

Network Working Group
Request for Comments: 5444
Category: Standards Track

T. Clausen
LIX, Ecole Polytechnique
C. Dearlove
BAE Systems ATC
J. Dean
Naval Research Laboratory
C. Adjih
INRIA Rocquencourt
February 2009

Обобщенный формат пакетов/сообщений MANET

Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format

Статус документа

Этот документ является проектом стандарта Internet (Internet Standards Track) и служит приглашением к дискуссии и внесению предложений с целью совершенствования протокола. Информацию о состоянии стандартизации и статусе протокола можно найти в текущей редакции документа «Internet Official Protocol Standards» (STD 1). Документ может распространяться свободно.

Авторские права

Авторские права (Copyright (c) 2009) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К этому документу применимы права и ограничения, перечисленные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно.

Аннотация

Этот документ задаёт форматы пакетов, способных передавать множество сообщений, которые могут использоваться протоколами маршрутизации в специализированных мобильных сетях.

Оглавление

1. Введение.....	2
2. Обозначения и терминология.....	2
2.1. Обозначения.....	2
2.1.1. Элементы.....	2
2.1.2. Переменные.....	3
2.2. Терминология.....	3
3. Заявление о применимости.....	3
4. Обзор и работа протокола.....	4
5. Спецификация синтаксиса.....	4
5.1. Пакеты.....	4
5.2. Сообщения.....	4
5.3. Адресные блоки.....	5
5.4. TLV и блоки TLV.....	6
5.4.1. TLV.....	7
5.4.2. Применение TLV.....	8
5.5. Элементы с некорректным форматом.....	8
6. Взаимодействие с IANA.....	8
6.1. Expert Review - рекомендации по оценке.....	8
6.2. Типы сообщений.....	9
6.2.1. Создание реестра Message-Type-Specific TLV.....	9
6.3. Типы Packet TLV.....	9
6.3.1. Создание реестра Packet TLV Type Extension.....	9
6.4. Типы Message TLV.....	9
6.4.1. Создание реестра Message TLV Type Extension.....	9
6.5. Типы Address Block TLV.....	9
6.5.1. Создание реестра Address Block TLV Type Extension.....	10
7. Вопросы безопасности.....	10
7.1. Предложения по аутентификации и защите целостности.....	10
7.2. Предложения по конфиденциальности.....	10
8. Участники работы.....	11
9. Благодарности.....	11
10. Литература.....	11
10.1. Нормативные документы.....	11
10.2. Дополнительная литература.....	11

Приложение А. Мультиплексирование и демуплексирование.....	11
Приложение В. Предусмотренное применение.....	12
Приложение С. Примеры.....	12
С.1. Примеры Address Block.....	12
С.2. Примеры TLV.....	14
Приложение D. Иллюстрации.....	15
D.1. Пакет.....	15
D.2. Сообщение.....	16
D.3. Тело сообщения.....	19
D.4. Адресный блок.....	20
D.5. Блок TLV.....	24
D.6. TLV.....	24
Приложение Е. Полный пример.....	27

1. Введение

Этот документ задаёт синтаксис формата пакетов, предназначенных для транспортировки множества сообщений протоколов маршрутизации, служащих для обмена информацией между маршрутизаторами MANET¹. Сообщения состоят из заголовка (Message Header), предназначенного для контроля распространения сообщений, и тела (Message Body) с протокольной информацией. Спецификация задаёт только синтаксис пакетов и сообщений.

В данном документе приведены спецификации перечисленных ниже элементов.

- Формат пакетов, позволяющих за одну передачу отправить более одного сообщения (могут передаваться также пакеты без сообщений, содержащие некую информацию только в заголовке).
- Формат сообщений, состоящих из заголовка (Message Header) и тела (Message Body).
- Формат заголовка, содержащего информацию, которой использующему эту спецификацию протоколу может быть достаточно для принятия решения о пересылке или обработке.
- Формат тела сообщения, содержащего атрибуты, связанные с сообщением или его инициатором, а также блоки адресов или адресные префиксы с соответствующими атрибутами.
- Формат блока адресов (Address Block), представляющего некое множество адресов или адресных префиксов в компактной форме с возможностью агрегирования.
- Обобщенный формат TLV² для представления атрибутов. Каждый TLV может быть связан с пакетом, сообщением, одним или множеством адресов или префиксов в одном Address Block. Возможно включение множества TLV, каждый из которых связан с пакетом, сообщением, а также одним, разными или пересекающимися наборами адресов или адресных префиксов.

Ниже приведён частичный список свойств, которые явно предусмотрены спецификацией.

Parsing logic - логика разбора

Используемые в этой спецификации обозначения усиливают базовую, протоколно-независимую логику разбора.

Extensibility - расширяемость

Пакеты и сообщения, определяемые использующим эту спецификацию протоколом, могут быть расширены путём добавления новых TLV. Протоколы, использующие эту спецификацию, способны корректно идентифицировать и пропускать неизвестные сообщения и TLV, корректно разбирая остальные пакеты и сообщения.

Efficiency - эффективность

Для случаев использования адресов с одинаковыми последовательностями битов (например, адресные префиксы или идентификаторы интерфейсов IPv6) Address Block обеспечивает компактное представление. Компактность заголовков сообщений обеспечивается за счёт включения в них только необходимых элементов. Структура пакетов с множеством сообщений позволяет снизить число передаваемых пакетов и октетов. Структура пакетов и представление сообщений позволяют разобрать, проверить и идентифицировать элементы за один проход.

Separation of forwarding and processing - разделение обработки и пересылки

Использующие эту спецификацию протоколы могут быть устроены так, что обнаружение дубликатов и управление областью пересылки могут выполняться на основе данных из заголовка сообщения без анализа его тела.

2. Обозначения и терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

Используемые в документе обозначения описаны в параграфе 2.1, а определения терминов - в параграфе 2.2.

2.1. Обозначения

Ниже приведены обозначения элементов и переменных, используемых в этом документе.

Этот формат использует сетевой порядок байтов (сначала передаётся старший октет) для всех полей. Для старшего бита в октете используется номер 0, для младшего - 7 [Stevens].

2.1.1. Элементы

Данная спецификация определяет элемент, как группу из произвольного числа последовательных битов, которые совместно формируют синтаксический объект, представляемый обозначением <element>. Каждый элемент в данной спецификации определяется одним из двух способов:

¹Mobile Ad hoc NETwork - специализированная мобильная сеть.

²Type-length-value - тип-размер-значение.

- битовое поле указанного размера;
- композитный элемент, состоящий из других <element>.

Композитный элемент определяется, как:

`<element> := specification`

где указанная справа от знака := спецификация представляется с использованием синтаксиса регулярных выражений, определённого в [SingleUNIX]. При этом используются лишь приведённые ниже обозначения.

`<element1><element2>` - за <element1> непосредственно следует <element2>;

`(<element1><element2>)` - указывает на группировку элементов, заключённых в скобки;

? - 0 или 1 включение предшествующего элемента или группы;

* - произвольное число (возможно, 0) включений предшествующего элемента или группы.

2.1.2. Переменные

Переменные введены в спецификацию исключительно в целях прояснения описаний. Используется два варианта обозначения переменных:

`<foo>` - если <foo> является целым числом без знака, обозначение <foo> используется также для представления значения этого поля;

`bar` - переменная, обычно получаемая путём расчёта на основе значений элементов.

2.2. Терминология

Ниже приведены определения используемых в документе терминов.

Packet - пакет

Объект (сущность) верхнего уровня в данной спецификации. Пакет содержит заголовок (Packet Header) и может также включать сообщения.

Message - сообщение

Фундаментальная единица передачи протокольной информации в форме адресных объектов и TLV.

Address - адрес

Множество октетов, образующих адрес, размер которого указан в инкапсулирующем заголовке сообщения. Значение адреса определяется использующим спецификацию протоколом.

Address Prefix - адресный префикс

Адрес и размер префикса, задаваемым числом битов адреса, отсчитываемых слева (старший бит).

Address Object - адресный объект

Адрес или адресный префикс, как указано для Address Block в данной спецификации.

TLV

Структура «тип-размер-значение» (type-length-value). Это базовый способ представления и корректного разбора атрибутов без понимания их сущности.

3. Заявление о применимости

Эта спецификация описывает базовый формат пакетов, предназначенный для использования протоколами маршрутизации MANET. Основой для спецификации послужил протокол OLSR¹ [RFC3626], который она расширяет.

Для сетей MANET, как правило, характерна возможность работы через беспроводные интерфейсы с ограниченными возможностями. Следовательно, для сетей MANET более критичен объем передаваемых данных, нежели для большинства кабельных сетей. Данная спецификация предлагает компромиссное решение с учётом эффективности, расширяемости и простоты использования.

Эффективность обеспечивается за счёт снижения размера пакетов и возможности передачи множества не связанных между собой сообщений в одном пакете. Снижение размера пакетов обеспечивается прежде всего за счёт агрегирования адресов, необязательности полей в заголовках пакетов и сообщений, а также полей в адресных блоках и TLV. Поддержка пакетов с множеством сообщений позволяет снизить число пакетов, каждый из которых может потреблять значимые ресурсы пропускной способности на транспортном, сетевом и нижележащих уровнях.

Спецификация обеспечивает как внутреннюю, так и внешнюю расширяемость. Внешняя расширяемость обеспечивается возможностью добавления Packet TLV и определения новых типов сообщений. Внутренняя расширяемость обеспечивается возможностью добавления Message TLV и Address Block TLV в имеющиеся сообщения. Протоколы могут определять новые типы TLV и, следовательно, значения полей Value в них, а также новые типы сообщений (см. параграф 6.1). Протоколы также могут пользоваться определениями типов TLV из других протоколов, соответствующих данной спецификации.

Целью этой спецификации является обеспечение достаточно выразительности и гибкости для восприятия различных классов протоколов маршрутизации MANET (например, проактивных, реактивных и гибридных), а также их расширений. Наличие общего формата пакетов и сообщений, а также единый способ представления адресов IP и связанных с ними атрибутов позволяют разрабатывать базовые средства синтаксического анализа, не зависящие от используемых протоколами алгоритмов.

Адреса в сообщении предполагаются имеющими один размер, указанный в Message Header. При смешанной адресации IPv6 и IPv4, последний тип представляется, как отображаемые на IPv4 адреса IPv6 [RFC4291].

Сообщения, определяемые данной спецификацией, предназначены для передачи сигналов протоколов маршрутизации MANET между маршрутизаторами MANET. Спецификация включает элементы, которые могут поддерживать лавинную рассылку в ограниченной области, а также пригодные для доставки сигнальных сообщений протоколов маршрутизации MANET по каналам «точка-точка» в сетях с множеством интервалов между

¹Optimized Link State Routing - оптимизированный протокол маршрутизации по состоянию каналов.

маршрутизаторами (multi-hop network). Пакеты могут быть групповыми или индивидуальными и могут использовать любой транспортный протокол или обходиться без него.

Протоколы маршрутизации MANET, использующие определённый здесь формат сообщений, могут ограничивать поддерживаемый ими синтаксис (например, требуя конкретного набора полей в Message Header). Протоколы с такими ограничениями должны быть способны разбирать все возможные структуры сообщений, определённые в данном документе, но при этом должны быть последовательными при генерации и приёме определяемых ими сообщений. Если протокол задаёт включаемые элементы, возможно прямое индексирование соответствующих полей с зависимости от вносимых протоколом синтаксических ограничений. Расширяемость таких протоколов может быть ограниченной.

4. Обзор и работа протокола

Данная спецификация не описывает протокол, задавая лишь формат сообщений, которые могут использоваться любым протоколом маршрутизации в специализированных (ad hoc) сетях.

5. Спецификация синтаксиса

В этом разделе приводится нормативная спецификация пакетов, представляемых элементом `<packet>` и составляющими пакет элементами. В спецификации используются обозначения, описанные в параграфе 2.1.

Графические представления элементов спецификации даны в Приложении D, а графическое представление полного примера (пакет, включающий сообщение с Address Block и TLV) - в Приложении E.

Данный формат использует сетевой порядок байтов, как указано в параграфе 2.1.

5.1. Пакеты

`<packet>` определяется, как

```
<packet> := <pkt-header>
          <message>*
```

где `<message>` определено в параграфе 5.2. Разбор пакета завершается успешно, когда использованы все октеты пакета (как указано в содержащей пакет дейтаграмме).

`<pkt-header>` определяется, как

```
<pkt-header> := <version>
                <pkt-flags>
                <pkt-seq-num>?
                <tlv-block>?
```

где

<version> - 4-битовое целое число без знака, указывающее версию спецификации, которой соответствует пакет и содержащееся в нем сообщение. Данный документ задаёт версию 0.

<pkt-flags> - 4-битовое поле, задающее интерпретацию оставшейся части Packet Header:

бит 0 (phasseqnum) - при сброшенном (0) бите `<pkt-seq-num>` не включается в `<pkt-header>`, при установленном (1) включается.

бит 1 (phastlv) - при сброшенном (0) бите `<tlv-block>` не включается в `<pkt-header>`, при установленном (1) включается.

Биты 2-3 - резервные биты, которые **следует** сбрасывать (0) при передаче и игнорировать при получении.

<pkt-seq-num> отсутствует, если флаг phasseqnum сброшен (0), а при установленном бите содержит 16-битовое целое число без знака, задающее порядковый номер пакета (Packet Sequence Number).

<tlv-block> отсутствует, если флаг phastlv сброшен (0) при установленном бите содержит TLV (см. параграф 5.4).

Предполагается, что сетевой уровень способен точно передать размер содержимого (payload length), поэтому размер пакета в нем не указывается.

5.2. Сообщения

В дополнение к заголовку (Packet Header) пакет может содержать одно или множество сообщений, включающих:

- заголовок (Message Header);
- блок Message TLV, который может включать TLV, связанные с сообщением в целом;
- могут включаться адресные блоки (Address Block), каждый из которых содержит один или множество адресных объектов;
- блок Address Block TLV, который может содержать TLV, за каждым из которых следует Address Block, с адресами которого могут быть связаны другие атрибуты.

`<message>` определяется, как

```
<message> := <msg-header>
            <tlv-block>
            (<addr-block><tlv-block>)*

<msg-header> := <msg-type>
                <msg-flags>
                <msg-addr-length>
                <msg-size>
```

```

<msg-orig-addr>?
<msg-hop-limit>?
<msg-hop-count>?
<msg-seq-num>?

```

где

<tlv-block> определён в параграфе 5.4.

<addr-block> определён в параграфе 5.3.

<msg-type> - 8-битовое целое число без знака, указывающее тип сообщения.

<msg-flags> - 4-битовое поле, определяющее интерпретацию оставшейся части Message Header:

бит 0 (mhasorig) - при сброшенном (0) флаге <msg-orig-addr> не включается в <msg-header>, при установленном (1) включается;

бит 1 (mhashoplimit) - при сброшенном (0) флаге <msg-hop-limit> не включается в <msg-header>, при установленном (1) включается;

бит 2 (mhashorcount) - при сброшенном (0) флаге <msg-hop-count> не включается в <msg-header>, при установленном (1) включается;

бит 3 (mhasseqnum) - при сброшенном (0) флаге <msg-seq-num> не включается в <msg-header>, при установленном (1) включается.

<msg-addr-length> - 4-битовое целое число без знака, представляющее размер всех адресов, включённых в это сообщение (<msg-orig-addr>, а также адреса в Address Block, как описано в параграфе 5.3):

<msg-addr-length> = размер адреса в октетах - 1

<msg-addr-length> - 3 для адресов IPv4, 15 для адресов IPv6.

Переменная address-length представляет размер адреса в октетах и рассчитывается по формуле:

address-length = <msg-addr-length> + 1

<msg-size> - 16-битовое целое число без знака, указывающее число октетов в <message> с учётом <msg-header>.

<msg-orig-addr> - не указывается при сброшенном (0) флаге mhasorig, а при установленном является идентификатором с размером address-length, который может служить для уникальной идентификации маршрутизатора MANET, создавшего это сообщение.

<msg-hop-limit> - не указывается при сброшенном (0) флаге mhashoplimit, а при установленном флаге содержит 8-битовое целое число без знака, которое может указывать максимальное число интервалов для последующей пересылки сообщения.

<msg-hop-count> - не указывается при сброшенном (0) флаге mhashorcount, а при установленном флаге содержит 8-битовое целое число без знака, которое может указывать число интервалов, через которые уже прошло сообщение.

<msg-seq-num> - не указывается при сброшенном (0) флаге mhasseqnum, а при установленном флаге содержит 16-битовое целое число без знака, которое может указывать порядковый номер, заданный маршрутизатором MANET, создавшим сообщение.

5.3. Адресные блоки

Address Block может задавать один или множество адресов, размер которых определяется октетами address-length с использованием <msg-addr-length> в <msg-header> сообщения, содержащего Address Block. Блок адресов может также задавать размер префикса, который может быть применён для всех адресов в Address Block, если это приемлемо. Это позволяет задавать с помощью Address Block как адреса, так и адресные префиксы. Протокол может указать, что адреса с максимальным размером префикса (размер адреса в битах или $8 * \text{address-length}$) рассматриваются, как адреса, а не префиксы - это позволяет включать в Address Block комбинации адресов и префиксов. Общий термин «адресный объект» служит для обозначения обоих типов объектов. Отметим, что такие объекты в Address Block всегда включают размер префикса.

Адреса задаются последовательностью октетов address-length в форме Head:Mid:Tail¹. Семантика для Head, Mid и Tail не задаётся - такое представление служит исключительно для обеспечения возможности агрегирования адресов, которые зачастую имеют общие части (например, общий префикс или множество адресов IPv6 на одном интерфейсе). Address Block содержит упорядоченный список адресов, имеющих общее значение компонент Head и Tail, но свои значения Mid. Компоненты Head и Tail могут быть пустыми (независимо друг от друга), что позволяет представлять адреса объектов, не имеющих общей части в голове (пустая компонента Head) или хвосте (пустая компонента Tail). Подробные примеры списков Address Blocks приведены в Приложении С.1.

Address Block может задавать адресные префиксы:

- с одним размером префикса для всех префиксов;
- с указанием размера для каждого префикса.

<address-block> определяется, как

```

<address-block> := <num-addr>
                  <addr-flags>
                  (<head-length><head>)?
                  (<tail-length><tail>)?
                  <mid>*
                  <prefix-length>*

```

¹«Голова:центральная часть:хвост»

где

<num-addr> - 8-битовое целое число без знака, указывающее количество адресов, представленных в Address Block (значение 0 **недопустимо**).

<addr-flags> - 8-битовое поле, определяющее интерпретацию остальной части Address Block:

бит 0 (ahashead) - при сброшенном (0) флаге <head-length> и <head> не включаются в <address-block>. Установленный флаг означает включение <head-length> в <address-block>, <head> включается в <address-block> для <head-length> отличных от 0.

бит 1 (ahasfulltail) и бит 2 (ahaszerotail) - интерпретируются в соответствии с таблицей 1. Не показанные в таблице комбинации **недопустимы**.

бит 3 (ahassingleprelen) и бит 4 (ahasmultiprelen) - интерпретируются в соответствии с таблицей 2. Не показанные в таблице комбинации **недопустимы**.

Биты 5-7 - резервные биты. Их **следует** сбрасывать (0) при передаче и игнорировать при получении.

Таблица 1. Интерпретация флагов ahasfulltail u ahaszerotail.

ahasfulltail	ahaszerotail	<tail-length>	<tail>
0	0	Не включено	Не включено
1	0	Включено	Включено, если <tail-length> отлична от 0
0	1	Включено	Не включено

Таблица 2. Интерпретация флагов ahassingleprelen u ahasmultiprelen.

ahassingleprelen	ahasmultiprelen	Число полей <prefix-length>	Размер n-го адресного префикса в битах
0	0	0	8 * address-length
1	0	1	<prefix-length>
0	1	<num-addr>	n-й <prefix-length>

<head-length> (при наличии) 8-битовое целое число без знака, указывающее количество октетов в компоненте Head всех адресов в Address Block (т. е., размер каждого элемента <head> в <head-length> в октетах).

Размер «головы» предполагается равным <head-length> (при наличии) или 0 (в противном случае).

<head> отсутствует при нулевом размере «головы», в противном случае содержит head-length старших (левых) октетов, одинаковых для всех адресов в Address Block.

<tail-length> (при наличии) 8-битовое целое число без знака, указывающее количество октетов в компоненте Head всех адресов в Address Block (т. е., размер каждого элемента <tail> в <tail-length> в октетах).

Размер «хвоста» предполагается равным <tail-length> (при наличии) или 0 (в противном случае).

<tail> отсутствует при нулевом размере «хвоста» или установленном флаге ahaszerotail, в противном случае содержит tail-length младших (правых) октетов, одинаковых для всех адресов в Address Block. Если установлен флаг ahaszerotail, значения tail-length младших октетов для всех адресов в Address Block имеют значение 0.

Размер средней части адреса (mid-length) **должен** быть неотрицательным и определяется выражением

$$\text{mid-length} := \text{address-length} - \text{head-length} - \text{tail-length}$$

т. е., каждый элемент <mid> включает mid-length октетов.

<mid> отсутствует при нулевом размере «середины», в противном случае каждая компонента <mid> представляет собой поле из mid-length октетов, представляющих среднюю часть соответствующего адреса в Address Block. При наличии этой компоненты Address Block содержит в точности <num-addr> полей <mid>.

<prefix-length> - 8-битовое целое число без знака, указывающее размер адресного префикса в битах. При установленном флаге ahassingleprelen включается единственное поле <prefix-length>, указывающее размер всех префиксов в Address Block. При установленном флаге ahasmultiprelen включается <num-addr> полей <prefix-length>, каждое из которых указывает размер соответствующего адресного префикса в Address Block (в порядке префиксов). В остальных случаях поле <prefix-length> не включается и каждый адресный объект может рассматриваться, как имеющий префикс размера 8 * address-length битов. Формат Address Block будет некорректным, если значение любого элемента <prefix-length> превышает 8 * address-length.

5.4. TLV и блоки TLV

TLV позволяют связать произвольный атрибут с сообщением или пакетом, а также с адресом или непрерывным блоком адресов в Address Block. Атрибут (значение) представляет собой целое число последовательных октетов. Различные атрибуты имеют разные типы, неизвестные атрибуты могут пропускаться при разборе.

TLV объединяются в блоки (TLV Block), где все TLV связываются с пакетом (TLV Block в Packet Header), сообщением (TLV Block сразу вслед за Message Header) или адресами в непосредственно предшествующем Address Block. Отдельные TLV в TLV Block, непосредственно следующие за Address Block могут связывать атрибуты с одним адресом, диапазоном адресов или всеми адресами в данном Address Block. Когда атрибут связывается с множеством адресов, TLV может включать одно значение для всех адресов или индивидуальные для каждого адреса. Детальные примеры TLV приведены в Приложении С.2.

TLV Block определяется, как

```
<tlv-block> := <tlvs-length>
               <tlv>*
```

где

<tlvs-length> - 16-битовое целое число без знака, указывающее общее количество октетов во всех непосредственно следующих элементах <tlv> (<tlvs-length> не учитываются).

<tlv> - см. параграф 5.4.1.

5.4.1. TLV

Существует три вида TLV, каждый из которых представляется элементом <tlv>:

- Packet TLV включается в Packet TLV Block заголовка пакета (Packet Header);
- Message TLV включается в Message TLV Block в теле сообщения до всех Address Block;
- Address Block TLV включается в Address Block TLV Block, следующий за Address Block; Address Block TLV применяется для любых из перечисленных ниже объектов в Address Block:
 - всех адресных объектов;
 - любой непрерывной последовательности адресных объектов;
 - одного адресного объекта.

<tlv> определяется, как

```
<tlv> := <tlv-type>
        <tlv-flags>
        <tlv-type-ext>?
        (<index-start><index-stop>?)?
        (<length><value>?)?
```

где

<tlv-type> - 8-битовое целое число без знака, указывающее тип TLV, зависящий от вида TLV (т. е. Packet TLV, Message TLV или Address Block TLV).

<tlv-flags> - 8-битовое поле, задающее интерпретацию оставшейся части TLV:

бит 0 (thastypeext) - при сброшенном (0) флаге <tlv-type-ext> не включается в <tlv>, при установленном включается;

бит 1 (thassingleindex) и бит 2 (thasmultiindex) - интерпретируются в соответствии с таблицей 3. Не указанные в таблице комбинации применять **недопустимо**; в Packet TLV и Message TLV оба флага должны быть сброшены (0);

бит 3 (thasvalue) и бит 4 (thasextlen) - интерпретируются в соответствии с таблицей 4. Не указанные в таблице комбинации применять **недопустимо**;

бит 5 (tismultivalue) - этот флаг указывает способ интерпретации поля <value>, описанный ниже. Этот флаг **должен** быть сброшен в Packet TLV и Message TLV, если сброшен флаг thasmultiindex или thasvalue;

биты 6-7: - резервные флаги, их **следует** сбрасывать при передаче и игнорировать при получении.

Таблица 3. Интерпретация флагов thassingleindex и thasmultiindex.

thassingleindex	thasmultiindex	<index-start>	<index-stop>
0	0	Не включено	Не включено
1	0	Включено	Не включено
0	1	Включено	Включено

Таблица 4. Интерпретация флагов thasvalue и thasextlen.

thasvalue	thasextlen	<length>	<value>
0	0	Не включено	Не включено
1	0	8 битов	Включено, если <length> отлично от 0
0	1	16 битов	Включено, если <length> отлично от 0

<tlv-type-ext> - 8-битовое целое число без знака, указывающее расширение TLV Type, специфическое для данного типа и вида TLV (Packet TLV, Message TLV, Address Block TLV).

Переменная tlv-type-ext имеет значение <tlv-type-ext> при наличии того или 0 при отсутствии.

Переменная tlv-fulltype определяется выражением

```
tlv-fulltype := 256 * <tlv-type> + tlv-type-ext
```

<index-start> и <index-stop> при их наличии (только в Address Block TLV) представляют собой 8-битовые числа без знака.

Переменные index-start и index-stop определяются в соответствии с таблицей 5. Переменная end-index определяется, как

- для Message TLV и Packet TLV

```
end-index := 0
```

- для Address Block TLV

```
end-index := <num-addr> - 1
```

Address Block TLV применяется к адресам объектов с позиции index-start до позиции index-stop (включительно) в Address Block, первый адрес имеет позицию 0.

Таблица 5. Интерпретация флагов thassingleindex и thasmultiindex.

thassingleindex	thasmultiindex	index-start :=	index-stop :=
0	0	0	end-index
1	0	<index-start>	<index-start>
0	1	<index-start>	<index-stop>

Переменная number-values определяется выражением

```
number-values := index-stop - index-start + 1
```

<length> опускается или может быть 8-ми или 16-битовым целым числом без знака, как указано в таблице 4. Если установлен (1) флаг `tismultivalued`, поле **<length>** **должно** быть целым кратным `number-values` и переменной `single-length`, определяемой выражением

```
single-length := <length> / number-values
```

Если флаг `tismultivalued` сброшен (0), переменная `single-length` определяется выражением

```
single-length := <length>
```

<value> при его наличии (см. таблицу 4) имеет размер `<length>` октетов. В Address Block TLV поле `<value>` связано с адресными объектами в позициях от `index-start` до `index-stop`, включительно. Если флаг `tismultivalued` сброшен (0), это поле целиком связано с каждым из указанных адресных объектов. При установленном (1) флаге `tismultivalued` это поле делится на `number-values` полей одного размера `single-length`, которые связываются в порядке следования с указанными адресными объектами.

5.4.2. Применение TLV

TLV связывает атрибут с пакетом, сообщением или одним или множеством последовательных адресных объектов в Address Block. Интерпретация и обработка этого атрибута, а также связи (включая порядок обработки) между разными атрибутами одного объекта **должны** определяться каждым протоколом, использующим данную спецификацию.

Любой протокол, соответствующий данной спецификации, **должен** определять своё поведение для случаев несогласованности информации - в частности, если два TLV одного типа задают разные значения для одного и того же объекта (пакета, сообщения или адреса), но это действие не имеет смысла. Протокол также **должен** задавать порядок обработки для TLV, связанных с одним объектом.

5.5. Элементы с некорректным форматом

Формат элемента считается некорректным, если его невозможно разобрать в соответствии со спецификацией синтаксиса (включая и нехватку октетов). Если некорректный элемент содержится в Packet Header, пакет **должен** отбрасываться без уведомления, а содержащееся в нем сообщение **недопустимо** обрабатывать или пересылать. Если некорректный элемент содержится в сообщении (Message TLV Block, Address Block или Address Block TLV Block), сообщение **должно** отбрасываться без уведомления, его обработка или пересылка **недопустимы**.

6. Взаимодействие с IANA

Этот документ вводит 4 новых регистрируемых пространства имён: Message Types, Packet TLV Types, Message TLV Types и Address Block TLV Types. Данная спецификация указывает реестры IANA для этих пространств и содержит рекомендации агентству IANA по части регистрации в этих пространствах имён.

Термины Namespace, Assigned Value, Registration, Unassigned, Reserved, Hierarchical Allocation и Designated Expert трактуются в соответствии с [BCP26].

Правила Private Use, Expert Review и Standards Action трактуются в соответствии с [BCP26].

6.1. Expert Review - рекомендации по оценке

Для регистрационных запросов, где требуется процедура Expert Review, назначенному эксперту (Designated Expert) **следует** принять во внимание перечисленные ниже рекомендации общего плана.

- Задачей этих реестров является поддержка стандартной и экспериментальной маршрутизации MANET, а также связанных с ней протоколов и расширений для них.
- Все регистрации должны сопровождаться публикацией RFC.
- Для регистрации до одобрения публикации RFC назначенный эксперт может выдать одобрение, если он уверен в предстоящей публикации RFC.
- Назначенный эксперт будет направлять в рабочую группу MANET или пришедшую ей на замену группу по указанию руководителя направления запрос для комментариев и рецензирования. Запрос будет включать ссылку на документ Internet-Draft, запрашивающий регистрацию.
- До истечения периода в 30 дней назначенный эксперт будет ожидать одобрения или отказа в регистрации и публиковать решение в списке рассылок рабочей группы MANET или заменившей её группы, а также информировать IANA и IESG. Сообщение об отказе **должно** сопровождаться объяснением причины и, по возможности, **следует** вносить предложения по изменению запроса для последующей попытки регистрации.

Рекомендации для реестра Message Types.

- Регистрация Message Type предполагает создание двух реестров для Message-Type-specific Message TLV и Message-Type-specific Address Block TLV. Документ, запрашивающий регистрацию Message Type, **должен** указывать, как эти Message-Type-specific TLV Type будут выделяться с использованием любого из указанных в [BCP26] вариантов и любым исходным распределением значений. Назначенному эксперту (Designated Expert) **следует** принять во внимание правила выделения, заданные для этих реестров, при рассмотрении запроса на выделение Message Type.

Рекомендации для реестров Packet TLV Types, Message TLV Types и Address Block TLV Types.

- Эти распределения являются иерархическими (Hierarchical Allocation), т. е., при выделении типа создаётся реестр для соответствующих этому типу расширенных типов. Запрашивающий регистрацию типа документ **должен** указывать способ выделения этих расширенных типов с использованием любого из указанных в [BCP26] вариантов и любым исходным распределением значений. Обычно для такого выделения требуется также экспертная оценка (Expert Review), но с возможностью выделения некоторых расширений типов, как резервных (Reserved), экспериментальных (Experimental) или для частных приложений (Private).

- Запрос для TLV Type **должен** включать спецификацию разрешённого размера, синтаксиса, любой внутренней структуры и толкования поля Value (при его наличии) в TLV.

Рекомендации для реестров Message TLV Types и Address Block TLV Types.

- Значения TLV Type от 0 до 127 являются общими для всех типов сообщений. TLV с типами из этого диапазона **следует** быть модульными для обеспечения использования разными протоколами.
- Значения TLV Type от 128 до 223 являются специфическими для определённых типов сообщений (Message-Type-specific TLV Type) и применимы лишь в контексте соответствующего Message Type. Регистрация значений TLV Type от 128 до 223 требует наличия реестра для конкретного значения Message Type (см. параграф 6.2.1) и регистрация выполняется в соответствии с правилами, заданными для реестров этого типа сообщений. Message-Type-specific TLV Type **следует** регистрировать для TLV, которые назначенный эксперт счёл излишне специфичными для регистрации в диапазоне от 0 до 127. В интервале от 128 до 233 одному значению TLV Type **может** соответствовать множество определений TLV в разных реестрах Message-Type-specific TLV Type. По возможности **следует** использовать имеющиеся определения глобальных TLV и модульных глобальных TLV для регистрации в диапазоне 0 - 127.

6.2. Типы сообщений

Для типов сообщений создаётся новый реестр Message Types, для которого начальное распределение и правила приведены в таблице 6.

Таблица 6. Типы сообщений.

Тип	Описание	Политика присвоения значений
0 - 223	Не распределены	Expert Review
224-255	Не распределены	Экспериментальное использование

6.2.1. Создание реестра Message-Type-Specific TLV

При регистрации Message Type **должны** указываться реестры для Message-Type-specific Message TLV (таблица 8) и Message-Type-specific Address Block TLV (таблица 10). Документ, задающий создание реестра Message-Type-specific TLV, **должен** также указывать механизм выделения Message-Type-specific TLV Type из числа описанных в [BCP26].

6.3. Типы Packet TLV

Для типов Packet TLV создаётся новый реестр Packet TLV Types, для которого начальное распределение и правила приведены в таблице 7.

Таблица 7. Типы Packet TLV.

Тип	Описание	Политика присвоения значений
0 - 223	Не распределены	Expert Review
224-255	Не распределены	Экспериментальное использование

6.3.1. Создание реестра Packet TLV Type Extension

При регистрации Packet TLV Type должен создаваться новый реестр для расширений данного типа пакетов. Документ, определяющий Packet TLV Type, **должен** также указывать механизм распределения расширений типа из числа описанных в [BCP26].

6.4. Типы Message TLV

Создаётся новый реестр для Message-Type-independent Message TLV Types с начальным распределением и правилами, показанными в таблице 8.

Таблица 8. Типы общих Message TLV.

Тип	Описание	Политика присвоения значений
0 - 127	Не распределены	Expert Review
128 - 223	Зависит от типа сообщения	Резерв, см. Таблицу 9
224-255	Не распределены	Экспериментальное использование

Значения типов Message TLV от 128 до 223 резервируются для специфических для типа сообщений Message TLV и для них создаётся новый реестр, начальное распределение и правила выделения для которого представлены в таблице 9.

Таблица 9. Типы специфических Message TLV.

Тип	Описание	Политика присвоения значений
0 - 127	Общие для всех типов сообщений	Резерв
128 - 223	Специфические для типа сообщений	См. ниже
224-255	Общие для всех типов сообщений	Резерв

Правила выделения значений зависящих от типа сообщений Message TLV **должны** указываться при создании реестра, связанного с соответствующим типом сообщений, как указано в параграфе 6.2.1.

6.4.1. Создание реестра Message TLV Type Extension

Если регистрируется тип Message TLV, для него должен создаваться новый реестр для расширений типа. Документ, определяющий Message TLV Type, **должен** также указывать механизм выделения расширений из числа определённых в [BCP26].

6.5. Типы Address Block TLV

Создаётся новый реестр Message-Type-independent Address Block TLV Types, начальное распределение и правила выделения для которого представлены в таблице 10.

Таблица 10. Типы Address Block TLV.

Тип	Описание	Политика присвоения значений
0 - 127	Не распределены	Expert Review
128 - 223	Специфические для типа сообщений	Резерв, см. Таблицу 11
224-255	Не распределены	Экспериментальное использование

Значения Address Block TLV Type от 128 до 223, включительно, зарезервированы для Message-Type-specific Address Block TLV и для них при регистрации соответствующих Message Type создаются новые реестры, начальное распределение и правила выделения для которых представлены в таблице 11.

Таблица 11. Типы Address Block TLV.

Тип	Описание	Политика присвоения значений
0 - 127	Общие для всех типов сообщений	Резерв
128 - 223	Специфические для типа сообщений	См. ниже
224-255	Общие для всех типов сообщений	Резерв

Правила выделения значений зависящих от типа сообщений Address Block TLV Type **должны** указываться при создании реестра, связанного с соответствующим типом сообщений, как указано в параграфе 6.2.1.

6.5.1. Создание реестра Address Block TLV Type Extension

При регистрации Address Block TLV Type создаётся новый реестр для расширений типа. Документ, определяющий Address Block TLV¹ Type, **должен** также указывать механизм выделения расширений из числа определённых в [BCP26].

7. Вопросы безопасности

Данная спецификация описывает не протокол, а форматы пакетов. По этой причине здесь не рассматриваются какие-либо вопросы безопасности - они относятся к протоколам, использующим данную спецификацию. Тем не менее, спецификация делает возможными некоторые механизмы защиты, которые могут послужить частью использующих спецификацию протоколов. Механизмы, которые могут использоваться для аутентификации и защиты целостности, рассмотрены в параграфе 7.1. Механизмы, которые могут использоваться протоколами для защиты конфиденциальности, рассмотрены в параграфе 7.2. Однако к протоколам, использующим данную спецификацию, не предъявляется требований в части применения описанных здесь механизмов.

7.1. Предложения по аутентификации и защите целостности

Предложения по аутентификации и защите целостности сделаны на основе предполагаемого применения, описанного в Приложении В, в частности с учётом того, что:

- сообщения предназначены для переноса протокольной информации и на каждом интервале **могут** пересылаться и/или обрабатываться протоколом, использующим данную спецификацию;
- пакеты предназначены для переноса множества сообщений между соседними маршрутизаторами MANET в один приём через один логический интервал (hop).

Следствия этого в плане аутентификации и защиты целостности приведены ниже.

- Для пересылаемых сообщений, которые не изменяются на пересылающих их маршрутизаторах MANET, **может** применяться сквозная аутентификация и защита целостности между маршрутизаторами MANET с установленными защищёнными связями путём включения подходящих Message TLV с криптографическими подписями сообщений. Поскольку при пересылке сообщений могут изменяться только поля <msg-hop-count> и <msg-hop-limit>, в расчёт контрольной суммы можно включать все сообщение с Message Header, принимая значения полей <msg-hop-count> и <msg-hop-limit> нулевыми (если эти поля имеются).
- **Может** быть реализована поэтапная (hop-by-hop) аутентификация и защита целостности между маршрутизаторами MANET с имеющимися защищёнными связями путём включения в пакеты подходящих TLV с криптографической подписью пакета. Поскольку пакеты в процессе передачи не меняются, подпись можно рассчитывать для всего пакета или его отдельных частей.

7.2. Предложения по конфиденциальности

Данная спецификация не включает явной защиты конфиденциальности пакетов и сообщений. При необходимости такая защита может обеспечиваться на каждом этапе пересылки (hop-by-hop) с помощью механизмов канального уровня или IP с использованием [RFC4301] и будет применяться только для пакетов.

Однако использующий эту спецификацию протокол может защитить конфиденциальность информации в пакетах, сообщениях или Address Block TLV, задавая шифрование полей Value в TLV с использованием указанного механизма шифрования.

В предельном случае можно шифровать всю информацию с помощью одного из описанных ниже способов.

- Пакет, содержащий только Packet Header (без сообщения) и Packet TLV, где Packet TLV Type указывает, что поле Value содержит одно или множество зашифрованных сообщений. При получении пакета и успешной расшифровке Packet TLV сообщения могут быть разобраны в соответствии с этой спецификацией и процедурами использующего спецификацию протокола.
- Сообщение, содержащее только Message Header и один экземпляр Message TLV, где Message TLV Type указывает, что поле Value содержит зашифрованную версию оставшихся Message TLV, Address Block и Address Block TLV из сообщения. При получении и успешной расшифровке Message TLV полное сообщение разбирается в соответствии с данной спецификацией и процедурами использующего спецификацию протокола.

¹В оригинале ошибочно сказано Message TLV. См. https://www.rfc-editor.org/errata_search.php?eid=4178. Прим. перев.

В обоих случаях протокол **должен** определить тип для шифруемого TLV, а также формат зашифрованного блока данных, содержащего поле Value этого TLV.

8. Участники работы

Эта спецификация является результатом работы группы OLSRv2 Design Team, перечисленных в алфавитном порядке:

- Cedric Adjih, INRIA, France, <Cedric.Adjih@inria.fr>;
- Emmanuel Baccelli, INRIA, France, <Emmanuel.Baccelli@inria.fr>;
- Thomas Heide Clausen, LIX, Ecole Polytechnique, France, <T.Clausen@computer.org>;
- Justin W. Dean, NRL, USA, <jdean@itd.nrl.navy.mil>;
- Christopher Dearlove, BAE Systems, UK, <chris.dearlove@baesystems.com>;
- Satoh Hiroki, Hitachi SDL, Japan, <hiroki.satoh.yj@hitachi.com>;
- Philippe Jacquet, INRIA, France, <Philippe.Jacquet@inria.fr>;
- Monden Kazuya, Hitachi SDL, Japan, <kazuya.monden.vw@hitachi.com>.

9. Благодарности

Авторы выражают свою благодарность команде OLSR [RFC3626], включая Anis Laouiti (INT, France), Pascale Minet, Laurent Viennot (оба из INRIA, France) и Amir Qayyum (Center for Advanced Research in Engineering, Pakistan) за их вклад, Elwyn Davies (Folly Consulting, UK), Lars Eggert (Nokia, Finland), Chris Newman (Sun Microsystems, USA), Tim Polk (NIST, USA) и Magnus Westerlund (Ericsson, Sweden) предоставили подробные рецензии и значимые комментарии.

Авторы также выражают искреннюю признательность перечисленным (в алфавитном порядке) ниже людям за обсуждение технических вопросов, рецензии и комментарии к спецификации и её частям.

- Brian Adamson (NRL);
- Teco Boot (Infinity Networks);
- Florent Brunneau (LIX);
- Ian Chakeres (CenGen);
- Alan Cullen (BAE Systems);
- Ulrich Herberg (LIX);
- Joe Macker (NRL);
- Yasunori Owada (Niigata University);
- Charlie E. Perkins (WiChorus);
- Henning Rogge (FGAN);
- Andreas Schjonhaug (LIX);
- и вся рабочая группа IETF MANET.

10. Литература

10.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), BCP 14, March 1997.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.
- [BCP26] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, [RFC 5226](#), May 2008.
- [SingleUNIX] IEEE Std 1003.1, The Open Group, and ISO/IEC JTC 1/SC22/WG15, "Single UNIX Specification, Version 3, 2004 Edition", April 2004.

10.2. Дополнительная литература

- [RFC3626] Clausen, T. and P. Jacquet, "The Optimized Link State Routing Protocol", RFC 3626, October 2003.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", [RFC 4301](#), December 2005.
- [Stevens] Stevens, W., "TCP/IP Illustrated Volume 1 - The Protocols", 1994.

Приложение А. Мультиплексирование и демультиплексирование

Формат пакетов и сообщений, описанный в этом документе, позволяет передавать в одном пакете несколько (включая 0) сообщений. Эти сообщения могут относиться к одному или множеству протоколов. Таким образом, **должен** присутствовать процесс мультиплексирования и демультиплексирования сообщений.

Мультиплексирование сообщений на данном маршрутизаторе MANET в один пакет вместо генерации пакетов для каждого сообщения снижает общий объем передаваемых данных и число пакетов, отправляемых маршрутизатором MANET.

Процесс мультиплексирования и демultipлексирования, работающий через данный порт UDP или с данным номером протокола IP и связанными с ним протоколами, **должен** соответствовать приведённым ниже условиям.

- Для каждого типа сообщений протокол (если явно не указано иное, один из зарезервированных IANA для данного Message Type) **должен** быть указан в качестве «владельца» этого Message Type.
- Поля заголовка пакета (Packet Header), включая Packet TLV Block, используются процессами мультиплексирования и демultipлексирования, которые **могут** сделать эту информацию доступной для использования в своих экземплярах протокола.
- Поле <pkt-seq-num> (при его наличии) содержит порядковый номер, который **следует** поддерживать для каждого¹ вовлечённого интерфейса и увеличивать на 1 в каждом пакете, генерируемом узлом для данного интерфейса. По достижении значения 65535 нумерация продолжается с 0. Иными словами, нумерация является циклической.
- Входящие сообщения **должны** отбрасываться без уведомления или **должны** доставляться экземпляру протокола, который владеет соответствующим Message Type. Входящие сообщения **не следует** доставлять какому-либо иному экземпляру протокола, а также **не следует** доставлять их нескольким экземплярам.
- Исходящие сообщения данного типа **должны** генерироваться только экземпляром протокола, владеющим Message Type и доставляться процессу мультиплексирования и демultipлексирования.
- Если два протокола хотят использовать один тип сообщений, такое взаимодействие **следует** включать в спецификацию протокола, указанного владельцем данного Message Type.

Приложение В. Предусмотренное применение

В этом приложении описано предполагаемое использование полей Message Header, а также их содержимое. Возможны и другие варианты применения данной спецификации.

Указанный в этом документе формат сообщений предназначен для передачи сигнальных сообщений протокола маршрутизации MANET между маршрутизаторами MANET и поддержки ограниченной лавинной рассылки, а также доставки в режиме «точка-точка».

Сообщения устроены так, чтобы обеспечивалась возможность пересылки через один или множество логических интервалов с использованием на каждом интервале нового пакета. Каждый логический интервал может включать один или множество интервалов пересылки IP.

В частности, лавинная рассылка в ограниченной области поддерживается для сообщений в указанных ниже случаях.

- Поле <msg-orig-addr> (при наличии) содержит уникальный идентификатор маршрутизатора MANET, создавшего сообщение.
- Поле <msg-seq-num> (при наличии) содержит порядковый номер. Отсчёт начинается с 0 при создании узлом первого сообщения данного типа, а затем номера увеличиваются на 1 для каждого следующего сообщения этого типа. После номера 65535 снова используется 0. Иными словами, нумерация является циклической.
- При наличии обоих полей <msg-orig-addr> и <msg-seq-num> заголовок сообщения обеспечивает подавление дубликатов с помощью идентификатора, состоящего из полей <msg-orig-addr>, <msg-seq-num> и <msg-type>. Это обеспечивает уникальную идентификацию сообщений MANET в течение цикла <msg-seq-num>.
- Поле <msg-hop-limit> (при наличии) содержит число интервалов пересылки (hop), через которые сообщению² разрешено пройти прежде, чем оно будет отброшено маршрутизатором MANET. Значение <msg-hop-limit> задаётся инициатором сообщения и служит для предотвращения бесконечной циркуляции сообщений в MANET. При пересылке сообщений маршрутизаторам MANET следует уменьшать <msg-hop-limit> на 1 и отбрасывать сообщения с <msg-hop-limit> = 0.
- Поле <msg-hop-count> (при наличии) указывает число интервалов пересылки, через которые сообщение² прошло в сети MANET. Это поле служит для предотвращения бесконечной циркуляции сообщений в MANET. Инициатор сообщения устанавливает <msg-hop-count> = 0 и каждому пересылающему сообщению маршрутизатору MANET следует увеличивать <msg-hop-count> на 1, а сообщения с <msg-hop-count> = 255 следует отбрасывать.
- При наличии обоих полей <msg-hop-limit> и <msg-hop-count> заголовок сообщения обеспечивает информацию для принятия решений о пересылке в режиме ограниченной лавинной рассылки (scope-limited flooding). Это может быть любой механизм рассылки, заданный протоколом, использующим данную спецификацию.

Приложение С. Примеры

В этом приложении приведены примеры для некоторых частей данной спецификации.

С.1. Примеры Address Block

Приведённые ниже примеры иллюстрируют некоторые комбинации адресов, которые могут быть включены в Address Block. Примеры даны для IPv4 с размером адреса 4. Обозначения a, b, c и т. д. представляют отличные от нуля значения октетов.

Отметим, что разрешено использовать менее эффективное представление, в частности, со сброшенными флагами ahashead и ahasfulltail (0) и, следовательно, head-length = 0, tail-length = 0, mid-length = address-length, Address Block (без адресных префиксов), включающий множество адресов, <addr-flags> = 0 и список неагрегированных адресов. Это

¹В оригинале по ошибке не указано, что это требование относится к каждому участвующему интерфейсу. См. https://www.rfc-editor.org/errata_search.php?eid=3496. Прим. перев.

²В оригинале ошибочно сказано «пакет». См. https://www.rfc-editor.org/errata_search.php?eid=4003. Прим. перев.

наиболее эффективная форма представления одного адреса и единственный вариант представить, например, a.b.c.d и e.f.g.h в одном Address Block.

Примеры

- Для включения a.b.c.d, a.b.e.f и a.b.g.h:
 - head-length = 2;
 - tail-length = 0;
 - mid-length = 2;
 - <addr-flags> имеет значение ahashead = 128;
 - <tail-length> и <tail> опущены.Address Block будет иметь вид 3 128 2 a b c d e f g h (11 октетов).
- Для включения a.b.c.g и d.e.f.g:
 - head-length = 0;
 - tail-length = 1;
 - mid-length = 3;
 - <addr-flags> имеет значение ahasfulltail = 64;
 - <head-length> и <head> опущены.Address Block будет иметь вид 2 64 1 g a b c d e f (10 октетов).
- Для включения a.b.d.e и a.c.d.e:
 - head-length = 1;
 - tail-length = 2;
 - mid-length = 1;
 - <addr-flags> имеет значения ahashead и ahasfulltail, равные 192.Address Block будет иметь вид 2 192 1 a 2 d e b c (9 октетов).
- Для включения a.b.0.0, a.c.0.0 и a.d.0.0:
 - head-length = 1;
 - tail-length = 2;
 - mid-length = 1;
 - <addr-flags> имеет значения ahashead и ahaszerotail, равные 160;
 - <tail> опущено.Address Block будет иметь вид 3 160 1 a 2 b c d (8 октетов).
- Для включения a.b.0.0 и c.d.0.0:
 - head-length = 0;
 - tail-length = 2;
 - mid-length = 2;
 - <addr-flags> имеет значение ahaszerotail = 32;
 - <head> и <tail> опущены.Address Block будет иметь вид 2 32 2 a b c d (7 октетов).
- Для включения a.b.0.0/n и c.d.0.0/n:
 - head-length = 0;
 - tail-length = 2;
 - mid-length = 2;
 - <addr-flags> имеет значения ahaszerotail и ahassingleprelen, равные 48;
 - <head> и <tail> опущены.Address Block будет иметь вид 2 48 2 a b c d n (8 октетов).
- Для включения a.b.0.0/n и c.d.0.0/m:
 - head-length = 0;
 - tail-length = 2;

- mid-length = 2;
- <addr-flags> имеет значения ahaszerotail и ahasmultiprelen, равные 40;
- <head> и <tail> опущены.

Address Block будет иметь вид 2 40 2 a b c d n m (9 октетов).

С.2. Примеры TLV

Предположим определение Address Block TLV с EXAMPLE1 (без расширения типа), имеющим однооктетные значения адресов. Существует много способов, которыми значения a, a, b и c можно связать с четырьмя адресами в предшествующем Address Block, где c является принятым по умолчанию значением, которм можно не указывать.

Примеры.

- Использование многозначного TLV для покрытия всех адресов:
 - <tlv-flags> имеет значения thasvalue и tismultivalue, равные 20;
 - <index-start> и <index-stop> опущены;
 - <length> = 4 (single-length = 1).TLV будет иметь вид EXAMPLE1 20 4 a a b c (7 октетов).
- Использование одного многозначного TLV без указания последнего адреса:
 - <tlv-flags> имеет значения thasmultiindex, thasvalue и tismultivalue, равные 52;
 - <index-start> = 0;
 - <index-stop> = 2;
 - <length> = 3 (single-length = 1).TLV будет иметь вид EXAMPLE1 52 0 2 3 a a b (8 октетов).
- Использование двух однозначных TLV без указания последнего адреса.

Первый:

- <tlv-flags> имеет значения thasmultiindex и thasvalue, равные 48;
 - <index-start> = 0;
 - <index-stop> = 1;
 - <length> = 1;
 - <value> = a.
- TLV будет иметь вид EXAMPLE1 48 0 1 1 a (6 октетов).

Второй:

- <tlv-flags> имеет значения thassingleindex и thasvalue, равные 80;
 - <index-start> = 2;
 - <index-stop> опущено;
 - <length> = 1;
 - <value> = b.
- TLV будет иметь вид EXAMPLE1 80 2 1 b (5 октетов).

Общий размер двух TLV - 11 октетов.

В этом случае первый вариант является наиболее эффективным. В иных случаях более предпочтительными могут оказаться другие варианты. В зависимости от эффективности можно использовать любой из них.

Предположим определение Address Block TLV с типом EXAMPLE2 (без расширения типа), не имеющим значения и связанным со вторым и третьим адресом в Address Block. Это можно выразить одним TLV:

- <tlv-flags> имеет значение thasmultiindex = 32;
- <index-start> = 1;
- <index-stop> = 2;
- <length> и <value> опущены.

TLV будет иметь вид EXAMPLE2 32 1 2 (4 октета).

Предположим определение Message TLV с типом EXAMPLE3 (без расширения типа), который может принимать поля Value любого размера. Для такого TLV с 8 октетами данных (a - h):

- <tlv-flags> имеет значение thasvalue = 16;
- <index-start> и <index-stop> опущены;

- <length> = 8.

TLV будет иметь вид EXAMPLE3 16 8 a b c d e f g h (11 октетов).

Если в этом примере число октетов составит 256 или больше, <tlv-flags> будет иметь значение thasextlen = 24. Для размера потребуется два октета (сначала старший). Размер TLV составит 4 + N октетов, где N - число октетов данных (3 + N октетов, если N не превышает 255).

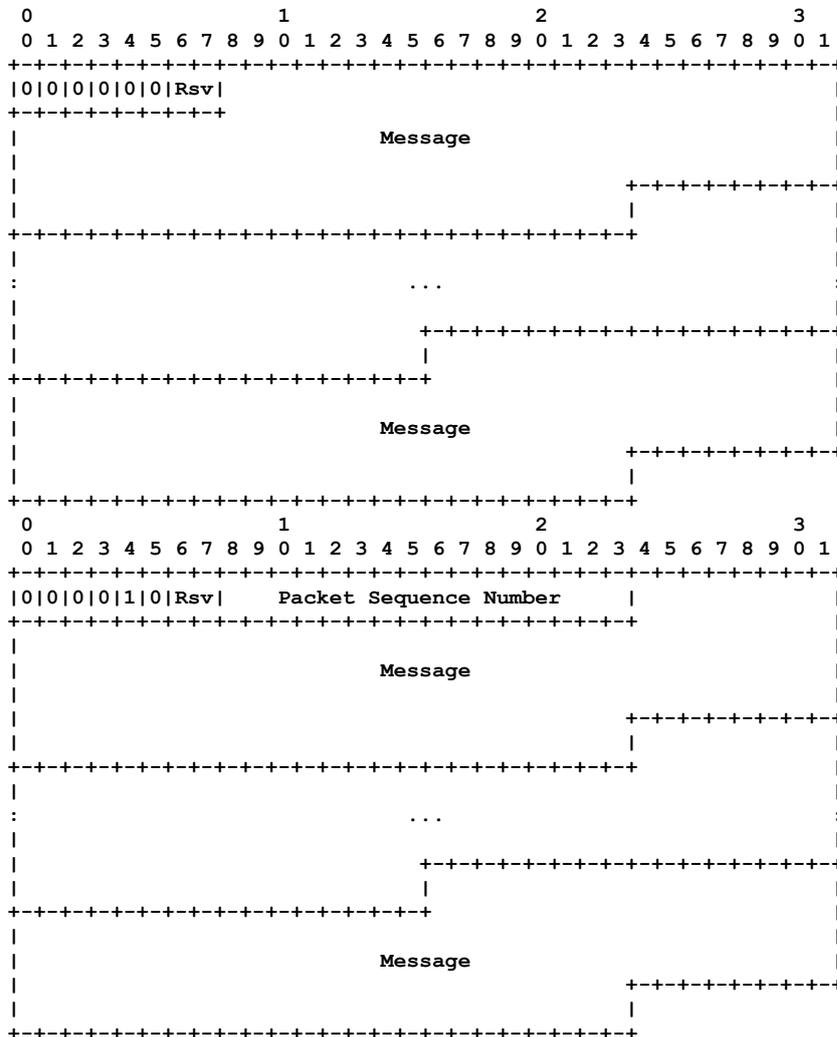
Приложение D. Иллюстрации

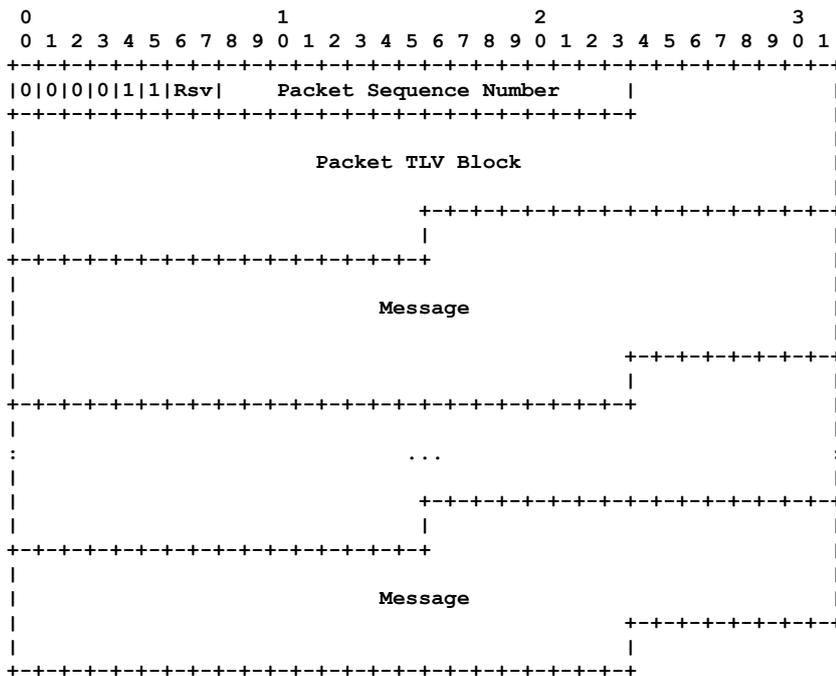
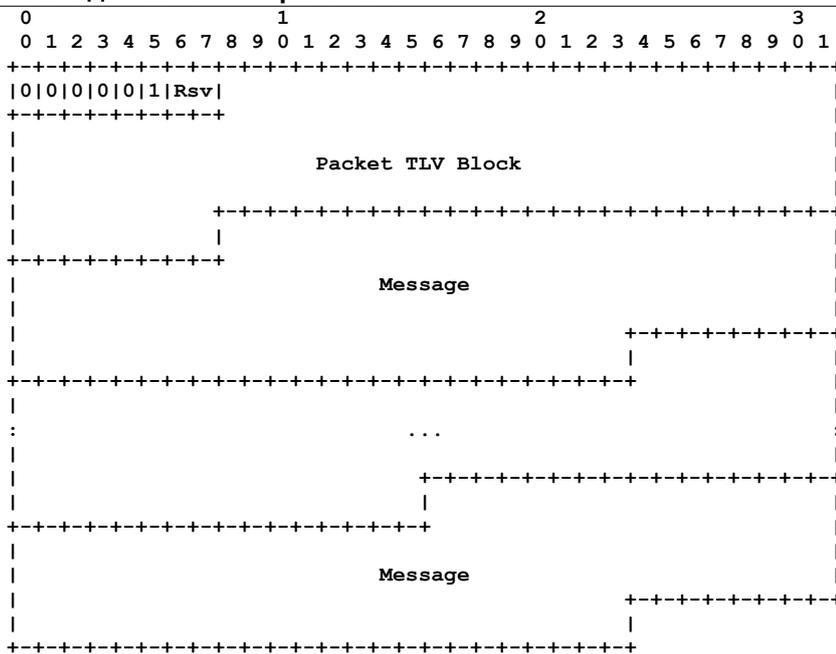
В этом информационном приложении показаны элементы, нормативная спецификация которых дана в разделе 5.

Резервные (Rsv0 следует сбрасывать (0), биты, помеченные M, могут быть сброшены (0) или установлены (1).

D.1. Пакет

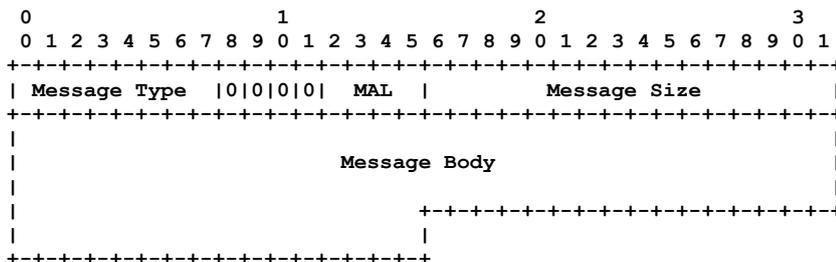
В этом параграфе показаны возможные варианты элемента <packet>, различающиеся полем флагов в первом октете. Пакет может включать любой (0 и более) число сообщений, которое определяется размером пакетов сетевого уровня.





D.2. Сообщение

В этом параграфе показаны возможные варианты элементов <message>, которые различаются октетами флагов (октет 2). Размер Message Body определяется с помощью поля Message Size, в котором указывается суммарный размер всех показанных полей. Поле MAL определяет размер всех адресов (включая Originator Address при его наличии и все адреса в Address Block) в октетах, который на единицу больше значения этого поля.



```

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
+-----+-----+-----+-----+
| Message Type |1|0|0|0| MAL |           Message Size |
+-----+-----+-----+-----+
|                                     Originator Address |
+-----+-----+-----+-----+
|                                     Message Body         |
|                                                         |
|                                                         |
+-----+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
| Message Type |0|1|0|0| MAL |           Message Size |
+-----+-----+-----+-----+
| Hop Limit    |                                     |
+-----+-----+-----+-----+
|                                     Message Body         |
|                                                         |
|                                                         |
+-----+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
| Message Type |1|1|0|0| MAL |           Message Size |
+-----+-----+-----+-----+
|                                     Originator Address |
+-----+-----+-----+-----+
| Hop Limit    |                                     |
+-----+-----+-----+-----+
|                                     Message Body         |
|                                                         |
|                                                         |
+-----+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
| Message Type |0|0|1|0| MAL |           Message Size |
+-----+-----+-----+-----+
| Hop Count    |                                     |
+-----+-----+-----+-----+
|                                     Message Body         |
|                                                         |
|                                                         |
+-----+-----+-----+-----+

```

```

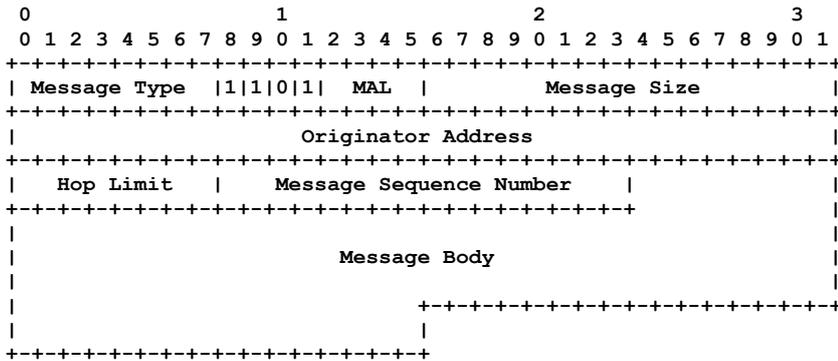
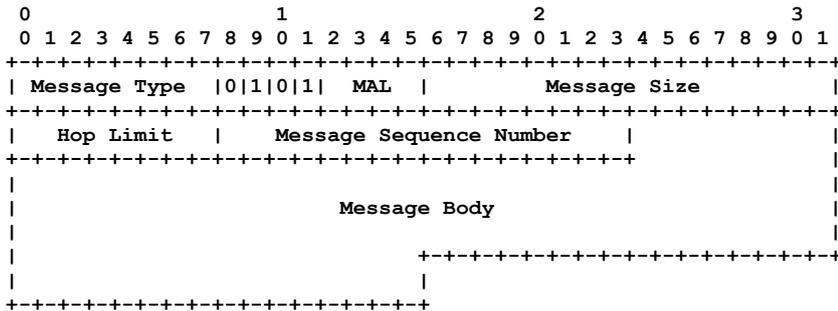
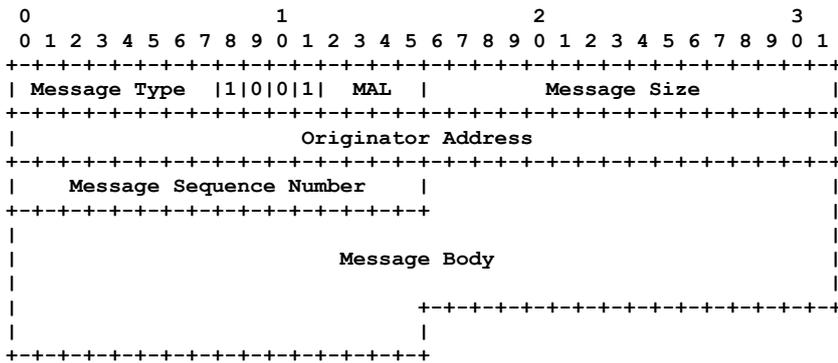
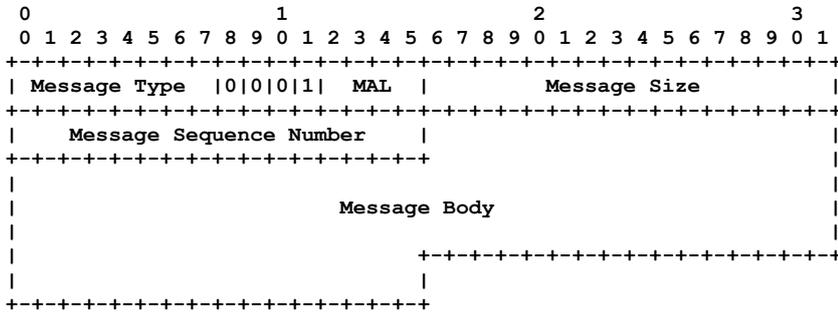
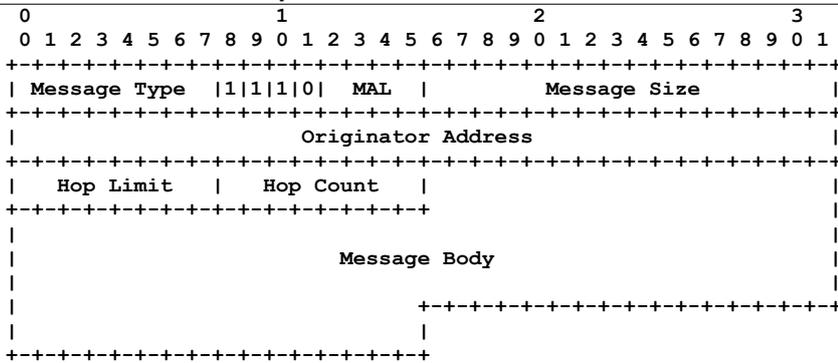
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
| Message Type |1|0|1|0| MAL |           Message Size |
+-----+-----+-----+-----+
|                                     Originator Address |
+-----+-----+-----+-----+
| Hop Count    |                                     |
+-----+-----+-----+-----+
|                                     Message Body         |
|                                                         |
|                                                         |
+-----+-----+-----+-----+

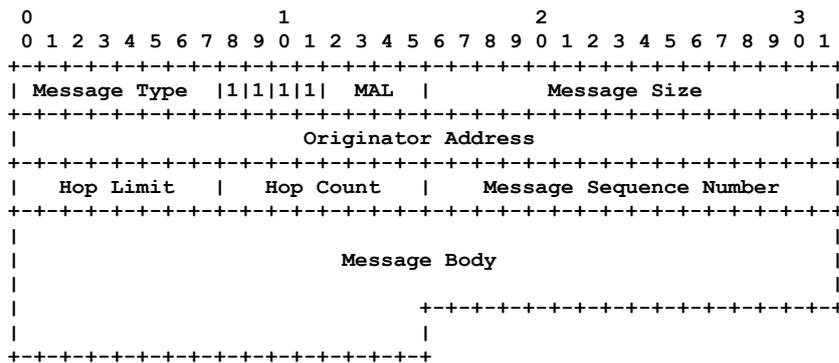
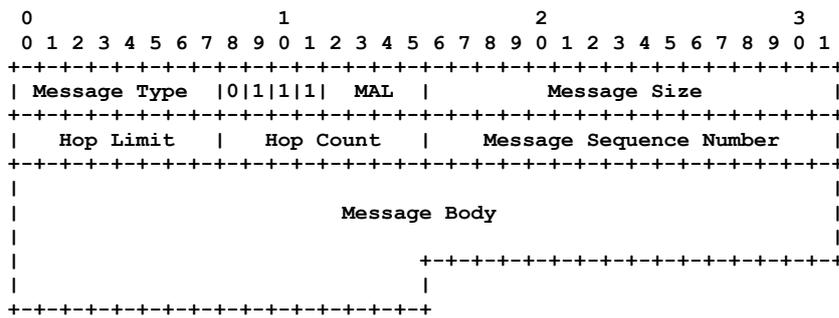
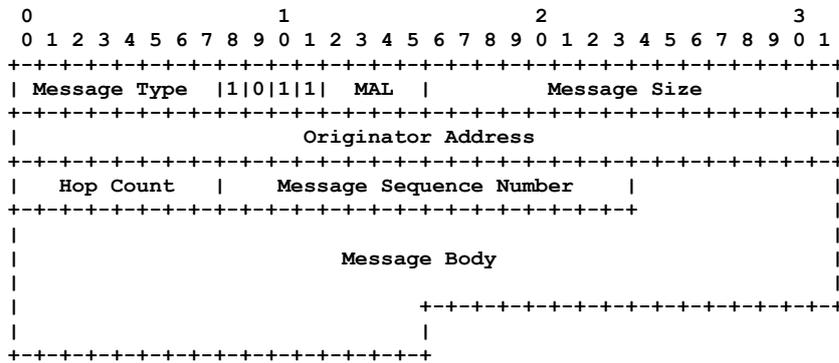
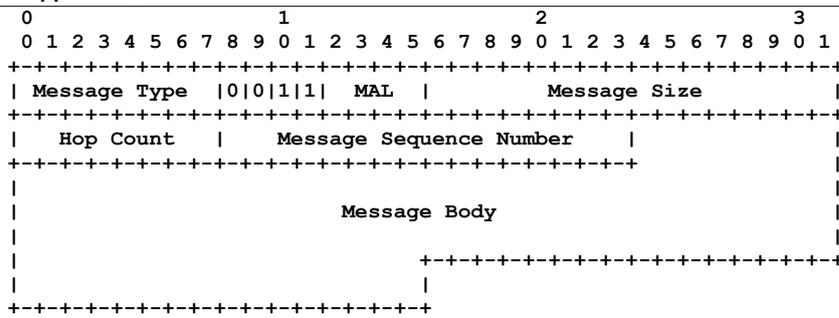
```

```

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
+-----+-----+-----+-----+
| Message Type |0|1|1|0| MAL |           Message Size |
+-----+-----+-----+-----+
| Hop Limit    | Hop Count |                                     |
+-----+-----+-----+-----+
|                                     Message Body         |
|                                                         |
|                                                         |
+-----+-----+-----+-----+

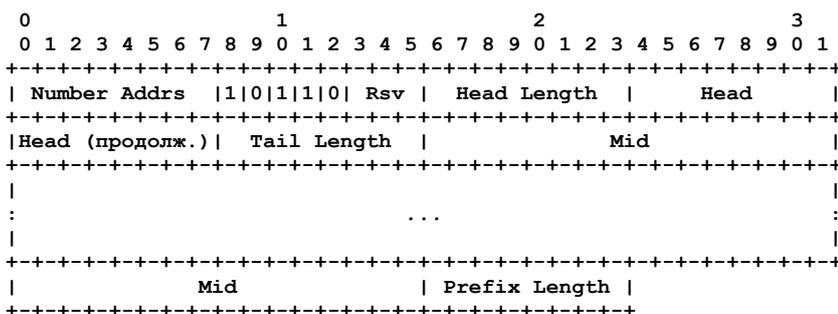
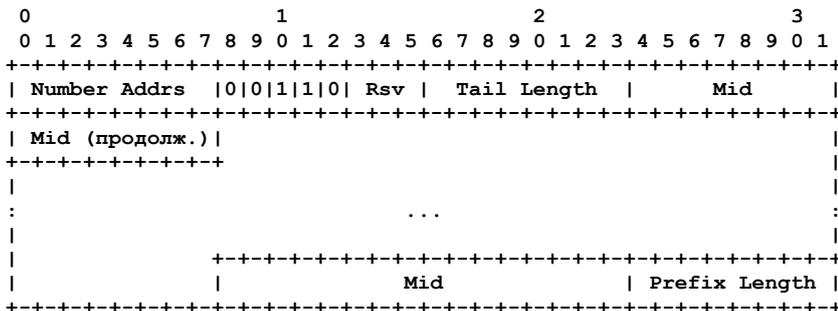
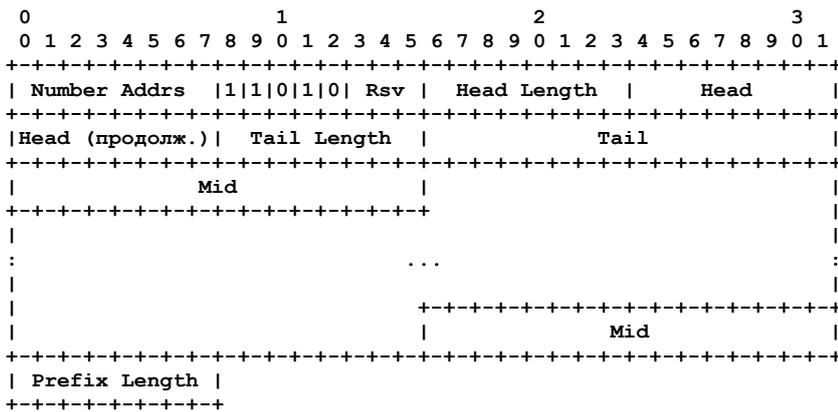
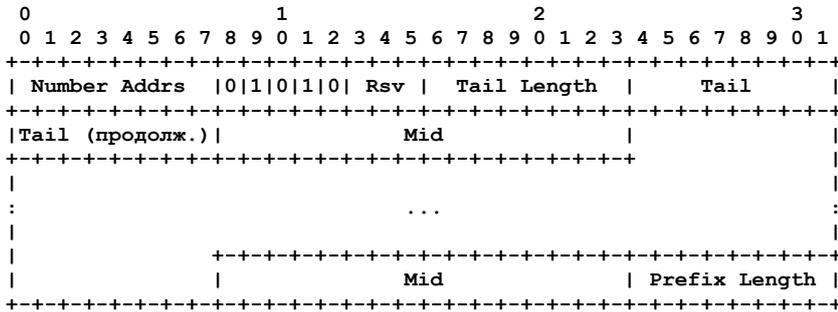
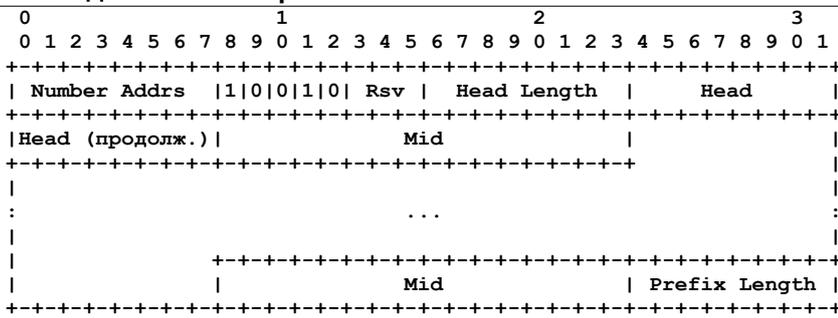
```





D.3. Тело сообщения

В этом параграфе показан формат Message Body (элемент <message> без начального <msg-header>). Message Body включает Message TLV Block (может содержать TLV) и может включать любое число (включая 0) пар Address Block и Address Block TLV Block.




```

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
+-----+-----+-----+-----+
|   Type       |0|1|0|0|0|0|Rsv| Index Start |
+-----+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
|   Type       |1|1|0|0|0|0|Rsv| Type Ext   | Index Start |
+-----+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
|   Type       |0|0|1|0|0|0|Rsv| Index Start | Index Stop  |
+-----+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
|   Type       |1|0|1|0|0|0|Rsv| Type Ext   | Index Start |
+-----+-----+-----+-----+
| Index Stop   |
+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
|   Type       |0|0|0|1|0|M|Rsv| Length   |
+-----+-----+-----+-----+
|
|                               Value
|
|                               +-----+-----+-----+-----+
|                               |
+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
|   Type       |1|0|0|1|0|M|Rsv| Type Ext   | Length   |
+-----+-----+-----+-----+
|
|                               Value
|
|                               +-----+-----+-----+-----+
|                               |
+-----+-----+-----+

```

```

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
+-----+-----+-----+-----+
|   Type       |0|1|0|1|0|0|Rsv| Index Start | Length   |
+-----+-----+-----+-----+
|
|                               Value
|
|                               +-----+-----+-----+-----+
|                               |
+-----+-----+-----+

```

```

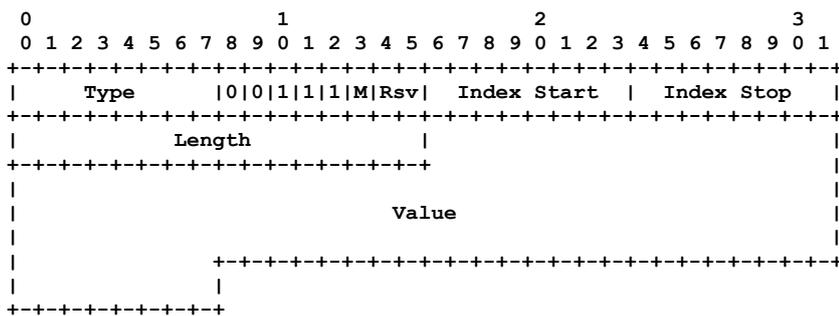
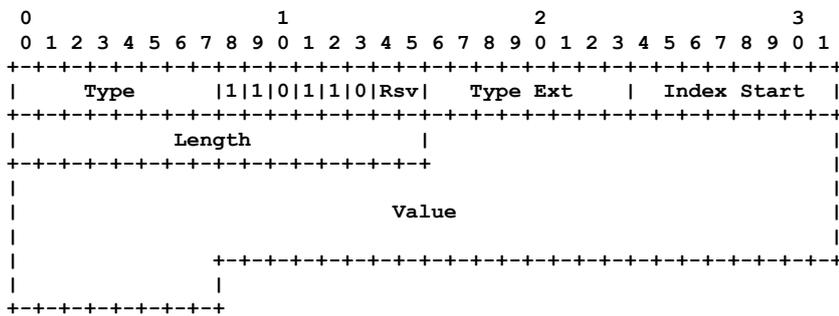
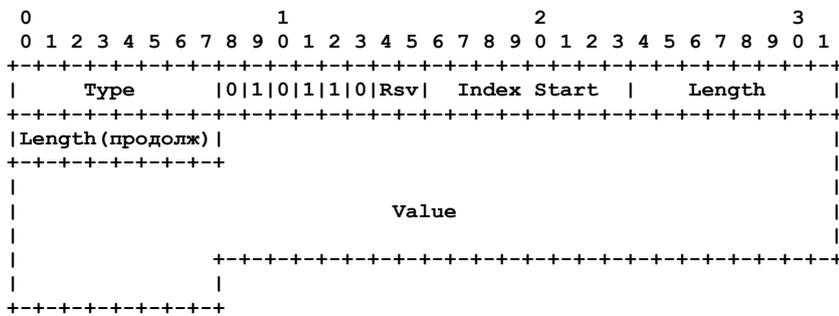
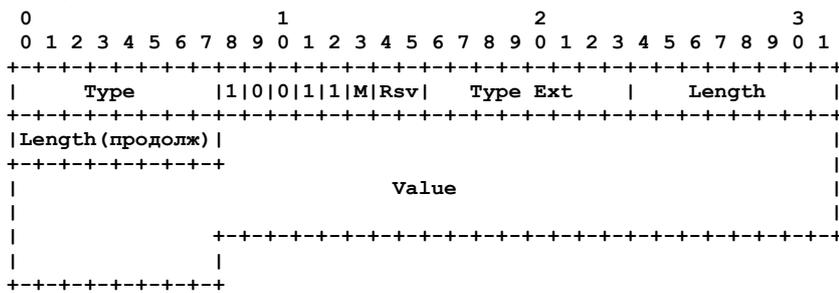
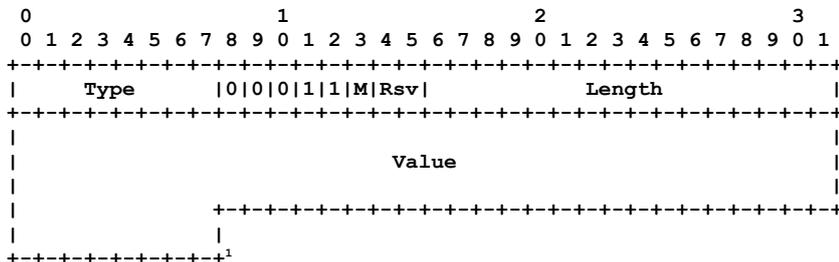
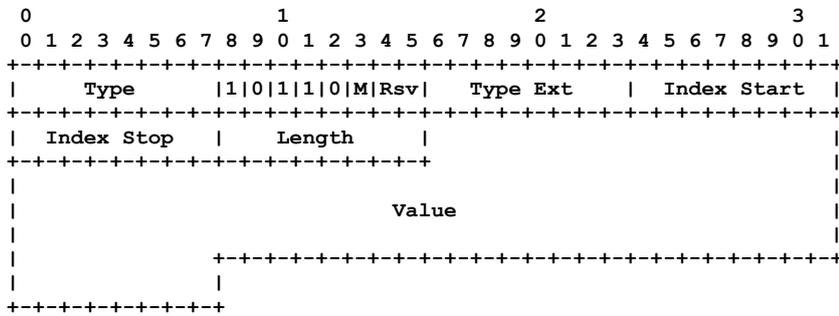
0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|   Type       |1|1|0|1|0|0|Rsv| Type Ext   | Index Start |
+-----+-----+-----+-----+
| Length       |
+-----+-----+-----+
|
|                               Value
|
|                               +-----+-----+-----+-----+
|                               |
+-----+-----+-----+

```

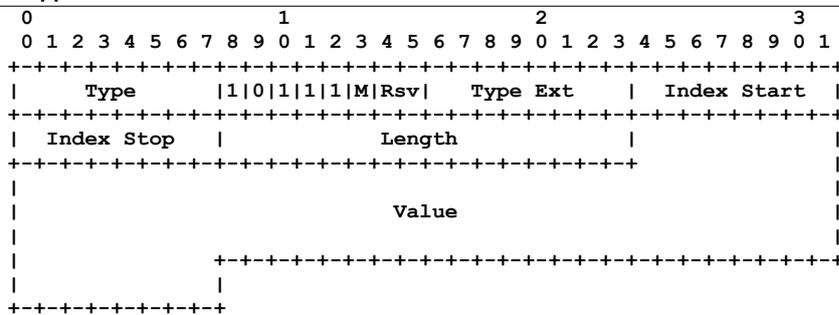
```

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
+-----+-----+-----+-----+
|   Type       |0|0|1|1|0|M|Rsv| Index Start | Index Stop |
+-----+-----+-----+-----+
| Length       |
+-----+-----+-----+
|
|                               Value
|
|                               +-----+-----+-----+-----+
|                               |
+-----+-----+-----+

```



¹В оригинале для бита 11 ошибочно указано значение 0. Прим. перев.



Приложение Е. Полный пример

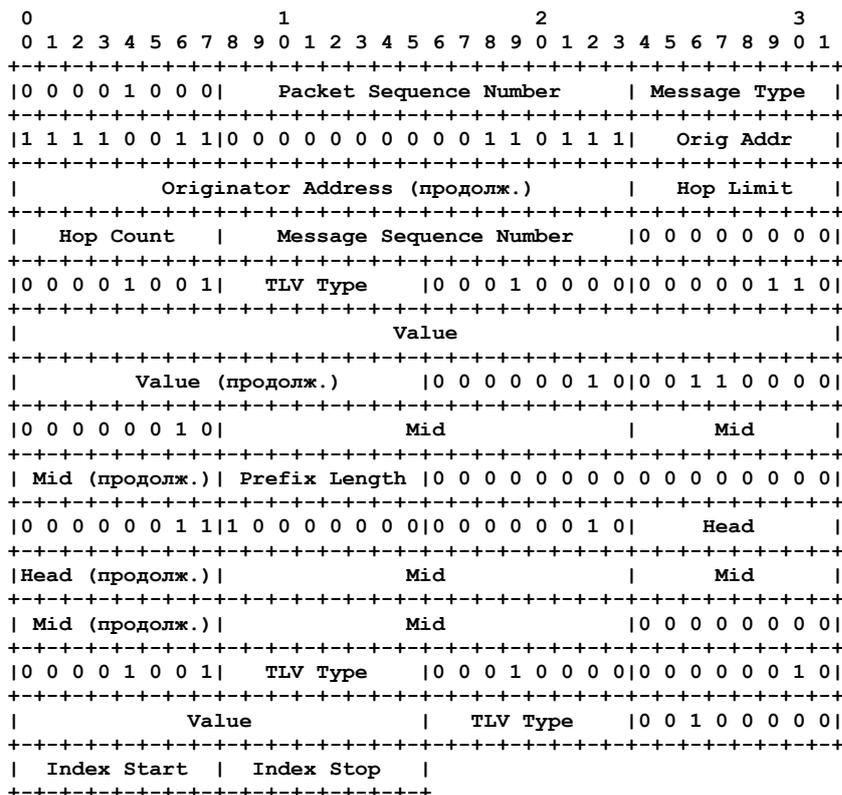
Ниже приведён пример, предназначенный для иллюстрации создания полей Packet Headers и Message Headers, а также представления адресов и атрибутов с помощью Address Block и TLV Block. При создании примера не ставилось цели снижения максимального размера сообщения или размера пакета.

Packet Header имеет флаг phasseqnum в поле флагов (значение 8) и, следовательно, имеет Packet Sequence Number, но не имеет Packet TLV Block.

Пакет содержит одно сообщение размером 55¹ октетов. Это сообщение имеет флаги mhasorig, mhashoplimit, mhashorcount и mhasseqnum в своём 4-битовом поле флагов (значение 15) и, следовательно, включает Originator Address, Hop Limit, Hop Count и Message Sequence Number (не зависит от типа). Четырёхбитовое поле Message Address Length имеет значение 3, следовательно адрес в сообщении имеет размер 4 октета и является адресом IPv4. Сообщение имеет Message TLV Block с размером содержимого в 9 октетов, включающий один Message TLV. В этом TLV установлен флаг thasvalue в октете флагов (значение 16), следовательно имеется поле Value с Value Length = 6. Сообщение имеет два Address Block, за каждым из которых следует блок Address Block TLV.

Первый блок адресов включает два адресных префикса. Он имеет флаги ahaszerotail и ahasingleprelen в октете флагов (значение 48) и, следовательно, не имеет Head (head-length = 0). Поле tail-length указывает 2 октета, следовательно mid-length также указывает 2 октета. Два октета Tail каждого из адресов не указываются (флаг ahaszerotail установлен) и имеют значение 0. Address Block имеет один размер префикса. Следующий блок Address Block TLV пуст (размер содержимого 0).

Второй Address Block включает 3 адреса, имеет флаг ahashead (октет флагов имеет значение 128), имеет head-length в 2 октета, не имеет Tail (tail-length = 0), следовательно mid-length = 2. За ним следует блок Address Block TLV с размером содержимого 9 октетов, включающий два Address Block TLV. В первом из этих TLV установлен флаг (значение октета флагов 16) и присутствует одно значение размеров 2 октета, которое применяется ко всем адресам в Address Block. Во втором TLV установлен флаг thasmultiindex (октет флагов 32) и, следовательно, нет полей Value Length и Value. Присутствуют два поля Index (Index Start и Index Stop), которые показывают адреса, к которым применяется TLV (включительный диапазон с отсчётом от 0).



Адреса авторов

Thomas Heide Clausen

¹В оригинале ошибочно сказано 54 октета. См. https://www.rfc-editor.org/errata_search.php?eid=1790. По этой причине значение бита 23 второго слова (перед Orig Addr) на рисунке ниже заменено при переводе на 1. Прим. перев.

LIX, Ecole Polytechnique

Phone: +33 6 6058 9349

E-Mail: T.Clausen@computer.org

URI: <http://www.thomasclausen.org/>

Christopher M. Dearlove

BAE Systems ATC

Phone: +44 1245 242194

E-Mail: chris.dearlove@baesystems.com

URI: <http://www.baesystems.com/>

Justin W. Dean

Naval Research Laboratory

Phone: +1 202 767 3397

E-Mail: jdean@itd.nrl.navy.mil

URI: <http://pf.itd.nrl.navy.mil/>

Cedric Adjih

INRIA Rocquencourt

Phone: +33 1 3963 5215

E-Mail: Cedric.Adjih@inria.fr

URI: <http://menetou.inria.fr/~adjih/>

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru