyvale, CA 94089

EMail: yakov@juniper.net

Guy Fedorkow

Cisco Systems, Inc.

250 Apollo Drive

Chelmsford, MA, 01824

EMail: fedorkow@cisco.com

Dino Farinacci

Procket Networks, Inc.

3910 Freedom Circle, Ste. 102A

Santa Clara, CA 95054

EMail: dino@procket.com

Tony Li

Procket Networks, Inc.

3910 Freedom Circle, Ste. 102A

Santa Clara, CA 95054

EMail: tli@procket.com

Alex Conta

TranSwitch Corporation

3 Enterprise Drive

Shelton, CT, 06484

EMail: aconta@txc.com

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

10. Литература

[1] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031[1], January 2001.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119[1], March 1997.

[3] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792[1], September 1981.

[4] Mogul, J. and S. Deering, "Path MTU Discovery", RFC 1191[1], November 1990.

[5] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[6] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661[1], July 1994.

[7] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 1885, December 1995.

[8] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[9] Davie, B., Lawrence, J., McCloghrie, K., Rekhter, Y., Rosen, E. and G. Swallow, "MPLS Using LDP and ATM VC Switching", RFC 3035, January 2001.

11. Полное заявление авторских прав

[1]Перевод этого документа имеется на сайте www.protocols.ru. *Прим. перев.*

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.