

Internet Engineering Task Force (IETF)  
Request for Comments: 7181  
Category: Standards Track  
ISSN: 2070-1721

T. Clausen  
LIX, Ecole Polytechnique  
C. Dearlove  
BAE Systems ATC  
P. Jacquet  
Alcatel-Lucent Bell Labs  
U. Herberg  
Fujitsu Laboratories of America  
April 2014

## Протокол маршрутизации OLSRv2

### The Optimized Link State Routing Protocol Version 2

#### Аннотация

Эта спецификация описывает версию 2 протокола OLSRv2<sup>1</sup> для сетей MANET<sup>2</sup>.

#### Статус документа

Этот документ является проектом стандарта Internet (Internet Standards Track).

Документ является результатом работы IETF<sup>3</sup> и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG<sup>4</sup>. Дополнительную информацию о стандартах Internet можно найти в разделе 2 документа RFC 5741.

Информация о текущем статусе данного документа, обнаруженных ошибках и способах обратной связи приведена на странице <http://www.rfc-editor.org/info/rfc7181>.

#### Авторские права

Авторские права (Copyright (c) 2014) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные 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. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Документ может содержать материалы из IETF Document или IETF Contribution, опубликованных или публично доступных до 10 ноября 2008 года. Лица, контролирующие авторские права на некоторые из таких документов, могли не предоставить IETF Trust права разрешать внесение изменений в такие документы за рамками процессов IETF Standards. Без получения соответствующего разрешения от лиц, контролирующих авторские права этот документ не может быть изменён вне рамок процесса IETF Standards, не могут также создаваться производные документы за рамками процесса IETF Standards за исключением форматирования документа для публикации или перевода с английского языка на другие языки.

## Оглавление

1. Введение.....	3
2. Терминология.....	4
3. Заявление о применимости.....	5
4. Обзор и функциональность протокола.....	5
4.1. Обзор.....	6
4.2. Маршрутизаторы и интерфейсы.....	7
4.3. Обзор информационных баз.....	7
4.3.1. Локальная информационная база.....	7
4.3.2. База информации об интерфейсах.....	7
4.3.3. База информации о соседях.....	7
4.3.4. База топологической информации.....	8
4.3.5. База информации о полученных сообщениях.....	8
4.4. Обзор сигнализации.....	8
4.5. Метрика каналов.....	9
4.6. Flooding MPR и Routing MPR.....	9
4.7. Использование Routing Set.....	9
5. Протокольные параметры и константы.....	10
5.1. Номера протокола и портов.....	10

<sup>1</sup>Optimized Link State Routing Protocol - оптимизированный протокол маршрутизации по состоянию каналов.

<sup>2</sup>Mobile Ad Hoc Network.

<sup>3</sup>Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

<sup>4</sup>Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

5.2. Групповой адрес.....	10
5.3. Параметр интерфейса.....	10
5.3.1. Срок корректности полученного сообщения.....	10
5.4. Параметры маршрутизатора.....	10
5.4.1. Время сохранения изменившейся локальной информации.....	10
5.4.2. Параметр метрики каналов.....	10
5.4.3. Интервалы между сообщениями.....	10
5.4.4. Анонсированные интервалы достоверности информации.....	11
5.4.5. Обработка и пересылка времени достоверности.....	11
5.4.6. Вариации задержек.....	11
5.4.7. Ограничение числа интервалов.....	11
5.4.8. Готовность.....	12
5.5. Ограничения на смену параметров.....	12
5.6. Константы.....	13
5.6.1. Константы метрики каналов.....	13
5.6.2. Константы готовности.....	13
5.6.3. Временная константа.....	13
6. Значения метрики каналов.....	13
6.1. Представление метрики канала.....	13
6.2. Сжатая форма метрики канала.....	14
7. Локальная информационная база.....	14
7.1. Originator Set.....	14
7.2. Local Attached Network Set.....	14
8. База информации об интерфейсах.....	15
8.1. Link Set.....	15
8.2. 2-Hop Set.....	15
9. База информации о соседях.....	15
10. База топологической информации.....	16
10.1. Advertising Remote Router Set.....	16
10.2. Router Topology Set.....	16
10.3. Routable Address Topology Set.....	16
10.4. Attached Network Set.....	17
10.5. Routing Set.....	17
11. База информации о полученных сообщениях.....	17
11.1. Received Set.....	17
11.2. Processed Set.....	17
11.3. Forwarded Set.....	18
12. Свойства информационных баз.....	18
12.1. Соответствие списков Protocol Tuple.....	18
12.2. Принадлежность адресов.....	18
13. Пакеты и сообщения.....	18
13.1. Сообщения.....	19
13.2. Пакеты.....	19
13.3. TLV.....	19
13.3.1. Message TLV.....	19
13.3.2. Address Block TLV.....	19
14. Обработка и пересылка сообщений.....	20
14.1. Действия при получении сообщения.....	20
14.2. Рассмотрение вопроса обработки сообщения.....	20
14.3. Рассмотрение вопроса пересылки сообщения.....	21
15. Сообщения HELLO.....	22
15.1. Генерация сообщения HELLO.....	22
15.2. Передача сообщения HELLO.....	22
15.3. Обработка сообщения HELLO.....	23
15.3.1. Отбрасывание сообщений HELLO.....	23
15.3.2. Использование сообщений HELLO.....	23
15.3.2.1. Обновление метрики.....	23
15.3.2.2. Обновление Willingness.....	24
15.3.2.3. Обновление состояний селекторов MPR.....	24
16. Сообщения TC.....	25
16.1. Генерация сообщений TC.....	25
16.2. Передача сообщений TC.....	25
16.3. Обработка сообщений TC.....	26
16.3.1. Отбрасывание сообщений TC.....	26
16.3.2. Определения для обработки сообщений TC.....	27
16.3.3. Первоначальная обработка сообщения TC.....	27
16.3.3.1. Заполнение Advertising Remote Router Set.....	27
16.3.3.2. Заполнение Router Topology Set.....	27
16.3.3.3. Заполнение Routable Address Topology Set.....	28
16.3.3.4. Заполнение Attached Network Set.....	28
16.3.4. Финальная обработка сообщения TC.....	29
16.3.4.1. Очистка Router Topology Set.....	29
16.3.4.2. Очистка Routable Address Topology Set.....	29
16.3.4.3. Очистка Attached Network Set.....	29
17. Изменения информационных баз.....	29
17.1. Изменение адреса инициатора.....	29
17.2. Изменения состояний каналов.....	29

17.3. Изменения состояний соседей.....	29
17.4. Изменение анонсируемых соседей.....	30
17.5. Завершение срока действия Advertising Remote Router.....	30
17.6. Изменения соседства и обновления MPR.....	30
17.7. Обновления Routing Set.....	31
18. Выбор MPR.....	31
18.1. Обзор.....	32
18.2. Граф соседей.....	32
18.3. Свойства MPR.....	32
18.4. Рассылающие MPR.....	33
18.5. Маршрутизирующие MPR.....	34
18.6. Рассчитывающий MPR.....	34
19. Расчёт Routing Set.....	34
19.1. Граф сетевой топологии.....	34
19.2. Заполнение Routing Set.....	35
20. Предлагаемые значения параметров.....	36
20.1. Время сохранения локальной истории.....	36
20.2. Интервалы между сообщениями.....	36
20.3. Параметры срока достоверности анонсированной информации.....	36
20.4. Параметры срока достоверности принятых сообщений.....	36
20.5. Параметры вариаций.....	36
20.6. Параметр Hop Limit.....	36
20.7. Параметры готовности.....	36
21. Порядковые номера.....	36
22. Расширения.....	37
23. Вопросы безопасности.....	37
23.1. Архитектура защиты.....	37
23.2. Целостность.....	37
23.3. Конфиденциальность.....	38
23.4. Взаимодействие с внешними доменами маршрутизации.....	38
23.5. Обязательные механизмы защиты.....	38
23.6. Управление ключами.....	38
24. Взаимодействие с IANA.....	39
24.1. Expert Review - рекомендации по оценке.....	40
24.2. Типы сообщений.....	40
24.3. Реестры специфических для типа сообщений типов TLV.....	40
24.4. Типы Message TLV.....	40
24.5. Типы Address Block TLV.....	40
24.6. Значения NBR_ADDR_TYPE и MPR.....	41
25. Разработчики документа.....	41
26. Благодарности.....	41
27. Литература.....	41
27.1. Нормативные документы.....	41
27.2. Дополнительная литература.....	42
Приложение А. Ограничения.....	42
Приложение В. Пример алгоритма для расчёта MPR.....	44
В.1. Дополнительные обозначения.....	44
В.2. Алгоритм выбора MPR.....	44
Приложение С. Пример алгоритма для расчёта Routing Set.....	45
С.1. Локальные интерфейсы и соседи.....	45
С.2. Добавление соседних маршрутизаторов.....	45
С.3. Добавление удалённых маршрутизаторов.....	45
С.4. Добавление адресов соседей.....	46
С.5. Добавление удалённых маршрутизируемых адресов.....	46
С.6. Добавление подключённых сетей.....	46
С.7. Добавление соседей 2-Нор.....	47
Приложение D. Пример сообщения TC.....	47
Приложение E. Управление потоком и контроль перегрузок.....	48

## 1. Введение

Протокол OLSRv2<sup>1</sup> является развитием протокола OLSR (версии 1), опубликованного в [RFC3626]. OLSRv2 при выборе кратчайшего пути использует те же механизмы и алгоритмы, что и [RFC3626], но усовершенствованные за счёт использования метрики, отличающейся от числа интервалов. В OLSRv2 также применяется более гибкая и эффективная модель сигнализации, а также внесено некоторое упрощение в процесс обмена сообщениями.

Протокол OLSRv2 разработан для сетей MANET<sup>2</sup>. Протокол работает на основе регулярного обмена топологической информацией между маршрутизаторами. OLSRv2 представляет собой оптимизацию классического протокола маршрутизации по состояниям каналов. Ключевым элементом концепции являются многоточечные трансляторы (MPR<sup>3</sup>). Каждый маршрутизатор выбирает два набора MPR, каждый из которых является множеством соседних маршрутизаторов, «покрывающим» все симметрично подключённые маршрутизаторы «второго интервала» (2-hop neighbor router). Наборы MPR называются «рассылающими MPR» (flooding MPR) и «маршрутизирующими MPR» (routing MPR) и применяются для снижения уровня лавинных рассылок и упрощения топологии, соответственно.

Снижение объёма лавинных рассылок обеспечивается путём контроля трафика, который будет рассылаться лавинообразно через сеть с использованием поэтапной (hop-by-hop) пересылки, при этом маршрутизатору нужно

<sup>1</sup>Optimized Link State Routing Protocol version 2.

<sup>2</sup>Mobile Ad Hoc Network - специализированная мобильная сеть.

<sup>3</sup>Multipoint relay.

пересылать лишь управляющий трафик, который был впервые получен напрямую от одного из маршрутизаторов, которые выбраны в качестве рассылающих MPR (flooding MPR selectors). Этот механизм, названный MPR flooding, обеспечивает эффективный способ распространения информации внутри сетей MANET за счёт снижения числа требуемых передач [MPR].

Упрощение топологии обеспечивается путём назначения особой ответственности для маршрутизаторов, выбранных в качестве routing MPR, при декларировании информации о состоянии каналов. Для OLSRv2 достаточным требованием для обеспечения кратчайших маршрутов ко всем получателям является декларирование информации о состоянии каналов маршрутизаторам из своего routing MPR (если они есть). Маршрутизаторам, не выбранным в routing MPR, не нужно передавать данные о состоянии каналов. На основе этой сокращённой информации о состоянии канала routing MPR используются в качестве промежуточных маршрутизаторов на многоэтапных (multi-hop) маршрутах.

Таким образом, использование MPR позволяет снизить число и размер сообщения о состоянии каналов, а также снижение объёма информации о состоянии каналов, поддерживаемой каждым маршрутизатором. При наличии возможности (в частности, при использовании числа интервалов в качестве метрики) одни и те же маршрутизаторы могут быть указаны в качестве рассылающих и маршрутизирующих MPR.

Маршрутизатор выбирает рассылающие и маршрутизирующие MPR из числа ближайших (one-hop) соседей, подключённых по симметричным (т. е. двунаправленным) каналам. Следовательно, выбор маршрутов с помощью маршрутизирующих MPR позволяет избежать проблем с передачей пакетов данных по односторонним каналам (например, проблема неполучения подтверждений канального уровня на каждом интервале для канальных уровней, поддерживающих это).

OLSRv2 использует и расширяет протокол NHDP<sup>1</sup>, определённый в [RFC6130], и также обобщенный формат пакетов и сообщений MANET [RFC5444] с описанными в [RFC5497] TLV, а также может использовать вариации message jitter, как описано в [RFC5148]. Все эти 4 протокола и спецификации создавались в рамках разработки OLSRv2, но были выделены в отдельные документы для более широкого применения.

OLSRv2 не включает предположение о природе канального уровня. Благодаря использованию [RFC6130], OLSRv2, может применять информацию и уведомления канального уровня, когда это доступно. В дополнение к этому OLSRv2 использует метрику каналов, создаваемую на основе данных канального уровня и другой информации. OLSRv2 не задаёт физический смысл метрики каналов, но определяет способы задания новых вариантов метрики в будущем для использования их протоколом OLSRv2 без его изменения.

OLSRv2, подобно OLSR [RFC3626], использует концепции пересылки и трансляции HIPERLAN<sup>2</sup> (протокол уровня MAC), стандартизованного ETSI [HIPERLAN] [HIPERLAN2]. Данный документ не отменяет действия [RFC3626], который сохраняется для дальнейших экспериментов.

## 2. 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].

Все термины, введённые в [RFC5444], включая packet, Packet Header, message, Message Header, Message Body, Message Type, message sequence number, hop limit, hop count, Address Block, TLV Block, TLV, Message TLV, Address Block TLV, type (для TLV), type extension (для TLV), value (для TLV), address, address prefix, address object, интерпретируются в соответствии с этим документом.

Все термины, введённые в [RFC6130], включая interface, MANET interface, network address, link, symmetric link, symmetric 1-hop neighbor, symmetric 2-hop neighbor, symmetric 1-hop neighborhood constant, interface parameter, router parameter, Information Base, HELLO message, интерпретируются в соответствии с этим документом.

В дополнение к этому данная спецификация использует описанные ниже термины.

### **Router - маршрутизатор**

Маршрутизатор MANET, реализующий данный протокол.

### **OLSRv2 interface - интерфейс OLSRv2**

Интерфейс MANET, на котором работает данный протокол. Маршрутизатор, на котором используется данный протокол, **должен** иметь хотя бы один интерфейс OLSRv2.

### **Routable address - маршрутизируемый адрес**

Сетевой адрес, который может использоваться в качестве адреса назначения для пакета данных. Поддерживающий этот протокол маршрутизатор должен отличать маршрутизируемые адреса от немаршрутизируемых путём из прямой проверки на основе данных IANA о глобальном распределении адресов и/или своей конфигурации (в части маршрутизации через сеть MANET). Широковещательные и групповые адреса, а также адреса с ограниченной областью действия, которая меньше сети MANET в целом, **недопустимо** считать маршрутизируемыми. Anycast-адреса могут рассматриваться в качестве маршрутизируемых.

### **Originator address - адрес инициатора**

Адрес, являющийся уникальным (в рамках MANET) для маршрутизатора. Маршрутизатор **должен** выбрать originator address и **может** использовать в этом качестве адрес одного из своих интерфейсов, который **может** быть как маршрутизируемым, так и немаршрутизируемым. **Недопустимо** использовать в качестве адреса инициатора широковещательные, групповые и anycast-адреса. Если маршрутизатор выбирает маршрутизируемый адрес, это **должен** быть адрес, воспринимаемый маршрутизатором в качестве адреса получателя. Для адреса инициатора **недопустимо** использовать размер префикса за исключением случаев, когда он включён в Address Block, где может быть связан с префиксом максимального размера (например, если адрес инициатора представляет собой адрес IPv6, он **должен** иметь размер префикса 128 или не иметь его совсем).

### **Message originator address - адрес инициатора сообщения**

<sup>1</sup>MANET Neighborhood Discovery Protocol - протокол обнаружения соседей MANET.

<sup>2</sup>High Performance Radio Local Area Network - высокопроизводительные локальные радиосети.

Адрес инициатора для маршрутизатора, создавшего сообщение, извлекаемый получателем этого сообщения. Для всех сообщений, используемых в данной спецификации, а также сообщений HELLO, определённых в [RFC6130], получатель **должен** быть способен выделить адрес инициатора. Обычно адрес инициатора сообщения включается в сообщение, как элемент <msg-orig-addr> в соответствии с определением [RFC5444]. Однако маршрутизаторы с единственным адресом могут не включать элемент <msg-orig-addr> в свои сообщения HELLO.

#### **Willingness - готовность**

Численное значение от WILL\_NEVER до WILL\_ALWAYS (включительно), которое представляет готовность маршрутизатора быть выбранным в качестве MPR. Маршрутизатор имеет разные значения готовности быть рассылающим и маршрутизирующим MPR.

#### **Willing symmetric 1-hop neighbor - готовый ближайший сосед с симметричным соединением**

Ближайший сосед с симметричным соединением и отличным от WILL\_NEVER значением готовности.

#### **Multipoint relay (MPR) - многоточечный транслятор**

Маршрутизатор X является MPR для маршрутизатора Y, если Y указал выбор маршрутизатора X в качестве MPR в своём недавнем сообщении HELLO. Маршрутизатор X может быть рассылающим MPR для Y, если он указан в качестве участника рассылки сообщений, полученных от Y, или маршрутизирующим MPR для Y, если он указан в качестве объявляющего информацию о состоянии канала от X до Y. Возможно также совмещение этих ролей.

#### **MPR selector - селектор MPR**

Маршрутизатор Y является селектором (рассылающего/маршрутизирующего) MPR для маршрутизатора X, если Y выбрал маршрутизатор X в качестве рассылающего/маршрутизирующего MPR.

#### **MPR flooding - лавинная рассылка MPR**

Оптимизированный механизм распространения информации масштаба сети MANET, реализуемый данным протоколом. Сообщения в этом случае транслируются лишь ограниченным набором маршрутизаторов в сети. По сути, MPR flooding представляет собой механизм снижения объёма лавинных рассылок.

#### **EXPIRED - время истекло**

Указывает, что для таймера установлено значения, явно предшествующее текущему времени (например, текущее время - 1).

В данной спецификации используются обозначения, принятые в [RFC5444] и [RFC6130].

### **3. Заявление о применимости**

Этот документ является спецификацией OLSRv2 - проактивного протокола, предназначенного для использования в сетях MANET [RFC2501]. Применимость протокола определяется его характеристиками, перечисленными ниже.

- Протокол разработан для применения в сетях с изменяющейся топологией и возможностью потери сообщений в результате конфликтов при доступе к беспроводной среде передачи.
- Поддерживаются маршрутизаторы, в которых один или множество интерфейсов принимают участие в работе OLSRv2 и все или некоторые из интерфейсов MANET используют протокол [RFC6130]. Множества интерфейсов OLSRv2 с маршрутизаторе, а также его множества интерфейсов MANET и иных типов могут меняться с течением времени. Каждый интерфейс может иметь один или множество сетевых адресов (у некоторых может присутствовать размер префикса), которые могут динамически меняться.
- Поддерживается поэтапная (hop-by-hop) маршрутизация - каждый маршрутизатор может использовать свою локальную информацию, обеспечиваемую данным протоколом, для маршрутизации пакетов.
- Постоянная поддержка маршрутов ко всем получателям в сети - все маршруты доступны в каждый момент времени и трафик передаётся без задержек на обнаружение маршрутов. Следовательно, буферизации трафика данных не требуется.
- Поддерживаются маршрутизаторы, имеющие интерфейсы не-OLSRv2, которые могут быть локальными для этого маршрутизатора или использоваться в качестве шлюзов в направлении других сетей.
- Поддерживается использование двухсторонней аддитивной метрики для нахождения кратчайших маршрутов (т. е., маршрутов с минимальной суммарной метрикой каналов). Входящие значения метрики определяются процессом, не рассматриваемым данной спецификацией.
- Протокол оптимизирован для больших сетей с высокой плотностью. Чем больше и плотнее сеть, тем сильнее она может быть оптимизирована за счёт использования MPR по сравнению с классическими алгоритмами маршрутизации по состоянию каналов [MPR].
- Используется [RFC5444], как указано в Приложении «Предусмотренное применение» к этому документу, и [RFC5498].
- Поддерживается «внутренняя» и «внешняя» расширяемость (добавление новых типов сообщений и добавление информации в имеющиеся сообщения), разрешённая [RFC5444].
- Протокол разработан для распределенной работы и не включает каких-либо центральных элементов.

### **4. Обзор и функциональность протокола**

Целью данного протокола является обеспечение каждому маршрутизатору возможности:

- идентифицировать всех получателей в сети;
- найти подмножество каналов в сети, достаточное для расчёта кратчайших путей ко всем адресатам;
- обеспечить множество Routing Set, содержащее кратчайшие пути от него до всех адресатов (маршрутизируемые адреса и локальные соединения).

#### **4.1. Обзор**

Означенные выше цели достигаются для каждого маршрутизатора с использованием перечисленных ниже мер.

- Использование протокола NHDP [RFC6130] для идентификации ближайших соседей с симметричным соединением и соседей второго уровня с симметричным соединением.
- Информирование об участии в работе OLSRv2 и готовности выступить в качестве рассылающего или маршрутизирующего MPR с использованием расширения сообщений HELLO, определённых в [RFC6130], путём добавления MPR\_WILLING Message TLV. Готовность маршрутизатора к рассылке показывает, что он может выступить в качестве рассылающего MPR. Готовность к маршрутизации показывает, что маршрутизатор готов выступить в качестве промежуточного устройства в системе маршрутизации. Отметим, что маршрутизатор может быть источником и получателем в системе маршрутизации, даже если он не указал готовности выполнять функции рассылающего или маршрутизирующего MPR.
- Расширение сообщений HELLO, определённых в [RFC6130], для поддержки добавления направленной метрики каналов для анонсирования другим маршрутизаторам, участвующим в работе OLSRv2, и для индикации типа канальной метрики, который будет использоваться этими маршрутизаторами. Передаваться может как входная, так и выходная метрика канала (последняя определяется анонсирующим маршрутизатором).
- Выбор рассылающих и маршрутизирующих MPR из числа выразивших готовность ближайших соседей с симметричным соединением так, чтобы для каждого набора MPR все соседи второго уровня с симметричным подключением были доступны напрямую или хотя бы через один маршрутизирующий MPR из числа выбранных с использованием пути с подходящей минимальной метрикой по выбору крайней мере для одного маршрутизирующего MPR. Анализ и примеры алгоритмов выбора MPR приведены в [MPR], предложенный алгоритм, подобающим образом адаптированный для каждого набора MPR, рассмотрен в приложении В к данной спецификации. Отметим, что маршрутизаторы не обязаны использовать один и тот же алгоритм для обеспечения взаимодействия в рамках одной сети MANET, но каждый из таких алгоритмов должен иметь определённые свойства, описанные в разделе 18.
- Информация о своём выборе рассылающих и маршрутизирующих MPR в расширенных (путём добавления MPR Address Block TLV с подходящими сетевыми адресами) сообщениях HELLO, определённых в [RFC6130].
- Извлечение информации о выборе рассылающих и маршрутизирующих MPR из MPR Address Block TLV в полученных сообщениях HELLO.
- Определение сообщений типа TC<sup>1</sup>, использующих формат из [RFC5444]. Сообщения TC служат для периодической сигнализации между маршрутизатором и выбравшими его в качестве маршрутизирующего MPR через сеть MANET. Сигнализация включает подходящую направленную метрику для соседей (лучшая канальная метрика в данном направлении для этого соседа).
- Возможность включать сообщений TC и HELLO в пакеты, определённые [RFC5444], с использованием номера протокола IP и порта UDP, заданных для MANET в [RFC5498].
- Распространение (диффузия) сообщений TC с использованием механизма ограниченной лавинной рассылки (MPR flooding), при которой пересылку сообщений от маршрутизатора (созданных или пересланных им) осуществляют только выбранные им рассылающие MPR.

Отметим, что указанные расширения [RFC6130] имеют формы, разрешаемые этой спецификацией.

Данная спецификация определяет перечисленные ниже аспекты.

- Требование использовать спецификацию [RFC6130] с параметрами, константами, сообщениями HELLO и информационными базами, которые расширяются данной спецификацией.
- Добавлены две информационные базы - топологическая информация (Topology Information Base) и полученные сообщения (Received Message Information Base).
- Сообщения TC, которые используются для сигнализации в масштабе MANET (с использованием рассылки MPR) о выбранной топологической (состояние каналов) информации.
- Для каждого маршрутизатора вводится требование включения адреса инициатора в сообщения HELLO и TC или возможности определить адрес из таких сообщений.
- Указаны новые Message TLV и Address Block TLV, которые используются в сообщениях HELLO и TC, включая передачу сведений о состоянии соседа, выборе MPR, внешних шлюзах, метрике каналов, готовности быть MPR и порядковых номерах содержимого. Отметим, что генерация значений (входной) метрики каналов выполняется процессом, не включённым в данную спецификацию, которая описывает лишь распространение и использование этой метрики.
- Сообщения TC генерируются из подходящей информации в базах Information Base.
- База топологической информации обновляется в соответствии с полученными сообщениями TC.
- Механизм лавинной рассылки flooding включает добавление адреса инициатора и порядкового номера сообщения для контроля дубликатов с использованием информации из Received Message Information Base.
- Отклики на другие события типа завершения срока действия информации в Information Bases.

Протокол наследует стабильность алгоритмов маршрутизации по состоянию каналов и обеспечивает дополнительные преимущества за счёт поддержки маршрутов, незамедлительно доступных при возникновении необходимости, благодаря его проактивной природе.

Данный протокол взаимодействует с IP только через управление таблицами маршрутизации и указывает IP-адрес передающей стороны в дейтаграммах IP с сообщениями, используемыми данной спецификацией.

<sup>1</sup>Topology Control - контроль топологии.

## 4.2. Маршрутизаторы и интерфейсы

Для того, чтобы маршрутизатор мог принимать участие в работе MANET, используя этот протокол, он должен иметь хотя бы один интерфейс OLSRv2. Свойства интерфейсов OLSRv2 перечислены ниже.

- Это должен быть интерфейс MANET в соответствии с [RFC6130]. В частности, на нем должен быть задан хотя бы один сетевой адрес. Это множество адресов должно быть специфичным для данного маршрутизатора и включать все адреса, которые будут использоваться в качестве адресов отправителей всех пакетов IP, передаваемых с данного интерфейса OLSRv2.
- Интерфейс должен иметь множество параметров в дополнение к заданным в [RFC6130].
- Интерфейс должен иметь Interface Information Base, расширенную по сравнению с [RFC6130].
- Интерфейс должен генерировать и обрабатывать сообщения HELLO в соответствии со спецификацией [RFC6130] и расширениями раздела 15.

В дополнение к описанным интерфейсам OLSRv2 каждый маршрутизатор обладает перечисленными ниже свойствами.

- Может иметь один или множество интерфейсов не-OLSRv2 (которые могут быть или не быть интерфейсами MANET) и/или подключённых локально сетей для которых этот маршрутизатор может воспринимать пакеты IP. Все маршрутизируемые адреса, для которых маршрутизатор воспринимает пакеты IP, должны быть адресами сетевых интерфейсов (OLSRv2 или других) маршрутизатора или адресами локально подключённых сетей.
- Имеет множество параметров в дополнение к заданным в спецификации [RFC6130].
- Имеет базу локальной информации (Local Information Base), расширяющую спецификацию [RFC6130] путём включения выбора адреса инициатора и записей для всех локально подключённых сетей.
- Имеет базу информации о соседях (Neighbor Information Base), расширяющую спецификацию [RFC6130] для записи выбора MPR и анонсов.
- Имеет базу топологической информации (Topology Information Base) с данными из полученных сообщений TC.
- Имеет базу принятых сообщений (Received Message Information Base) с информацией о полученных сообщениях для того, чтобы каждое принятое сообщение TC обрабатывалось только один раз и пересылалось не более одного раза через каждый интерфейс OLSRv2 данного маршрутизатора.
- Генерирует, получает и обрабатывает сообщения TC.

## 4.3. Обзор информационных баз

Каждый маршрутизатор поддерживает информационные базы, описанные в последующих параграфах. Эти базы применяются для описания протокола в данной спецификации. Реализации протокола могут поддерживать информацию в описанной здесь форме или выбрать иной вариант, который будет обеспечивать доступ к данным. В частности, отметим, что не требуется удалять упорядоченные списки (Tuple) из конфигурации (Set) точно в указанное время, достаточно просто вести себя так, будто эти списки удалены в нужный момент.

### 4.3.1. Локальная информационная база

База Local Information Base определена в [RFC6130] и содержит данные о локальной конфигурации маршрутизатора. Данная спецификация расширяет это определение для включения адреса инициатора и перечисленных свойств маршрутизатора:

- конфигурация Originator Set, содержащая адреса, которые были недавно использованы в качестве адресов инициатора для данного маршрутизатора вместе с текущим адресом инициатора для него, чтобы обеспечить маршрутизатору возможность распознавания и отбрасывания управляющего трафика от самого себя;
- конфигурация Local Attached Network Set, содержащая сетевые адреса сетей, для которых данный маршрутизатор может выступать в качестве шлюза (т. е., анонсируемых им в сообщениях TC).

### 4.3.2. База информации об интерфейсах

База Interface Information Base для каждого интерфейса OLSRv2, определённая в [RFC6130], расширена здесь путём записи в каждую конфигурацию Link Set значений канальной метрики (входящей или исходящей) и данных о выборе flooding MPR.

### 4.3.3. База информации о соседях

База Neighbor Information Base определена в [RFC6130] и расширена здесь путём добавления в Neighbor Tuple записи для каждого соседа:

- адрес инициатора;
- значения метрики соседа (минимальные значения канальной метрики в указанном направлении для всех симметричных непосредственных (1-hop) соединений с данным соседом).
- Готовность быть рассылающим и маршрутизирующим MPR;
- выбран ли этот сосед в качестве рассылающего или маршрутизирующего MPR данным маршрутизатором и является ли он селектором routing MPR для данного маршрутизатора (запись о состоянии селектора flooding MPR для данного соседа хранится в Interface Information Base);
- анонсируется ли этот сосед в сообщениях TC, передаваемых данным маршрутизатором.

#### 4.3.4. База топологической информации

База Topology Information Base в каждом маршрутизаторе включает перечисленные ниже данные.

- Конфигурация Advertising Remote Router Set, содержащая каждый удалённый маршрутизатор, от которого было получено сообщение TC. Эти данные используются для проверки актуальности информации в получаемых сообщениях TC - устаревшие сообщения игнорируются.
- Конфигурация Router Topology Set, содержащая данные о каналах между маршрутизаторами в сети MANET из полученных сообщений TC.
- Конфигурация Rutable Address Topology Set, содержащая информацию о маршрутизируемых адресах в сети MANET (IP-адреса получателей пакетов) и удалённых маршрутизаторах, через которые эти маршрутизирующие адреса доступны напрямую (т. е., находятся в одном интервале маршрутизации IP от удалённого маршрутизатора), из полученных сообщений TC.
- Конфигурация Attached Network Set, содержащая информацию о сетях, для которых удалённые маршрутизаторы анонсируют себя в качестве шлюзов. Эти сети могут находиться на удалении в один или множество интервалов от удалённого маршрутизатора.
- Конфигурация Routing Set, содержащая маршруты от данного маршрутизатора ко всем доступным получателям. Таблица маршрутизации IP обновляется с использованием Routing Set (маршрутизатор может выбрать для использования любые или все адреса сетей в Routing Set для обновления своей таблицы маршрутизации IP, механизмы такого выбора выходят за рамки этой спецификации).

Назначение Topology Information Base состоит в записи информации, используемой в дополнение к содержимому Local Information Base, Interface Information Bases и Neighbor Information Base для создания конфигурации Routing Set (которая тоже включается в Topology Information Base).

Данная спецификация описывает расчёт Routing Set на основе графа топологии (Topology Graph), создаваемого в два этапа. Сначала создаётся опорный (backbone) граф, представляющий маршрутизаторы сети MANET и связи между ними, на основе данных из Local Information Base, Neighbor Information Base и Router Topology Set. Далее этот граф «декорируется» дополнительными адресами сетей получателей с использованием данных из Local Information Base, Rutable Address Topology Set и Attached Network Set.

Topology Graph не требуется записывать в Topology Information Base, поскольку он может быть создан при изменении конфигурации Routing Set или возникновении явной необходимости в его создании (см. Приложение С). Однако реализация может создавать и хранить Topology Graph, если считает это нужным.

#### 4.3.5. База информации о полученных сообщениях

Полученная база Message Information в каждом маршрутизаторе включает перечисленные ниже данные.

- Конфигурация Received Set для каждого интерфейса OLSRV2, описывающая сообщения TC, полученные данным маршрутизатором от этого интерфейса OLSRV2.
- Конфигурация Processed Set, описывающая обработанные данным маршрутизатором сообщения TC.
- Конфигурация Forwarded Set, описывающая сообщения TC, пересланные данным маршрутизатором.

Полученная Message Information Base обслуживает механизм лавинной рассылки MPR, обеспечивая пересылку полученных сообщений не более одного раза и однократную обработку полученных маршрутизатором сообщений. Полученная Message Information Base может также включать информацию о других типах сообщений, которые используются механизмом лавинной рассылки MPR.

### 4.4. Обзор сигнализации

Протокол генерирует и обрабатывает сообщения HELLO в соответствии с [RFC6130]. В сообщения HELLO, передаваемые интерфейсами OLSRV2, в соответствии с разделом 15 этой спецификации включается адрес инициатора, канальная метрика и данные о выборе MPR.

Данная спецификация определяет один тип сообщений - TC. Сообщения TC передаются создавшими их маршрутизаторами с упреждающей манере с регулярными интервалами через все интерфейсы OLSRV2. Интервал между сообщениями может быть постоянным или меняться динамически - например, он может уменьшаться в результате перегрузки или стабильности сети. Сообщения TC могут также передаваться в результате изменений в самом маршрутизаторе или анонсируемой им ближайшей окрестности с симметричными соединениями (symmetric 1-hop neighborhood) - например, в результате выбора этого маршрутизатора впервые в качестве маршрутизирующего MPR.

Поскольку сообщения TC передаются периодически, протокол устойчив к негарантированной доставке сообщений TC. Потери сообщений могут быть достаточно частыми в беспроводных сетях по причине конфликтов или других проблем при передаче. Протокол может использовать вариации (jitter) времени передачи сообщений за счёт внесений случайных задержек, что снижает вероятность конфликтов, как описано в [RFC5148].

Этот протокол устойчив к нарушению порядка доставки сообщений TC в результате изменения порядка в процессе передачи. Каждый маршрутизатор сохраняет значение анонсированного соседом порядкового номера (ANSN<sup>1</sup>), которое инкрементируется при изменении записанной информации от соседа, которая включается в сообщения TC. Значение ANSN включается в сообщения TC данного маршрутизатора. Получатель сообщения TC может использовать полученное значение ANSN для проверки актуальности содержащейся в сообщении информации даже при нарушении порядка доставки сообщений. Используется только свежая информация, а устаревшие данные отбрасываются.

Сообщения TC могут быть полными и неполными. Полные сообщения TC анонсируют все селекторы MPR передавшего сообщения маршрутизатора и могут включать также анонсы других ближайших соседей с симметричным

<sup>1</sup>Advertised Neighbor Sequence Number - анонсированный соседом порядковый номер.



подключением. Полные сообщения TC генерируются периодически (а также могут генерироваться в ответ на изменения ближайших соседей с симметричным соединением). Неполные сообщения TC могут использоваться для анонсирования дополнительной информации без включения не изменившихся данных.

Сообщения TC и HELLO расширены данной спецификацией путём добавления адреса инициатора (явное включение или дедукция при наличии единственного адреса) для маршрутизатора, создавшего сообщение. В сообщениях TC указывается адрес инициатора и маршрутизируемые адреса его анонсируемых соседей в двух разных Address Block TLV (адрес может быть одновременно маршрутизируемым и адресом инициатора). В сообщениях TC указываются также локально подключённые к инициатору сети.

Сообщения TC рассылаются MPR в лавинном режиме через сеть MANET. Маршрутизатор пересылает сообщение TC, принятое на интерфейсе OLSRv2, тогда и только тогда, когда это сообщение не порождено данным маршрутизатором и не пересылалось им ранее, т. е. сообщение получено впервые на данном интерфейсе OLSRv2 и принято от одного из селекторов flooding MPR (порождено или переслано этим селектором).

Для некоторых сообщений TC лавинная рассылка может осуществляться только в части сети, например, для того, чтобы ближние маршрутизаторы обеспечивались более актуальными данными, нежели удалённые, как это происходит в Fisheye State Routing [FSR] и Fuzzy Sighted Link State routing [FSLs]. Для этого используется [RFC5497].

Сообщения TC включают исходящую метрику соседей, которая будет использоваться при выборе маршрутов.

## 4.5. Метрика каналов

OLSRv1 [RFC3626] создаёт маршруты к получателям с минимальным числом интервалов (hop). Однако во многих, если не в большинстве случаев можно создать лучшие маршруты (в плане качества обслуживания конечных пользователей) на основе метрики каналов.

OLSRv2, как определено в данной спецификации, поддерживает маршрутизацию на основе метрики, т. е. позволяет для каждого канала задавать значение выбранной метрики. Метрика каналов в OLSRv2 является аддитивной и маршруты создаются с учётом минимальной суммы значений метрики для всех каналов данного маршрута.

Метрика канала является направленной, т. е. значения метрики для каналов между парой маршрутизаторов могут различаться для разных направлений. Метрика канала оценивается на приёмной стороне, поскольку на (обычно) беспроводном соединении приёмная сторона лучше информирована о состоянии канала. В OLSRv2 используется информация о входящих и исходящих каналах и спецификация явно различает эти случаи.

Эта спецификация определяет также входящую и исходящую метрику для каждого ближайшего соседа с симметричным подключением, которая будет равна минимальному из значений канальной метрики того же направления среди всех симметричных каналов к данному соседу. Отметим, что в результате все значения метрики соседей являются значениями метрики каналов и указание, например, значения метрики канала означает также указание значения метрики соседа.

Спецификация не определяет природу метрики канала. Однако за счёт применения расширения типов определённых Address Block TLV спецификация позволяет определять канальную метрику со специфическим значением через распределение агентством IANA или для частного использования. Каждое сообщение HELLO или TC, указывающее метрику канала (или соседа), показывает какая метрика передаётся, что позволяет маршрутизаторам убедиться в возможности взаимодействия. Если канальная метрика требует дополнительной сигнализации для определения её значений через сообщения HELLO или иным способом, это допускается но выходит за рамки данной спецификации.

Следует аккуратно рассмотреть использование метрики каналов. В частности, желательно не просто использовать для всех каналов принятую по умолчанию метрику (число интервалов), а внимательно разобраться с тем, какая метрика лучше подойдёт.

## 4.6. Flooding MPR и Routing MPR

Данная спецификация использует два типа MPR - рассылающие (flooding MPR) и маршрутизирующие (routing MPR). Трансляторы выбираются отдельно, поскольку:

- рассылающие MPR могут использовать метрику, маршрутизирующие MPR должны её использовать;
- когда рассылающие MPR используют метрику, это исходящая канальная метрика, маршрутизирующие MPR используют входящую метрику соседей;
- рассылающие MPR должны выбираться по интерфейсам OLSRv2, маршрутизирующие MPR не требуется выбирать по интерфейсам OLSRv2.

## 4.7. Использование Routing Set

Целью конфигурации Routing Set является определение и запись маршрутов (адресов сетей на локальных интерфейсах и интерфейсах следующего интервала) для всех возможных маршрутизируемых адресов, анонсируемых этим протоколом, а также для всех локальных адресатов (независимо от того, являются ли их адреса маршрутизируемыми). Для таких маршрутов используются только симметричные соединения.

Предполагается, что Routing Set могут применяться для маршрутизации IP путём использования содержимого этих конфигураций для обновления таблиц маршрутизации IP. Такое обновление и вопросы использования Routing Tuple при обновлении таблиц маршрутизации IP выходят за рамки этой спецификации.

Сигнализация в данной спецификации разработана так, что «опорный» граф топологии ("backbone" Topology Graph) для маршрутизаторов, каждый из которых идентифицируется своим адресом инициатора, а для любой пары маршрутизаторов имеется по крайней мере одно прямое соединение, может быть создан (из Neighbor Set и Router Topology Set) с использованием подходящего алгоритма минимального пути. В Topology Graph могут после этого добавляться адреса сетей (маршрутизируемые или от ближайших соседей с симметричным подключением) с использованием данных из Interface Information Base, Routable Address Topology Set и Attached Network Set.

## 5. Протокольные параметры и константы

В этой спецификации используются параметры и константы, определённые в [RFC6130] и последующих параграфах данного документа. В этой спецификации также используется деление параметров [RFC6130] на параметры интерфейсов, параметры маршрутизаторов и константы.

Подобно параметрам [RFC6130], определённые в этой спецификации параметры **могут** динамически изменяться маршрутизатором и не обязаны совпадать для разных маршрутизаторов даже в одной сети MANET или, для параметров интерфейса, на разных интерфейсах одного маршрутизатора.

### 5.1. Номера протокола и портов

Этот протокол задаёт сообщения TC, которые включаются в пакеты, определённые в [RFC5444]. Эти пакеты **должны** использовать номер протокола `manet` или общеизвестный номер порта UDP `manet`, заданный в спецификации [RFC5498].

Оба типа сообщений TC и HELLO [RFC6130] **должны** в данной сети MANET использовать протокол IP или UDP для того, чтобы сообщения обоих протоколов можно было помещать в один пакет [RFC5444] для передачи.

### 5.2. Групповой адрес

Данный протокол задаёт сообщения TC, которые включаются в пакеты, определённые в [RFC5444]. Эти пакеты **могут** передаваться с использованием локального для соединения (Link-Local) группового адреса LL-MANET-Routers, как указано в [RFC5498].

### 5.3. Параметр интерфейса

Задан один дополнительный параметр интерфейса, используемый только для интерфейсов OLSRv2.

#### 5.3.1. Срок корректности полученного сообщения

Описанный ниже параметр определяет срок достоверности записанной информации из принятого сообщения.

##### *RX\_HOLD\_TIME*

Интервал времени после получения сообщение соответствующим интерфейсом OLSRv2 (для которого записана эта информация) данного маршрутизатора, по истечении которого сообщение будет считаться ранее полученным на этом интерфейсе OLSRv2.

Параметр имеет показанные ниже ограничения.

- `RX_HOLD_TIME > 0`;
- значение `RX_HOLD_TIME SHOULD` следует задавать больше максимальной разницы времени прохождения сообщениями сети MANET с учётом вариаций при пересылке сообщения, времени в пути, задержек в очередях и при обработке.

### 5.4. Параметры маршрутизатора

Ниже описаны параметры, задаваемые для маршрутизаторов.

#### 5.4.1. Время сохранения изменившейся локальной информации

Приведённый ниже параметр определяет время, в течение которого сохраняется локальная информация.

##### *O\_HOLD\_TIME*

Время, в течение которого недавно использованный и потом изменённый адрес инициатора продолжает использоваться для распознавания принадлежащих этому маршрутизатору сообщений.

Параметр имеет показанное ниже ограничение.

`O_HOLD_TIME > 0`

#### 5.4.2. Параметр метрики каналов

На всех маршрутизаторах, соответствующих этой спецификации, используется один тип метрики канала, указываемый параметром `LINK_METRIC_TYPE`, который может принимать значения от 0 до 255, включительно.

#### 5.4.3. Интервалы между сообщениями

Здесь описаны параметры, регулирующие передачу маршрутизаторами сообщений TC. Такие сообщения обычно передаются периодически, но **могут** отправляться также в ответ на изменение в маршрутизаторе конфигурации Neighbor Set и/или Local Attached Network Set. В сильно динамичных сетях при большом значении `TC_INTERVAL` и малом значении `TC_MIN_INTERVAL` вызванные изменениями сообщения TC **могут** передаваться чаще периодических. Однако, по причине отсутствия у маршрутизатора сведений, например, о подключающихся к сети удалённых (более, чем на 2 интервала) маршрутизаторов, **недопустимо** ограничиваться передачей лишь вызванных событиями сообщений TC.

##### *TC\_INTERVAL*

Максимальный интервал между двумя последовательными сообщениями TC от этого маршрутизатора. Когда сообщения TC не передаются в ответ на локальные изменения (в силу особенностей протокола или отсутствия локальных изменений в сети), сообщения TC **должны** отправляться регулярно с интервалом `TC_INTERVAL`, который может меняться за счёт вариаций, как описано в [RFC5148].

##### *TC\_MIN\_INTERVAL*

Минимальный интервал между двумя последовательными сообщениями TC от этого маршрутизатора (этот интервал может меняться за счёт вариаций, как описано в [RFC5148]).

Для параметров установлен ряд ограничений:

- TC\_INTERVAL > 0;
- 0 <= TC\_MIN\_INTERVAL <= TC\_INTERVAL;
- если TLV с Type = INTERVAL\_TIME (как определено в [RFC5497]) включаются в сообщения TC, значение TC\_INTERVAL **должно** быть представимо в экспоненциальной форме, как описано в разделе 5 [RFC5497].

#### 5.4.4. Анонсированные интервалы достоверности информации

Приведённые ниже параметры определяют время достоверности информации в сообщениях TC.

##### T\_HOLD\_TIME

Используется в качестве минимального значения в TLV с Type = VALIDITY\_TIME, включаемых во все сообщения TC, которые передаёт данный маршрутизатор. Если используется одно значение параметра TC\_HOP\_LIMIT (см. параграф 5.4.7), в данном TLV будет только одно значение.

##### A\_HOLD\_TIME

Период, в течение которого передаются сообщения TC после того, как в них уже нет никакой анонсируемой информации. Целью такой передачи является ускоренное удаление устаревших данных на других маршрутизаторах.

Для параметров установлен ряд ограничений:

- T\_HOLD\_TIME > 0;
- A\_HOLD\_TIME >= 0;
- T\_HOLD\_TIME >= TC\_INTERVAL;
- если сообщения TC могут теряться, значения T\_HOLD\_TIME и A\_HOLD\_TIME **следует** делать существенно больше TC\_INTERVAL; **рекомендуется** использовать коэффициент не менее 3;
- значение T\_HOLD\_TIME **должно** быть представимо в экспоненциальной форме, как описано в разделе 5 [RFC5497].

#### 5.4.5. Обработка и пересылка времени достоверности

Приведённые ниже параметры управляют обработкой и пересылкой данных о достоверности записанной из сообщений информации.

##### P\_HOLD\_TIME

Интервал времени после приёма обрабатываемого маршрутизатором сообщения с записываемой информацией для этого маршрутизатора, в течение которого данное сообщение не будет обрабатываться заново при его повторном получении.

##### F\_HOLD\_TIME

Интервал времени после приёма пересылаемого маршрутизатором сообщения с записываемой информацией, в течение которого данное сообщение не будет пересылаться заново при его повторном получении.

Для параметров установлен ряд ограничений:

- P\_HOLD\_TIME > 0
- F\_HOLD\_TIME > 0
- значения обоих параметров **следует** делать больше максимальной разницы времени, в течение которого пакеты могут проходить через сеть MANET с учётом вариация, собственно распространения через сеть и задержек в очередях и при обработке.

#### 5.4.6. Вариации задержек

При использовании вариаций, как описано в [RFC5148], значение этих вариаций определяются приведёнными ниже параметрами.

##### TP\_MAXJITTER

Представляет значение MAXJITTER, используемое в [RFC5148] для периодически генерируемых сообщений TC от этого маршрутизатора.

##### TT\_MAXJITTER

Представляет значение MAXJITTER, используемое в [RFC5148] для вызываемых событиями сообщений TC от этого маршрутизатора.

##### F\_MAXJITTER

Представляет принятое по умолчанию значение MAXJITTER, используемое в [RFC5148] для пересылаемых этим маршрутизатором сообщений. Отметим, что до начала использования F\_MAXJITTER маршрутизатор **может** предпринять попытку вывести более подходящее значение MAXJITTER (например, на основе TLV с Type = INTERVAL\_TIME или Type = VALIDITY\_TIME в сообщении, которое будет пересылаться).

Ограничения для этих параметров рассмотрены в [RFC5148].

#### 5.4.7. Ограничение числа интервалов

Параметр TC\_HOP\_LIMIT задаёт предел числа интервалов пересылки, устанавливаемый в каждом сообщении TC. TC\_HOP\_LIMIT **может** использовать одно фиксированное значение, а **может** раздельно устанавливаться в каждом сообщении TC даже от одного маршрутизатора. Однако каждый из других маршрутизаторов на любом удалении **должен** видеть регулярную картину сообщений TC для того, чтобы осмысленные значения TLV с Type = INTERVAL\_TIME и Type = VALIDITY\_TIME на каждом удалении могли включаться в сообщения в соответствии с [RFC5497]. Для этого **должна** использоваться определённая картина TC\_HOP\_LIMIT. Например, повторяющаяся последовательность (255 4 4) удовлетворяет этому требованию (период TC\_INTERVAL для числа интервалов до 4, включительно, и 3 x TC\_INTERVAL при числе интервалов более 4), а (255 255 4 4) не удовлетворяет, поскольку при

числе интервалов пересылки более 4 интервалы между сообщениями будут принимать чередующиеся значения TC\_INTERVAL и 3 x TC\_INTERVAL.

Для параметра установлен ряд ограничений:

- максимальное значение TC\_HOP\_LIMIT не меньше диаметра сети в числе интервалов пересылки (hop0; **рекомендуется** значение 255; отметим, что при использовании упомянутой выше картины разных значений TC\_HOP\_LIMIT ограничивается только наибольшее значение в последовательности;
- во всех случаях TC\_HOP\_LIMIT >= 2.

### 5.4.8. Готовность

Каждый маршрутизатор имеет два параметра готовности WILL\_FLOODING и WILL\_ROUTING, каждый из которых **должен** принимать значение из диапазона от WILL\_NEVER до WILL\_ALWAYS, включительно.

WILL\_FLOODING представляет готовность маршрутизатора быть выбранным в качестве рассылающего MPR и, следовательно, участвовать в лавинной рассылке MPR (в частности, для сообщений TC).

WILL\_ROUTING представляет готовность маршрутизатора быть выбранным в качестве маршрутизирующего MPR и, следовательно, быть промежуточным маршрутизатором.

Во всех случаях большее значение параметра указывает на более высокую готовность маршрутизатора выступать в роли рассылающего или маршрутизирующего MPR. Если значение параметра готовности маршрутизатора составляет WILL\_NEVER (минимально возможное), он не выполняет соответствующей задачи. Используя этот протокол сеть MANET со слишком большим числом маршрутизаторов, для которых параметры WILL\_FLOODING или WILL\_ROUTING имеют значения WILL\_NEVER, просто не будет работать - **должны** быть приняты административные или иные меры для предотвращения таких ситуаций.

Отметим, что доля маршрутизаторов с параметром готовности, равным WILL\_NEVER, которая будет «слишком велика» зависит от топологии сети, которая в MANET может меняться динамически. Параметры готовности предназначены для того, чтобы некоторые маршрутизаторы (имеющие достаточно ресурсов, например, питание от электросети) можно было настроить так, чтобы они выполняли больше сетевых операций, чем другие (имеющие более ограниченные ресурсы, например, батарейное питание). В качестве общей рекомендации следует отметить, что для маршрутизаторов, не способных решать ту или иную задачу (рассылка или маршрутизация), следует устанавливать значение WILL\_NEVER для соответствующего параметра готовности.

Если для маршрутизатора параметр готовности имеет значение WILL\_ALWAYS (максимально возможное), он всегда будет выбираться в качестве маршрутизирующего или рассылающего MPR всеми ближайшими соседями с симметричным подключением.

В сетях MANET, где все маршрутизаторы имеют WILL\_FLOODING = WILL\_ALWAYS, ограничение лавинной рассылки будет по сути дела отключено и лавинная рассылка будет выполняться «вслепую».

В сетях MANET, где все маршрутизаторы имеют WILL\_ROUTING = WILL\_ALWAYS, упрощение топологии (topology reduction) будет по сути дела отключено и все маршрутизаторы будут анонсировать все свои соединения в сообщениях TC.

Маршрутизатор с WILL\_ROUTING = WILL\_NEVER не будет выполнять функций промежуточного маршрутизатора в сети MANET. Такой маршрутизатор может быть лишь источником, получателем или шлюзом в другой домен маршрутизации.

Разные маршрутизаторы **могут** иметь различные значения WILL\_FLOODING и/или WILL\_ROUTING.

Для параметров установлен ряд ограничений:

```
WILL_NEVER <= WILL_FLOODING <= WILL_ALWAYS
WILL_NEVER <= WILL_ROUTING <= WILL_ALWAYS
```

## 5.5. Ограничения на смену параметров

Если параметры протокола изменяются динамически, эти изменения будут ограничены описанными ниже параметрами.

### RX\_HOLD\_TIME

Если изменяется значение RX\_HOLD\_TIME для интерфейса OLSRv2, срок действия всех списков Received Tuple для этого интерфейса OLSRv2 **может** быть изменён.

### O\_HOLD\_TIME

Если изменяется значение O\_HOLD\_TIME, срок действия всех списков Originator Tuple **может** быть изменён.

### TC\_INTERVAL

Если значение TC\_INTERVAL увеличивается, следующее сообщение TC, генерируемое данным маршрутизатором, **должно** создаваться в соответствии с предыдущим (меньшим) значением TC\_INTERVAL. Последующие сообщения TC также **могут** генерироваться с использованием прежнего (меньшего) значения TC\_INTERVAL.

Если значение TC\_INTERVAL уменьшается, последующие сообщения TC от этого маршрутизатора **должны** генерироваться в соответствии с текущим (уменьшенным) значением TC\_INTERVAL.

### P\_HOLD\_TIME

Если изменяется значение P\_HOLD\_TIME, срок действия всех списков Processed Tuple **может** быть изменён.

### F\_HOLD\_TIME

Если изменяется значение F\_HOLD\_TIME, срок действия всех списков Forwarded Tuple **может** быть изменён.

### TP\_MAXJITTER

Если изменяется значение TP\_MAXJITTER, планирование периодических сообщений TC на данном маршрутизаторе **может** быть изменено незамедлительно.

### TT\_MAXJITTER

Если изменяется значение TT\_MAXJITTER, вызываемые внешними событиями сообщения TC на данном маршрутизаторе **могут** быть перепланированы.

**F\_MAXJITTER**

Если изменяется значение F\_MAXJITTER, сообщения TC, ожидающие пересылки с задержкой на основе этого параметра, **могут** быть перепланированы.

**TC\_HOP\_LIMIT**

Если изменяется значение TC\_HOP\_LIMIT и маршрутизатор после изменения использует множество значений, **должны** соблюдаться интервалы между сообщениями и время достоверности в сообщениях TC. Простейшим способом выполнения этого условия будет запуск новой повторяющейся последовательности TC\_HOP\_LIMIT с наибольшим значением.

**LINK\_METRIC\_TYPE**

Если изменяется значение LINK\_METRIC\_TYPE, все метрические данные каналов, записанные этим маршрутизатором, становятся неприменимыми. Маршрутизатор **должен** выполнить перечисленные ниже действия и последующие операции, описанные в разделе 17 и [RFC6130].

- Для каждого Link Tuple во всех Link Set для данного интерфейса OLSRv2 требуется обновить L\_in\_metric (**может** использоваться значение MAXIMUM\_METRIC) или удалить Link Tuple из Link Set.
- Для каждого списка Link Tuple, который не будет удалён, нужно установить:
  - L\_out\_metric := UNKNOWN\_METRIC;
  - L\_SYM\_time := EXPIRED;
  - L\_MPR\_selector := false.
- Удалить все списки Router Topology Tuple, Rutable Address Topology Tuple, Attached Network Tuple и Routing Tuple из соответствующих конфигураций Protocol Set в Topology Information Base.

**5.6. Константы**

Приведённые ниже константы задаются для маршрутизаторов. В отличие от параметров маршрутизатора менять константы **недопустимо** и они **должны** быть одинаковыми для всех маршрутизаторов.

**5.6.1. Константы метрики каналов**

Константы для минимального и максимального значений метрики каналов определяются, как:

```
MINIMUM_METRIC := 1
MAXIMUM_METRIC := 16776960
```

Символическое значение UNKNOWN\_METRIC определено в параграфе 6.1.

**5.6.2. Константы готовности**

Константы для минимального, **рекомендуемого** значения по умолчанию и максимального значения готовности:

```
WILL_NEVER := 0
WILL_DEFAULT := 7
WILL_ALWAYS := 15
```

**5.6.3. Временная константа**

Константа C (единица отсчёта времени) используется в соответствии с [RFC5497]. Она **должна** совпадать с используемой в [RFC6130], **рекомендуемое** значение составляет (в секундах):

```
C := 1/1024
```

Отметим, что эта константа используется для представления временных интервалов. Значения времени (например, в списках Protocol Tuple) представляются иначе. Уровня разрешения C достаточно (но не требуется) для таких значений.

**6. Значения метрики каналов**

Маршрутизатор записывает значение метрики канала для каждого известного ему направления канала. Значения канальной метрики используются для создания метрики маршрутов, дополняющей значения метрики каналов.

**6.1. Представление метрики канала**

Метрика каналов передаётся в сообщениях с использованием сжатого представления в форме 12-битового поля, состоящего из 4-битовой и 8-компонент. Сжатая форма обеспечивает представление положительных целых чисел с минимальным значением 1 и максимальным значением, которое несколько меньше максимального 24-битового числа в обычном представлении. Для метрики применяются лишь те значения, которые имеют точно представление в сжатой форме. Метрика маршрута определяется суммой (не более 256) значений канальной метрики и, следовательно, представляется с использованием не более 32битов.

Метрика каналов и маршрутов, используемая в информационных базах, которые определены в данной спецификации, основана на несжатых значениях и арифметические операции со значениями метрики также выполняются с несжатыми значениями и полной точностью результата. Запись значений метрики определяется реализацией и не входит в данную спецификацию, равно как и поведение реализации при записи несжатых значений. Например, реализация может использовать 32-битовые значения для метрики каналов и маршрутов.

В некоторых случаях метрика канала может быть не известна. Это указывается в данной спецификации символьным значением UNKNOWN\_METRIC. Реализация может использовать любое представление UNKNOWN\_METRIC, никогда не включая его в сообщения и не используя в расчётах. Возможными вариантами представления являются 0 или любое значение, превышающее максимальное значение метрики.

**6.2. Сжатая форма метрики канала**

Сжатая 12-битовая форма представления метрики каналов использует модифицированный формат представления с 8-битовой мантиссой (обозначена a) и 4-битовым показателем степени (обозначена b). Отметим, что для представления в виде 12-битового значения  $256b+a$  будет сохраняться такой же порядок значений, как для сжатого представления.

Значение метрики определяется, как  $(257+a)2^b - 256$ , где  $^$  означает возведение в степень. Минимальное значение (при  $a = 0$  и  $b = 0$ ) составит  $MINIMUM\_METRIC = 1$ , а максимальное ( $a = 255$  и  $b = 15$ ) -  $MAXIMUM\_METRIC = 2^{24} - 256$ .

Алгоритм расчёта значений  $a$  и  $b$  для представления наименьшего значения, которое не меньше реального значения канальной метрики  $v$  ( $\text{MINIMUM\_METRIC} \leq v \leq \text{MAXIMUM\_METRIC}$ ) имеет вид:

1. найти наименьшее целое число  $b$ , для которого выполняется условие  $v + 256 \leq 2^{(b + 9)}$ ;
2. установить  $a := (v - 256(2^b - 1)) / (2^b) - 1$  с округлением до ближайшего целого числа.

## 7. Локальная информационная база

База Local Information Base, определённая для каждого маршрутизатора в [RFC6130], расширяется данным протоколом.

- Запись адреса инициатора для маршрутизатора. Адрес инициатора **должен** быть уникальным для данного маршрутизатора и его **недопустимо** использовать в качестве адреса инициатора для любого другого маршрутизатора. Этот адрес может входить в адресный блок любой сети из любого `I_local_iface_addr_list` в данном маршрутизаторе; **недопустимо** использование адреса из любого адреса сети в любом `I_local_iface_addr_list` любого другого маршрутизатора. Адрес инициатора **может** входить в `AL_net_addr` в любом списке Local Attached Network Tuple этого или любого другого маршрутизатора, но точное совпадение **недопустимо**.
- Добавлены конфигурация Originator Set, определённая в параграфе 7.1, и Local Attached Network Set, определённая в параграфе 7.2.

Все маршрутизируемые адреса, для адреса которых маршрутизатор воспринимает в качестве IP-адреса получателя, **должны** быть включены в конфигурацию Local Interface Set или Local Attached Network Set.

### 7.1. Originator Set

Конфигурация Originator Set в маршрутизаторе содержит адреса, которые недавно использовались этим маршрутизатором в качестве адреса инициатора. Если маршрутизатор не меняет адрес инициатора, конфигурация Originator Set всегда будет пустой и **может** быть опущена. Конфигурация состоит из упорядоченных списков Originator Tuple:

`(O_orig_addr, O_time)`

где

**O\_orig\_addr** - недавно использованный адрес инициатора (без включения размера префикса);

**O\_time** указывает время завершения срока действия Originator Tuple, после чего этот список **должен** удаляться.

### 7.2. Local Attached Network Set

Конфигурация Local Attached Network Set в маршрутизаторе содержит записи о его локальных интерфейсах, не относящихся к OLSRv2, через которые он может выполнять функции шлюза в другие сети. Конфигурация Local Attached Network Set **должна** обеспечиваться для этого протокола и **должна** отражать любые изменения в состоянии маршрутизатора (если конфигурация маршрутизатора не изменяется, Local Attached Network Set также сохраняется без изменений, а в случае наличия на маршрутизаторе только интерфейсов OLSRv2 множество Local Attached Network Set может быть пустым). Этот протокол не меняет конфигурации Local Attached Network Set и лишь реагирует на (внешние по отношению к протоколу) изменения в Local Attached Network Set. Конфигурация содержит упорядоченные списки of Local Attached Network Tuple:

`(AL_net_addr, AL_dist, AL_metric)`

где

**AL\_net\_addr** - адрес сети, подключённой к интерфейсу, которая доступна через данный маршрутизатор; в этом поле **следует** указывать маршрутизируемый адрес, ограничения рассмотрены ниже;

**AL\_dist** - число интервалов пересылки до сети с адресом `AL_net_addr` от данного маршрутизатора.

**AL\_metric** - метрика соединения с подключённой через интерфейс сетью с адресом `AL_net_addr` от данного маршрутизатора.

Сети с локальным подключением только к данному маршрутизатору (т. е., недоступные иными путями), **следует** трактовать, как локальные интерфейсы не-MANET и добавлять в Local Interface Set, как указано в [RFC6130], а не в Local Attached Network Set.

Поскольку подключённая сеть не является специфичной для маршрутизатора и может находиться за пределами MANET, такая сеть может быть подключена и к другим маршрутизаторам. Маршрутизация в `AL_net_addr` будет использоваться соответствие с максимальной длиной префикса, следовательно, `AL_net_addr` **может** включать адрес любой сети на любом интерфейсе любого маршрутизатора (т. е., включенный в любой список `I_local_iface_addr_list`) но **недопустимо** совпадение с таким адресом или его частью (включение в адресный блок).

Этот протокол не отвечает за поддержку маршрутов от данного маршрутизатора в сети, указанные в Local Attached Network Set.

Списки Local Attached Network Tuple исключаются из конфигурации Local Attached Network Set только в случае изменения набора локально подключённых к данному маршрутизатору сетей, т. е., срок их жизни не ограничен какими-либо параметрами или информацией из принятых сообщений.

## 8. База информации об интерфейсах

Interface Information Base, как определено в [RFC6130], поддерживается для каждого интерфейса MANET. Конфигурации Link Set и 2-Hop Set в Interface Information Base для интерфейсов OLSRv2 изменены данным протоколом. В некоторых случаях эти конфигурации удобнее рассматривать, как содержащие также эти

дополнительные элементы для других интерфейсов MANET, принимая указанные элементы при создании и не меняя их.

## 8.1. Link Set

Конфигурация Link Set изменена путём добавления приведённых ниже элементов в каждый список Link Tuple.

**L\_in\_metric** - метрика канала от интерфейса OLSRv2 с адресом L\_neighbor\_iface\_addr\_list до данного интерфейса OLSRv2;

**L\_out\_metric** - метрика канала к интерфейсу OLSRv2 с адресом L\_neighbor\_iface\_addr\_list от данного интерфейса OLSRv2;

**L\_mpr\_selector** - флаг, указывающий выбран ли этот сосед данным маршрутизатором в качестве рассылающего MPR (т. е., данный маршрутизатор является для того селектором flooding MPR).

L\_in\_metric будет задаваться процессом, который по отношению к данной спецификации является внешним. Любой список Link Tuple с L\_status = HEARD или L\_status = SYMMETRIC **должен** иметь указанное значение L\_in\_metric, если оно используется данным протоколом.

В списке Link Tuple, определённом в [RFC6130] (до изменения), **должно** устанавливаться:

```
L_in_metric := UNKNOWN_METRIC
L_out_metric := UNKNOWN_METRIC
L_mpr_selector := false
```

## 8.2. 2-Hop Set

Конфигурация 2-Hop Set изменяется путём добавления перечисленных ниже элементов в каждый список 2-Hop Tuple.

**N2\_in\_metric** - метрика соседа от маршрутизатора с адресом N2\_2hop\_iface\_addr до маршрутизатора с интерфейсом OLSRv2, имеющим адрес N2\_neighbor\_iface\_addr\_list;

**N2\_out\_metric** - метрика соседа к маршрутизатору с адресом N2\_2hop\_iface\_addr от маршрутизатора с интерфейсом OLSRv2, имеющим адрес N2\_neighbor\_iface\_addr\_list.

В списке 2-Hop Tuple, определённом в [RFC6130] (до изменения), **должно** устанавливаться:

```
N2_in_metric := UNKNOWN_METRIC
N2_out_metric := UNKNOWN_METRIC
```

## 9. База информации о соседях

База Neighbor Information Base в соответствии с [RFC6130] поддерживается для каждого маршрутизатора. Данный протокол меняет её путём добавления дополнительных элементов в каждый список Neighbor Tuple конфигурации Neighbor Set. В некоторых случаях эти конфигурации удобней рассматривать, как содержащие также эти дополнительные элементы для других интерфейсов MANET, принимая указанные элементы при создании и не меняя их.

**N\_orig\_addr** - соседский адрес инициатора, который может быть не известен (отметим, что этот адрес не включает размер префикса);

**N\_in\_metric** - метрика соседа для любого канала от этого соседа к интерфейсу OLSRv2 на данном маршрутизаторе (т. е., минимальное из всех соответствующих значений L\_in\_metric с L\_status = SYMMETRIC и L\_in\_metric != UNKNOWN\_METRIC, UNKNOWN\_METRIC, если нет таких Link Tuple);

**N\_out\_metric** - метрика соседа для любого канала от интерфейса OLSRv2 данного маршрутизатора к этому соседу (т. е., минимальное из всех соответствующих значений L\_out\_metric с L\_status = SYMMETRIC и L\_out\_metric != UNKNOWN\_METRIC, UNKNOWN\_METRIC, если нет таких Link Tuple);

**N\_will\_flooding** - готовность соседа быть выбранным в качестве рассылающего MPR в диапазоне от WILL\_NEVER до WILL\_ALWAYS, включительно, или WILL\_NEVER при отсутствии информации OLSRv2 от этого соседа;

**N\_will\_routing** - готовность соседа быть выбранным в качестве маршрутизирующего MPR в диапазоне от WILL\_NEVER до WILL\_ALWAYS, включительно, или WILL\_NEVER при отсутствии информации OLSRv2 от этого соседа;

**N\_flooding\_mpr** - флаг, показывающий выбран ли данный сосед этим маршрутизатором в качестве рассылающего MPR;

**N\_routing\_mpr** - флаг, показывающий выбран ли данный сосед этим маршрутизатором в качестве маршрутизирующего MPR;

**N\_mpr\_selector** - флаг, показывающий выбрал ли данный сосед этот маршрутизатор в качестве маршрутизирующего MPR (т. е., является ли сосед селектором routing MPR для данного маршрутизатора);

**N\_advertised** - флаг, показывающий будет ли данный маршрутизатор анонсировать канал к этому соседу в своих сообщениях TC.

В списке Neighbor Tuple, определённом в [RFC6130] (до изменения), **должно** устанавливаться:

```
N_orig_addr := unknown
N_in_metric := UNKNOWN_METRIC
N_out_metric := UNKNOWN_METRIC
N_will_flooding := WILL_NEVER
N_will_routing := WILL_NEVER
N_routing_mpr := false
N_flooding_mpr := false
N_mpr_selector := false
```

```
N_advertised := false
```

Neighbor Information Base включает также переменную ANSN<sup>1</sup>, значение которой помещается в сообщения TC для индикации «свежести» передаваемой информации. Значение ANSN инкрементируется при каждом изменении анонсируемой информации (адрес инициатора и маршрутизируемые адреса, включённые в списки Neighbor Tuple с N\_advertised = true и адреса локально подключённых сетей в Local Attached Network Set базы Local Information Base), включая добавление или удаление данных.

## 10. База топологической информации

В Topology Information Base, определяемой данной спецификацией для каждого маршрутизатора, хранится информация из полученных сообщений TC, представленная там в конфигурациях Advertising Remote Router Set, Router Topology Set, Routable Address Topology Set и Attached Network Set.

В дополнение к этому поддерживается конфигурация Routing Set, создаваемая на основе информации из Local Information Base, Interface Information Bases, Neighbor Information Base и остальной части Topology Information Base.

### 10.1. Advertising Remote Router Set

Конфигурация маршрутизатора Advertising Remote Router Set хранит данные, описывающие каждый удалённый маршрутизатор в сети, который передаёт сообщения TC, что позволяет распознавать и отбрасывать устаревшие сообщения TC. Конфигурация состоит из упорядоченных списков Advertising Remote Router Tuple

```
(AR_orig_addr, AR_seq_number, AR_time)
```

где

**AR\_orig\_addr** - адрес инициатора полученного сообщения TC (этот адрес не включает размер префикса);

**AR\_seq\_number** - наибольшее значение номера ANSN во всех сообщениях TC, полученных от маршрутизатора с адресом инициатора AR\_orig\_addr (т. е., представляющего информацию в данном Tuple);

**AR\_time** задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

### 10.2. Router Topology Set

Конфигурация маршрутизатора Topology Set содержит топологические данные о соединениях между маршрутизаторами в сети MANET и состоит из списков Router Topology Tuple

```
(TR_from_orig_addr, TR_to_orig_addr, TR_seq_number, TR_metric, TR_time)
```

где

**TR\_from\_orig\_addr** - адрес инициатора для маршрутизатора, которому маршрутизатор с адресом инициатора TR\_to\_orig\_addr доступен через один интервал (hop) (этот адрес не включает размер префикса);

**TR\_to\_orig\_addr** - адрес инициатора для маршрутизатора, который может быть доступен для маршрутизатора с адресом инициатора TR\_from\_orig\_addr через один интервал (этот адрес не включает размер префикса);

**TR\_seq\_number** - наибольшее значение номера ANSN во всех сообщениях TC, полученных от маршрутизатора с адресом инициатора TR\_from\_orig\_addr (т. е., представляющего информацию в данном Tuple);

**TR\_metric** - метрика соседа от маршрутизатора с адресом инициатора TR\_from\_orig\_addr для маршрутизатора с адресом инициатора TR\_to\_orig\_addr;

**TR\_time** задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

### 10.3. Routable Address Topology Set

Конфигурация маршрутизатора Routable Address Topology Set содержит топологические данные о маршрутизируемых адресах в сети MANET, включая адреса, доступные через другие маршрутизаторы. Конфигурация состоит из упорядоченных списков Routable Address Topology Tuple

```
(TA_from_orig_addr, TA_dest_addr, TA_seq_number, TA_metric, TA_time)
```

где

**TA\_from\_orig\_addr** - адрес инициатора для маршрутизатора, которому маршрутизатор с адресом инициатора TA\_dest\_addr доступен через один интервал (этот адрес не включает размер префикса);

**TA\_dest\_addr** адрес инициатора для маршрутизатора, который может быть доступен для маршрутизатора с адресом инициатора TA\_from\_orig\_addr через один интервал;

**TA\_seq\_number** - наибольшее значение номера ANSN во всех сообщениях TC, полученных от маршрутизатора с адресом инициатора TA\_from\_orig\_addr (т. е., представляющего информацию в данном Tuple);

**TA\_metric** - метрика соседа от маршрутизатора с адресом инициатора TA\_from\_orig\_addr до маршрутизатора с интерфейсом OLSRV2, имеющим адрес TA\_dest\_addr;

**TA\_time** задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

### 10.4. Attached Network Set

Конфигурация маршрутизатора Attached Network Set содержит информацию о сетях (которые могут не входить в MANET), подключённых к другим маршрутизаторам, и маршрутизируемым адресам этих сетей. Конфигурация состоит из упорядоченных списков Attached Network Tuple

```
(AN_orig_addr, AN_net_addr, AN_seq_number, AN_dist, AN_metric, AN_time)
```

где

<sup>1</sup>Advertised Neighbor Sequence Number - порядковый номер анонса.



**AN\_orig\_addr** - адрес инициатора для маршрутизатора, который может служить шлюзом в сеть с адресом AN\_net\_addr (этот адрес не включает размер префикса);

**AN\_net\_addr** - адрес подключённой сети, которая может быть доступна через маршрутизатор с адресом инициатора AN\_orig\_addr;

**AN\_seq\_number** - наибольшее значение номера ANSN во всех сообщениях TC, полученных от маршрутизатора с адресом инициатора AN\_orig\_addr (т. е., представляющего информацию в данном Tuple);

**AN\_dist** - число интервалов пересылки до сети с адресом AN\_net\_addr от маршрутизатора с адресом инициатора AN\_orig\_addr;

**AN\_metric** - метрика канала от маршрутизатора с адресом инициатора AN\_orig\_addr до подключённой сети с адресом AN\_net\_addr;

**AN\_time** задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

## 10.5. Routing Set

Конфигурация маршрутизатора Routing Set содержит информацию о первом интервале на выбранном пути к каждому получателю, для которого такой путь известен. Конфигурация состоит из упорядоченных списков Routing Tuple

(R\_dest\_addr, R\_next\_iface\_addr, R\_local\_iface\_addr, R\_dist, R\_metric)

где

**R\_dest\_addr** - адрес сети для получателя, который может быть адресом сети на интерфейсе целевого маршрутизатора или адресом подключённой сети;

**R\_next\_iface\_addr** - адрес сети следующего интервала (next hop) на выбранном пути к получателю;

**R\_local\_iface\_addr** - адрес локального интерфейса, через пакеты IP **должны** передаваться для доставки получателю по выбранному пути;

**R\_dist** - число интервалов пересылки (hop) на выбранном пути к получателю;

**R\_metric** - метрика маршрута к получателю с адресом R\_dest\_addr.

Конфигурация Routing Set для маршрутизатора создаётся на основе содержимого конфигураций Protocol Sets этого маршрутизатора (множество Link Set, Neighbor Set, Router Topology Set, Routable Address Topology Set, Attached Network Set и, **возможно**, конфигурации 2-Hop Set). Конфигурация Routing Set обновляется (списки Routing Tuple добавляются и удаляются или Routing Set пересчитывается полностью) при расчёте маршрутных путей на основе изменений в других Protocol Set. Действие списков Routing Tuple не ограничивается по времени.

## 11. База информации о полученных сообщениях

База Received Message Information Base, определяемая данной спецификацией, содержит информацию, которая требуется для того, чтобы сообщения обрабатывались не более одного раза и пересылались не более одного раза не более, чем через один интерфейс OLSRv2 маршрутизатора, обеспечивающего пересылку MPR. Сообщения записываются с помощью «ключей» (signature), содержащих тип сообщения, адрес инициатора и порядковый номер.

### 11.1. Received Set

Маршрутизатор имеет конфигурацию Received Set для каждого интерфейса OLSRv2. Каждая конфигурация Received Set содержит ключи сообщений, полученных через данный интерфейс OLSRv2. Каждая конфигурация состоит из упорядоченных списков Received Tuple

(RX\_type, RX\_orig\_addr, RX\_seq\_number, RX\_time)

где

**RX\_type** - тип полученного сообщения (Message Type);

**RX\_orig\_addr** - адрес инициатора полученного сообщения (этот адрес не включает размер префикса);

**RX\_seq\_number** - порядковый номер полученного сообщения;

**RX\_time** задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

### 11.2. Processed Set

Маршрутизатор имеет одну конфигурацию Processed Set, в которой содержатся ключи обработанных маршрутизатором сообщений. Конфигурация состоит из упорядоченных списков Processed Tuple

(P\_type, P\_orig\_addr, P\_seq\_number, P\_time)

где

**P\_type** - тип полученного сообщения (Message Type);

**P\_orig\_addr** - адрес инициатора полученного сообщения (этот адрес не включает размер префикса);

**P\_seq\_number** - порядковый номер полученного сообщения;

**P\_time specifies** задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

### 11.3. Forwarded Set

Маршрутизатор имеет одну конфигурацию Forwarded Set, в которой содержатся ключи пересланных маршрутизатором сообщений. Конфигурация состоит из упорядоченных списков Forwarded Tuple

(F\_type, F\_orig\_addr, F\_seq\_number, F\_time)

где

*F\_type* - тип полученного сообщения (Message Type);

*F\_orig\_addr* - адрес инициатора полученного сообщения (этот адрес не включает размер префикса);

*F\_seq\_number* - порядковый номер полученного сообщения;

*F\_time* задаёт время окончания срока действия данного Tuple, после чего список **должен** удаляться.

## 12. Свойства информационных баз

В этом разделе описываются некоторые дополнительные свойства информационных баз и их содержимого.

### 12.1. Соответствие списков Protocol Tuple

Как часть спецификации во многих случаях возникает естественное соответствие списка Protocol Tuple в одной конфигурации Protocol Set списку Protocol Tuple в другой конфигурации Protocol Set той же или иной информационной базы. В таких случаях второй список Protocol Tuple называется «соответствующим» первому списку Protocol Tuple.

Конкретные примеры соответствующих списков Protocol Tuple перечислены ниже.

- Список Local Interface Tuple, соответствующий каждому Link Tuple, где Link Tuple является частью конфигурации Link Set для интерфейса MANET, а Local Interface Tuple представляет этот интерфейс MANET.
- Список Neighbor Tuple, соответствующий каждому списку Link Tuple, для которого время L\_HEARD\_time не **истекло**, так что N\_neighbor\_addr\_list содержит L\_neighbor\_iface\_addr\_list.
- Список Link Tuple (в конфигурации Link Set той же базы Interface Information Base), соответствующий каждому списку 2-Hop Tuple, так что L\_neighbor\_iface\_addr\_list = N2\_neighbor\_iface\_addr\_list.
- Список Neighbor Tuple, соответствующий каждому списку 2-Hop Tuple, так что N\_neighbor\_addr\_list содержит N2\_neighbor\_iface\_addr\_list (имеется список Neighbor Tuple, соответствующий списку Link Tuple, который соответствует списку 2-Hop Tuple).
- Список Advertising Remote Router Tuple, соответствующий каждому списку Router Topology Tuple, так что AR\_orig\_addr = TR\_from\_orig\_addr.
- Список Advertising Remote Router Tuple, соответствующий каждому списку Routable Address Topology Tuple, так что AR\_orig\_addr = TA\_from\_orig\_addr.
- Список Advertising Remote Router Tuple, соответствующий каждому списку Attached Network Tuple, так что AR\_orig\_addr = AN\_orig\_addr.
- Список Neighbor Tuple, соответствующий каждому списку Routing Tuple, так что N\_neighbor\_addr\_list содержит R\_next\_iface\_addr.

### 12.2. Принадлежность адресов

Адреса или адреса сетей с любым из перечисленных ниже свойств считаются «полностью принадлежащими» маршрутизатору при обработке полученного сообщения.

- Совпадающие с его адресом инициатора;
- совпадающие с O\_orig\_addr в списке Originator Tuple;
- совпадающие или являющиеся подмножеством I\_local\_iface\_addr\_list в списке Local Interface Tuple;
- совпадающие или являющиеся подмножеством IR\_local\_iface\_addr в списке Removed Interface Address Tuple;
- совпадающие с AL\_net\_addr в списке Local Attached Network Tuple.

Адреса или адреса сетей с любым из перечисленных ниже свойств считаются «частично принадлежащими» (сюда могут включаться «полностью принадлежащие» адреса) маршрутизатору при обработке полученного сообщения.

- Перекрывающие (совпадение или надмножество) его адрес инициатора;
- перекрывающие (совпадение или надмножество) O\_orig\_addr в списке Originator Tuple;
- перекрывающие I\_local\_iface\_addr\_list в списке Local Interface Tuple;
- перекрывающие IR\_local\_iface\_addr в списке Removed Interface Address Tuple;
- совпадающие или включающие AL\_net\_addr в списке Local Attached Network Tuple.

## 13. Пакеты и сообщения

Формат используемых этим протоколом пакетов и сообщений определён в [RFC5444]. Если не указано иное, опции, определённые в [RFC5444] также могут использоваться, включая дополнительные форматы, определённые для пакетов, сообщений, Address Block и флагов TLV.

В этом параграфе описано использование данной спецификацией пакетов и сообщений, определённых в [RFC5444], а также определённых здесь TLV.

## 13.1. Сообщения

Использующие этот протокол маршрутизаторы обмениваются информацией через сообщения. Протокол использует сообщения типов HELLO и TC. Сообщения HELLO определены в [RFC6130] и расширены данной спецификацией (см. раздел 15). Сообщения TC определяются данной спецификацией (см. раздел 16).

## 13.2. Пакеты

Одно или множество одновременно передаваемых маршрутизатором сообщений **следует** объединять в один пакет, для которого **могут** действовать ограничения на размер (например, в соответствии со значением MTU на локальном интерфейсе). Это могут быть сообщения от самого маршрутизатора или сообщения других маршрутизаторов, которые он пересылает. Сообщения от разных инициаторов **можно** объединять в одном пакете. Сообщения других протоколов, использующие форматы [RFC5444], включая (но не ограничиваясь) протокол NHDP [RFC6130], также **могут** комбинироваться для передачи в одном пакете. Данная спецификация не определяет и не использует содержимого заголовка пакетов (Packet Header).

Для пересылаемых сообщений **могут** добавляться вариации, описанные в [RFC5148], с учётом того, что вариации для всех сообщений, принятых в одном пакете, **следует** делать одинаковыми. Значение MAXJITTER, используемые при задании вариаций для пересылки сообщений, **может** основываться на информации из данного сообщения (в частности, всех Message TLV с Type = INTERVAL\_TIME или Type = VALIDITY\_TIME), в остальных случаях **следует** использовать максимальную задержку F\_MAXJITTER. Маршрутизатор **может** изменять вариации для сообщений с целью более эффективного объединения сообщений в пакет при условии того, что значение вариации не превысит разрешенный максимум.

## 13.3. TLV

Данная спецификация определяет два Message TLV и четыре Address Block TLV.

Во всех упоминаниях TLV в данной спецификации, где не указан тип расширения, предполагается Type Extension = 0. TLV в обрабатываемых сообщениях с типом расширения, отличным от указанного нулевого значения и 0, игнорируются.

Отметим, что в соответствии с [RFC5444] и сетевым порядком байтов, биты октетов нумеруются от 0 (старший) до 7 (младший).

### 13.3.1. Message TLV

MPR\_WILLING TLV используется в сообщениях HELLO. В сообщении **недопустимо** включать более одного MPR\_WILLING TLV.

Таблица 1. Определение MPR\_WILLING TLV.

Тип	Размер	Значение
MPR_WILLING	1 октет	Биты 0-3 представляют параметр WILL_FLOODING, биты 4-7 - WILL_ROUTING.

CONT\_SEQ\_NUM TLV используются в сообщениях TC. В сообщении **недопустимо** включать более одного CONT\_SEQ\_NUM TLV.

Таблица 2. Определение CONT\_SEQ\_NUM TLV.

Тип	Размер	Значение
CONT_SEQ_NUM	2 октета	Порядковый номер ANSN из Neighbor Information Base.

### 13.3.2. Address Block TLV

LINK\_METRIC TLV применяются в сообщениях HELLO и TC и **могут** использовать любые расширения типов. Данная спецификация использует LINK\_METRIC TLV только с расширением LINK\_METRIC\_TYPE. Адрес **недопустимо** связывать более, чем с одним значением метрики канала для любого конкретного расширения типа, вида (канал или сосед) или направления при использовании этого TLV.

Таблица 3. Определение LINK\_METRIC TLV.

Тип	Размер	Значение
LINK_METRIC	2 октета	Биты 0-3 указывают вид и направление; биты 4-7 - показатель степени (b), а биты 8-15 - мантиссу (a) значения метрики.

Для показателя степени и мантиссы используется формат представления, описанный в разделе 6. Биты в вида и направления **могут** присутствовать в любой комбинации, их трактовка указана в таблице 4.

Таблица 4. Типы и направления LINK\_METRIC TLV.

Бит	Тип	Направление
0	Метрика канала	Входящее
1	Метрика канала	Исходящее
2	Метрика соседа	Входящее
3	Метрика соседа	Исходящее

MPR TLV используются в сообщениях HELLO для индикации того, что связанный адрес относится к ближайшему соседу с симметричным подключением, выбранному в качестве MPR.

Таблица 5. Определение MPR TLV.

Тип	Размер	Значение
MPR	1 октет	FLOODING указывает, что соответствующий адрес принадлежит соседу, выбранному в качестве рассылающего MPR, ROUTING указывает, что адрес принадлежит соседу, выбранному в качестве маршрутизирующего MPR, FLOOD_ROUTE указывает, что адрес принадлежит соседу, выбранному в качестве маршрутизирующего и рассылающего MPR (см. параграф 24.6).

NBR\_ADDR\_TYPE TLV используются в сообщениях TC.

Таблица 6. Определение NBR\_ADDR\_TYPE TLV.

Тип	Размер	Значение
NBR_ADDR_TYPE	1 октет	ORIGINATOR указывает, что соответствующий адрес (которых <b>должен</b> иметь максимальный размер префикса) является адресом инициатора, ROUTABLE указывает, что соответствующий адрес сети является маршрутизируемым, ROUTABLE_ORIG указывает на то, что адрес является маршрутизируемым адресом инициатора ( см. параграф 24.6).

Если адрес принадлежит инициатору и является маршрутизируемым, он может быть связан с одним Address Block TLV с Type := NBR\_ADDR\_TYPE и Value := ROUTABLE\_ORIG или двумя Address Block TLV с Type:= NBR\_ADDR\_TYPE, и Value := ORIGINATOR для одного и Value := ROUTABLE для другого.

GATEWAY TLV используются в сообщениях TC. Адрес **недопустимо** связывать более, чем с одним значением счётчика интервалов, используемым в этом TLV.

Таблица 7. Определение GATEWAY TLV.

Тип	Размер	Значение
GATEWAY	1 октет	Число интервалов пересылки (hop) до подключённой сети.

Все адресные объекты, включаемые в сообщения TC в соответствии с данной спецификацией, **должны** связываться хотя бы с одним TLV с Type := NBR\_ADDR\_TYPE или TLV с Type := GATEWAY, но не с обоими сразу. В Address Block сообщений TC **могут** включаться любые адресные объекты, но данная спецификация будет игнорировать их.

## 14. Обработка и пересылка сообщений

В этом разделе описаны операции оптимизированной лавинной рассылки (MPR flooding), используемой в тех случаях, когда управляющие сообщения, как экземпляры [RFC5444], предназначены для рассылки в масштабе всей сети MANET. Этот механизм рассылки определяет, когда принятое сообщение должно (или не должно) обрабатываться и пересылаться.

Этот механизм рассылки используется данным протоколом и **может** применяться расширениями протокола, которые определяют (и, следовательно, владеют ими) другие типы сообщений, для управления обработкой и/или пересылкой сообщений. Данная спецификация включает элементы (P\_type, RX\_type, F\_type), требуемые только для такого применения.

Этот механизм рассылки всегда применяется для сообщений TC (см. раздел 16). Полученные сообщения HELLO (см. раздел 15) всегда обрабатываются (если они корректны) и никогда не пересылаются с использованием данного механизма. По этой причине такие сообщения не требуется записывать в базу Received Message Information Base.

Механизмы выбора обработки и пересылки устроены так, что требуется разбирать лишь Message Header для того, чтобы определить потребность в обработке и/или пересылке сообщения и не нужно разбирать Message Body для пересылаемых без обработки сообщений. Реализация **может** разбирать Message Body только при необходимости, а **может** разбирать тело сообщения и отвергать сообщения, которые содержат ошибки или не понятны.

Реализация **должна** отбрасывать без уведомлений сообщения при невозможности разобрать заголовок или тело (если такой разбор происходит) или сообщение (отличное от HELLO) не включает порядкового номера.

### 14.1. Действия при получении сообщения

При получении на интерфейсе OLSRv2 сообщения типа, указанного использующим механизм, который включает определённые в этой спецификации сообщения TC, маршрутизатор **должен** выполнить приведённые ниже операции

1. Если маршрутизатор по адресу инициатора в сообщении определяет что сообщение относится к числу порождённых этим маршрутизатором сообщений (т. е., адрес инициатора в сообщении является адресом инициатора для данного маршрутизатора или указан в поле O\_orig\_addr списка Originator Tuple), сообщение **должно** быть отброшено без уведомления.
2. В остальных случаях:
  1. Если тип сообщения относится к числу обрабатываемых, рассматривается вопрос обработки этого сообщения в соответствии с параграфом 14.2.
  2. Если данный тип сообщений может пересылаться и выполняются условия:
    - <msg-hop-limit> > 1;
    - <msg-hop-count> отсутствует или < 255,
рассматривается вопрос о пересылке сообщения в соответствии с параграфом 14.3.

### 14.2. Рассмотрение вопроса обработки сообщения

Если (текущее) сообщение рассматривается для обработки, **должны** быть выполнены перечисленные ниже операции.

1. Если адрес отправителя (адрес в заголовке дейтаграммы IP, содержащей текущее сообщение) не соответствует (с учётом размера префикса) адресу сети в поле L\_neighbor\_iface\_addr\_list списка Link Tuple с L\_status = SYMMETRIC в конфигурации Link Set для интерфейса OLSRv2, принявшего (полученное) сообщение, обработка такого сообщения является **необязательной**. Если обработка текущего сообщения не будет производиться, последующие операции не выполняются.
2. Если имеется Processed Tuple, для которого выполняются все приведённые ниже условия:
  - P\_type = Message Type текущего сообщения;
  - P\_orig\_addr = адрес инициатора текущего сообщения;
  - P\_seq\_number = порядковый номер текущего сообщения,

обработка текущего сообщения **недопустима**.

3. В остальных случаях
  1. создаётся Processed Tuple в Processed Set со значениями
    - P\_type := Message Type текущего сообщения;
    - P\_orig\_addr := адрес инициатора текущего сообщения;
    - P\_seq\_number := порядковый номер текущего сообщения;
    - P\_time := текущее время + P\_HOLD\_TIME;
  2. выполняется обработка текущего сообщения в соответствии с Message Type.  
Для сообщений TC обработка описана в параграфе 16.3.

### 14.3. Рассмотрение вопроса пересылки сообщения

Если (текущее) сообщение рассматривается для пересылки, **должны** быть выполнены перечисленные ниже операции.

1. Если адрес отправителя (адрес в заголовке дейтаграммы IP, содержащей текущее сообщение) не соответствует (с учётом размера префикса) адресу сети в поле L\_neighbor\_iface\_addr\_list списка Link Tuple с L\_status = SYMMETRIC в конфигурации Link Set для интерфейса OLSRv2, принявшего (полученное) сообщение, такое сообщение **должно** отбрасываться без уведомления.
2. В остальных случаях
  1. если имеется Received Tuple в конфигурации Received Set для принявшего сообщение интерфейса и выполняются все перечисленные ниже условия:
    - RX\_type = Message Type текущего сообщения;
    - RX\_orig\_addr = адрес инициатора текущего сообщения;
    - RX\_seq\_number = порядковый номер текущего сообщения,текущее сообщение **должно** отбрасываться без уведомления.
  2. В остальных случаях
    1. Создаётся список Received Tuple в конфигурации Received Set для принявшего сообщение интерфейса со значениями:
      - RX\_type := Message Type текущего сообщения;
      - RX\_orig\_addr := адрес инициатора текущего сообщения;
      - RX\_seq\_number := порядковый номер текущего сообщения;
      - RX\_time := текущее время + RX\_HOLD\_TIME.
    2. Если имеется список Forwarded Tuple для которого выполняются все приведённые ниже условия:
      - F\_type = Message Type текущего сообщения;
      - F\_orig\_addr = адрес инициатора текущего сообщения;
      - F\_seq\_number = порядковый номер текущего сообщения,текущее сообщение **должно** отбрасываться без уведомления.
    3. В остальных случаях, если адрес отправитель (с учётом любого префикса) соответствует любому адресу сети в поле L\_neighbor\_iface\_addr\_list списка Link Tuple в конфигурации Link Set для принявшего сообщение интерфейса OLSRv2, который имеет L\_status = SYMMETRIC и L\_mpr\_selector = true:
      1. создаётся список Forwarded Tuple в конфигурации Forwarded Set со значениями:
        - F\_type := Message Type текущего сообщения;
        - F\_orig\_addr := адрес инициатора текущего сообщения;
        - F\_seq\_number := порядковый номер текущего сообщения;
        - F\_time := текущее время + F\_HOLD\_TIME;
      2. Message Header текущего сообщения изменяется, как показано ниже:
        - значение <msg-hop-limit> в Message Header уменьшается на 1;
        - при наличии <msg-hop-count> в Message Header это поле увеличивается на 1;
      3. сообщение передаётся через все интерфейсы OLSRv2, как описано в разделе 13.
    4. В противном случае текущее сообщение **должно** отбрасываться без уведомления.

## 15. Сообщения HELLO

Тип сообщений HELLO принадлежит протоколу NHDP [RFC6130] и эти сообщения, таким образом, генерируются, передаются, принимаются и обрабатываются протоколом NHDP. Данный протокол, в соответствии с [RFC6130], тоже использует сообщения HELLO, включая их генерацию и реализацию дополнительной обработки на основе полученных сообщений HELLO. Сообщения HELLO не пересылаются с помощью NHDP [RFC6130] или OLSRv2.

### 15.1. Генерация сообщения HELLO

Сообщения HELLO, передаваемые через интерфейсы OLSRv2, генерируются в соответствии с [RFC6130] и после этого изменяются, как описано в этом параграфе. Сообщения HELLO, передаваемые через другие интерфейсы MANET, данная спецификация не меняет.

В сообщения HELLO, передаваемые через интерфейсы OLSRv2, добавляются описанные ниже элементы.

- Адрес инициатора сообщения, в качестве которого используется адрес инициатора на маршрутизаторе. Если не выполняется ни одного из перечисленных ниже условий, **должен** использоваться элемент <msg-orig-addr>:
  - если сообщение указывает единственный адресный элемент (т. е., содержит лишь один адресный объект, который связан с Address Block TLV с Type = LOCAL\_IF и не имеет размера префикса или использует максимальный размер), этот элемент используется в качестве адреса инициатора сообщения;
  - если сообщение не включает ни одного адреса сетей на локальных интерфейсах (т. е., не имеет адресных объектов, которые связаны с Address Block TLV с Type = LOCAL\_IF), как указано в спецификации [RFC6130], маршрутизатор, генерирующий данное сообщение HELLO, имеет только один адрес интерфейса и будет использовать его в качестве адреса отправителя дейтаграммы IP, в которой содержится сообщение HELLO - этот адрес будет использоваться и в качестве адреса инициатора сообщения.
- **Должно** включаться Message TLV с Type := MPR\_WILLING.
- В приведённых ниже случаях Address Block TLV связываются с одним или множеством списков Link Tuple или Neighbor Tuple, если такие списки включены в сообщение HELLO. В каждом случае TLV **должен** связываться хотя бы с одним адресным объектом для адреса из подходящего Tuple; TLV **может** связываться со множеством таких адресов (включая копии данного адресного объекта, сами по себе возможно не связанные ни с какими другими из указанных TLV в том же или другом Address Block). Эти дополнительные TLV **недопустимо** связывать с какими-либо иными адресами в сообщении HELLO, которые будут обрабатываться протоколом NHDP [RFC6130].
  - Для каждого списка Link Tuple с L\_in\_metric != UNKNOWN\_METRIC, в котором один или множество адресов в его L\_neighbor\_iface\_addr\_list включены в качестве адресных объектов со связанными Address Block TLV с Type = LINK\_STATUS и Value = HEARD или Value = SYMMETRIC, хотя бы один из таких адресов **должен** быть связан с Address Block TLV с Type := LINK\_METRIC, указывающим метрику входного канала со значением L\_in\_metric.
  - Для каждого списка Link Tuple с L\_in\_metric != UNKNOWN\_METRIC, в котором один или множество адресов в его L\_neighbor\_iface\_addr\_list включены в качестве адресных объектов со связанными Address Block TLV с Type = LINK\_STATUS и Value = SYMMETRIC, хотя бы один из таких адресов **должен** быть связан с Address Block TLV с Type := LINK\_METRIC, указывающим метрику выходного канала со значением L\_out\_metric.
  - Для каждого списка Neighbor Tuple с N\_symmetric = true, в котором один или множество адресов в его N\_neighbor\_addr\_list включены в качестве адресных объектов со связанными Address Block TLV с Type = LINK\_STATUS или Type = OTHER\_NEIGHB и Value = SYMMETRIC, хотя бы один из таких адресов **должен** быть связан с Address Block TLV с Type := LINK\_METRIC, указывающим входную метрику соседа со значением N\_in\_metric.
  - Для каждого списка Neighbor Tuple с N\_symmetric = true, в котором один или множество адресов в его N\_neighbor\_addr\_list включены в качестве адресных объектов со связанными Address Block TLV с Type = LINK\_STATUS или Type = OTHER\_NEIGHB и Value = SYMMETRIC, хотя бы один из таких адресов **должен** быть связан с Address Block TLV с Type := LINK\_METRIC, указывающим выходную метрику соседа со значением N\_out\_metric.
  - Для каждого списка Neighbor Tuple с N\_flooding\_mpr = true, в котором один или множество адресов в его N\_neighbor\_addr\_list включены в качестве адресных объектов в сообщении HELLO со связанным Address Block TLV с Type = LINK\_STATUS и Value = SYMMETRIC, хотя бы один из таких адресов **должен** быть связан с Address Block TLV с Type := MPR и Value := FLOODING или Value := FLOOD\_ROUTE.
  - Для каждого списка Neighbor Tuple с N\_routing\_mpr = true, в котором один или множество адресов в его N\_neighbor\_addr\_list включены в качестве адресных объектов в сообщении HELLO со связанным Address Block TLV с Type = LINK\_STATUS и Value = SYMMETRIC, хотя бы один из таких адресов **должен** быть связан с Address Block TLV с Type := MPR и Value := ROUTING или Value := FLOOD\_ROUTE.

### 15.2. Передача сообщения HELLO

Сообщения HELLO планируются и передаются протоколом NHDP [RFC6130]. Этот протокол **может** требовать передачи дополнительного сообщения HELLO на каждом интерфейсе OLSRv2 при любом изменении множества MPR в маршрутизаторе в дополнение к случаям, указанным в [RFC6130]. В этом случае применяются ограничения, заданные в [RFC6130] (в частности, для минимальных интервалов передачи сообщений HELLO).

### 15.3. Обработка сообщения HELLO

При получении интерфейсом OLSRv2 сообщения HELLO это сообщение становится доступным данному протоколу одним из двух способов, разрешённых [RFC6130]:

- принятое сообщение HELLO **должно** быть сделано доступным этому протоколу при получении, что позволяет отбросить его до обработки протоколом NHDP [RFC6130] (например, если добавленная данной спецификацией в сообщение HELLO информация противоречива);
- принятое сообщение HELLO **должно** быть сделано доступным OLSRv2 для обработки после того, как NHDP [RFC6130] завершит его обработку, если оно не отброшено NHDP, как некорректно сформированное.

#### 15.3.1. Отбрасывание сообщений HELLO

В дополнение к указанному в [RFC6130] причинам отбрасывания полученных сообщений HELLO отбрасываться **должны** также перечисленные ниже сообщения HELLO, полученные на интерфейсе OLSRv2, до их обработки NHDP [RFC6130] или данным протоколом.

- В сообщении имеется более одного Message TLV с Type = MPR\_WILLING;
- в сообщении имеется адрес его инициатора или адрес сети, соответствующие адресному объекту, связанному с Address Block TLV с Type = LOCAL\_IF, т. е., частично принадлежащие данному маршрутизатору (некоторые из таких сообщений уже отбрасываются в соответствии с [RFC6130]);
- сообщение включает какой-либо адресный объект, связанный с Address Block TLV с Type = LINK\_STATUS или Type = OTHER\_NEIGHB, перекрывающий адрес инициатора сообщения;
- сообщение содержит адреса, которые будут обрабатываться NHDP [RFC6130], связанные с использованием того же или иного адресного объекта, с двумя разными значениями канальной метрики одного вида и направления, использующие TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE; это применимо также к разным адресам, относящимся к интерфейсу OLSRv2, принявшему сообщение HELLO;
- сообщение включает какой-либо адресный объект, связанный с Address Block TLV с Type = MPR, но не связанный также с Address Block TLV с Type = LINK\_STATUS и Value = SYMMETRIC (включая использование разных копий адресного объекта в одном или разных Address Block).

#### 15.3.2. Использование сообщений HELLO

Сообщения HELLO сначала обрабатываются в соответствии с [RFC6130]. Эта обработка включает идентификацию (или создание) списков Link Tuple и Neighbor Tuple, соответствующих инициатору сообщения HELLO (current Link Tuple и current Neighbor Tuple). После этого **должны** выполняться перечисленные ниже действия, если сообщение HELLO принято на интерфейсе OLSRv2 и содержит TLV с Type = MPR\_WILLING.

1. Если сообщение HELLO имеет точно указанный (well-defined) адрес инициатора (т. е. включает элемент <msg-orig-addr>) или имеет не более одного адреса сети, связанного с TLV с Type = LOCAL\_IF, выполняются следующие действия.
  1. Удаляются все Neighbor Tuple за исключением текущего списка Neighbor Tuple, где N\_orig\_addr совпадает с адресом инициатора сообщения, после чего выполняются следующие из этого действия (включая возможное удаление списков Link Tuple), указанные в [RFC6130].
  2. Для текущего списка Link Tuple выполняются следующие операции:
    1. обновляются L\_in\_metric и L\_out\_metric в соответствии с параграфом 15.3.2.1;
    2. обновляется L\_mpr\_selector в соответствии с параграфом 15.3.2.3.
  3. Для текущего списка Neighbor Tuple выполняются следующие операции:
    1. N\_orig\_addr присваивается значение адреса инициатора сообщения;
    2. обновляются N\_in\_metric и N\_out\_metric в соответствии с параграфом 15.3.2.1;
    3. обновляются N\_will\_flooding и N\_will\_routing в соответствии с параграфом 15.3.2.2;
    4. обновляется N\_mpr\_selector в соответствии с параграфом 15.3.2.3.
  4. Для всех списков 2-Hop Tuple, обновлённых в соответствии с [RFC6130] выполняются следующие операции:
    1. обновляются N2\_in\_metric и N2\_out\_metric в соответствии с параграфом 15.3.2.1.
2. При наличии каких-либо изменений в информационных базах маршрутизатора выполняются действия, описанные в разделе 17.

##### 15.3.2.1. Обновление метрики

Для каждого адреса в принятом сообщении HELLO со связанным TLV с Type = LINK\_STATUS и Value = HEARD или Value = SYMMETRIC определяется значение входящей (к инициатору сообщения) метрики канала. Если сообщение HELLO включает TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE, связывающее это значение адреса со значением метрики подходящего вида (канал) и направления (входящее), входящей метрикой канала будет это значение метрики; в противном случае метрика определяется значением UNKNOWN\_METRIC.

Для каждого адреса в полученном сообщении HELLO со связанным TLV с Type = LINK\_STATUS и Value = SYMMETRIC определяется значение исходящей (от инициатора сообщения) метрики канала. Если сообщение HELLO содержит TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE, связывающее это значение адреса со значением

метрики подходящего вида (канал) и направления (исходящее), это значение будет исходящей метрикой канала, в противном случае для исходящей метрики устанавливается значение UNKNOWN\_METRIC.

Для каждого адреса в полученном сообщении HELLO со связанным TLV с Type = LINK\_STATUS или Type = OTHER\_NEIGHB и Value = SYMMETRIC определяется значение входящей (к инициатору сообщения) метрики соседа. Если сообщение HELLO содержит TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE, связывающее это значение адреса со значением метрики подходящего вида (сосед) и направления (исходящее), это значение будет входящей метрикой соседа, в противном случае для входящей метрики устанавливается значение UNKNOWN\_METRIC.

Для каждого адреса в полученном сообщении HELLO со связанным TLV с Type = LINK\_STATUS или Type = OTHER\_NEIGHB и Value = SYMMETRIC определяется значение исходящей (от инициатора сообщения) метрики соседа. Если сообщение HELLO содержит TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE, связывающее это значение адреса со значением метрики подходящего вида (сосед) и направления (исходящее), это значение будет исходящей метрикой соседа, в противном случае для исходящей метрики устанавливается значение UNKNOWN\_METRIC.

Элементы метрики канала L\_in\_metric и L\_out\_metric в Link Tuple обновляются, как показано ниже:

- Для любого Link Tuple в L\_in\_metric **может** быть установлено любое представительное значение внешним по отношению к данной спецификации процессом в любой момент. Значение L\_in\_metric **должно** устанавливаться всякий раз, когда L\_status становится равным HEARD или SYMMETRIC (при недоступности других значений **должно** устанавливаться значение MAXIMUM\_METRIC MUST). При установке L\_in\_metric **может** использоваться информация, основанная на получении пакета с сообщением HELLO, вызвавшего создание или обновление Link Tuple.
- При обновлении Link Tuple, как указано в [RFC6130], (возможно, сразу после создания) в результате получения сообщения HELLO для L\_out\_metric устанавливается значение входящей метрики канала для любого из включённых адресов интерфейса, через который было получено сообщение HELLO, если L\_status = SYMMETRIC (отметим, что правила отбрасывания сообщений HELLO из параграфа 15.3.1 делают это значение однозначным). Если такой адрес имеется, но нет метрики канала, для L\_out\_metric устанавливается значение UNKNOWN\_METRIC.

Элементы метрики соседа N\_in\_metric и N\_out\_metric в Neighbor Tuple обновляются в соответствии с параграфом 17.3.

Элементы метрики N2\_in\_metric и N2\_out\_metric в любом 2-Hop Tuple, обновлённые в соответствии с [RFC6130], обновляются, соответственно, значениями входящей и исходящей метрики соседа, связанного с соответствующим N2\_2hop\_addr. При отсутствии такой метрики для этих элементов устанавливаются значения UNKNOWN\_METRIC.

### 15.3.2.2. Обновление Willingness

Значения N\_will\_flooding и N\_will\_routing в текущем списке Neighbor Tuple обновляются с использованием Message TLV с Type = MPR\_WILLING (должно присутствовать), как показано ниже:

- N\_will\_flooding := биты 0-3 значения TLV;
- N\_will\_routing := биты 4-7 значения TLV.

Каждое из этих значений будет находиться в диапазоне от 0 до 15 (т. е., от WILL\_NEVER до WILL\_ALWAYS.)

### 15.3.2.3. Обновление состояний селекторов MPR

L\_mpr\_selector обновляется следующим образом:

1. Если маршрутизатор находит адресный объект, представляющий любой из имеющих отношение к делу адресов сетей его локальных интерфейсов (т. е., содержащийся в I\_local\_iface\_addr\_list его интерфейса OLSRv2), со связанным Address Block TLV с Type = MPR и Value = FLOODING или Value = FLOOD\_ROUTE в сообщении HELLO (показывающем, что его инициатор выбран получившим сообщение маршрутизатором в качестве рассылающего MPR), для текущего Link Tuple устанавливается

I\_mpr\_selector := true

2. В противном случае (т. е., не найдено таких адресов и Address Block TLV), если маршрутизатор находит адресный объект, представляющий любой из имеющих отношение к делу адресов сетей его локальных интерфейсов (т. е., содержащийся в I\_local\_iface\_addr\_list его интерфейса OLSRv2), со связанным Address Block TLV с Type = LINK\_STATUS и Value = SYMMETRIC в сообщении HELLO, для текущего Link Tuple устанавливается

I\_mpr\_selector := false

N\_mpr\_selector обновляется следующим образом:

1. Если маршрутизатор находит адресный объект, представляющий любой из имеющих отношение к делу адресов сетей его локальных интерфейсов (т. е., содержащийся в I\_local\_iface\_addr\_list его интерфейса OLSRv2), со связанным Address Block TLV с Type = MPR и Value = ROUTING или Value = FLOOD\_ROUTE в сообщении HELLO (показывающем, что его инициатор выбран получившим сообщение маршрутизатором в качестве маршрутизирующего MPR), для текущего Neighbor Tuple устанавливается

N\_mpr\_selector := true

N\_advertised := true

2. В противном случае (т. е., не найдено таких адресов и Address Block TLV), если маршрутизатор находит адресный объект, представляющий любой из имеющих отношение к делу адресов сетей его локальных интерфейсов (т. е., содержащийся в I\_local\_iface\_addr\_list его интерфейса OLSRv2), со связанным Address Block TLV с Type = LINK\_STATUS и Value = SYMMETRIC в сообщении HELLO, для текущего Neighbor Tuple устанавливается

N\_mpr\_selector := false



маршрутизатор **может** также установить `N_advertised := false`

## 16. Сообщения TC

Протокол определяет (и, следовательно, владеет им) TC Message Type (см. раздел 24). Таким образом, как указано в [RFC5444], этот протокол генерирует и передаёт все сообщения TC, получает все сообщения TC и отвечает за условия и способы обработки (обновления Topology Information Base) каждого сообщения TC, а также за их пересылку в соответствии с данной спецификацией.

### 16.1. Генерация сообщений TC

Сообщения TC соответствуют определению [RFC5444]. Генерируемые сообщения TC **должны** содержать перечисленные ниже элементы, соответствующие спецификации [RFC5444]:

- Адрес инициатора сообщения (маршрутизатор, создавший сообщений). Это поле **должно** использовать элемент `<msg-orig-addr>`.
- Элемент `<msg-seq-num>` с порядковым номером сообщения.
- Элемент `<msg-hop-limit>`, содержащий `TC_HOP_LIMIT`. Маршрутизатор **может** одно или разные значения `TC_HOP_LIMIT` в своих сообщениях TC (см. параграф 5.4.7).
- Элемент `<msg-hop-count>` со значением 0, если сообщение содержит TLV с `Type = VALIDITY_TIME` или `Type = INTERVAL_TIME` (как указано в [RFC5497]), указывающее более одного значения времени. Сообщение TC **может** включать элемент `<msg-hop-count>` даже если он не требуется.
- Один экземпляр Message TLV с `Type := CONT_SEQ_NUM` и `Value := ANSN` из базы Neighbor Information Base. Если сообщение TC является полным, для этого Message TLV **должно** устанавливаться `Type Extension := COMPLETE`, в противном случае **должно** быть `Type Extension := INCOMPLETE` (если сообщение TC не включает никакие адресных объектов со связанным Address Block TLV с `Type = NBR_ADDR_TYPE` или `Type = GATEWAY`, Message TLV **может** отсутствовать).
- Один экземпляр Message TLV с `Type := VALIDITY_TIME`, как указано в [RFC5497]. Если все сообщения TC передаются с одинаковым предельным числом интервалов, этот экземпляр TLV **должен** иметь значение, представляющее период `T_HOLD_TIME`. В противном случае (более одного значения `TC_HOP_LIMIT`) этот экземпляр TLV **должен** задавать время, которое зависит от числа интервалов применительно к выбранной схеме ограничения числа интервалов, как указано в [RFC5497]; в качестве таких значений **следует** выбирать числа, кратные `T_HOLD_TIME`. Эта опция включена в [RFC5497] для представления того, что **недопустимо** использовать интервалы нулевой и бесконечной продолжительности.
- Если сообщение TC является полным, все адреса сетей, которые являются `N_orig_addr` в списках Neighbor Tuple с `N_advertised = true`, **должны** быть представлены адресами объектов в одном или множестве Address Block. В неполные сообщения TC **могут** включаться любые такие адресные объекты. Хотя бы одна копия каждого такого включённого адресного объекта **должна** быть связана с Address Block TLV с `Type := NBR_ADDR_TYPE` и `Value := ORIGINATOR` или `Value := ROUTABLE_ORIG`, если этот адресный объект также связан с `Value = ROUTABLE`.
- Если сообщение TC является полным, все маршрутизируемые адреса сетей из списков `N_neighbor_addr_list` в Neighbor Tuple с `N_advertised = true` **должны** быть представлены адресными объектами в одном или множестве Address Block. В неполные сообщения TC **могут** включаться любые такие адресные объекты. Хотя бы одна копия каждого такого адресного объекта **должна** быть связана с Address Block TLV с `Type = NBR_ADDR_TYPE` и `Value = ROUTABLE` или `Value = ROUTABLE_ORIG`, если этот адресный объект также связан с `Value = ORIGINATOR`. Хотя бы одна копия каждого такого адресного объекта **должна** быть связана с Address Block TLV с `Type = LINK_METRIC` и `Type Extension = LINK_METRIC_TYPE`, показывающим исходящую метрику соседа со значением, равным соответствующему значению `N_out_metric`.
- Если сообщение TC является полным, все адреса сетей, которые являются `AL_net_addr` в Local Attached Network Tuple, **должны** быть представлены адресами объектов в одном или множестве Address Block. В неполные сообщения TC **могут** включаться любые такие адресные объекты. Хотя бы одна копия каждого такого адресного объекта **должна** быть связана с Address Block TLV с `Type := GATEWAY` и `Value := AN_dist`. В неполные сообщения TC **могут** включаться любые такие адресные объекты. Хотя бы одна копия каждого такого адресного объекта **должна** быть связана с Address Block TLV с `Type = LINK_METRIC` и `Type Extension = LINK_METRIC_TYPE`, показывающим исходящую метрику соседа со значением, равным соответствующему значению `AL_metric`.

Сообщения TC **могут** включать:

- Один экземпляр Message TLV с `Type := INTERVAL_TIME`, как указано в [RFC5497]. Если все сообщения TC передаются с одинаковым предельным числом интервалов, этот экземпляр TLV **должен** иметь значение, представляющее период `TC_INTERVAL`. Если в сообщениях TC используются разные предельные значения числа интервалов, этот экземпляр TLV **должен** задавать время, которое зависит от числа интервалов применительно к выбранной схеме ограничения числа интервалов, как указано в [RFC5497]; в качестве таких значений **следует** выбирать числа, кратные `TC_INTERVAL`. Эта опция включена в [RFC5497] для представления того, что **недопустимо** использовать интервалы нулевой и бесконечной продолжительности.

### 16.2. Передача сообщений TC

Маршрутизатор с одним или множеством интерфейсов OLSRv2 и любыми Neighbor Tuple с `N_advertised = true` или непустой конфигурацией Local Attached Network Set **должен** генерировать сообщения TC. Маршрутизатор, не имеющий такой информации для анонсирования, **должен** генерировать «пустые» сообщения TC в интервале `A_HOLD_TIME` с момента создания последнего непустого сообщения TC.

Полные сообщения генерируются и передаются периодически через все интерфейсы OLSRv2 с используемым по умолчанию интервалом TC\_INTERVAL между двумя последовательными сообщениями TC от этого маршрутизатора.

Сообщения TC **могут** генерироваться в ответ на изменения информации, которую они анонсируют, указываемые изменением ANSN в Neighbor Information Base. В таких случаях маршрутизатор **может** передавать полные сообщения TC, а **может** перезапустить своё планирование сообщений TC. Кроме того, маршрутизатор **может** передавать неполные сообщения TC, содержащие по крайней мере недавно анонсированные адреса сетей (т. е., не предшествующие, а текущие N\_orig\_addr или из списка N\_neighbor\_addr\_list в Neighbor Tuple с N\_advertised = true или AL\_net\_addr) в своих Address Block со связанными Address Block TLV. Отметим, что маршрутизатор не может сообщать об удалении анонсированных данных с помощью неполных сообщений TC.

При отправке сообщения TC в ответ на изменение анонсируемых адресов сетей маршрутизатор **должен** соблюдать минимальный интервал TC\_MIN\_INTERVAL между передаваемыми сообщениями TC (полными или неполными) и максимальный интервал TC\_INTERVAL между полными сообщениями TC. Таким образом, маршрутизатору **недопустимо** передавать неполное сообщение TC, если в интервале TC\_MIN\_INTERVAL запланирована передача полного сообщения TC.

При генерации сообщений TC (периодических или вызванных изменением содержимого) **могут** вноситься вариации, описанные в [RFC5148]. При этом вариации **должны** быть ограничены сверху (MAXJITTER) значениями:

- TP\_MAXJITTER для периодических сообщений TC;
- TT\_MAXJITTER для вызванных изменениями сообщений TC.

### 16.3. Обработка сообщений TC

При получении сообщения TC на интерфейсе OLSRv2 принимающий маршрутизатор **должен** следовать процедурам обработки и пересылки, определенным в разделе 14.

Если сообщение рассматривается для обработки (параграф 14.2), маршрутизатор **должен** сначала проверить его приемлемость для обработки данным маршрутизатором, как указано в параграфе 16.3.1. Аналогичную проверку маршрутизатор **может** выполнять и для сообщений, рассматриваемых для пересылки - он **должен** проверить элементы в заголовке сообщения.

Если сообщение TC приемлемо, оно **должно** обрабатываться в соответствии с TC Message Type, как описано в параграфе 16.3.2, с обновлением информации в соответствующих базах Interface Information Base и Router Information Base. При внесении изменений в информационные базы **должна** выполняться обработка, описанная в разделе 17.

#### 16.3.1. Отбрасывание сообщений TC

При выполнении любого из приведённых ниже условий принятое сообщение TC является неприемлемым для обработки данным маршрутизатором.

- Размер адреса в Message Header не соответствует используемому этим маршрутизатором размеру адреса.
- В сообщении отсутствует адрес инициатора и порядковый номер.
- В сообщении нет счётчика интервалов пересылки и присутствует многозначный экземпляр TLV с Type = VALIDITY\_TIME или Type = INTERVAL\_TIME, как определено в [RFC5497].
- В сообщении не содержится в точности один экземпляр Message TLV с Type = VALIDITY\_TIME.
- В сообщении присутствует более одного экземпляра Message TLV с Type = INTERVAL\_TIME.
- В сообщении нет Message TLV с Type = CONT\_SEQ\_NUM и Type Extension = COMPLETE или Type Extension = INCOMPLETE, но есть хотя бы один адресный объект, связанный с Address Block TLV с Type = NBR\_ADDR\_TYPE или Type = GATEWAY.
- В сообщении присутствует более одного экземпляра Message TLV с Type = CONT\_SEQ\_NUM и Type Extension = COMPLETE или Type Extension = INCOMPLETE.
- Адрес инициатора сообщения частично принадлежит данному маршрутизатору.
- Сообщение включает какой-либо адресный объект с отличным от максимального размером префикса (размер префикса в битах), связанный с Address Block TLV с Type = NBR\_ADDR\_TYPE и Value = ORIGINATOR или Value = ROUTABLE\_ORIG.
- Сообщение включает какой-либо адресный объект, представляющий немаршрутизируемый адрес, связанный с Address Block TLV с Type = NBR\_ADDR\_TYPE и Value = ROUTABLE или Value = ROUTABLE\_ORIG.
- Сообщение включает какой-либо адресный объект, связанный с Address Block TLV с Type = NBR\_ADDR\_TYPE или Type = GATEWAY, который представляет адрес инициатора сообщения.
- Сообщение включает какой-либо адресный объект (включая разные копии адресного объекта в одном или разных Address Block), связанный с Address Block TLV с Type = NBR\_ADDR\_TYPE или Type = GATEWAY, который также связан с множеством исходящих метрик соседей, использующих TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE.
- Связывает какой-либо адресный объект (включая разные копии адресного объекта в одном или разных Address Block) с множеством значений single hop count, использующих один или множество Address Block TLV с Type = GATEWAY.
- Связывает какой-либо адресный объект (включая разные копии адресного объекта в одном или разных Address Block) с Address Block TLV с Type = NBR\_ADDR\_TYPE и Type = GATEWAY.

Маршрутизатор **может** признать сообщение неприемлемым и по иным причинам. Неприемлемые сообщения **должны** отбрасываться без уведомления отправителя и обновления информационных баз маршрутизатора.

Отметим, что маршрутизатор, действующий противоречиво (например, «случайно» отбрасывающий сообщения TC), может сделать часть сети неспособной к взаимодействию с остальной сетью. Поэтому **рекомендуется** согласовывать «дополнительные причины» признания сообщений неприемлемыми на уровне сети в целом (например, в политике безопасности) и реализовать это на всех участвующих маршрутизаторах.

### 16.3.2. Определения для обработки сообщений TC

Когда, в соответствии с параграфом 14.2, сообщение TC «обрабатывается в соответствии с его типом», это означает:

- если сообщение TC содержит множество Message TLV с Type = CONT\_SEQ\_NUM и Type Extension = COMPLETE, оно обрабатывается в соответствии с параграфом 16.3.3, а затем выполняется финальная обработка в соответствии с параграфом 16.3.4;
- если сообщение TC содержит Message TLV с Type = CONT\_SEQ\_NUM и Type Extension = INCOMPLETE, оно обрабатывается лишь в соответствии с параграфом 16.3.3.

При обработке сообщений в соответствии с параграфами 16.3.3 и 16.3.4:

- «время достоверности» определяется из VALIDITY\_TIME Message TLV в сообщении TC, согласно спецификации [RFC5497]; это время одинаково для всех данных в сообщении TC;
- «полученный ANSN» определяется значением Message TLV с Type = CONT\_SEQ\_NUM;
- «связанное значение метрики» определяется для любого адреса в сообщении TC, как значение метрики соседа, указанное TLV с Type = LINK\_METRIC и Type Extension = LINK\_METRIC\_TYPE, которое связано с любым адресным объектом в сообщении TC, эквивалентным этому адресу, или UNKNOWN\_METRIC (отметим, что правила параграфа 16.3.1 делают это определение однозначным);
- сравнение порядковых номеров выполняется в соответствии с разделом 21.

### 16.3.3. Первоначальная обработка сообщения TC

Обработка сообщения TC выполняется в несколько этапов, перечисленных ниже.

1. Обновляется конфигурация Advertising Remote Router Set в соответствии с параграфом 16.3.3.1. Если сообщение TC при этом будет признано отбрасываемым, остальные этапы не выполняются.
2. Обновляется конфигурация Router Topology Set в соответствии с параграфом 16.3.3.2.
3. Обновляется конфигурация Rutable Address Topology Set в соответствии с параграфом 16.3.3.3.
4. Обновляется конфигурация Attached Network Set в соответствии с параграфом 16.3.3.4.

#### 16.3.3.1. Заполнение Advertising Remote Router Set

Маршрутизатор **должен** обновить свою конфигурацию Advertising Remote Router Set, как показано ниже.

1. Если имеется список Advertising Remote Router Tuple, для которого выполняются оба условия:
  - AR\_orig\_addr = адрес инициатора сообщения;
  - AR\_seq\_number > полученный ANSN,сообщение TC **должно** быть отброшено.
2. В остальных случаях выполняются следующие операции:
  1. если нет Advertising Remote Router Tuple с
    - AR\_orig\_addr = адрес инициатора сообщения,создаётся Advertising Remote Router Tuple с
    - AR\_orig\_addr := адрес инициатора сообщения.
  2. Список Advertising Remote Router Tuple (существующий или новый) изменяется, как показано ниже:
    - AR\_seq\_number := полученный ANSN;
    - AR\_time := текущее время + время достоверности.

#### 16.3.3.2. Заполнение Router Topology Set

Маршрутизатор **должен** обновить свою конфигурацию Router Topology Set, как показано ниже.

1. Для каждого адреса (далее, анонсируемый адрес), соответствующего одному или множеству адресных объектов со связанным Address Block TLV с Type = NBR\_ADDR\_TYPE и Value = ORIGINATOR или Value = ROUTABLE\_ORIG, который не принадлежит частично данному маршрутизатору, выполняются следующие действия.
  1. Если связанная метрика имеет значение UNKNOWN\_METRIC, удаляются все Router Topology Tuple, для которых выполняются два условия:
    - TR\_from\_orig\_addr = адрес инициатора сообщения;
    - TR\_to\_orig\_addr = анонсируемый адрес.

2. В остальных случаях, если нет Router Topology Tuple, для которого выполняются оба условия:
  - TR\_from\_orig\_addr = адрес инициатора сообщения;
  - TR\_to\_orig\_addr = анонсируемый адрес,создаётся новый список Router Topology Tuple с:
  - TR\_from\_orig\_addr := адрес инициатора сообщения;
  - TR\_to\_orig\_addr := анонсируемый адрес.
3. Этот список Router Topology Tuple (существующий или новый) изменяется, как показано ниже:
  - TR\_seq\_number := полученный ANSN;
  - TR\_metric := метрика связанного канала;
  - TR\_time := текущее время + время достоверности.

### 16.3.3.3. Заполнение Routable Address Topology Set

Маршрутизатор **должен** обновить свою конфигурацию Routable Address Topology Set, как показано ниже.

1. Для каждого адреса (далее, анонсируемый адрес), соответствующего одному или множеству адресных объектов со связанным Address Block TLV с Type = NBR\_ADDR\_TYPE и Value = ROUTABLE\_ORIG, который не принадлежит частично данному маршрутизатору, выполняются следующие действия.
  1. Если связанная метрика имеет значение UNKNOWN\_METRIC, удаляются все Routable Address Topology Tuple, для которых выполняются два условия:
    - TA\_from\_orig\_addr = адрес инициатора сообщения;
    - TA\_dest\_addr = анонсируемый адрес.
  2. В остальных случаях, если нет Routable Address Topology Tuple, для которого выполняются оба условия:
    - TA\_from\_orig\_addr = адрес инициатора сообщения;
    - TA\_dest\_addr = анонсируемый адрес,создаётся новый список Routable Address Topology Tuple с:
    - TA\_from\_orig\_addr := адрес инициатора сообщения;
    - TA\_dest\_addr := анонсируемый адрес.
  3. Этот список Routable Address Topology Tuple (существующий или новый) изменяется, как показано ниже:
    - TA\_seq\_number := полученный ANSN;
    - TA\_metric := метрика связанного канала;
    - TA\_time := текущее время + время достоверности.

### 16.3.3.4. Заполнение Attached Network Set

Маршрутизатор **должен** обновить свою конфигурацию Attached Network Set, как показано ниже.

1. Для каждого адреса (далее, анонсируемый адрес), соответствующего одному или множеству адресных объектов со связанным Address Block TLV с Type = GATEWAY, который не принадлежит полностью данному маршрутизатору, выполняются следующие действия.
  1. Если связанная метрика имеет значение UNKNOWN\_METRIC, удаляются все Attached Network Tuple, для которых выполняются два условия:
    - AN\_net\_addr = присоединённый адрес;
    - AN\_orig\_addr = адрес инициатора сообщения.
  2. В остальных случаях, если нет Attached Network Tuple, для которого выполняются оба условия:
    - AN\_net\_addr = присоединённый адрес;
    - AN\_orig\_addr = адрес инициатора сообщения,создаётся новый список Attached Network Tuple с:
    - AN\_net\_addr := присоединённый адрес;
    - AN\_orig\_addr := адрес инициатора сообщения.
  3. Этот список Attached Network Tuple (существующий или новый) изменяется, как показано ниже:
    - AN\_seq\_number := полученный ANSN;
    - AN\_dist := значение (Value) из связанного GATEWAY TLV;
    - AN\_metric := метрика связанного канала;
    - AN\_time := текущее время + время достоверности.

### 16.3.4. Финальная обработка сообщения ТС

Окончательная обработка сообщения ТС выполняется в несколько этапов, перечисленных ниже.

1. Обновляется конфигурация Router Topology Set в соответствии с параграфом 16.3.4.1.
2. Обновляется конфигурация Ratable Address Topology Set в соответствии с параграфом 16.3.4.2.
3. Обновляется конфигурация Attached Network Set в соответствии с параграфом 16.3.4.3.

#### 16.3.4.1. Очистка Router Topology Set

Конфигурация Router Topology Set **должна** быть обновлена, как показано ниже.

1. Все списки Router Topology Tuples, для которых выполняются оба условия:
  - TR\_from\_orig\_addr = адрес инициатора сообщения;
  - TR\_seq\_number < полученный ANSN,**должны** быть удалены.

#### 16.3.4.2. Очистка Ratable Address Topology Set

Конфигурация Ratable Address Topology Set **должна** быть обновлена, как показано ниже.

1. Все списки Ratable Address Topology Tuples, для которых выполняются оба условия:
  - TA\_from\_orig\_addr = адрес инициатора сообщения;
  - TA\_seq\_number < полученный ANSN,**должны** быть удалены.

#### 16.3.4.3. Очистка Attached Network Set

Конфигурация Attached Network Set **должна** быть обновлена, как показано ниже.

1. Все списки Attached Network Tuples, для которых выполняются оба условия:
  - AN\_orig\_addr = адрес инициатора сообщения;
  - AN\_seq\_number < полученный ANSN,**должны** быть удалены.

## 17. Изменения информационных баз

Описанные в последующих параграфах изменения **должны** вноситься всякий раз при индикации изменения в любой из информационных баз.

### 17.1. Изменение адреса инициатора

При изменении на маршрутизаторе адреса инициатора выполняются перечисленные ниже действия<sup>1</sup>.

1. Если существует список Originator Tuple с
  - O\_orig\_addr = **прежний** адрес инициатора
 этот список удаляется.
 

Создаётся Originator Tuple с

  - O\_orig\_addr := **новый** адрес инициатора
  - O\_time := **текущее время** + O\_HOLD\_TIME

### 17.2. Изменения состояний каналов

**Должна** поддерживаться непротиворечивость Link Tuple в соответствии с приведёнными ниже правилами, дополняющими правила [RFC6130].

- Если L\_status = HEARD или L\_status = SYMMETRIC, **должен** устанавливаться флаг L\_in\_metric (внешним по отношению к этой спецификации процессом).
- Если L\_status != SYMMETRIC, устанавливается L\_mpr\_selector := false.
- Если L\_out\_metric = UNKNOWN\_METRIC, **недопустимо** значение L\_status = SYMMETRIC; устанавливается L\_SYM\_time := EXPIRED, если был указан симметричный канал.

### 17.3. Изменения состояний соседей

**Должна** поддерживаться непротиворечивость Neighbor Tuple MUST в соответствии с приведёнными ниже правилами, дополняющими правила [RFC6130].

1. Если N\_symmetric = true, то N\_in\_metric **должно** быть равно минимальному из значений L\_in\_metric в соответствующих Link Tuple с L\_status = SYMMETRIC и L\_in\_metric != UNKNOWN\_METRIC. Если таких списков нет, **должно** устанавливаться N\_in\_metric = UNKNOWN\_METRIC.

<sup>1</sup>В оригинале допущена ошибка (см. [http://www.rfc-editor.org/errata\\_search.php?rfc=7181&eid=4874](http://www.rfc-editor.org/errata_search.php?rfc=7181&eid=4874)). Здесь приведен один из возможных вариантов исправления, но окончательное решение остается за авторами документа. *Прим. перев.*

2. Если `N_symmetric = true`, то `N_out_metric` **должно** быть равно минимальному из значений `L_out_metric` в соответствующих `Link Tuple` с `L_status = SYMMETRIC` и `L_out_metric != UNKNOWN_METRIC`. Если таких списков нет, **должно** устанавливаться `N_in_metric = UNKNOWN_METRIC`.
3. Если `N_symmetric = false`, то для `N_flooding_mpr`, `N_routing_mpr`, `N_mpr_selector` и `N_advertised` **должно** устанавливаться значение `false`.
4. Если `N_mpr_selector = true`, то для `N_advertised` **должно** устанавливаться значение `true`.
5. Если `N_symmetric = true`, `N_out_metric != UNKNOWN_METRIC` и `N_mpr_selector = false`, маршрутизатор **может** выбрать `N_advertised = true` или `N_advertised = false`. При росте числа анонсируемых соседей увеличивается размер сообщений TC, но при этом растёт и уровень избыточности (резервирования) маршрутизации. Маршрутизаторам **следует** принимать во внимание природу своей сети при выборе решения, а также **следует** избегать ненужных изменения в анонсируемых состояниях, которые могут приводить к неоправданным изменениям в маршрутизации.

## 17.4. Изменение анонсируемых соседей

Маршрутизатор **должен** инкрементировать ANSN в Neighbor Information Base в перечисленных ниже случаях.

1. В любом списке Neighbor Tuple меняется значение `N_advertised` или список Neighbor Tuple с `N_advertised = true` удаляется.
2. В любом списке Neighbor Tuple меняется значение `N_orig_addr` или добавляется/удаляется любой маршрутизируемый адрес в `N_neighbor_addr_list`.
3. В любом списке Neighbor Tuple с `N_advertised = true` меняется значение `N_out_metric`.
4. Возникают какие-либо изменения в Local Attached Network Set.

## 17.5. Завершение срока действия Advertising Remote Router

Конфигурации Router Topology Set, Routable Address Topology Set и Attached Network Set **должны** меняться при завершении срока действия Advertising Remote Router Tuple (достижения времени `AR_time`). Перед удалением Advertising Remote Router Tuple должны выполняться перечисленные ниже изменения.

1. Все списки Router Topology Tuple с
  - `TR_from_orig_addr = AR_orig_addr` из Advertising Remote Router Tuple удаляются.
2. Все списки Routable Address Topology Tuple с
  - `TA_from_orig_addr = AR_orig_addr` из Advertising Remote Router Tuple удаляются.
3. Все списки Attached Network Tuple с
  - `AN_orig_addr = AR_orig_addr` из Advertising Remote Router Tuple удаляются.

## 17.6. Изменения соседства и обновления MPR

Ближайшие соседи с симметричным подключением, выбранные в качестве рассылающих и маршрутизирующих MPR **должны** удовлетворять условиям, указанным в разделе 18. Для этого следует выполнить перечисленные ниже условия.

1. Набор рассылающих MPR для маршрутизатора **должен** определяться заново при возникновении любого из перечисленных условий:
  - добавлен список Link Tuple с `L_status = SYMMETRIC` и `L_out_metric != UNKNOWN_METRIC`;
  - удалён список Link Tuple с `L_status = SYMMETRIC` и `L_out_metric != UNKNOWN_METRIC`;
  - список Link Tuple с `L_status = SYMMETRIC` и `L_out_metric != UNKNOWN_METRIC` изменяется так, что `L_status = HEARD`, `L_status = LOST` или `L_out_metric = UNKNOWN_METRIC`;
  - список Link Tuple с `L_status = HEARD` or `L_status = LOST` изменяется так, что `L_status = SYMMETRIC` и `L_out_metric != UNKNOWN_METRIC`;
  - процесс выбора рассылающего MPR использует значения метрики (см. параграф 18.4) и меняется `L_out_metric` в любом из списков Link Tuple с `L_status = SYMMETRIC`;
  - значение `N_will_flooding` в Neighbor Tuple с `N_symmetric = true` и `N_out_metric != UNKNOWN_METRIC` меняется с `WILL_NEVER` на любое значение;
  - значение `N_will_flooding` в Neighbor Tuple с `N_flooding_mpr = true` меняется на `WILL_NEVER`;
  - значение `N_will_flooding` в Neighbor Tuple с `N_symmetric = true`, `N_out_metric != UNKNOWN_METRIC` и `N_flooding_mpr = false` меняется на `WILL_ALWAYS`;
  - добавляется или удаляется 2-Hop Tuple с `N2_out_metric != UNKNOWN_METRIC`;
  - значение `N2_out_metric` в любом 2-Hop Tuple изменяется и процесс выбора рассылающего MPR использует значения метрики (см. параграф 18.4) или меняет метрику на значение `UNKNOWN_METRIC` или наоборот.
2. В остальных случаях набор рассылающих MPR маршрутизатора **может** рассчитываться заново, если `N_will_flooding` в Neighbor Tuple с `N_symmetric = true` меняется каким-либо иным способом; этот набор **следует**

рассчитывать заново, если `N_flooding_mpr = false` и это увеличивает `N_will_flooding` или `N_flooding_mpr = true` и это уменьшает `N_will_flooding`.

3. Набор маршрутизирующих MPR для маршрутизатора **должен** определяться заново при возникновении любого из перечисленных условий:
  - добавлен список Neighbor Tuple с `N_symmetric = true` и `N_in_metric != UNKNOWN_METRIC`;
  - удалён список Neighbor Tuple с `N_symmetric = true` и `N_in_metric != UNKNOWN_METRIC`;
  - список Neighbor Tuple с `N_symmetric = true` и `N_in_metric != UNKNOWN_METRIC` меняется так, что `N_symmetric = false`;
  - список Neighbor Tuple с `N_symmetric = false` меняется так, что `N_symmetric = true` и `N_in_metric != UNKNOWN_METRIC`;
  - меняется значение `N_in_metric` в любом из Neighbor Tuple с `N_symmetric = true`;
  - значение `N_will_routing` в Neighbor Tuple с `N_symmetric = true` и `N_in_metric != UNKNOWN_METRIC` меняется с `WILL_NEVER` на любое другое значение;
  - значение `N_will_routing` в Neighbor Tuple с `N_routing_mpr = true` меняется на `WILL_NEVER`;
  - значение `N_will_routing` в Neighbor Tuple с `N_symmetric = true`, `N_in_metric != UNKNOWN_METRIC` и `N_routing_mpr = false` меняется на `WILL_ALWAYS`;
  - добавляется или удаляется 2-Hop Tuple с `N2_in_metric != UNKNOWN_METRIC`;
  - меняется `N2_in_metric` в любом 2-Hop Tuple.
4. В остальных случаях набор маршрутизирующих MPR для маршрутизатора **может** рассчитываться заново, если `N_will_routing` в Neighbor Tuple с `N_symmetric = true` меняется иным способом; этот набор **следует** рассчитывать заново, если `N_routing_mpr = false` и это увеличивает `N_will_routing` или `N_routing_mpr = true` и это уменьшает `N_will_routing`.

При повторном расчёте набора MPR для маршрутизатора **должны** выполняться требования раздела 18.

## 17.7. Обновления Routing Set

Конфигурация Routing Set **должна** обновляться (см. раздел 19), когда изменения в Local Information Base, Neighborhood Information Base или Topology Information Base указывают на изменения (включая любые потенциально используемые значения исходящей метрики соседей) известных симметричных соединений и/или подключённых сетей в MANET и, следовательно, изменения топологического графа (Topology Graph). Достаточно принимать во внимание изменения, оказывающие влияние по крайней мере на перечисленные ниже аспекты.

- Конфигурация Local Interface Set для интерфейса OLSRv2 при удалении любого адреса сети из `I_local_iface_addr_list`. В таких случаях, если не удаляется интерфейс OLSRv2, может не потребоваться ничего сверх замены адреса сети (если он используется) на другой адрес сетей из того же `I_local_iface_addr_list`.
- Конфигурация Local Attached Set при удалении любого `AL_net_addr`, который является также `AN_net_addr`. В этом случае может не потребоваться ничего, кроме добавления списков Routing Tuple с `R_dest_addr` равным этому `AN_net_addr`.
- Конфигурация Link Set любого интерфейса OLSRv2 с учётом только списков Link Tuple, которые имеют или имели `L_status = SYMMETRIC` и `L_out_metric != UNKNOWN_METRIC` (включая удаление таких списков Link Tuple).
- Конфигурация Neighbor Set с учётом только списков Neighbor Tuple, которые имеют или имели `N_symmetric = true` и `N_out_metric != UNKNOWN_METRIC`, но не имеют `N_orig_addr = unknown`.
- Конфигурация 2-Hop Set любого интерфейса OLSRv2, если она используется при создании Routing Set и изменения воздействуют на какой-либо 2-Hop Tuple с `N2_out_metric != UNKNOWN_METRIC`.
- Конфигурация Router Topology Set.
- Конфигурация Routable Address Topology Set.
- Конфигурация Attached Network Set.

## 18. Выбор MPR

Каждый маршрутизатор **должен** выбрать из числа выразивших готовность ближайших соседей с симметричным соединением два подмножества маршрутизаторов - flooding MPR и routing MPR. Этот выбор записывается в конфигурацию маршрутизатора Neighbor Set и указывается в сообщениях HELLO от этого маршрутизатора. Маршрутизаторы **могут** выбирать для себя устройства MPR с помощью любого процесса, который удовлетворяет приведённым далее условиям и может (но не обязан) использовать описанную здесь организацию данных. Взаимодействие маршрутизаторов не зависит от использования ими общего или разных алгоритмов выбора MPR.

Только рассылающие MPR пересылают сообщения, передаваемые в лавинном режиме через сеть MANET. Это позволяет снизить объем лавинных рассылок и оптимизировать механизмы такой рассылки (MPR flooding). Маршрутизирующие Routing MPR служат для упрощения топологии в сетях MANET (если такое упрощение не требуется, маршрутизатор может выбрать всех подходящих соседей в качестве routing MPR). Минимизация этих двух наборов MPR не имеет существенного значения для работы сети, однако позволяет снизить издержки протокола.

## 18.1. Обзор

MPR выбирается в соответствии с перечисленными ниже этапами, определёнными в последующих параграфах.

- Определяется структура данных Neighbor Graph.
- Определяются свойства MPR Set на основе Neighbor Graph. Любой алгоритм, создающий MPR Set с нужными свойствами, будет пригоден для выбора MPR. Пример алгоритма создания MPR Set дан в Приложении В.
- Определяется способ создания графа соседей (Neighbor Graph) для каждого интерфейса на основе соответствующей Interface Information Base и способ комбинирования полученных MPR Set для определения рассылающего MPR для данного маршрутизатора, а также запись соответствующей информации в Neighbor Set данного маршрутизатора.
- Определяется способ создания единого Neighbor Graph на основе информации из всех Interface Information Bases и Neighbor Information Base, а также запись полученного MPR Set в качестве маршрутизирующих MPR в Neighbor Set данного маршрутизатора.
- Приведена спецификация условий, когда **должен** выполняться расчёт MPR.

При выборе маршрутизатором своих MPR он **может** принимать во внимание любые, известные ему характеристики соседей. В частности, **следует** учитывать готовность соседа, указанную в соответствующем значении N\_will\_flooding или N\_will\_routing, предпочитая соседей в более высоком уровне готовности (отметим, что уровни готовности WILL\_NEVER и WILL\_ALWAYS принимаются во внимание всегда). Однако маршрутизатор **может** принимать во внимание другие характеристики, считая их более важными.

Каждый маршрутизатор **может** выбирать свои маршрутизирующие и рассылающие MPR независимо или скоординированно. Маршрутизатор **может** выбирать свои MPR независимо от выбора MPR другими маршрутизаторами или **может**, например, отдать предпочтение MPR, которые уже выбраны (или не выбраны) другими маршрутизаторами.

## 18.2. Граф соседей

Neighbor Graph представляет собой (определённую здесь) структуру, состоящую из множеств N1 и N2, а также связанных с ними значений метрики и готовности. Элементы множества N1 представляют указавших свою готовность ближайших соседей с симметричным подключением, а N2 представляет адреса соседей второго уровня с симметричным соединением.

Свойства Neighbor Graph перечислены ниже.

- Граф состоит из двух множеств N1 и N2.
- Для каждого элемента x в N1 имеются связанное с ним значение готовности W(x) и  $WILL\_NEVER < W(x) \leq WILL\_ALWAYS$ .
- Для каждого элемента x в N1 имеется связанное с ним значение метрики  $d1(x) > 0$ .
- Для некоторых элементов y в N2 имеется значение связанной с элементом метрики  $d1(y) > 0$  (остальные элементы y в N2 имеют неопределённое значение  $d1(y)$ , которое можно рассматривать, как бесконечное).
- Для каждого элемента x в N1 существует подмножество N2(x) элементов N2 (которое может быть пустым), такое, что для каждого x в N1 и каждого y в N2(x) имеется связанное с ними значение метрики  $d2(x,y) > 0$  (для других x в N1 и y в N2 значение  $d2(x,y)$  не определено и может считаться бесконечным).
- N2 является объединением всех N2(x) для всех x в N1 (т. е., для каждого y в N2 имеется хотя бы один элемент x в N1, такой, что y входит в N2(x)).

Удобно также определить приведённые ниже свойства.

- Для каждого y в N2 существует множество N1(y), включающее x из N1 тогда и только тогда, когда y входит в N2(x); из последнего свойства предыдущей группы следует, что множество N1(y) не пусто.
- Для каждого x в N1 и y в N2, если определено  $d2(x,y)$ , то  $d(x,y) := d1(x) + d2(x,y)$ ; в противном случае метрика  $d(x,y)$  не определена (таким образом, метрика  $d(x,y)$  определена, если y входит в N2(x) или, что эквивалентно, если x входит в N1(y)).
- Для любого подмножества S в N1 и для каждого y в N2 метрика  $d(y,S)$  представляет собой значение  $d1(y)$ , если оно определено и все  $d(x,y)$  для x входят в N1(y) и S. Если нет минимального значения метрики,  $d(y,S)$  не определена (может считаться бесконечной). Из последнего свойства предыдущей группы следует, что метрика  $d(y,N1)$  определена для всех y.

## 18.3. Свойства MPR

С учётом определения Neighbor Graph в параграфе 18.2 множество MPR Set для данного Neighbor Graph представляет собой подмножество M множества N1, удовлетворяющее приведённым ниже условиям:

- если x из N1 имеет  $W(x) = WILL\_ALWAYS$ , x входит в M;
- для любого y в N2, не имеющего определённой метрики  $d1(y)$ , существует хотя бы один элемент в M, входящий также в N1(y) (это эквивалентно требованию определённости  $d(y,M)$ );
- для любого y в N2 выполняется условие  $d(y,M) = d(y,N1)$ .

Эти свойства отражают то, что MPR Set состоит из набора ближайших соседей с симметричными подключениями, покрывающего всех соседей второго уровня с симметричным подключением и обеспечивающего маршруты минимальной протяжённости (1 или 2 интервала) к каждому соседу второго уровня с симметричным подключением.



Отметим, что если множество  $M$  является MPR Set, то и любое подмножество  $N1$ , содержащее  $M$ , также будет MPR Set, а  $N1$  всегда является MPR Set. Множество MPR Set может быть пустым, если  $N2$  не содержит ни одного элемента с неопределённой метрикой  $d1(y)$ .

## 18.4. Рассылающие MPR

Всякий раз, когда нужно рассчитать рассылающие MPR, реализация **должна** определить и записать множество рассылающих MPR, эквивалентное рассчитанному в соответствии с этим параграфом.

При расчёте рассылающих MPR не требуется использовать метрику каналов или, эквивалентно, можно использовать метрику с фиксированным значением, в качестве которого принимается 1. Однако каналы с неизвестной метрикой ( $L\_out\_metric = UNKNOWN\_METRIC$ ) использовать **недопустимо**, даже если метрики каналов не применяются.

Маршрутизаторы **могут** принимать отдельные решения по части применения метрики для расчёта рассылающих MPR. Маршрутизатор **должен** использовать одинаковый подход в части применения метрики для всех каналов (т. е., в случаях А и В выбор **должен** совпадать).

Для каждого интерфейса OLSRv2 (текущий интерфейс) определяется граф соседей (Neighbor Graph), как описано в параграфе 18.2, в соответствии с приведёнными ниже правилами.

- Определить доступный Link Tuple в качестве Link Tuple набора Link Set для текущего интерфейса с  $L\_status = SYMMETRIC$  и  $L\_out\_metric \neq UNKNOWN\_METRIC$ .
- Определить разрешенный Link Tuple в качестве доступного Link Tuple, для которого соответствующий Neighbor Tuple имеет  $N\_will\_flooding > WILL\_NEVER$ .
- Определить разрешенный 2-Hop Tuple в качестве 2-Hop Tuple в наборе 2-Hop Set для текущего интерфейса с  $N2\_out\_metric \neq UNKNOWN\_METRIC$  и наличием разрешённого Link Tuple с  $L\_neighbor\_iface\_addr\_list = N2\_neighbor\_iface\_addr\_list$ .
- Определить элемент  $N1$  для каждого разрешённого Link Tuple. Это будет определять соответствующий Link Tuple для каждого элемента  $N1$  и соответствующий Neighbor Tuple для каждого элемента  $N1$ , являющийся Neighbor Tuple для Link Tuple.
- Для каждого элемента  $x$  в  $N1$  определить  $W(x) := N\_will\_flooding$  соответствующего Neighbor Tuple.
- Для каждого элемента  $x$  в  $N1$  определить  $d1(x)$  как одно из приведённых ниже значений:
  - A.  $L\_out\_metric$  соответствующего Link Tuple;
  - B. 1.
- Определить элемент  $N2$  для каждого сетевого адреса, который является  $N2\_2hop\_addr$  одного или нескольких разрешённых 2-Hop Tuple. Это будет определять соответствующий адрес каждого элемента  $N2$ .
- Если соответствующий адрес присутствует в  $N\_neighbor\_addr\_list$  кортежа Neighbor, соответствующего одному или множеству доступных Link Tuple, для каждого элемента  $y$  в  $N2$  определяется  $d1(y)$  как одно из
  - A. минимальное из значений  $L\_out\_metric$  в этих Link Tuple;
  - B. 1.
- В остальных случаях  $d1(y)$  не определяется. При  $d1(y) := 1$  все такие  $y$  в  $N2$  могут быть удалены из  $N2$ .
- Для каждого элемента  $x$  в  $N1$  определяется  $N2(x)$  как набор элементов  $y$  в  $N2$ ,  $y$  которых соответствующий адрес имеет значение  $N2\_2hop\_addr$  из разрешённого 2-Hop Tuple с  $N2\_neighbor\_iface\_addr\_list = L\_neighbor\_iface\_addr\_list$  кортежа Link Tuple, соответствующего  $x$ .

Для всех таких  $x$  и  $y$  определяется  $d2(x,y)$  как одно из двух значений

- A.  $N2\_out\_metric$  из данного 2-Hop Tuple;
- B. 1.

Решение о способе маркировки каждого элемента  $N1$  или  $N2$  зависит от реализации. Например, элемент  $N1$  может быть помечен одним ли несколькими адресами из соответствующего  $L\_neighbor\_iface\_addr\_list$ , указателем или ссылкой на соответствующий Link Tuple.

С использованием этих Neighbor Graph использующие лавинную рассылку MPR выбираются и записываются, как указано ниже.

- Для каждого интерфейса OLSRv2 определяется MPR Set as в соответствии с параграфом 18.3.
- Neighbor Tuple представляет использующий лавинную рассылку MPR и имеет  $N\_flooding\_mpr := true$  (иначе,  $N\_flooding\_mpr := false$ ) тогда и только тогда, когда данный Neighbor Tuple соответствует элементу в MPR Set, создаваемому для любого интерфейса, как описано выше. Т. е. общий набор использующих лавинную рассылку MPR является объединением наборов рассылающих MPR для всех интерфейсов OLSRv2.

Маршрутизатор **может** выбрать свои рассылающие MPR независимо для каждого интерфейса OLSRv2 или **может** координировать выбор MPR среди интерфейсов OLSRv2, если для каждого интерфейса OLSRv2 выполняются требуемые условия. Одним из вариантов координации является последовательная обработка интерфейсов OLSRv2 и выбор для каждого интерфейса OLSRv2 рассылающего MPR, начиная с уже выбранных (и не удаляемых) рассылающих MPR, если сосед выбран в качестве MPR для интерфейса OLSRv2, который уже обработан. Этот алгоритм описан в приложении В и может применяться таким способом.

## 18.5. Маршрутизирующие MPR

Всякий раз, когда рассчитываются MPR для маршрутизации, реализация **должна** определить и записать набор маршрутизирующих MPR, который эквивалентен одному из рассчитанных в соответствии с этим параграфом.

Определяется один Neighbor Graph в соответствии с параграфом 18.2, как описано ниже.

- Определяется достижимый Neighbor Tuple для установки в качестве Neighbor Tuple с  $N\_symmetric = true$  и  $N\_in\_metric \neq UNKNOWN\_METRIC$ .
- Определяется разрешенный Neighbor Tuple в качестве достижимого Neighbor Tuple с  $N\_will\_routing > WILL\_NEVER$ .
- Определяется разрешенный 2-Hop Tuple в качестве 2-Hop Tuple в 2-Hop Set для любого интерфейса OLSRv2 с  $N2\_in\_metric \neq UNKNOWN\_METRIC$ , для которого имеется разрешенный Neighbor Tuple с  $N\_neighbor\_addr\_list$ , содержащим  $N2\_neighbor\_iface\_addr\_list$ .
- Определяется элемент N1 для каждого разрешённого Neighbor Tuple. Затем он определяет соответствующий Neighbor Tuple для каждого элемента N1.
- Для каждого элемента x в N1 определяется  $W(x) := N\_will\_routing$  соответствующего Neighbor Tuple.
- Для каждого элемента x в N1 определяется  $d1(x) := N\_in\_metric$  соответствующего Neighbor Tuple.
- Определяется элемент N2 для каждого сетевого адреса, который является  $N2\_2hop\_addr$  одного или нескольких 2-Hop Tuple. Затем он определяет соответствующий адрес для каждого элемента N2.
- Для каждого элемента y в N2 определяется  $d1(y) = N\_in\_metric$  данного Neighbor Tuple, если соответствующий адрес имеется в списке  $N\_neighbor\_addr\_list$  достижимого Neighbor Tuple. В остальных случаях  $d1(y)$  не определяется.
- Для каждого элемента x в N1 определяется N2(x) как набор элементов y в N2, соответствующие адреса которых являются  $N2\_2hop\_addr$  в разрешённом 2-Hop Tuple, который имеет  $N2\_neighbor\_iface\_addr\_list$ , содержащийся в  $N\_neighbor\_addr\_list$  кортежа Neighbor Tuple, соответствующего x. Для всех таких x и y определяется  $d2(x,y) := N2\_out\_metric$  из данного 2-Hop Tuple.

Решение о способе маркировки каждого элемента N1 или N2 определяется реализацией. Например, элемент N1 можно пометить одним или несколькими адресами из соответствующего  $N\_neighbor\_addr\_list$ , указателем или ссылкой на соответствующий Neighbor Tuple.

С использованием этих Neighbor Graph выбираются и записываются маршрутизирующие MPR, как указано ниже.

- Определяется MPR Set в соответствии с параграфом 18.3.
- Neighbor Tuple представляет маршрутизирующий MPR и имеет  $N\_routing\_mpr := true$  (иначе  $N\_routing\_mpr := false$ ), тогда и только тогда, когда этот Neighbor Tuple соответствует элементу в MPR Set, созданному как указано выше.

## 18.6. Рассчитывающий MPR

Маршрутизатор **должен** заново рассчитывать свои наборы MPR всякий раз, когда текущий набор MPR не соответствует требованиям. Он **может** пересчитать MPR в поисках более эффективного набора, даже если текущий набор соответствует требованиям. Условия, достаточные для пересчёта наборов MPR маршрутизатором приведены в параграфе 17.6.

## 19. Расчёт Routing Set

Routing Set маршрутизатора заполняется Routing Tuple, которые представляют пути от маршрутизатора ко всем адресатам в сети. Эти пути рассчитываются на основе графа Network Topology Graph, который создаётся из данных в Information Base, полученных при обмене сообщениями HELLO и TC.

Изменения в Routing Set не требуют передачи каких-либо сообщений. Однако состояние Routing Set **следует** отражать в таблице маршрутизации IP путём добавления или удаления записей в этой таблице. В таблице маршрутизации IP нужно отражать только подходящие Routing Tuple (в частности, только те, которые представляют локальные каналы или пути к маршрутизируемым адресам).

### 19.1. Граф сетевой топологии

Network Topology Graph формируется из содержащейся в маршрутизаторе информации Local Interface Set, Link Set для интерфейсов OLSRv2, Neighbor Set, Router Topology Set, Routable Address Topology Set и Attached Network Set. Network Topology Graph **может** также использовать информацию из имеющихся в маршрутизаторе 2-Hop Set для интерфейсов OLSRv2. Network Topology Graph формирует представление маршрутизатора для топологии сети в форме направленного графа. С каждым ребром этого графа связано значение метрики. Network Topology Graph имеет «магистраль» (создаётся из маршрутов с минимальной суммарной метрикой), содержащую перечисленные ниже ребра.

- Ребра  $X \rightarrow Y$  для всех возможных Y по одному X для Y, для которых выполняются **все** приведённые ниже условия:
  - $Y = N\_orig\_addr$  из Neighbor Tuple;
  - $N\_orig\_addr$  неизвестно;
  - X входит в  $I\_local\_iface\_addr\_list$  из Local Interface Tuple;

- имеется Link Tuple с  $L\_status = SYMMETRIC$  и  $L\_out\_metric \neq UNKNOWN\_METRIC$ , такой, что этот Neighbor Tuple и этот Local Interface Tuple соответствуют ему. Сетевой адрес из  $L\_neighbor\_iface\_addr\_list$  будет в этом случае обозначаться R.

По возможности **следует** предпочитать выбор  $R = Y$  и выбор X из Local Interface Tuple, соответствующего Link Tuple, из которого был выбран R. Метрикой для такого ребра будет соответствующее значение  $N\_out\_metric$ .

- Все ребра  $W \rightarrow U$ , для которых выполняются **все** приведённые ниже условия:
  - W является  $TR\_from\_orig\_addr$  из Router Topology Tuple;
  - U является  $TR\_to\_orig\_addr$  из того же Router Topology Tuple.

Метрикой такого ребра является соответствующее значение  $TR\_metric$ .

Network Topology Graph дополнительно «декорируется» описанными далее рёбрами. Если сетевой адрес S, V, Z или T равен сетевому адресу Y или W, ребро, завершающееся адресом S, V, Z или T, **недопустимо** использовать в пути.

- Ребра  $X \rightarrow S$  для всех возможных S и одного X на S, для которых выполняются **все** приведённые ниже условия:
  - S входит в  $N\_neighbor\_addr\_list$  из Neighbor Tuple;
  - X входит в  $I\_local\_iface\_addr\_list$  из Local Interface Tuple;
  - Имеется Link Tuple с  $L\_status = SYMMETRIC$  и  $L\_out\_metric \neq UNKNOWN\_METRIC$ , такой, что этот Neighbor Tuple и этот Local Interface Tuple соответствуют ему. Сетевой адрес из  $L\_neighbor\_iface\_addr\_list$  будет в этом случае обозначаться R.

По возможности **следует** предпочитать выбор  $R = S$  и выбор X из Local Interface Tuple, соответствующего Link Tuple, из которого был выбран R. Метрикой для такого ребра будет соответствующее значение  $N\_out\_metric$ .

- Все ребра  $W \rightarrow V$ , для которых выполняются **все** приведённые ниже условия:
  - W является  $TA\_from\_orig\_addr$  из Rutable Address Topology Tuple;
  - V является  $TA\_dest\_addr$  из того же Rutable Address Topology Tuple.

Метрикой такого ребра является соответствующее значение  $TA\_metric$ .

- Все ребра  $W \rightarrow T$ , для которых выполняются **все** приведённые ниже условия:
  - W является  $AN\_orig\_addr$  из Attached Network Tuple;
  - T является  $AN\_net\_addr$  из того же Attached Network Tuple.

Метрикой такого ребра является соответствующее значение  $AN\_metric$ .

- (Необязательно) Все ребра  $Y \rightarrow Z$ , для которых выполняются **все** приведённые ниже условия:
  - Z является маршрутизируемым адресом и  $N2\_2hop\_addr$  из 2-Hop Tuple с  $N2\_out\_metric \neq UNKNOWN\_METRIC$ ;
  - Y является  $N\_orig\_addr$  из соответствующего Neighbor Tuple;
  - This Neighbor Tuple имеет значение  $N\_will\_routing$ , отличное от  $WILL\_NEVER$ .

Пути, завершающиеся такими рёбрами., **недопустимо** предпочитать какому-либо иному пути. Метрикой такого ребра является соответствующее значение  $N2\_out\_metric$ .

Ни одна часть Topology Graph, которая не соединена с локальным сетевым адресом X, не применяется. **Следует** выбирать лишь одно значение X из каждого  $I\_local\_iface\_addr\_list$  и лишь одно значение R **следует** выбирать из любого  $L\_neighbor\_iface\_addr\_list$ . Все ребра имеют значение счётчика интервалов 1, за исключением рёбер  $W \rightarrow T$ , у которых счётчик интервалов имеет значение соответствующего  $AN\_dist$ .

## 19.2. Заполнение Routing Set

Routing Set **должен** содержать кратчайшие пути ко всем адресатам от локальных интерфейсов OLSRV2, использующих Network Topology Graph. Для расчёта **можно** использовать любой алгоритм, включающий те или иные способы выбора среди путей с одинаковой метрикой (при наличии двух путей с одинаковой метрикой, но разным числом интервалов, **следует** выбирать путь с меньшим числом интервалов).

Используя нотацию из параграфа 19.1, сначала строятся «магистральные» пути, использующие только ребра  $X \rightarrow Y$  и  $W \rightarrow U$  (с помощью алгоритма определения минимальной дистанции). Затем могут быть добавлены пути, использующие лишь финальное ребро. Их **недопустимо** применять вместо магистральных путей к тем же адресатам (а пути, завершающиеся ребром  $Y \rightarrow Z$ , **не следует** применять взамен путей с другой формой завершающего ребра).

Каждый путь будет соответствовать Routing Tuple. Пути могут быть двух типов. Первый тип будет представлять пути с одним ребром типа  $X \rightarrow S$  и  $X \rightarrow Y$  в виде:

- $R\_local\_iface\_addr := X$ ;
- $R\_next\_iface\_addr := R$ ;
- $R\_dest\_addr := S$  или  $Y$ ;
- $R\_dist := 1$ ;
- $R\_metric :=$  метрика ребра,

где R определён в параграфе 19.1 для этих типов рёбер.

Второй тип будет представлять пути с множеством рёбер, которые всегда будут иметь первое ребро типа X -> Y и финальное ребро типа W -> U, W -> V, W -> T или Y -> Z. Routing Tuple будет иметь вид:

- R\_local\_iface\_addr := X;
- R\_next\_iface\_addr := Y;
- R\_dest\_addr := U, V, T или Z;
- R\_dist := суммарный счётчик интервалов (hop) на всех рёбрах пути;
- R\_metric := суммарная метрика на всех рёбрах пути.

Routing Tuple второго типа с немаршрутизируемым адресом R\_dest\_addr **можно** отбрасывать.

Пример алгоритма для расчёта Routing Set в маршрутизаторе представлен в приложении C.

## 20. Предлагаемые значения параметров

Этот протокол использует все параметры, определённые в [RFC6130], и дополнительные параметры из данной спецификации. Все дополнительные параметры, за исключением RX\_HOLD\_TIME, являются параметрами маршрутизатора, как определено в [RFC6130]. Предлагаемые значения параметров, приведённые в следующих параграфах, подходят для случаев, когда каждый параметр (включая определённые в [RFC6130]) имеет одно значение. Значения, предлагаемые для параметров, которые определены в [RFC6130], даны в этом документе.

Предлагаемые ниже значения основаны на опыте развёртывания [RFC3626] (описанном, например, в [McCabe]) и считаются типовыми. Они могут изменяться с учётом требований конкретного развёртывания. Например, предполагается, что сеть с ограниченной пропускной способностью интерфейсов будет использовать более продолжительные интервалы между соединениями, а сети с высокой мобильностью будут применять более короткие интервалы. При выборе этих значений **должны** соблюдаться ограничения, приведённые в разделе 5.

Отметим, что маршрутизаторы в сети MANET не обязаны применять одинаковые наборы параметров и параметры интерфейсов не обязаны быть одинаковыми для всех интерфейсов OLSRV2 в одном маршрутизаторе.

### 20.1. Время сохранения локальной истории

O\_HOLD\_TIME := 30 (секунд)

### 20.2. Интервалы между сообщениями

TC\_INTERVAL := 5 (секунд)

TC\_MIN\_INTERVAL := TC\_INTERVAL/4

### 20.3. Параметры срока достоверности анонсированной информации

T\_HOLD\_TIME := 3 x TC\_INTERVAL

A\_HOLD\_TIME := T\_HOLD\_TIME

### 20.4. Параметры срока достоверности принятых сообщений

RX\_HOLD\_TIME := 30 (секунд)

P\_HOLD\_TIME := 30 (секунд)

F\_HOLD\_TIME := 30 (секунд)

### 20.5. Параметры вариаций

TP\_MAXJITTER := HP\_MAXJITTER

TT\_MAXJITTER := HT\_MAXJITTER

F\_MAXJITTER := TT\_MAXJITTER

### 20.6. Параметр Hop Limit

TC\_HOP\_LIMIT := 255

### 20.7. Параметры готовности

WILL\_FLOODING := WILL\_DEFAULT

WILL\_ROUTING := WILL\_DEFAULT

## 21. Порядковые номера

Порядковые номера используются в этой спецификации для принятия решения об отбрасывании «старой» информации (т. е. сообщений, полученных с нарушением порядка). Однако ограниченное число битов в порядковых номерах с неизбежностью приведёт к «переполнению», когда после максимального значения нумерация продолжается с нуля. Для предотвращения нарушений в работе протокола в результате циклической нумерации при определении упорядоченности номеров **должны** выполняться приведённые ниже требования.

Термин MAXVALUE обозначает, что следующее значение (на 1 больше) будет превышать максимально допустимое. Для 16-битовых порядковых номеров (используются в данной спецификации) MAXVALUE = 65536.

О номере S1 говорят, что он больше номера S2, если выполняется **любое** из приведённых ниже условий.

S1 > S2 и S1 - S2 < MAXVALUE/2

S2 > S1 и S2 - S1 > MAXVALUE/2

Когда порядковые номера отличаются на MAXVALUE/2 их порядок следования становится неопределённым. В такой ситуации (которой не должно возникать) можно выбирать любой порядок.

Таким образом при сравнении двух порядковых номеров (даже в случае перехода через 0) всегда можно определить, какое из сообщений содержит более свежую информацию.

## 22. Расширения

Расширения этого протокола должны взаимодействовать с данной спецификацией и, возможно, с [RFC6130]. Устройство протокола обеспечивает такое взаимодействие разными способами, включая перечисленные ниже.

- Через доступ и, возможно, расширение данных в Information Base. Для обновления элементов, заданных в этой спецификации, действуют нормативные ограничения, указанные в [RFC6130] и приложении А. Отметим, что указанная в данном документе гарантирует выполнение этих ограничений.
- Через доступ к исходящим сообщениям до их отправки через любой из интерфейсов OLSRv2 и добавление информации в сообщения, как указано в [RFC5444]. Это **может** включать Message TLV и/или сетевые адреса со связанными Address Block TLV (адреса без вновь привязанных TLV **не следует** добавлять в сообщения). Это позволяет, например, протоколу защиты (см. раздел 23) добавить TLV с криптографической подписью сообщения.
- Через доступ к входящим сообщениям с возможностью их отбрасывания до обработки данным протоколом. Это позволяет, например, протоколу защиты (см. раздел 23) проверять подписи сообщений и предотвращать обработку и/или пересылку не поддающихся проверке сообщений.
- Через доступ к входящим сообщениям после завершения их протокольной обработки. В частности, это может позволить протоколу, добавившему информацию путём включения подходящих TLV или сетевых адресов, связанных с новыми TLV, получить доступ к такой информации после того, как соответствующие изменения будут записаны в Information Base данного протокола.
- Через запросы генерации сообщений в заданное время. В этом случае при генерации сообщений **должны** соблюдаться ограничения, заданные в [RFC6130] и параграфе 5.4.3.

## 23. Вопросы безопасности

Как протокол упреждающей маршрутизации, OLSRv2 является потенциальной мишенью для различных атак. В этом разделе представлена архитектура возможной защиты OLSRv2 и даны рекомендации по защите целостности и конфиденциальности, а также встраиванию во внешние домены маршрутизации. Обобщены определённые отдельно механизмы защиты и приведены некоторые замечания по управлению ключами.

### 23.1. Архитектура защиты

OLSRv2 встраивается в архитектуру, указанную в приложении А к [RFC5444], [RFC5498] и разделе 16 в [RFC6130], причём последняя использует и расширяет сообщения и базы Information Base.

В рамках этой архитектуры OLSRv2 и NHDP [RFC6130] принимают наличие внешних причин, позволяющих отвергнуть сообщения, которые будут сочтены некорректно сформированными или небезопасными (например, значение ICV<sup>1</sup>, включённое внешним механизмом, не может быть проверено). Эта архитектура позволяет выбрать способ реализации функций защиты, отражая ситуацию, когда домены развёртывания протокола маршрутизации MANET имеют разные требования безопасности от самых строгих до практического отсутствия. Это позволяет спецификации протокола маршрутизации MANET оставаться достаточно общей с возможностью расширения самого протокола и/или процесса мультиплексирования и демultipлексирования, описанного в приложении А к [RFC5444], обеспечивая механизмы защиты, подходящие для данного домена развёртывания.

В последующих параграфах даны рекомендации по защите целостности и конфиденциальности, а также встраиванию во внешние домены маршрутизации с помощью таких расширений.

### 23.2. Целостность

Каждый маршрутизатор отдаёт топологическую информацию в сеть, передавая сообщения HELLO, а в некоторых случаях также сообщения TC. Если маршрутизатор по той или иной причине (умышленно или в результате ошибки) вносит некорректный управляющий трафик, целостность сети может быть нарушена. Поэтому настоятельно рекомендуется проверка подлинности пакетов или сообщений.

Ниже перечислены некоторые возможные варианты некорректной работы маршрутизаторов.

1. Генерация сообщений TC, анонсирующих каналы к маршрутизаторам, не являющимся соседними.
2. Генерация сообщений TC, выдающих его за другой маршрутизатор.
3. Генерация сообщений HELLO, анонсирующих каналы к маршрутизаторам, не являющимся соседними.
4. Генерация сообщений HELLO, выдающих его за другой маршрутизатор.
5. Пересылка изменённых управляющих сообщений.
6. Отказ от пересылки управляющих сообщений.
7. Некорректный выбор многоточечных трансляторов.
8. Пересылка широковещательных управляющих сообщений без пересылки трафика данных с индивидуальными адресами.
9. Воспроизведение (replay) записанного ранее управляющего трафика от другого маршрутизатора.

Аутентификация исходного маршрутизатора (для случаев 2, 4, 5) и отдельных каналов, анонсированных в управляющих сообщениях (для случаев 1 и 3) может служить противодействием. Однако для предотвращения передачи маршрутизаторами старой (корректно аутентифицированной) информации (случай 9) нужны дополнительные данные (например, временная метка или порядковый номер), позволяющие обнаружить воспроизводимые сообщения.

<sup>1</sup>Integrity Check Value - значение для проверки целостности.

В общем случае ICV (например, цифровые подписи) и другая нужная для защиты информация могут передаваться в сообщениях HELLO и TC или в заголовках пакетов с использованием механизма TLV. Любой из этих вариантов обеспечивает при желании использовать в сети одновременно несколько разных уровней защиты.

Важно понимать, что все управляющие сообщения (HELLO и TC) передаются всем маршрутизаторам в радиусе одного интервала (1-hop neighborhood), а некоторые сообщения (TC) лавинно рассылаются всем маршрутизаторам сети. Это выполняется с помощью пакетов, передаваемых всем маршрутизаторам в радиусе одного интервала, текущий набор которых может быть неизвестен. Таким образом, управляющее сообщение или пакет, используемые этим протоколом, всегда содержатся в отprawке, адресованной множеству получателей и важно, чтобы применяемый механизм аутентификации позволял любому принимающему маршрутизатору проверить подлинность сообщения или пакета.

В [RFC7182] задан общий формат обмена для криптографической информации в виде Packet TLV, Message TLV и Address Block TLV, как указано в [RFC5444]. Эти данные могут использоваться (сообща) разными защитными расширениями протокола маршрутизации MANET. В частности, [RFC7182] задаёт формат TLV для включения ICV, т. е. подписей для защиты целостности, а также TLV с временной информацией для предотвращения атак с воспроизведением. В [RFC7182] указаны реестры для использования разных шифров и форматов временной информации. При использовании ICV TLV в среде OLSRv2 невозможность проверить включённое значение ICV требует отвергать входящее сообщение или пакет как непригодные в соответствии с параграфом 12.1 [RFC6130] и параграфом 16.3.1 данной спецификации, когда применяется процесс мультиплексирования и демупльтиплексирования, описанный в приложении А к [RFC5444].

[RFC7182] указывает, как помещать ICV в созданные сообщения, как проверять входящие сообщения и как отвергать «небезопасные» (т. е. сообщения без ICV или с ICV, которое невозможно проверить). Разные развёртывания MANET могут в зависимости от целей использования требовать разных алгоритмов ICV и временных меток или ключей, поэтому защитные расширения могут применять любые из опций, представленных в [RFC7182].

### 23.3. Конфиденциальность

В OLSRv2 трансляторы MPR периодически выполняют лавинную рассылку топологической информации всем маршрутизаторам в сети. Поэтому при использовании незащищённой сети (в частности, беспроводной) сетевая топология раскрывается для всех, кто прослушивает управляющие сообщения. Эта информация может помочь атакующему получить подробные сведения о топологии и организовать более эффективную атаку на маршрутизаторы домена, например, путём подмены адресов маршрутизаторов и перевода трафика с них на себя. Отметим, что речь идёт не о защите трафика данных, а лишь о защите управляющих сообщений, т. е. данных о сетевой топологии.

В ситуациях, где нужна конфиденциальность данных о топологии сети, могут применяться обычные криптографические методы, такие как защита групповых управляющих пакетов OLSRv2 с помощью IPsec (например, с общим секретным ключом), чтобы гарантировать доступность управляющего трафика лишь уполномоченным сторонам. Можно также применять защитное расширение, задающее механизм защиты конфиденциальности для управляющих сообщений и/или пакетов. Однако, если конфиденциальность информации о топологии сети не нужно защищать, защиты целостности (см. параграф 23.2) будет достаточно для допуска в сеть лишь доверенных маршрутизаторов (т. е. маршрутизаторов с действительными учётными данными).

### 23.4. Взаимодействие с внешними доменами маршрутизации

Протокол обеспечивает базовый механизм внесения внешней маршрутной информации в домен маршрутизации данного протокола. Маршрутные данные могут также извлекаться из протокольных Information Base, в частности, из Routing Set и выводиться во внешние домены маршрутизации, если управляющие доменом протоколы позволяют это.

Когда работающие маршрутизаторы подключают домен маршрутизации, использующий этот протокол, к внешнему домену маршрутизации, **должны** приниматься меры предотвращения вывода во внешний домен потенциально небезопасной и недостоверной информации. Также **должны** приниматься меры для проверки корректности информации перед её включением, чтобы предотвратить загрязнение таблиц маршрутизации недействительными данными.

Рекомендуемым способом расширения доступа из внешнего домена маршрутизации в домен, использующий данный протокол, является назначение префикса IP (под контролем маршрутизаторов и шлюзов, соединяющих домен с внешним доменом маршрутизации) исключительно для данного домена маршрутизации и настройка шлюзов на анонсирование этого префикса во внешний домен.

### 23.5. Обязательные механизмы защиты

Соответствующая спецификации реализация OLSRv2 **должна**, как минимум, поддерживать механизмы, указанные в [RFC7183], для защиты целостности и предотвращения повторного использования управляющих сообщений OLSRv2, включая сообщения HELLO, указанные в [RFC6130] и используемые OLSRv2, путём включения TLV с временной меткой и ICV TLV. Эти ICV TLV используют HMAC на основе SHA-256 и один или множество управляемых вручную общих секретных ключей. TLV временных меток основаны на времени POSIX<sup>1</sup>, предполагающем синхронизацию часов в маршрутизаторах.

Базовым вариантом использования, для которого механизмы обеспечивают адекватную защиту без смены ключей, является краткосрочное (например, до нескольких месяцев) развёртывание OLSRv2.

При любом развёртывании OLSRv2 **следует** применять механизм защиты, указанный в [RFC7183], но **можно** использовать другой механизм, если он лучше подходит для среды OLSRv2. Например, при долгосрочном развёртывании OLSRv2 **следует** рассмотреть дополнительные механизмы защиты (например, смену ключей).

### 23.6. Управление ключами

Эта спецификация и [RFC7183] не предписывают автоматизированное управление ключами (AKM<sup>2</sup>) как часть архитектуры защиты OLSRv2. Хотя в некоторых случаях OLSRv2 может требовать АКМ, предполагается, что в

<sup>1</sup>Portable Operating System Interface - интерфейс переносимых операционных систем.

<sup>2</sup>Automated key management.

большинстве случаев этого не требуется по описанным ниже причинам.

Начальная загрузка ключа сложна в радиосетях, где обычно невозможно определить источник принятого сигнала или разделить сигналы от разных источников.

Широкое применение радиосетей и мобильных телефонов основано на предположениях о том, что (i) секретная информация встраивается в телефон производителем и (ii) централизованная база этой информации доступна в течение срока службы сети.

Поскольку основным применением MANET является обеспечения связности без централизованных элементов и с минимальным управлением, решение, подобное описанному в предыдущем параграфе, не представляется возможным. Во многих случаях в сети MANET может не быть криптографического органа, поскольку он был бы слишком уязвим. Динамическая топология MANET может делать криптографический орган недоступным для части или всех маршрутизаторов MANET без предварительного уведомления.

В [BCP107] даны рекомендации по управлению криптографическими ключами. В частности, параграф 2.1 задаёт условия, когда нужно использовать АКМ, а в параграфе 2.2 указано, когда возможно ручное управление ключами.

В параграфе 2.1 [BCP107] сказано: «Автоматизированное управление ключами **должно** применяться при выполнении любого из [задан набор] условий». Эти условия приведены ниже с обоснованием их применимости для базового варианта развёртывания OLSRv2.

- Участник должен поддерживать  $n^2$  статических ключей, где значение  $n$  может быть велико.  
В базовом варианте OLSRv2 применяется один или несколько управляемых вручную общих секретов для всей MANET.
- Используется любой потоковый шифр (например, RC4 [RFC6229][RC4], AES-CTR [RFC3610][NIST-SP-800-38A], или AES-CCM [RFC3686][NIST-SP-800-38C]).  
Потоковый шифр не подходит для создания ICV в управляющих сообщениях OLSRv2.
- Вектор инициализации ( $IV^1$ ) может использоваться неоднократно (особенно неявный IV). Отметим, что случайные или псевдослучайные IV не создают проблемы при невысокой вероятности повторения.  
IV не подходит для создания ICV в управляющих сообщениях OLSRv2.
- Может потребоваться шифрование большого объёма данных за короткое время, что будет требовать частой смены краткосрочного сеансового ключа.  
ICV требуются только для управляющих сообщений OLSRv2 и объём данных невелик.
- Долгосрочные сеансовые ключи используются более чем двумя сторонами. Групповая рассылка является необходимым исключением, но в будущем появятся стандарты группового управления ключами для решения этой проблемы. Совместного использования долгосрочных сеансовых ключей обычно следует избегать.  
Управляющие сообщения OLSRv2 передаются с использованием групповых адресов link-local.
- В вероятной рабочей среде будет часто происходить смена персонала (или устройств), что вызовет частую смену краткосрочных сеансовых ключей.  
Это не является предусмотренным применением OLSRv2. Для долгосрочных систем OLSRv2 **следует** предусматривать дополнительные механизмы защиты (например, смену ключей).

В параграфе 2.2 [BCP107] сказано: «Ручное управление ключами может быть разумно в любой из [указан набор] ситуаций». Эти условия приведены ниже с обоснованием их применимости для базового варианта развёртывания OLSRv2.

- Среда имеет очень ограниченную пропускную способность и большой период кругового обхода. Для систем с открытым ключом нужны большие сообщения и громоздкие расчёты, симметричные ключи (например, Kerberos) зачастую требуют нескольких циклов кругового обхода и взаимодействия с внешними элементами.  
Выше была отмечена возможность отсутствия нужной инфраструктуры (криптографического органа) или её недоступности в MANET. Пропускная способность MANET обычно ограничена радиосредой и необходимостью минимального расхода полосы на сигнализацию, а время кругового обхода может быть значительным.
- Ценность защищаемой информации невысока.  
Зависит от использования OLSRv2, но защищаемая информация является управляющим трафиком OLSRv2, который имеет определённую ценность, поэтому данный случай неприменим.
- Общий объём трафика в течение времени действия долгосрочного ключа очень мал.  
ICV нужны лишь для управляющих сообщений OLSRv2, которые имеют очень малые размеры.
- Масштаб каждого развёртывания очень ограничен.  
Типичный вариант OLSRv2 может включать десятки устройств и даже крупные системы OLSRv2 невелики по стандартам Internet.

## 24. Взаимодействие с IANA

Эта спецификация определяет один тип сообщений из реестра Message Types [RFC5444], 2 типа Message TLV из реестра Message TLV Types [RFC5444] и 4 типа Address Block TLV из реестра Address Block TLV Types [RFC5444].

<sup>1</sup>Initialization vector.

## 24.1. Expert Review - рекомендации по оценке

Для реестров с политикой Expert Review назначенному эксперту **следует** учитывать общие рекомендации [RFC5444].

## 24.2. Типы сообщений

Эта спецификация определяет одно значение Message Type из диапазона 0-223 в пространстве имён Message Types, определённом [RFC5444], как показано в таблице 8.

Таблица 8. Выделение типов сообщений.

Тип	Описание
1	TC (Topology Control - контроль топологии) - сигнализация в масштабе MANET

## 24.3. Реестры специфических для типа сообщений типов TLV

Агентство IANA создало реестр для зависящих от типа сообщения Message TLV в сообщениях TC согласно параграфу 6.2.1 [RFC5444] с начальными значениями, приведёнными в таблице 9.

Таблица 9. Типы специфических для типа сообщений TLV.

Тип	Описание	Политика назначения
128 - 223	Не распределены	Expert Review

Агентство IANA создало реестр для зависящих от типа сообщения Block TLV в сообщениях TC согласно параграфу 6.2.1 [RFC5444] с начальными значениями, приведёнными в таблице 10.

Таблица 10. Типы специфических для типа сообщений Address Block TLV.

Тип	Описание	Политика назначения
128 - 223	Не распределены	Expert Review

## 24.4. Типы Message TLV

Эта спецификация определяет два типа Message TLV, значения которых выделены из пространства имён Message TLV Types, определённого в [RFC5444]. Агентство IANA выделило эти значения из диапазона 0-127. Были созданы два новых реестра расширений типов (Type Extension), начальные значения которых показаны в таблицах 11 и 12. Спецификации этих TLV приведены в параграфе 13.3.1. Любой из этих TLV **недопустимо** включать в Message TLV Block более одного раза.

Таблица 11. Распределение типов Message TLV - MPR\_WILLING.

Имя	Тип	Расширение типа	Описание	Политика назначения
MPR_WILLING	7	0	Биты 0-3 указывают желание исходного маршрутизатора служить рассылающим MPR, биты 4-7 - маршрутизирующим MPR.	
MPR_WILLING	7	1 - 255	Не распределены	Expert Review

Таблица 12. Распределение типов Message TLV - CONT\_SEQ\_NUM.

Имя	Тип	Расширение типа	Описание	Политика назначения
CONT_SEQ_NUM	8	0	COMPLETE - порядковый номер этого полного сообщения.	
CONT_SEQ_NUM	8	1	INCOMPLETE - порядковый номер этого неполного сообщения.	
CONT_SEQ_NUM	8	2 - 255	Не распределены	Expert Review

Для расширений типов с политикой Expert Review эксперту **следует** учитывать [RFC5444] в соответствии с процедурой Expert Review, определённой в [RFC5226].

## 24.5. Типы Address Block TLV

Эта спецификация определяет 4 типа Address Block TLV из пространства имён Address Block TLV Types, определённого в [RFC5444]. Агентство IANA выделило для них значения из диапазона 8-127. Созданы 4 новых реестра расширений типов (Type Extension), начальные значения которых приведены в таблицах 13 - 16. Спецификации этих TLV указаны в параграфе 13.3.2.

Для реестра LINK\_METRIC Address Block TLV Type Extensions применяется процедура Expert Review.

Таблица 13. Распределение типов Address Block TLV - LINK\_METRIC.

Имя	Тип	Расширение типа	Описание
LINK_METRIC	7	0	Метрика канала назначена административно
LINK_METRIC	7	1 - 223	Не распределены
LINK_METRIC	7	224 - 255	Резерв для экспериментов

Все LINK\_METRIC TLV независимо от расширения типа **должны** использовать своё поле value для указания типа и значения (от MINIMUM\_METRIC до MAXIMUM\_METRIC, включительно) метрики канала, как указано в разделе 6 и параграфе 13.3.2. Назначение типов расширений LINK\_METRIC TLV **должно** указывать физический смысл метрики канала и его отображение на значения указанного интервала.

Таблица 14. Распределение типов Address Block TLV - MPR.

Имя	Тип	Расширение типа	Описание	Политика назначения
MPR	8	0	Указывает, что данный сетевой адрес относится к маршрутизатору, выбранному рассылающим MPR (FLOODING = 1), маршрутизирующим MPR (ROUTING = 2) или обоими сразу (FLOOD_ROUTE = 3).	
MPR	8	1 - 255	Не распределены	Expert Review

Таблица 15. Распределение типов Address Block TLV - NBR\_ADDR\_TYPE.

Имя	Тип	Расширение типа	Описание	Политика назначения
-----	-----	-----------------	----------	---------------------



NBR_ADDR_TYPE	9	0	Указывает, что данный сетевой адрес относится к соседу, доступному через исходный маршрутизатор, если это адрес источника (ORIGINATOR = 1), маршрутизируемый адрес (ROUTABLE = 2) или то и другое вместе (ROUTABLE_ORIG = 3).	
NBR_ADDR_TYPE	9	1 - 255	Не распределены	Expert Review

Таблица 16. Распределение типов Address Block TLV - GATEWAY.

Имя	Тип	Расширение типа	Описание	Политика назначения
GATEWAY	10	0	Указывает, что данный сетевой адрес доступен через шлюз на исходном маршрутизаторе со значением равным числу интервалов пересылки (hop).	
GATEWAY	10	1 - 255	Не распределены	Expert Review

Для расширений типов с политикой Expert Review эксперту **следует** учитывать [RFC5444] в соответствии с процедурой Expert Review, определённой в [RFC5226].

## 24.6. Значения NBR\_ADDR\_TYPE и MPR

**Примечание.** Этот параграф не требует действий IANA, поскольку нужная информация включена в описания MPR и NBR\_ADDR\_TYPE Address Block TLV, выделенные в параграфе 24.5. Эта информация приведена здесь для ясности и использования в этой спецификации.

Ниже приведены поля Value, которые может использовать MPR Address Block TLV:

```
FLOODING := 1;
ROUTING := 2;
FLOOD_ROUTE := 3.
```

Ниже приведены поля Value, которые может использовать NBR\_ADDR\_TYPE Address Block TLV:

```
ORIGINATOR := 1;
ROUTABLE := 2;
ROUTABLE_ORIG := 3.
```

## 25. Разработчики документа

Эта спецификация является результатом работы перечисленных в алфавитном порядке людей.

Cedric Adjih, INRIA, France, <[Cedric.Adjih@inria.fr](mailto:Cedric.Adjih@inria.fr)>  
 Emmanuel Baccelli, INRIA, France, <[Emmanuel.Baccelli@inria.fr](mailto:Emmanuel.Baccelli@inria.fr)>  
 Thomas Heide Clausen, LIX, France, <[T.Clausen@computer.org](mailto:T.Clausen@computer.org)>  
 Justin Dean, NRL, USA, <[jdean@itd.nrl.navy.mil](mailto:jdean@itd.nrl.navy.mil)>  
 Christopher Dearlove, BAE Systems, UK, <[chris.dearlove@baesystems.com](mailto:chris.dearlove@baesystems.com)>  
 Ulrich Herberg, Fujitsu Laboratories of America, USA, <[ulrich@herberg.name](mailto:ulrich@herberg.name)>  
 Satoh Hiroki, Hitachi SDL, Japan, <[hiroki.satoh.yi@hitachi.com](mailto:hiroki.satoh.yi@hitachi.com)>  
 Philippe Jacquet, Alcatel Lucent Bell Labs, France, <[philippe.jacquet@alcatel-lucent.fr](mailto:philippe.jacquet@alcatel-lucent.fr)>  
 Monden Kazuya, Hitachi SDL, Japan, <[kazuya.monden.vw@hitachi.com](mailto:kazuya.monden.vw@hitachi.com)>  
 Kenichi Mase, Niigata University, Japan, <[mase@ie.niigata-u.ac.jp](mailto:mase@ie.niigata-u.ac.jp)>  
 Ryuji Wakikawa, Toyota, Japan, <[ryuji@sfc.wide.ad.jp](mailto:ryuji@sfc.wide.ad.jp)>

## 26. Благодарности

Авторы благодарят команду людей за рамками OLSRv1, указанных в RFC 3626, включая Anis Laouiti (INT), Pascale Minet (INRIA), Paul Muhlethaler (INRIA), Amir Qayyum (M.A. Jinnah University), Laurent Viennot (INRIA) за вклад в работу.

Авторы благодарны за технические дискуссии, рецензии и комментарии к спецификации и её компонентам Khaldoun Al Agha (LRI), Teco Boot (Infinity Networks), Ross Callon (Juniper), Song-Yean Cho (Samsung), Alan Cullen (BAE Systems), Louise Lamont (CRC), Li Li (CRC), Joseph Macker (NRL), Richard Ogier (SRI), Charles E. Perkins (Futurewei), Henning Rogge (Frauenhofer FKIE) (указаны в алфавитном порядке) и всю рабочую группу IETF MANET.

В заключение авторы хотели бы выразить свою признательность руководителям направлений за ценные комментарии в процессе оценки IESG, в частности, (по алфавиту) Benoit Claise, Adrian Farrel, Stephen Farrell, Barry Leiba, Pete Resnick, Martin Stiemering.

## 27. Литература

### 27.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", [RFC 5148](#), February 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, [RFC 5226](#), May 2008.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", [RFC 5444](#), February 2009.
- [RFC5497] Clausen, T. and C. Dearlove, "Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)", RFC 5497, March 2009.
- [RFC5498] Chakeres, I., "IANA Allocations for Mobile Ad Hoc Network (MANET) Protocols", [RFC 5498](#), March 2009.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", [RFC 6130](#), April 2011.

[RFC7182] Herberg, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", RFC 7182, April 2014.

[RFC7183] Herberg, U., Dearlove, C., and T. Clausen, "Integrity Protection for the Neighborhood Discovery Protocol (NHDP) and Optimized Link State Routing Protocol Version 2 (OLSRv2)", RFC 7183, April 2014.

## 27.2. Дополнительная литература

- [BCP107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.
- [FSLs] Santivanez, C., Ramanathan, R., and I. Stavrakakis, "Making Link-State Routing Scale for Ad Hoc Networks", MobiHoc '01, Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, 2001.
- [FSR] Pei, G., Gerla, M., and T. Chen, "Fisheye State Routing in Mobile Ad Hoc Networks", ICDCS Workshop on Wireless Networks and Mobile Computing, 2000.
- [HIPERLAN] ETSI, "Radio Equipment and Systems (RES); High Performance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification", ETSI 300-652, June 1996.
- [HIPERLAN2] Jacquet, P., Minet, P., Muhlethaler, P., and N. Rivierre, "Increasing Reliability in Cable-Free Radio LANs: Low Level Forwarding in HIPERLAN", Wireless Personal Communications, Volume 4, Issue 1, 1997.
- [MPR] Qayyum, A., Viennot, L., and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless Networks", INRIA, No. 3898, March 2000.
- [McCabe] McCabe, A., Dearlove, C., Fredin, M., and L. Axelsson, "Scalability modelling of ad hoc routing protocols - a comparison of OLSR and DSR", Scandinavian Wireless Adhoc Networks '04, 2004.
- [NIST-SP-800-38A] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: Methods and Techniques", Special Publication 800-38A, December 2001.
- [NIST-SP-800-38C] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality", Special Publication 800-38C, May 2004.
- [RC4] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C", Second Edition, John Wiley and Sons, New York, 1996.
- [RFC2501] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.
- [RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.
- [RFC3626] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, October 2003.
- [RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.
- [RFC6229] Strombergson, J. and S. Josefsson, "Test Vectors for the Stream Cipher RC4", RFC 6229, May 2011.

## Приложение А. Ограничения

Обновления Local Information Base, Neighborhood Information Base или Topology Information Base **должны** обеспечивать соблюдение всех ограничений, заданных этим приложением, а также [RFC6130]. Это вариант обработки, заданный в данном документе. Любое расширение протокола или внешний процесс, обновляющие Neighborhood Information Base или Topology Information Base, также **должны** соблюдать эти ограничения.

В каждом Originator Tuple:

- O\_orig\_addr **недопустимо** совпадать с любым другим O\_orig\_addr;
- O\_orig\_addr **недопустимо** совпадать с адресом источника для этого маршрутизатора.

В каждом Local Attached Network Tuple:

- AL\_net\_addr **недопустимо** совпадать с любым другим AL\_net\_addr;
- AL\_net\_addr **недопустимо** совпадать или быть поддиапазоном любого сетевого адреса в l\_local\_iface\_addr\_list любого Local Interface Tuple;
- AL\_net\_addr **недопустимо** быть адресом источника для этого маршрутизатора или O\_orig\_addr в любом Originator Tuple;
- AL\_dist **недопустимо** быть меньше 0.

В каждом Link Tuple:

- L\_neighbor\_iface\_addr\_list **недопустимо** включать какой-либо сетевой адрес, совпадающий с AL\_net\_addr любого Local Attached Network Tuple или являющийся его поддиапазоном;
- Если L\_in\_metric != UNKNOWN\_METRIC, значение L\_in\_metric **должно** должно быть представимо в определённой сжатой форме;
- Если L\_out\_metric != UNKNOWN\_METRIC, значение L\_out\_metric **должно** должно быть представимо в определённой сжатой форме;

- Если L\_mpr\_selector = true, то L\_status = SYMMETRIC.

В каждом Neighbor Tuple:

- N\_orig\_addr **недопустимо** менять на unknown;
- N\_orig\_addr **недопустимо** совпадать с адресом источника для этого маршрутизатора или O\_orig\_addr в любом Originator Tuple;
- N\_orig\_addr **недопустимо** совпадать с AL\_net\_addr в любом Local Attached Network Tuple;
- Если N\_orig\_addr != unknown, то N\_orig\_addr **недопустимо** быть N\_orig\_addr из любого другого Neighbor Tuple;
- N\_neighbor\_addr\_list **недопустимо** содержать какой-либо сетевой адрес, который включает адрес источника для этого маршрутизатора, O\_orig\_addr из любого Originator Tuple, совпадать или быть частью AL\_net\_addr в любом Local Attached Network Tuple;
- Если N\_orig\_addr = unknown, то N\_will\_flooding = WILL\_NEVER, N\_will\_routing = WILL\_NEVER, N\_flooding\_mpr = false, N\_routing\_mpr = false, N\_mpr\_selector = false, N\_advertised = false;
- N\_in\_metric **должно** совпадать с минимальным L\_in\_metric из всех соответствующих Link Tuple с L\_status = SYMMETRIC и L\_in\_metric != UNKNOWN\_METRIC (при наличии), иначе N\_in\_metric = UNKNOWN\_METRIC;
- N\_out\_metric **должно** совпадать с минимальным L\_out\_metric из всех соответствующих Link Tuple с L\_status = SYMMETRIC and L\_out\_metric != UNKNOWN\_METRIC (при наличии), иначе N\_out\_metric = UNKNOWN\_METRIC;
- N\_will\_flooding и N\_will\_routing должны быть из диапазона WILL\_NEVER - WILL\_ALWAYS, включительно;
- Если N\_flooding\_mpr = true, то N\_symmetric **должно** быть true, N\_out\_metric **недопустимо** быть UNKNOWN\_METRIC, а N\_will\_flooding **недопустимо** быть WILL\_NEVER;
- If N\_routing\_mpr = true, то N\_symmetric **должно** быть true, N\_in\_metric **недопустимо** быть UNKNOWN\_METRIC, N\_will\_routing **недопустимо** быть WILL\_NEVER;
- Если N\_symmetric = true и N\_flooding\_mpr = false, то N\_will\_flooding **недопустимо** быть WILL\_ALWAYS;
- Если N\_symmetric = true и N\_routing\_mpr = false, то N\_will\_routing **недопустимо** быть WILL\_ALWAYS;
- Если N\_mpr\_selector = true, то N\_advertised **должно** быть true;
- Если N\_advertised = true, то N\_symmetric **должно** быть true, а N\_out\_metric **недопустимо** быть UNKNOWN\_METRIC.

В каждом Lost Neighbor Tuple:

- NL\_neighbor\_addr **недопустимо** быть адресом источника для этого маршрутизатора, O\_orig\_addr любого другого Originator Tuple, адресом или частью AL\_net\_addr в любом Local Attached Network Tuple.

В каждом 2-Hop Tuple:

- N2\_2hop\_addr **недопустимо** быть адресом источника для этого маршрутизатора, O\_orig\_addr любого другого Originator Tuple, адресом или частью AL\_net\_addr в любом Local Attached Network Tuple;
- Если N2\_in\_metric != UNKNOWN\_METRIC, значение N2\_in\_metric **должно** должно быть представимо в определённой сжатой форме;
- Если N2\_out\_metric != UNKNOWN\_METRIC, значение N2\_out\_metric **должно** должно быть представимо в определённой сжатой форме.

В каждом Advertising Remote Router Tuple:

- AR\_orig\_addr **недопустимо** быть адресом из l\_local\_iface\_addr\_list с любым Local Interface Tuple или IR\_local\_iface\_addr в любом Removed Interface Address Tuple;
- AR\_orig\_addr **недопустимо** быть адресом источника для этого маршрутизатора или O\_orig\_addr любого Originator Tuple;
- AR\_orig\_addr **недопустимо** быть AL\_net\_addr из любого Local Attached Network Tuple;
- AR\_orig\_addr **недопустимо** быть AR\_orig\_addr из любого другого Advertising Remote Router Tuple.

В каждом Router Topology Tuple:

- **Должен** быть Advertising Remote Router Tuple с AR\_orig\_addr = TR\_from\_orig\_addr.
- TR\_to\_orig\_addr **недопустимо** быть адресом из l\_local\_iface\_addr\_list с любым Local Interface Tuple или IR\_local\_iface\_addr в любом Removed Interface Address Tuple;
- TR\_to\_orig\_addr **MUST NOT недопустимо** быть адресом источника для этого маршрутизатора или O\_orig\_addr любого Originator Tuple;
- TR\_to\_orig\_addr **недопустимо** быть AL\_net\_addr из любого Local Attached Network Tuple;
- Упорядоченной паре (TR\_from\_orig\_addr, TR\_to\_orig\_addr) **недопустимо** быть соответствующей парой любого другого Router Topology Tuple;
- TR\_seq\_number **недопустимо** быть больше AR\_seq\_number в Advertising Remote Router Tuple с AR\_orig\_addr = TR\_from\_orig\_addr;

- Значение TR\_metric **должно** должно быть представимо в определённой сжатой форме.

В каждом Routable Address Topology Tuple:

- **Должен** быть Advertising Remote Router Tuple с AR\_orig\_addr = TA\_from\_orig\_addr;
- Адрес TA\_dest\_addr **должен** быть маршрутизируемым;
- TA\_dest\_addr **недопустимо** перекрываться с любым сетевым адресом из I\_local\_iface\_addr\_list в любом Local Interface Tuple или IR\_local\_iface\_addr в любом Removed Interface Address Tuple;
- TA\_dest\_addr **недопустимо** включать адрес источника для этого маршрутизатора или O\_orig\_addr из любого Originator Tuple;
- TA\_dest\_addr **недопустимо** быть адресом или частью AL\_net\_addr любого Local Attached Network Tuple;
- Упорядоченной паре (TA\_from\_orig\_addr, TA\_dest\_addr) **недопустимо** быть соответствующей парой любого другого Attached Network Tuple;
- TA\_seq\_number **недопустимо** быть больше AR\_seq\_number в Advertising Remote Router Tuple с AR\_orig\_addr = TA\_from\_orig\_addr;
- Значение TA\_metric **должно** должно быть представимо в определённой сжатой форме.

В каждом Attached Network Tuple:

- **Должен** быть Advertising Remote Router Tuple с AR\_orig\_addr = AN\_orig\_addr;
- AN\_net\_addr **недопустимо** быть адресом или частью любого сетевого адреса из I\_local\_iface\_addr\_list в любом Local Interface Tuple, адресом или частью IR\_local\_iface\_addr в любом Removed Interface Address Tuple;
- AN\_net\_addr **недопустимо** быть адресом источника для этого маршрутизатора или O\_orig\_addr любого Originator Tuple;
- Упорядоченной паре (AN\_orig\_addr, AN\_net\_addr) **недопустимо** быть соответствующей парой любого другого Attached Network Tuple;
- AN\_seq\_number **недопустимо** быть больше AR\_seq\_number в Advertising Remote Router Tuple с AR\_orig\_addr = AN\_orig\_addr;
- AN\_dist **недопустимо** быть меньше 0;
- Значение AN\_metric **должно** должно быть представимо в определённой сжатой форме.

## Приложение В. Пример алгоритма для расчёта MPR

Ниже описан алгоритм, который **можно** применять для выбора MPR Set с учётом Neighbor Graph, как указано в параграфах 18.2 и 18.3.

Этот алгоритм выбирает MPR Set M, являющийся подмножеством набора N1, который является частью Neighbor Graph. Алгоритм предполагает, что подмножество I набора N1 заранее выбрано как MPR, т. е. M будет включать I.

### В.1. Дополнительные обозначения

Ниже приведены дополнительные обозначения (см. параграф 18.2), используемые алгоритмом.

**N**

Подмножество N2, состоящее из элементов y в N2 таких, что либо d1(y) не определено, либо в N1 имеется хотя бы одно x, для которого определено d(x,y) и d(x,y) < d1(y).

**D(x)**

Для элемента x в N1 число элементов y в N, для которых d(x,y) определено и имеет минимальное значение среди d(z,y) для всех z в N1.

**R(x,M)**

Для элемента x в N1 число элементов y в N, для которых d(x,y) определено минимальным среди d(z,y) для всех z в N1 и ни одно из таких минимальных значений не имеет z в M (отметим, что обозначением пустого множества 0 будет D(x) = R(x,0)).

### В.2. Алгоритм выбора MPR

Создание MPR Set M начинается с M := I.

1. В M добавляются все элементы x из N1, которые имеют W(x) = WILL\_ALWAYS.
2. Для каждого y в N, имеющего лишь 1 элемент x в N1, такой, что d2(x,y) определено, x добавляется в M.
3. Пока в N1 имеются элементы x с R(x,M) > 0:
  1. выбирается элемент x из N1 с R(x,M) > 0 и добавляется к M в приведённом ниже порядке:
    - наибольшее W(x);
    - наибольшее R(x,M);
    - наибольшее D(x);
    - любой выбор, который **может** основываться на других критериях (например, маршрутизатор **может** предпочесть соседа как MPR, если этот сосед уже выбран маршрутизатором как MPR того же типа,

**может** предпочесть соседа по свежести информации, **может** предпочесть соседа по продолжительности предыдущего выбора в качестве MPR), или **может** быть случайным.

4. **Необязательно.** Рассмотрим каждый элемент  $x$  в  $M$ , не входящий в  $I$ . Если  $x$  можно удалить из  $M$ , оставив его соответствующим определению MPR Set, этот элемент  $x$  удаляется из  $M$ . Элементы **можно** рассматривать в любом порядке, например, по возрастанию  $W(x)$ .

## Приложение С. Пример алгоритма для расчёта Routing Set

Описанная здесь процедура приведена в качестве примера расчёта Routing Set с использованием варианта алгоритма Dijkstra. Сначала все Routing Tuple удаляются, а затем с помощью выбора и определений приложения С.1 применяются процедуры из последующих параграфов (каждый считается «этапом» обработки).

### С.1. Локальные интерфейсы и соседи

Ниже перечислены принятые определения и варианты выбора.

1. Для каждого Local Interface Tuple выбирается сетевой адрес из его  $I\_local\_iface\_addr\_list$ .
2. Для каждого Link Tuple выбранный адрес из его соответствующего Local Interface Tuple определяется как выбранный локальный адрес для этого Link Tuple.
3. Для каждого Neighbor Tuple с  $N\_symmetric = true$  и  $N\_out\_metric \neq UNKNOWN\_METRIC$  выбирается Link Tuple с  $L\_status = SYMMETRIC$ , для которого это соответствующий Neighbor Tuple и  $L\_out\_metric = N\_out\_metric$ . Это будет выбранный Link Tuple для данного Neighbor Tuple.
4. Для каждого сетевого адреса ( $N\_orig\_addr$  или из списка  $N\_neighbor\_addr\_list$ , «адрес соседа») из Neighbor Tuple с  $N\_symmetric = true$  и  $N\_out\_metric \neq UNKNOWN\_METRIC$  выбирается Link Tuple («выбранный Link Tuple») из тех, для которых это соответствующий Neighbor Tuple,  $L\_status = SYMMETRIC$  и  $L\_out\_metric = N\_out\_metric$ , как показано ниже.
  1. Если есть Link Tuple, где  $L\_neighbor\_iface\_addr\_list$  содержит адрес соседа, выбирается этот Link Tuple.
  2. В остальных случаях берётся выбранный Link Tuple для этого Neighbor Tuple.

Затем для этого сетевого адреса выполняются указанные ниже действия.

3. Выбранный локальный адрес определяется как выбранный локальный адрес для выбранного Link Tuple.
4. Выбранный адрес канала определяется как адрес из  $L\_neighbor\_iface\_addr\_list$  выбранного Link Tuple, по возможности совпадающий с этим соседним адресом.
5. Предпочтение Routing Tuple определяется предпочтением для минимального  $R\_metric$ , затем для минимального  $R\_dist$ , а затем для соответствующих Neighbor Tuple, как указано ниже.
  - Больше значение  $N\_will\_routing$ .
  - $N\_mpr\_selector = true$  предпочтительней  $N\_mpr\_selector = false$ .

Отметим, что использовать **следует** предпочтительные Routing Tuple. **Должны** использоваться Routing Tuple с минимальной метрикой  $R\_metric$  MUST, это задано за пределами определения предпочтений. Реализация **может** изменить определение предпочтений (включая минимум для  $R\_dist$ ) без иного влияния на алгоритм.

### С.2. Добавление соседних маршрутизаторов

Приведённая ниже процедура выполняется однократно.

1. Для каждого Neighbor Tuple с  $N\_symmetric = true$  и  $N\_out\_metric \neq UNKNOWN\_METRIC$  добавляется Routing Tuple с:
  - $R\_dest\_addr := N\_orig\_addr$ ;
  - $R\_next\_iface\_addr :=$  выбранный адрес канала для  $N\_orig\_addr$ ;
  - $R\_local\_iface\_addr :=$  выбранный локальный адрес для  $N\_orig\_addr$ ;
  - $R\_metric := N\_out\_metric$ ;
  - $R\_dist := 1$ .

### С.3. Добавление удалённых маршрутизаторов

Приведённая ниже процедура выполняется однократно.

1. Добавляется метка *used* (применяется) или *unused* (не применяется) к каждому Routing Tuple. Начальным значением является *unused* (эти метки нужны лишь для алгоритма).
2. Если нет неиспользуемых Routing Tuple, этап завершается, иначе выполняется указанное ниже.
  1. Находится неиспользуемый Routing Tuple с минимальной  $R\_metric$  (при наличии нескольких выбирается любой) и помечается как текущий *current* Routing Tuple.
  2. Текущий Routing Tuple помечается как используемый (*used*).
  3. Для каждого Router Topology Tuple с  $TR\_from\_orig\_addr = R\_dest\_addr$  из текущего Routing Tuple:
    1. определяется
      - $new\_metric := R\_metric$  из текущего Routing Tuple +  $TR\_metric$ ;

- $new\_dist := R\_dist$  из текущего Routing Tuple + 1.
2. Если нет Routing Tuple с  $R\_dest\_addr = TR\_to\_orig\_addr$ , создаётся неиспользуемый Routing Tuple с
    - $R\_dest\_addr := TR\_to\_orig\_addr$ ;
    - $R\_next\_iface\_addr := R\_next\_iface\_addr$  из текущего Routing Tuple;
    - $R\_local\_iface\_addr := R\_local\_iface\_addr$  из текущего Routing Tuple;
    - $R\_metric := new\_metric$ ;
    - $R\_dist := new\_dist$ .
  3. В остальных случаях при наличии Routing Tuple с  $R\_dest\_addr = TR\_to\_orig\_addr$  и  $new\_metric < R\_metric$  или  $new\_metric = R\_metric$  и обновленный Routing Tuple будет предпочтительней, обновляется Routing Tuple, как показано ниже
    - $R\_next\_iface\_addr := R\_next\_iface\_addr$  из текущего Routing Tuple;
    - $R\_local\_iface\_addr := R\_local\_iface\_addr$  из текущего Routing Tuple;
    - $R\_metric := new\_metric$ ;
    - $R\_dist := new\_dist$ .

#### С.4. Добавление адресов соседей

Приведённая ниже процедура выполняется однократно.

1. Для каждого Neighbor Tuple с  $N\_symmetric = true$  и  $N\_out\_metric \neq UNKNOWN\_METRIC$ :
  1. Для каждого сетевого адреса («адрес соседа») в  $N\_neighbor\_addr\_list$  при несовпадении адреса соседа с  $R\_dest\_addr$  из какого-либо Routing Tuple добавляется Routing Tuple
    - $R\_dest\_addr := neighbor\ address$ ;
    - $R\_next\_iface\_addr :=$  выбранный адрес канала для адреса соседа;
    - $R\_local\_iface\_addr :=$  выбранный локальный адрес для адреса соседа;
    - $R\_metric := N\_out\_metric$ ;
    - $R\_dist := 1$ .

#### С.5. Добавление удалённых маршрутизируемых адресов

Приведённая ниже процедура выполняется однократно.

1. Для каждого Routable Address Topology Tuple при выполнении **обоих** приведённых ниже условий
  - $TA\_dest\_addr$  не совпадает с  $R\_dest\_addr$  ни одного из Routing Tuple, добавленных на предыдущих этапах;
  - $TA\_from\_orig\_addr$  совпадает с  $R\_dest\_addr$  из Routing Tuple («предыдущий Routing Tuple»), добавляется новый Routing Tuple с
    - $R\_dest\_addr := TA\_dest\_addr$ ;
    - $R\_next\_iface\_addr := R\_next\_iface\_addr$  из предыдущего Routing Tuple;
    - $R\_local\_iface\_addr := R\_local\_iface\_addr$  из предыдущего Routing Tuple;
    - $R\_metric := R\_metric$  из предыдущего Routing Tuple +  $TA\_metric$ ;
    - $R\_dist := R\_dist$  из предыдущего Routing Tuple + 1.

Может существовать несколько Routing Tuple, которые могут быть добавлены для  $R\_dest\_addr$  на этом этапе. В таких случаях, для каждого такого  $R\_dest\_addr$  **должен** добавляться Routing Tuple с минимальной  $R\_metric$  MUST, в остальных случаях **следует** добавлять предпочтительный Routing Tuple.

#### С.6. Добавление подключённых сетей

Приведённая ниже процедура выполняется однократно.

1. Для каждого Attached Network Tuple при выполнении **обоих** приведённых ниже условий
  - $AN\_net\_addr$  не совпадает с  $R\_dest\_addr$  ни одного из Routing Tuple, добавленных на предыдущих этапах;
  - $AN\_orig\_addr$  совпадает с  $R\_dest\_addr$  из Routing Tuple («предыдущий Routing Tuple»), добавляется новый Routing Tuple с
    - $R\_dest\_addr := AN\_net\_addr$ ;
    - $R\_next\_iface\_addr := R\_next\_iface\_addr$  из предыдущего Routing Tuple;
    - $R\_local\_iface\_addr := R\_local\_iface\_addr$  из предыдущего Routing Tuple;
    - $R\_metric := R\_metric$  из предыдущего Routing Tuple +  $AN\_metric$ ;

- R\_dist := R\_dist из предыдущего Routing Tuple + AN\_dist.

Может существовать несколько Routing Tuple, которые могут быть добавлены для R\_dest\_addr на этом этапе. В таких случаях, для каждого такого R\_dest\_addr **должен** добавляться Routing Tuple с минимальной R\_metric MUST, в остальных случаях **следует** добавлять предпочтительный Routing Tuple.

## С.7. Добавление соседей 2-Нор

Описанная ниже процедура **необязательна** в соответствии с параграфом 19.1 и **может** быть выполнена один раз.

1. Для каждого 2-Нор Tuple с N2\_out\_metric != UNKNOWN\_METRIC при выполнении **всех** приведённых условий
  - N2\_2hop\_addr является маршрутизируемым адресом;
  - N2\_2hop\_addr не совпадает с R\_dest\_addr ни одного из Routing Tuple, добавленных на предыдущих этапах;
  - Routing Tuple с R\_dest\_addr = N\_orig\_addr соответствующего Neighbor Tuple («предыдущий Routing Tuple») имеет R\_dist = 1,

добавляется новый Routing Tuple с

- R\_dest\_addr := N2\_2hop\_addr;
- R\_next\_iface\_addr := R\_next\_iface\_addr из предыдущего Routing Tuple;
- R\_local\_iface\_addr := R\_local\_iface\_addr из предыдущего Routing Tuple;
- R\_metric := R\_metric из предыдущего Routing Tuple + N\_out\_metric из соответствующего Neighbor Tuple;
- R\_dist := 2.

Может существовать несколько Routing Tuple, которые могут быть добавлены для R\_dest\_addr на этом этапе. В таких случаях, для каждого такого R\_dest\_addr **должен** добавляться Routing Tuple с минимальной R\_metric MUST, в остальных случаях **следует** добавлять предпочтительный Routing Tuple.

## Приложение D. Пример сообщения TC

Сообщения TC являются экземплярами сообщений [RFC5444]. Эта спецификация требует наличия в TC полей <msg-hop-limit> и <msg-orig-addr>. Поддерживаются сообщения TC с любой комбинацией оставшихся опций заголовка сообщения и кодирования адресов, разрешённых [RFC5444], которые переносят нужную информацию. В результате этого не существует способа представить, как выглядят все сообщения TC. Это приложение служит иллюстрацией сообщения TC. Точные значения полей приведены и разъяснены в тексте.

В TC 4-битовое поле флагов (Message Flags - MF) имеет значение 15, указывающее, что заголовок сообщения содержит адрес инициатора, ограничение числа интервалов и порядковый номер сообщения. 4-битовое поле размера адреса (Message Address Length - MAL) имеет значение 3, указывающее, что адреса в сообщении имеют размер 4 октета (в данном случае это IPv4). Общий размер сообщения составляет 75 октетов.

Сообщение имеет Message TLV Block с размером содержимого 17 октетов, включающий 4 TLV. Первые два TLV указывают срок действия и интервал времени для сообщения. Третий TLV указывает порядковый номер и передаёт 2 октета ANSN и (с нулевым типом расширения по умолчанию - COMPLETE) указывает полноту сообщения TC. Четвёртый TLV указывает намерения исходного маршрутизатора в части пересылки и маршрутизации (FWILL и RWILL, соответственно). Каждый блок TLV использует октет флагов (MTLVF) со значением 16, указывающим наличие Value, и отсутствие расширения типа или индексов начала и завершения. Первые 2 TLV имеют Value Length в 1 октет, остальные - 2.

Сообщение имеет два Address Block (это необязательно — информацию можно передать в одном Address Block, использование двух, что тоже разрешено, выбрано для иллюстрации). Первый Address Block содержит 3 адреса с октетом флагов (ABF) 128, поэтому включает раздел Head (2 октета), но не имеет раздела Tail, а разделы Mid имеют размер по 2 октета. Следующий TLV Block (13 октетов) содержит 2 TLV. Первым является NBR\_ADDR\_TYPE TLV с октетом флагов (ATLVF) 16, оказывающим одно поле Value без индексов. Таким образом, все эти адреса связаны с Value (Value Length = 1) ROUTABLE\_ORIG, т. е. являются адресами инициатора анонсов соседей, которые также являются маршрутизируемыми. Вторым является LINK\_METRIC TLV с октетом флагов (ATLVF) 20, указывающим Value для каждого адреса, т. е. общий размер Value Length = 6 октетов и каждый адрес, связанный с Value, имеет размер 2 октета. Эти поля Value показаны как имеющие 4 бита, указывающие, что они являются значениями метрики исходящих соседей и имеют 12 битов со значением метрики (первые 4 бита - показатель степени, остальные 8 - мантисса).

Второй Address Block содержит 1 адрес с октетом флагов (ATLVF) 176, указывающим наличие раздела Head (2 октета), разделом Tail (2 октета) со значениями 0 (не включены) и наличие размера префикса 16. Сетевой адрес в результате будет иметь вид Head.0.0/16. Следующий TLV Block (9 октетов содержимого) включает 2 TLV. Первый имеет октет флагов (ATLVF) 16, снова указывающий ненужность индексов, но с наличием поля Value (Value Length = 1), указывающего удалённость адреса числом интервалов (hop). Вторым является LINK\_METRIC TLV, как в первом Address TLV Block, но с октетом флагов (ATLVF) 16, указывающим наличие одного поля 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|          TC          | MF=15 | MAL=3 |          Message Length = 75 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Originator Address          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Hop Limit | Hop Count |          Message Sequence Number |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Message TLV Block Length = 17 | VALIDITY_TIME | MTLVF = 16 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Value Len = 1 | Value (Time) | INTERVAL_TIME | MTLVF = 16 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Value Len = 1 | Value (Time) | CONT_SEQ_NUM | MTLVF = 16 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Value Len = 2 |          Value (ANSN)          | MPR_WILLING |
+-----+-----+-----+-----+-----+-----+-----+-----+
| MTLVF = 16 | Value Len = 1 | FWILL | RWILL | Num Addrs = 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ABF = 128 | Head Len = 2 |          Head          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Mid          |          Mid          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Mid          | Address TLV Block Length = 13 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| NBR_ADDR_TYPE | ATLVF = 16 | Value Len = 1 | ROUTABLE_ORIG |
+-----+-----+-----+-----+-----+-----+-----+-----+
| LINK_METRIC | ATLVF = 20 | Value Len = 6 |0|0|0|1|Metric |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Metric (cont) |0|0|0|1|          Metric          |0|0|0|1|Metric |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Metric (cont) | Num Addrs = 1 | ABF = 176 | Head Len = 2 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Head          | Tail Len = 2 | Pref Len = 16 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Address TLV Block Length = 9 |          GATEWAY          | ATLVF = 16 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Value Len = 1 | Value (Hops) | LINK_METRIC | ATLVF = 16 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Value Len = 2 |0|0|0|1|          Metric          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## Приложение E. Управление потоком и контроль перегрузок

Проактивный характер протокола обеспечивает естественный контроль над потоком трафика управления. Маршрутизаторы передают управляющие сообщения с заданной частотой и ограниченными интервалами.

Этот протокол применяет [RFC6130] для локальной сигнализации, встраивая анонсы выбора MPR через простой Address Block TLV и анонсы намерений маршрутизатора (при наличии) как один блок Message TLV. Поэтому локальная сигнализации имеет характеристики и ограничения, описанные в [RFC6130].

Кроме того, применение MPR может существенно снизить издержки на сигнализацию за счёт распространения информации о состоянии каналов двумя способами, снижающими объем лавинной рассылки и топологических данных. Во-первых, при использовании лавинной рассылки MPR затраты на распространение информации о состоянии каналов по сети снижаются по сравнению с лавинной рассылкой «вслепую», поскольку сообщения о состоянии каналов передают лишь маршрутизаторы MPR. Во-вторых, снижается объем информации, которую маршрутизатор должен объявлять, - это лишь селекторы MPR данного маршрутизатора. Это снижает объем объявлений о состоянии каналов по сравнению с полной информацией о состоянии каналов. В частности, некоторым маршрутизаторам не нужно декларировать такую информацию. В плотных сетях снижение объема управляющего трафика может составлять несколько порядков по сравнению с протоколами маршрутизации, использующими лавинную рассылку вслепую [MPR]. Это свойство обеспечивает расширение полосы, доступной для трафика данных и снижает уровень насыщения сети.

Поскольку трафик управления является непрерывным и периодическим, это обеспечивает более стабильное качество каналов, применяемых для маршрутизации. Однако при использовании некоторых опций отдельные управляющие сообщения (HELLO или TC) могут быть преднамеренно отправлены раньше крайнего срока для повышения чувствительности протокола к изменениям топологии. Это может вызывать незначительный, кратковременный и локальный рост трафика управления, однако рост будет ограничен минимальным интервалом между сообщениями.

Маршрутизатор, понимающий, что сеть перегружена, может увеличить интервал между своими сообщениями. Если это делает большая часть или все маршрутизаторы в сети, общий трафик управления снизится. При использовании этой возможности маршрутизаторы должны ограничивать рост интервалов между сообщениями, чтобы не потерять информацию об изменении топологии. Следует отметить, что маршрутизаторы могут принимать такие решения независимо и не обязательно всем использовать одинаковые параметры, а также нет необходимости менять параметры всех маршрутизаторов одновременно.

### Адреса авторов

**Thomas Heide Clausen**

LIX, Ecole Polytechnique

Phone: +33 6 6058 9349

EMail: [T.Clausen@computer.org](mailto:T.Clausen@computer.org)



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

**Christopher Dearlove**

BAE Systems Advanced Technology Centre

West Hanningfield Road

Great Baddow, Chelmsford

United Kingdom

Phone: +44 1245 242194

E-Mail: [chris.dearlove@baesystems.com](mailto:chris.dearlove@baesystems.com)

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

**Philippe Jacquet**

Alcatel-Lucent Bell Labs

Phone: +33 6 7337 1880

E-Mail: [philippe.jacquet@alcatel-lucent.com](mailto:philippe.jacquet@alcatel-lucent.com)

**Ulrich Herberg**

Fujitsu Laboratories of America

1240 E. Arques Ave.

Sunnyvale, CA 94085

USA

E-Mail: [ulrich@herberg.name](mailto:ulrich@herberg.name)

URI: <http://www.herberg.name/>

**Перевод на русский язык**

Николай Малых

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)