

IPv6

4.0.1.1

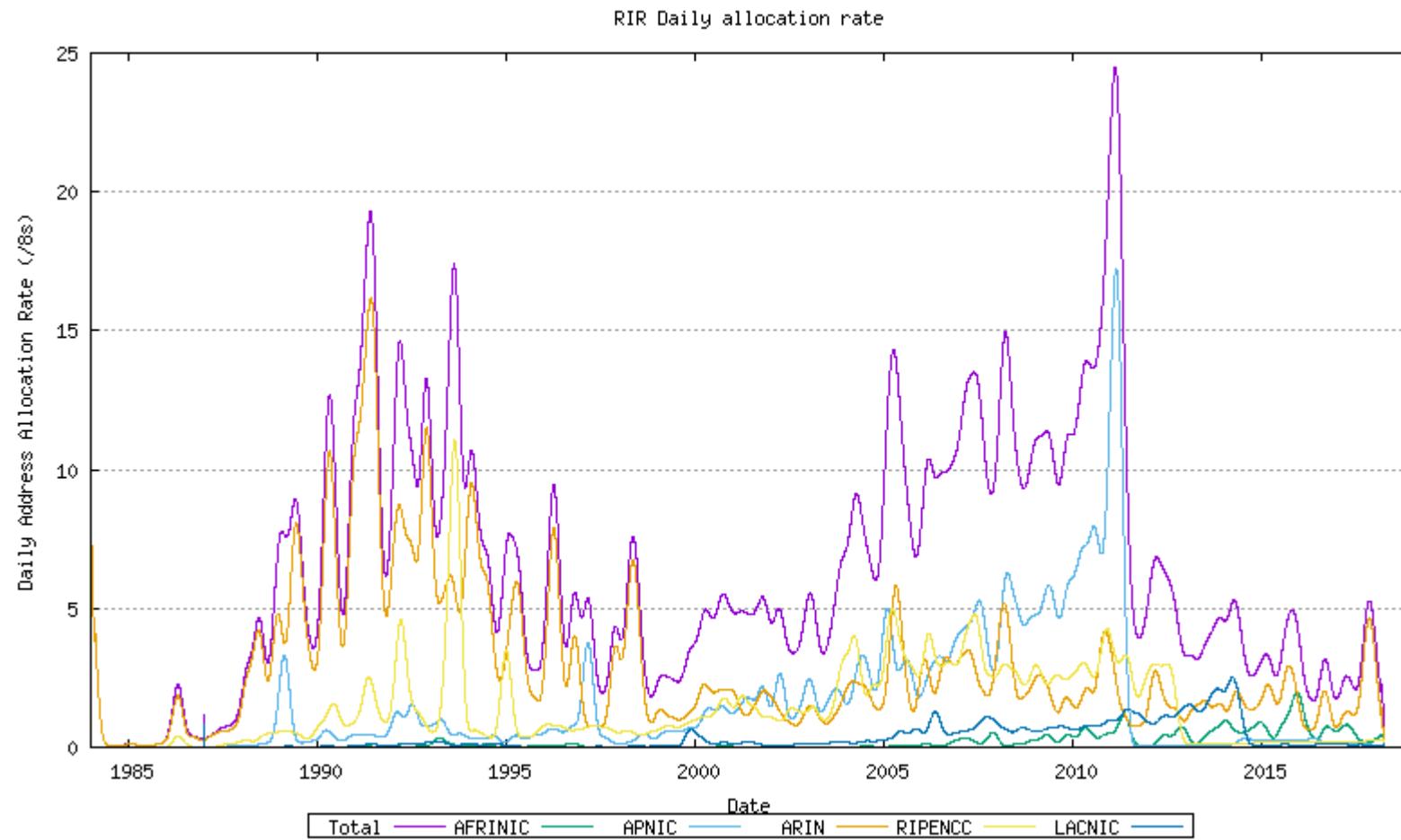
С февраля 2008 г. начат массированный переход к полномасштабному использованию адресации и маршрутизации IPv6.

Основной причиной этого перехода является резкий рост пропускной способности СрПД и производительности сетевого оборудования. Следует упомянуть также:

1. Необходимость учета опыта эксплуатации IPv4.
2. Необходимость поддержки новых применений.
3. Исчерпание публичных адресов IPv4.

Датой исчерпания адресов IPv4 принято считать 3 февраля 2011 г. (хотя в разных регионах ситуация разная).

4.0.1.2



Распределение адресов IPv4 [IANA]

4.0.1.3

Для правильного понимания IPv6 необходимо учесть «место» этого протокола в КС.

Подготовлен план перехода от IPv4 dominated Internet к IPv6 dominated Internet, предусматривающий три фазы (RFC 5211):

1. Preparation (подготовительная) -- до декабря 2009 г.
2. Transition (собственно переходная) -- январь 2010 г. -- декабрь 2011 г.
3. Post-Transition (пост-переходная) -- с января 2012 г. (окончание не оговаривается).

IPv6 продолжают дорабатывать (постоянно публикуют новые RFCs).

4.0.2.1

Наряду с общим сохранением преемственности, технологии IPv6 все-таки существенно отличаются от технологий IPv4.

Изменены как длина, так и формат адреса.

Формат представления и примеры записи одного и того же адреса IPv6:

X:X:X:X:X:X:X

1234:abcd:CDEF:0000:abEF:0000:0000:09aF

1234:abcd:cdef:0:abef::9af

где X – шестнадцатеричное (любой регистр) шестнадцатибитное число.

То есть общая длина адреса составляет 128 битов.

Поскольку часто встречаются длинные последовательности нулей, одно либо более рядом стоящих нулевых чисел можно сокращать как два двоеточия. Но нужно помнить об однозначности интерпретации адреса. Также можно не писать лидирующие нули в тетрадах.

4.0.2.2

По причине применения двоеточий в адресах IPv6 возникает необходимость устранения конфликтов при разборе адресов с указанием портов.

Пример URL с адресом IPv6 и портом:

`http://[fd00:0:0:1::80]:81...`

4.0.3.1

Изменен формат заголовка пакета.

Вместо заголовка фиксированной длины с фиксированными полями применяется гибкий базовый заголовок плюс набор необязательных заголовков различного формата.

4.0.3.2

IPv4 & IPv6 Header Comparison

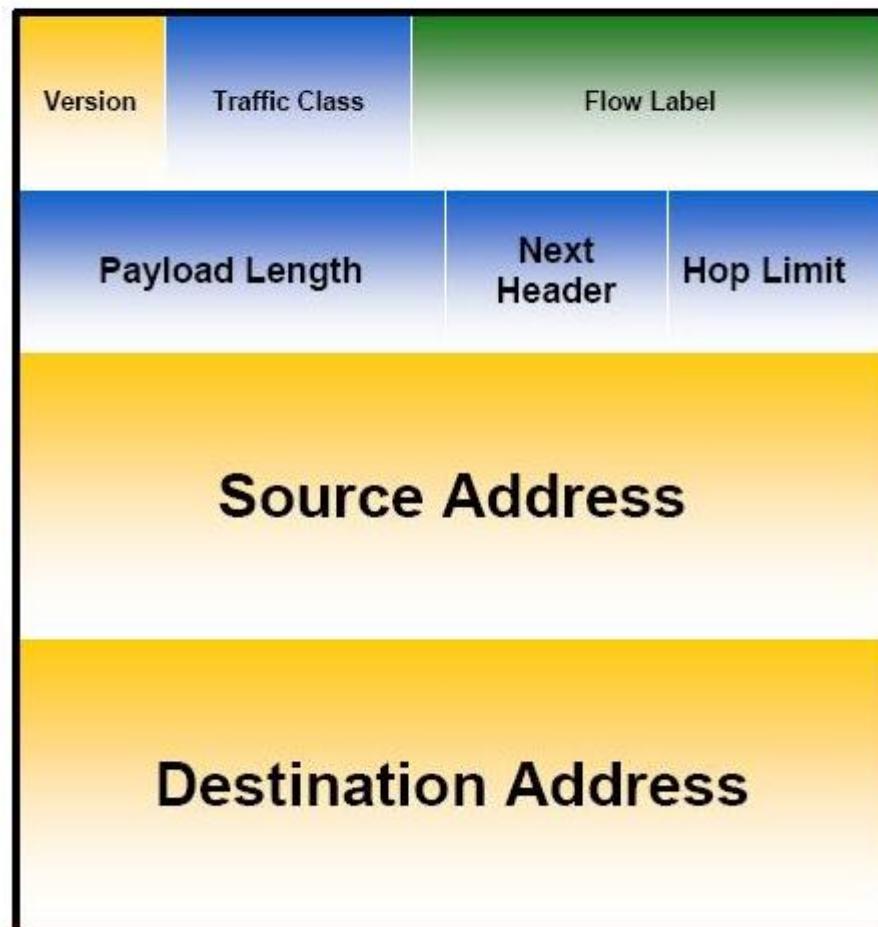
IPv4 Header



Legend

- field's name kept from IPv4 to IPv6
- fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6

IPv6 Header



[Cisco]

4.0.4.1

Изменена иерархия адресного пространства.

Применительно к IPv6-адресации механизм классов упразднен. Вместо классов широко применяется механизм *адресных префиксов* (address prefixes).

Имеются три базовых типа адресов:

1. Юникаст.
2. Мультикаст.
3. Эникаст.

Бродкаст-адресов нет вообще.

Базовые типы, как таковые, не используются. Их делят на виды согласно специфике применения. Принадлежность к тому либо иному виду определяется по адресному префиксу -- фиксированным начальным битам адреса. Для всех допустимых видов определены собственные форматы, выражющиеся в стандартных полях. Однозначно идентифицировать адрес можно только разобрав его полностью.

4.0.5.1

Изменен подход к назначению адресов сетевым интерфейсам.

Одному и тому же сетевому интерфейсу могут быть назначены несколько адресов различных типов. Допускается даже назначение более одного адреса одного типа и это вполне нормально. Если по правилам IPv4-адресации такой подход был скорей исключением, то здесь ситуация противоположная. Смысл: «Отдельная задача -- отдельный адрес». Приложение может использовать столько адресов, сколько ему нужно -- персональных либо разделяемых с другими приложениями.

Стандартные виды адресов имеют «предустановленный» смысл.

4.0.6.1

Модифицированы понятия сети и подсети.

Если в случае с IPv4 предусматривалась только одна глобальная сеть, то на базе IPv6 предполагается возможность построения нескольких независимых глобальных сетей.

Понятие подсети расширено. Стандартизированы следующие виды подсетей (RFC 7346), что, в частности, отражается в значениях специального введенных 4-битных полей Scope в форматах адресов некоторых видов:

0, F -- Reserved.

1 -- Interface-local.

2 -- Link-local.

3 -- Realm-local.

4 -- Admin-local.

5 -- Site-local.

6, 7, 9, A, B, C, D -- Unassigned (по своему усмотрению).

8 -- Organization-local.

E -- Global scope.

Таким образом «очерчивается круг», в пределах которого адрес валиден.

4.0.6.2

Особо следует выделить *линк* (link) -- подсеть размером в один сегмент.

4.0.7.1

Модифицировано понятие станции (узла).

Для ссылки на любой из видов пользовательских станций в основном используют обобщенный термин *хост* (host).

Вместо термина «шлюз» используют обобщенный термин *маршрутизатор* (router).

Частными случаями маршрутизатора являются *маршрутизатор следующего звена* (next-hop router) и *маршрутизатор по умолчанию* (default router).

4.0.8.1

Введены новые правила задания размера подсети.

Маска подсети, как таковая, аннулирована.

Размер подсети определяется по специальному варианту префикса -- префиксу *подсети* (subnet prefix) -- фиксированным начальным битам адресов из диапазона описываемой подсети.

Пример указания префикса IPv6 -- в данном случае префикса подсети (не CIDR):

fd00:0:0:6:: / 63 -- число фиксированных битов

4.0.8.2

Как и раньше, размер подсети либо определяется автоматически, либо его задают принудительно.

Автоматическое определение осуществляется согласно полям соответствующего формата вида адреса.

В ряде случаев, когда поля допускают «разбежку» (например, при проектировании собственной инфраструктуры), задание префикса подсети обязательно.

При выборе размера подсетей настоятельно рекомендуется не отходить от стандартных правил.

Например, минимальная физическая подсеть -- линк стандартизована только с префиксом подсети равным 64.

Но, например, Cisco в исключительных случаях допускает еще меньшие подсети (nibble boundary).

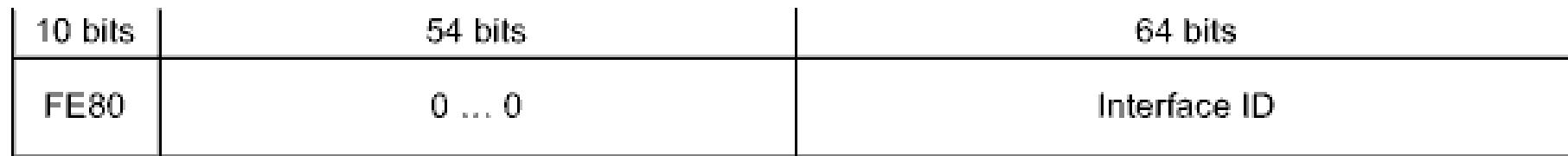
4.0.9.1

Локальное адресное пространство.

При IPv6-адресации приватные адреса, как таковые, не выделяют. Обобщенно их заменяют локальные (local) адреса.

Адрес вида Link-local Unicast ($FE80::/10$) (RFC 4291) предназначен для использования в пределах линка.

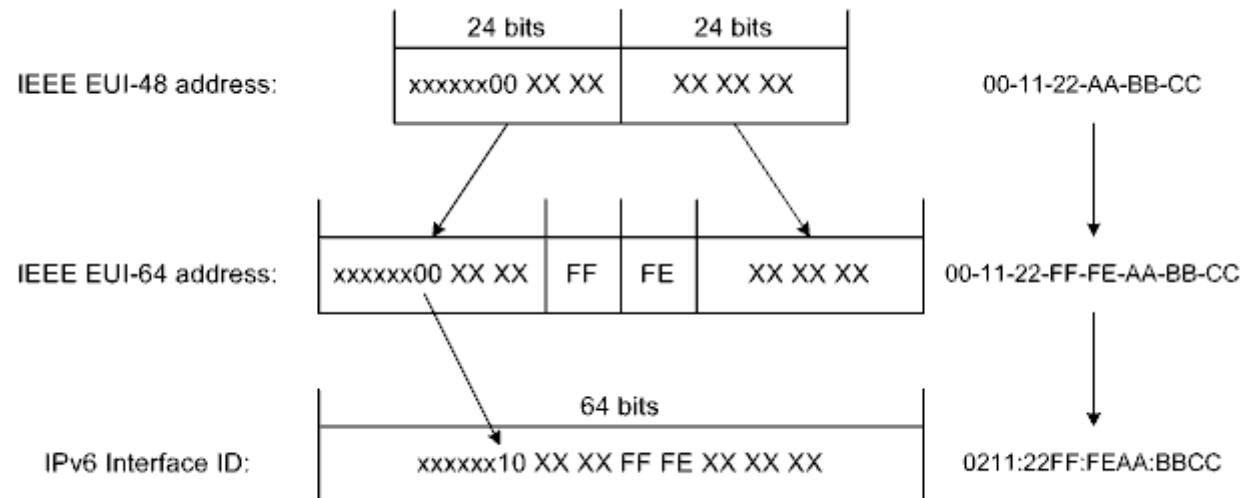
Выход пакетов с адресами Link-local Unicast (хотя бы в одном из полей) за пределы линков должен подавляться маршрутизаторами.



Как и в других юникаст-адресах, имеется четкое разделение на топологическую и интерфейсную части.

4.0.9.2

Адреса Link-local Unicast автоматически генерируются на базе MAC-адресов (что гарантирует их уникальность) **следующим образом.**



В результате, интерфейсная часть соответствует нотации EUI-64 (точнее, модифицированной нотации EUI-64).

От приведенного правила можно отступать, но это не рекомендуется.

4.0.9.3

А что делать если MAC-адреса нет?

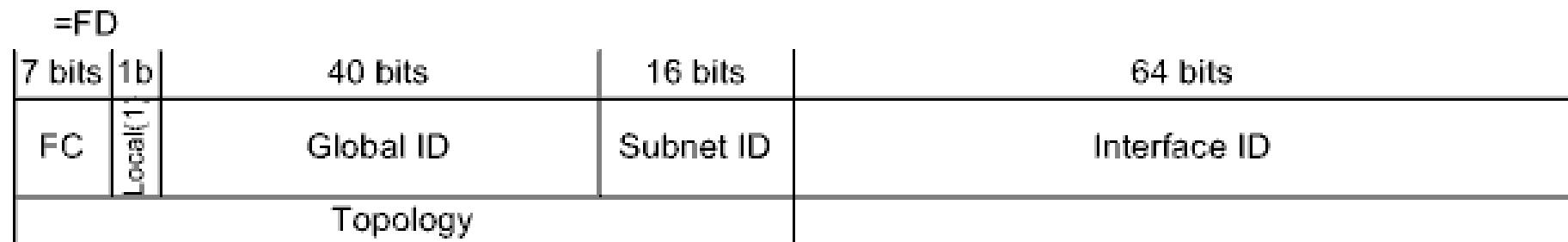
4.0.9.4

Если у сетевого интерфейса нет MAC-адреса, то в качестве «затравки» рекомендуется использовать MAC-адрес другого сетевого интерфейса либо какой-нибудь уникальный идентификатор станции (по возможности).

4.0.9.5

Для всех организаций, имеющих более или менее иерархическую подсетевую структуру и не испытывающих потребность во внешнем трафике, в качестве основной замены приватных адресов IPv4 позиционируют адреса вида Unique Local Unicast ($\text{FC}00::/7$) (RFC 4193).

Пакеты с адресами Unique Local Unicast должны подавляться всеми маршрутизаторами кроме внутренних.



Для всех юникаст-адресов, в том числе Unique Local Unicast, приемлема (но не всегда удобна) EUI-64-нотация интерфейсной части.

4.0.9.6

Выглядит немного странно, однако предусмотрен глобальный идентификатор в связке с флагом локальности -- может быть востребовано для связывания разрозненных частей внутренней сети через публичную.

Согласно стандарту глобальный идентификатор должен быть случайным числом (что с высокой степенью вероятности обеспечивает уникальность адресов). Но, в цельных внутренних сетях, учитывая, что в настоящее время разрешено использование адресов Unique Local Unicast только с установленным флагом локальности, с целью обеспечения удобства администрирования глобальные идентификаторы часто обнуляют.

4.0.9.7

Одно время выделялись еще адреса вида Site-local Unicast ($\text{FEC}0::/10$) (RFC 4291, RFC 3879), но теперь они отменены и их считают невалидными.

4.0.10.1

Глобальное адресное пространство.

В качестве основной замены публичных адресов IPv4 предлагают адреса вида Global Unicast (RFC 4291).

| 3 bits | 45 bits | 16 bits | 64 bits |
|----------|-----------------------------------|----------------------|--------------|
| 001 | Global Routing Prefix (public) | Subnet ID (local) | Interface ID |
| Topology | | | |

Формат адресов Global Unicast претерпел эволюцию. Выше приведена современная трактовка -- собственно Global Unicast ($2000::/3$) (RFC 3587). Непосредственными предшественниками были адреса Aggregatable Global Unicast (так же $2000::/3$) (RFC 2374), которые еще раньше сменили адреса Provider-based Unicast ($4000::/3$) (RFC 2073).

4.0.10.2

IPv6 Global Unicast Address Assignments

Last Updated
2015-03-24

Registration Procedure(s)
Allocations to RIRs are made in line with the Global Policy published at [<http://www.icann.org/en/resources/policy/global-addressing>]. All other assignments require IETF Review.

Description
The allocation of Internet Protocol version 6 (IPv6) unicast address space is listed here. References to the various other registries detailing the use of the IPv6 address space can be found in the [[IPv6 Address Space registry](#)].

Reference
[[RFC7249](#)]

Note
The assignable Global Unicast Address space is defined in [[RFC4291](#)] as being the address block defined by the prefix 2000::/3. All address space in this block not listed in the table below is reserved by IANA for future allocation.

Available Formats

[CSV](#) [XML](#) [HTML](#) [Plain text](#)

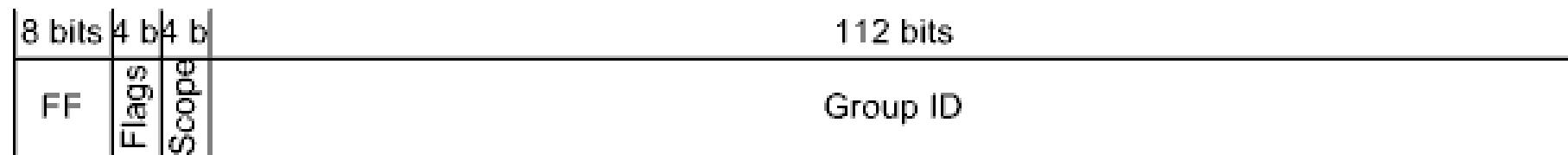
| Prefix | Designation | Date | WHOIS | RDAP | Status | Note |
|----------------|-------------|------------|------------------|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2001:0000::/23 | IANA | 1999-07-01 | whois.iana.org | | ALLOCATED | 2001:0000::/23 is reserved for IETF Protocol Assignments [RFC2928]. 2001:0000::/32 is reserved for TEREDO [RFC4380]. 2001:0002::/48 is reserved for Benchmarking [RFC5180]. 2001:10::/28 is reserved for ORCHID [RFC4843]. For complete registration details, see [IANA registry iana-ipv6-special-registry]. |
| 2001:0200::/23 | APNIC | 1999-07-01 | whois.apnic.net | | ALLOCATED | |
| 2001:0400::/23 | ARIN | 1999-07-01 | whois.arin.net | | ALLOCATED | |
| 2001:0600::/23 | RIPE NCC | 1999-07-01 | whois.ripe.net | | ALLOCATED | |
| 2001:0800::/23 | RIPE NCC | 2002-05-02 | whois.ripe.net | | ALLOCATED | |
| 2001:0a00::/23 | RIPE NCC | 2002-11-02 | whois.ripe.net | | ALLOCATED | |
| 2001:0c00::/23 | APNIC | 2002-05-02 | whois.apnic.net | | ALLOCATED | 2001:db8::/32 reserved for Documentation [RFC3849]. For complete registration details, see [IANA registry iana-ipv6-special-registry]. |
| 2001:0e00::/23 | APNIC | 2003-01-01 | whois.apnic.net | | ALLOCATED | |
| 2001:1200::/23 | LACNIC | 2002-11-01 | whois.lacnic.net | | ALLOCATED | |
| 2001:1400::/23 | RIPE NCC | 2003-02-01 | whois.ripe.net | | ALLOCATED | |
| 2001:1600::/23 | RIPE NCC | 2003-07-01 | whois.ripe.net | | ALLOCATED | |
| 2001:1800::/23 | ARIN | 2003-04-01 | whois.arin.net | | ALLOCATED | |
| 2001:1a00::/23 | RIPE NCC | 2004-01-01 | whois.ripe.net | | ALLOCATED | |
| 2001:1c00::/22 | RIPE NCC | 2004-05-04 | whois.ripe.net | | ALLOCATED | |
| 2001:2000::/20 | RIPE NCC | 2004-05-04 | whois.ripe.net | | ALLOCATED | |
| 2001:3000::/21 | RIPE NCC | 2004-05-04 | whois.ripe.net | | ALLOCATED | |
| 2001:3800::/22 | RIPE NCC | 2004-05-04 | whois.ripe.net | | ALLOCATED | |
| 2001:3c00::/22 | IANA | | | | RESERVED | 2001:3c00::/22 is reserved for possible future allocation to the RIPE NCC. |
| 2001:4000::/23 | RIPE NCC | 2004-06-11 | whois.ripe.net | | ALLOCATED | |

4.0.11.1

Мультикасты.

Адрес типа Multicast ($\text{FF00}::/8$) (RFC 4291) предназначен для использования в пределах подсети определенного вида и представляет собой уникальный в пределах таковой подсети групповой идентификатор.

Мультикаст-адреса могут присутствовать в пакетах только в поле Destination Address.



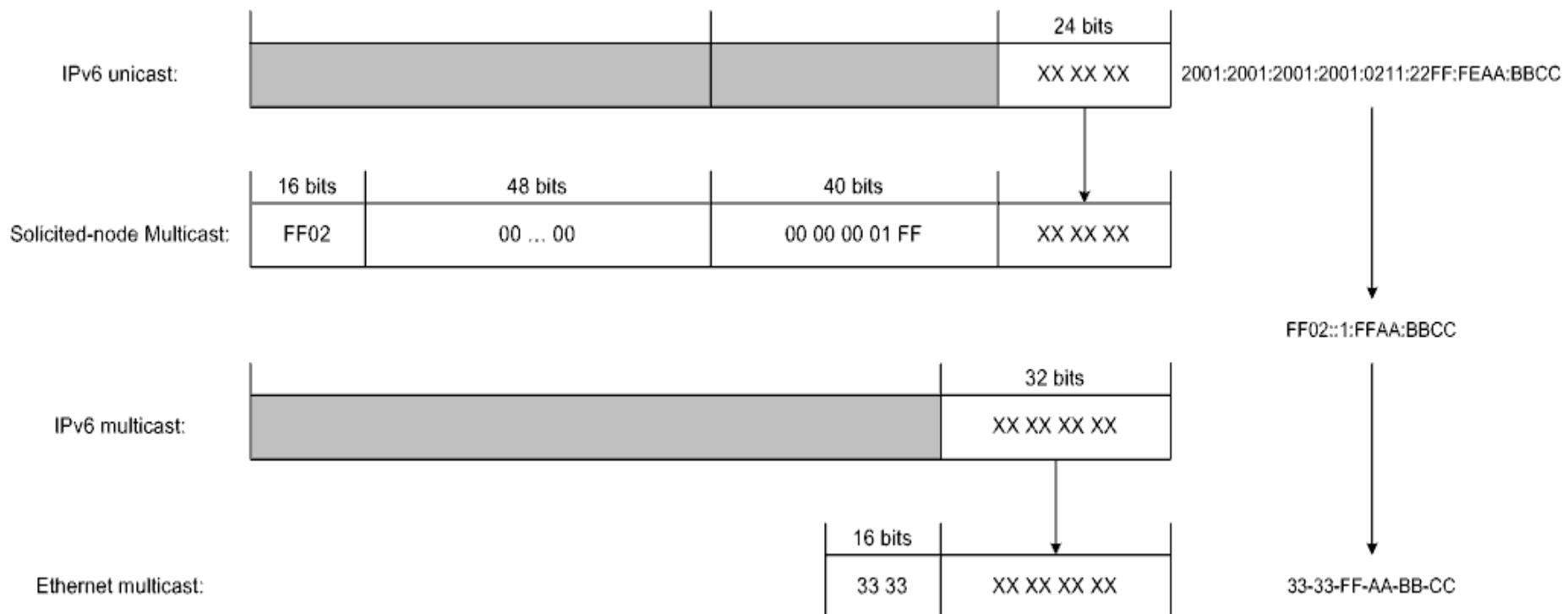
Примеры стандартных видов: Link-local All Nodes Multicast ($\text{FF02}::1/128$), Link-local All Routers Multicast ($\text{FF02}::2/128$), Site-local All Routers Multicast ($\text{FF05}::2/128$) и так далее.

4.0.11.2

Применительно к линку, в качестве замены широковещательных адресов IPv4 позиционируют адреса вида Link-local All Nodes Multicast.

4.0.11.3

Кроме того, при автоконфигурировании в пределах линка используются специальные адреса вида Solicited-node Multicast ($FF02::1:FF00/104$) (RFC 4291), строящиеся на основе адресов Link-local Unicast и других юниказст-адресов, из которых переносятся последние 24 бита.



4.0.12.1

Эникасты.

Применительно к IPv6 эникаст-адреса обладают двумя специфическими свойствами (так задумывалось):

Во-первых, если юникаст-адрес присвоить более чем одному сетевому интерфейсу в подсети, то он превращается в эникаст-адрес.

Во-вторых, критерием выбора эникаст-адреса является кратчайшее расстояния при маршрутизации.

Адрес типа Anycast (RFC 2526) предназначен для использования в пределах подсети и получается на основе префикса подсети.

| x bits | 121 – x bits | 7 bits |
|---------------|--------------|----------|
| Subnet Prefix | 1 ... 1 | Group ID |

Соответствующие приведенному выше формату, одному из двух форматов Reserved Subnet Anycast, виды пока применения не нашли.

4.0.12.2

Единственным используемым на практике видом является Subnet-router Anycast (RFC 4291).

| x bits | 128 – x bits |
|---------------|--------------|
| Subnet Prefix | 0 ... 0 |

Такие адреса разрешено назначать только сетевым интерфейсам маршрутизаторов и они могут присутствовать только в соответствующих служебных пакетах, причем только в поле Destination Address.

4.0.13.1

Специальные адреса.

Соглашения в области IPv6-адресации:

1. Unspecified (: : /128) (RFC 4291) -- адрес всех глобальных сетей.
2. Loopback (: : 1/128) (RFC 4291) -- адрес сетевого интерфейса -- заглушки.

4.0.14.1

Следует учитывать факт наличия и специальных сетей IPv6.

Протокол IPv6 прошел две основные фазы тестирования.

Первую -- собственно IPv6 Test Network (`5F00::/8`) (RFC 1897, RFC 2471).

И вторую -- очень известную в свое время экспериментальную сеть 6Bone (`3FFE::/16`) (RFC 2471, RFC 3701), интенсивно развивавшуюся в 2003 -- 2004 годах и свернутую 6 июня 2006 г.



Другие примеры.

IPv6 Benchmarking (`2001:0200::/48`) (RFC 5180).

IPv6 Documentation (`2001:DB8::/32`) (RFC 3849).

4.0.15.1

Internet Protocol Version 6 Address Space

(last updated 2008-05-13)

| IPv6 Prefix | Allocation | Reference | Note |
|-------------|----------------------|-----------|------|
| 0000::/8 | Reserved by IETF | [RFC4291] | |
| 0100::/8 | Reserved by IETF | [RFC4291] | |
| 0200::/7 | Reserved by IETF | [RFC4048] | |
| 0400::/6 | Reserved by IETF | [RFC4291] | |
| 0800::/5 | Reserved by IETF | [RFC4291] | |
| 1000::/4 | Reserved by IETF | [RFC4291] | |
| 2000::/3 | Global Unicast | [RFC4291] | |
| 4000::/3 | Reserved by IETF | [RFC4291] | |
| 6000::/3 | Reserved by IETF | [RFC4291] | |
| 8000::/3 | Reserved by IETF | [RFC4291] | |
| A000::/3 | Reserved by IETF | [RFC4291] | |
| C000::/3 | Reserved by IETF | [RFC4291] | |
| E000::/4 | Reserved by IETF | [RFC4291] | |
| F000::/5 | Reserved by IETF | [RFC4291] | |
| F800::/6 | Reserved by IETF | [RFC4291] | |
| FC00::/7 | Unique Local Unicast | [RFC4193] | |
| FE00::/9 | Reserved by IETF | [RFC4291] | |
| FE80::/10 | Link Local Unicast | [RFC4291] | |
| FEC0::/10 | Reserved by IETF | [RFC3879] | |
| FF00::/8 | Multicast | [RFC4291] | |

[IANA]

4.0.16.1

Введено понятие зоны.

Под зоной (zone) (RFC 4007) понимают некоторую условно выделенную, с целью обеспечения удобства администрирования, виртуальную подобласть в пределах области, очерченной подсетью определенного вида. Зоны не могут пересекаться.

Пример указания зоны IPv6:

fd00:0:0:1::11%5 -- число либо строка -- идентификатор зоны

В реализациях идентификаторы зон (zone, scope identifiers) обычно совпадают с внутренними идентификаторами либо названиями соответствующих сетевых интерфейсов.

Например, с помощью идентификатора зоны можно сослаться на нужный локальный адрес, если во внутренней сети локальные адреса повторяются.

4.0.17.1

Вопросы совместимости версий.

Все IP-станции делят на следующие стандартные типы (RFC 4213):

IPv4-only.

IPv6-only.

IPv6/IPv4.

IPv4 (IPv4-only либо IPv6/IPv4).

IPv6 (IPv6-only либо IPv6/IPv4).

4.0.17.2

Широко применяют туннелирование.

Фактически туннели выражаются в виде отдельных специализированных сетевых интерфейсов на взаимодействующих станциях.

Туннельный интерфейс должен быть привязан к физическому.

Туннели могут конфигурироваться как полностью «вручную» (manual tunnels), так и частично автоматически (automatic tunnels).

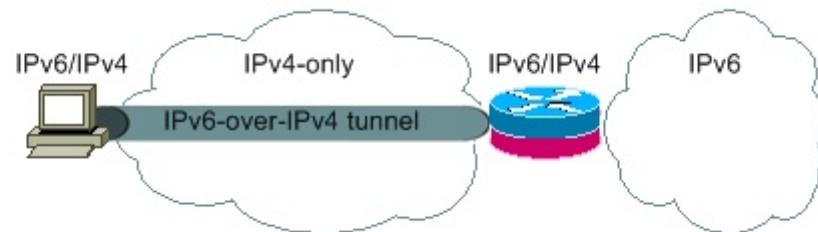
Сама природа туннеля подразумевает наличие у него граничных точек (endpoints). Для автоматических туннелей граничные точки задаются не «вручную», а определяются автоматически с помощью скрытых механизмов (но это не отменяет «ручное» конфигурирование других параметров).

Использование туннеля предполагает определенность всех необходимых параметров на всех задействованных станциях. Таким образом, туннельный интерфейс, в добавок к собственному адресу, имеет еще адрес граничной точки источника и, если туннель типа point-to-point, адрес граничной точки назначения.

4.0.17.3

В контексте совместимости IPv4 и IPv6, практический интерес представляет лишь возможность передавать трафик IPv6 посредством трафика IPv4 (RFC 4213), то есть организовывать туннели IPv6-over-IPv4. Таковые туннели делят на три типа:

1. Host-to-host.
2. Host-to-router (слева направо) и router-to-host (справа налево).



3. Router-to-router.

4.0.17.4

Для обеспечения совместимости с IPv4 стандартизированы следующие виды адресов IPv6.

Адрес вида IPv4-compatible (`::D.D.D.D/128`) (RFC 4291).

Включает публичный адрес IPv4.

В настоящее время использование этих адресов не рекомендуется.

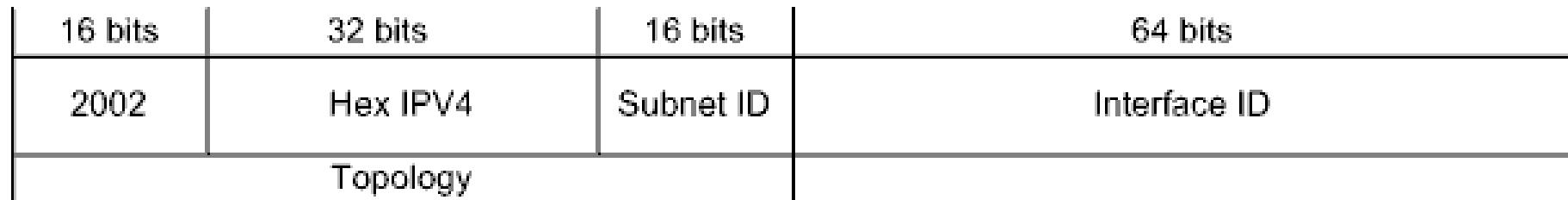
Адрес вида IPv4-mapped (`::FFFF:D.D.D.D/128`) (RFC 4291, RFC 4038).

Предназначен для использования при работе с виртуальной станцией IPv4 внутри станции IPv6.

В физических пакетах эти адреса недопустимы и в основных реализациях не поддерживаются.

4.0.17.5

Адрес вида 6to4 Unicast (RFC 3056).



Включает шестнадцатеричное представление публичного адреса IPv4 и предназначен для формирования автоматических туннелей.

Это один из видов туннельных адресов, поддерживаемый всеми основными реализациями.

4.0.17.6

Адрес вида ISATAP Unicast (RFC 5214).

`unicast_prefix::0:5EFE:D.D.D.D/96` -- с приватным адресом IPv4

`unicast_prefix::200:5EFE:D.D.D.D/96` -- с публичным адресом IPv4

Предназначен для использования одноименным протоколом (Intra-Site Automatic Tunnel Addressing Protocol) при формировании автоматических туннелей.

Это второй вид туннельных адресов, поддерживаемый всеми основными реализациями.

4.0.17.7

Адрес вида Teredo Unicast ($2001::/32$) (RFC 4380, RFC 5991, RFC 6081). Предназначен для реализации одноименной сложной клиент-серверной модели при формировании автоматических туннелей с учетом возможности многократной подмены самих адресов IPv4.

4.0.17.8

В свое время были стандартизированы и виды адресов IPv6 для туннелирования некоторых сторонних протоколов (например, IPX).

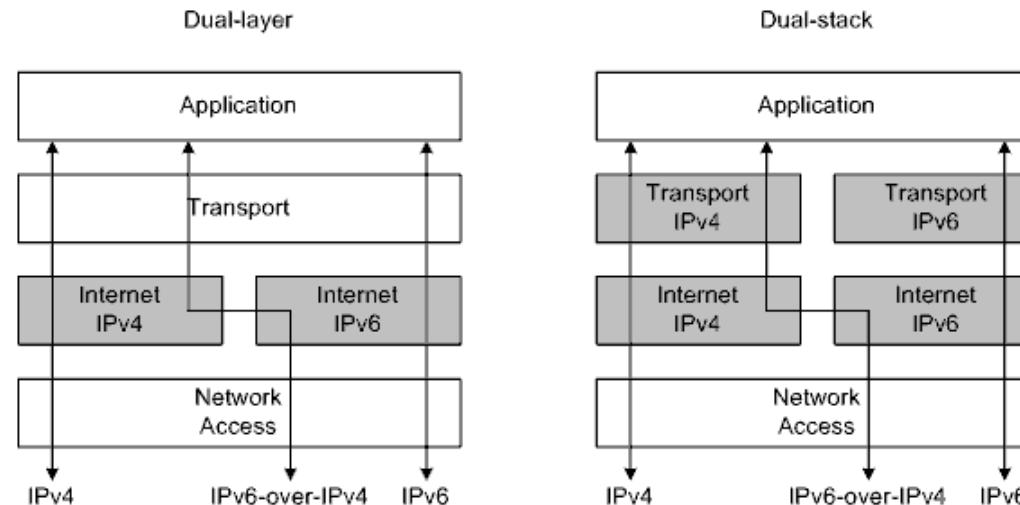
4.0.18.1

Замечание о стеках протоколов.

Вопросы совместимости IPv4 и IPv6 затрагивают работу всего семейства протоколов TCP/IPv6.

Выделяют две архитектуры (RFC 4038):

1. Dual-layer -- IPv4 и IPv6 разделены только на Internet-уровне модели TCP/IP.
2. Dual-stack -- стеки TCP/IPv4 и TCP/IPv6 полноценны и независимы.



В оптимальном случае трафик IPv6 полностью отделен от трафика IPv4.

4.0.19.1

Автоконфигурирование.

В сравнении с IPv4, возможности динамической IPv6-адресации значительно расширены и усовершенствованы, вплоть до полного автоконфигурирования.

Предусмотрены две базовых модели:

1. Stateless (RFC 4862) (часто используют сокращение SLAAC -- StateLess Address AutoConfiguration) -- распределенное управление, адреса и другие параметры конфигурируют с помощью служебных сообщений, базируется на ICMPv6.

2. Stateful -- централизованное управление, адреса и другие параметры передаются по специальному протоколу, базируется на DHCPv6.

Причем, в качестве приоритетной модели рассматривают первую, а не вторую.

4.0.19.2

ICMPv6 (RFC 4443), кроме всего прочего, включает в себя два мощных функционала:

1. Neighbor Discovery (ND) (RFC 4861) -- граничное обнаружение.
2. Multicast Listener Discovery (MLD) (RFC 2710) -- обнаружение мультикаст-станции-потребителя.

4.0.19.3

Для защиты от атак, связанных с перебором адресов, предусмотрены временные (temporary) юникаст-адреса (RFC 4861).

Интерфейсная часть временных адресов (и, опционально, постоянных) генерируется случайно для использования в течение ограниченного времени (privacy extensions).

Такие адреса имеют смысл только на стороне клиентов.

4.0.19.4

При разработке ND были четко сформулированы девять задач для решения в границах линка:

1. Обнаружение соседних маршрутизаторов.
2. Восстановление значений префиксов подсетей.
3. Восстановление значений некоторых других параметров (например, MTU).
4. Автоконфигурирование адресов.
5. Восстановление MAC-адресов соседних станций (вместо IPv4 ARP).
6. Обнаружение маршрутизаторов следующего звена (включая маршрутизатор по умолчанию).
7. Проверка достижимости соседних станций.
8. Проверка конфликтов адресов.
9. Оптимизация маршрутов (вместо ICMPv4 redirects).

4.0.19.5

Важно, что задачи ND решают именно в пределах линка.

Для обеспечения ND предусмотрены пять типов ICMPv6-сообщений:

- 133. RS (Router Solicitation).
- 134. RA (Router Advertisement).
- 135. NS (Neighbor Solicitation).
- 136. NA (Neighbor Advertisement).
- 137. Redirect.

Под *адвертейзингом* (advertising) понимают «предлагать услуги», а *солиситингом* (soliciting) -- «спрашивать об услугах».

RAs и Redirects передаются только маршрутизаторами, остальные ICMPv6-сообщения -- любыми станциями (и хостами, и маршрутизаторами).

По сути, протокол ND предназначен для пересылки значений требующихся ND-параметров и ND-опций. ND-опции вкладываются как унифицированные структуры (часто называют TLV -- от type, length, value).

4.0.19.6

| | | | |
|---------------------------|---------------------------|---------------------------|---------------------------|
| 0 | 1 | 2 | 3 |
| 0 1 2 3 4 5 6 7 8 9 0 | 1 2 3 4 5 6 7 8 9 0 | 1 2 3 4 5 6 7 8 9 0 | 1 |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |
| Type Code Checksum | | | |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |
| Reserved | | | |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |
| | | | |
| + | | | + |
| | | | |
| + | Target Address | | + |
| | | | |
| + | | | + |
| | | | |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |
| Options ... | | | |
| +-----+-----+-----+-----+ | | | |

| | | | |
|---------------------------|---------------------------|---------------------------|---------------------------|
| 0 | 1 | 2 | 3 |
| 0 1 2 3 4 5 6 7 8 9 0 | 1 2 3 4 5 6 7 8 9 0 | 1 2 3 4 5 6 7 8 9 0 | 1 |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |
| Type Length ... | | | |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |
| ~ | ... | ~ | ~ |
| +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ | +-----+-----+-----+-----+ |

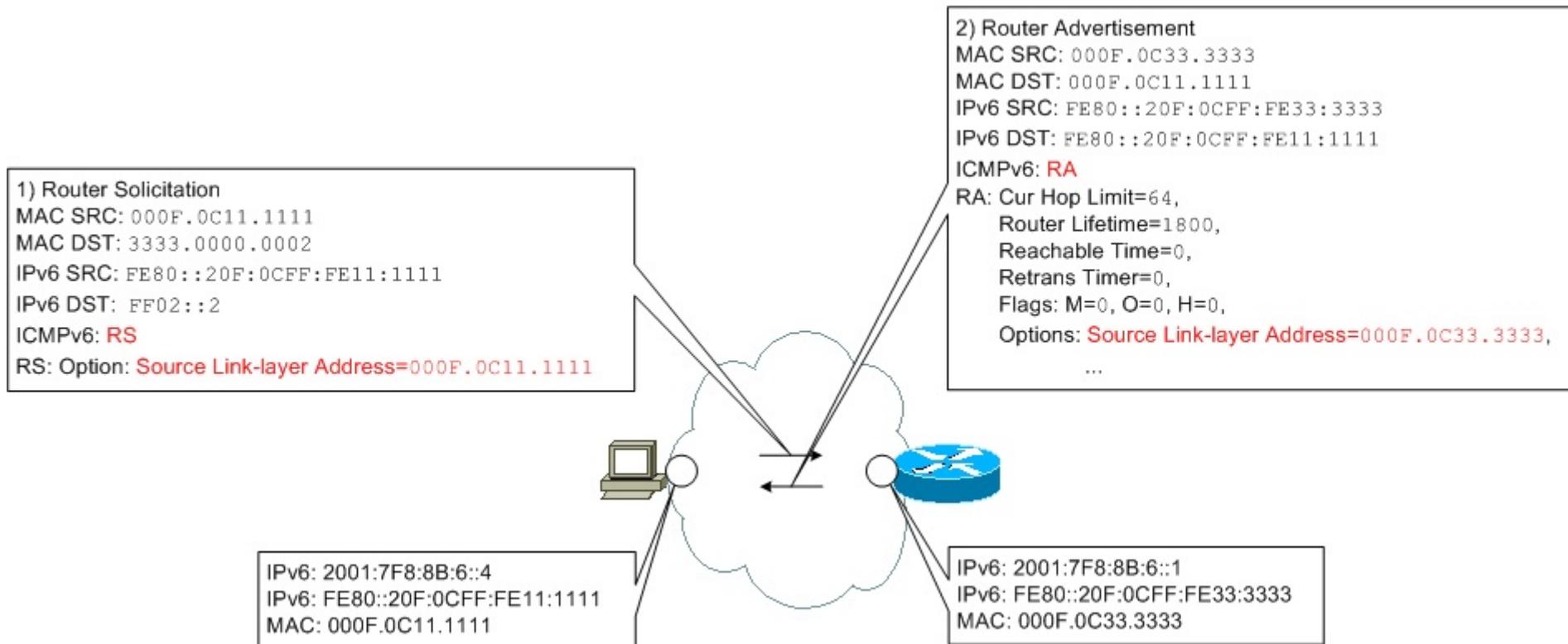
ICMPv6 message (ND Neighbor Solicitation) and ICMPv6 ND option [RFC]

4.0.19.7

ND -- это механизм, вполне допускающий конфигурирование. Многие параметры могут быть заданы.

4.0.19.8а

Для решения первой задачи используется связка RS и RA.



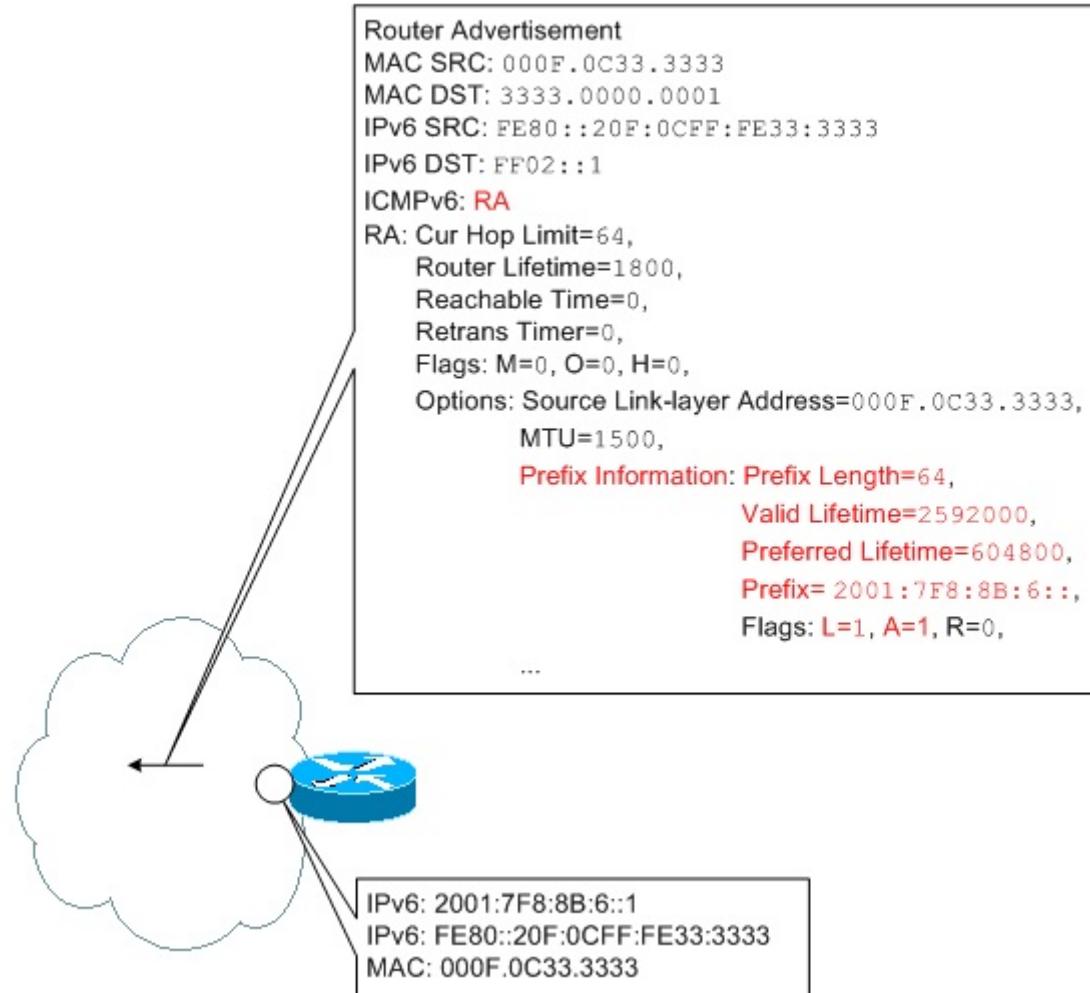
4.0.19.8b

Согласно стандарту ND, маршрутизаторы должны не только отвечать на RSes, а и периодически передавать RAs «на упреждение», анонсируя свое присутствие в линке (с IPv6-адресом назначения FF02::1).

Интенсивность передачи контролируется двумя таймерами: MaxRtrAdvInterval (не реже) и MinRtrAdvInterval (не чаще), согласно стандарту период может быть в диапазоне от 3 до 1800 s (по умолчанию 600 s).

4.0.19.9а

Хост (маршрутизатор) восстанавливает значения префиксов подсетей путем анализа RA.



4.0.19.9b

Отдельный префикс подсети анонсируется маршрутизатором в виде отдельной ND-опции Prefix Information со следующими ключевыми полями: Prefix Length -- длина префикса; Valid Lifetime -- общее время жизни (FFFFFFFFFFh -- бесконечность); Preferred Lifetime -- интервал времени, в течение которого адрес, сгенерированный на основе данного префикса подсети, будет считаться предпочтительным (FFFFFFFFFFh -- бесконечность); Prefix -- собственно префикс подсети; включая флаги: L (On-Link) -- данный префикс подсети относится к текущему линку; A (Autonomous Address-configuration) -- данный префикс подсети может быть использован для генерирования адресов.

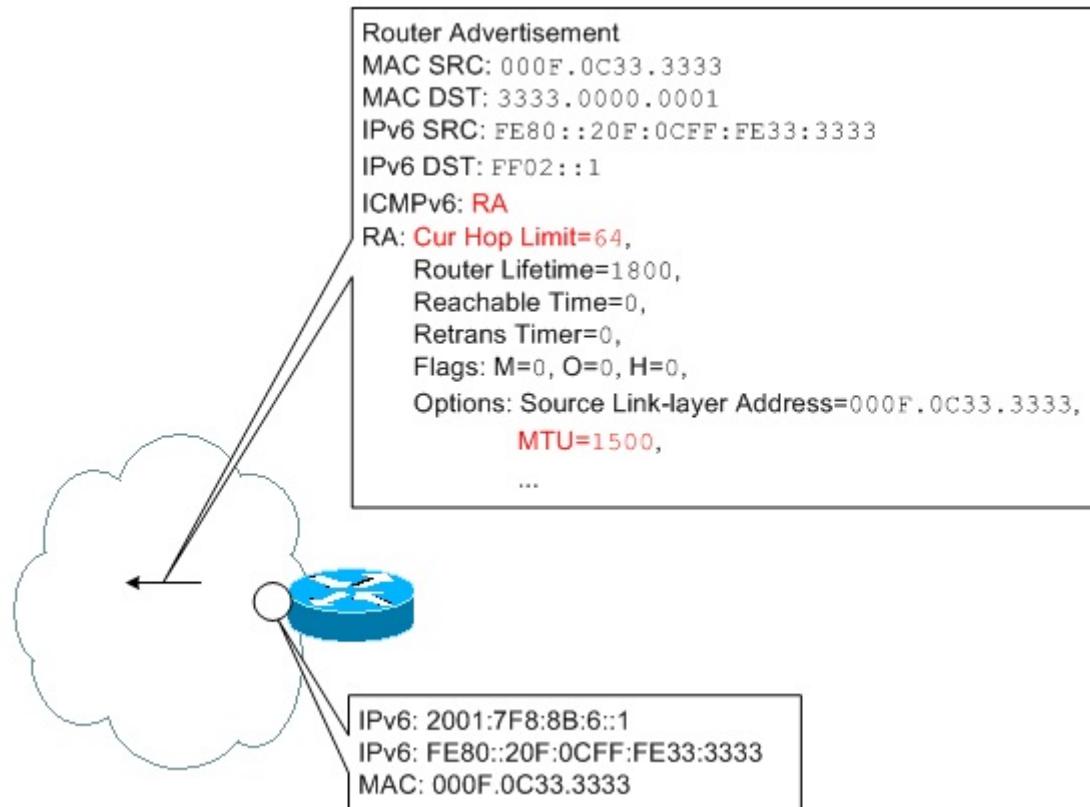
В RA вкладывается столько ND-опций, сколько нужно. Анонсируются все префиксы подсетей из привязанного к сетевому интерфейсу списка AdvPrefixList. Существует настоятельная рекомендация о том, что на маршрутизаторе в этот список по умолчанию вносятся префиксы всех подсетей, к которым относится сетевой интерфейс, исключая префиксы подсетей Link-local Unicast. При необходимости, список может быть дополнен «вручную».

4.0.19.9c

Какова же цель. В результате анализа RA, маршруты ко всем соответствующим подсетям автоматически вносятся в таблицу маршрутизации -- как маршруты к своим подсетям.

4.0.19.10a

Хост (маршрутизатор) восстанавливает значение еще двух важных параметров, опять же, путем анализа RA.



4.0.19.10b

Первым таковым параметром является Cur Hop Count. Значение будет вписываться в поле Hop Limit заголовка IPv6 каждого передаваемого маршрутизатору пакета (0 -- не определено).

Вторым параметром является MTU. В линках с вариативным MTU, например Ethernet, маршрутизатор обязан указывать (ND-опция).

4.0.19.11а

Важно, что в контексте SLAAC под автоконфигурированием адресов (в рамках автоконфигурирования) понимают автоматическое назначение сетевому интерфейсу юникаст-адресов (например, Global Unicast), не затрагивая адреса Link-local Unicast. Адреса Link-local Unicast также назначаются автоматически (на каждом сетевом интерфейсе IPv6), но вне рамок автоконфигурирования.

Топологическая часть адреса берется из ND-опции Prefix Information в RA от маршрутизатора, а для интерфейсной части используется нотация EUI-64 (гарантируется уникальность). При этом воспринимаются только префиксы подсети длиной 64 бита. Если маршрутизаторов несколько, то воспринимаются префиксы (и сопутствующие параметры) от всех маршрутизаторов.

Автоконфигурирование позиционируют прежде всего в отношении хостов, однако и сетевые интерфейсы маршрутизаторов могут быть подвержены автоконфигурированию. При этом соответствующие префиксы подсетей в RAs не включаются.

4.0.19.11b

SLAAC и DHCPv6 вполне совместимы друг с другом.

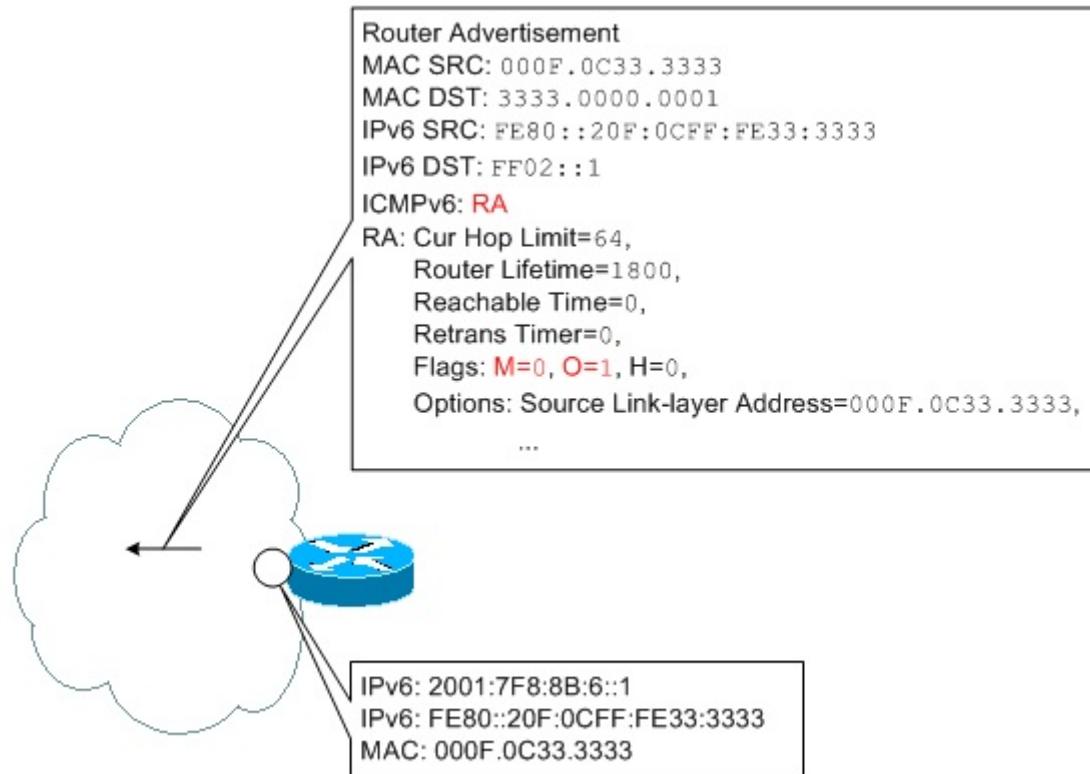
«Разделение обязанностей» контролируется двумя флагами в RA: M (Managed Address Configuration) -- адреса доступны посредством DHCPv6; O (Other Configuration) -- другие параметры доступны посредством DHCPv6 (кроме адресов) (если флаг M установлен, то флаг O игнорируется).

Конфигурации с установленным флагом M иногда называют stateful DHCPv6, а конфигурации с установленным флагом O -- stateless DHCPv6.

Автоконфигурирование адресов подразумевает и автоматическое нахождение маршрутизатора по умолчанию.

Адреса DNS-серверов автоматически могут быть получены только посредством DHCPv6.

4.0.19.11c



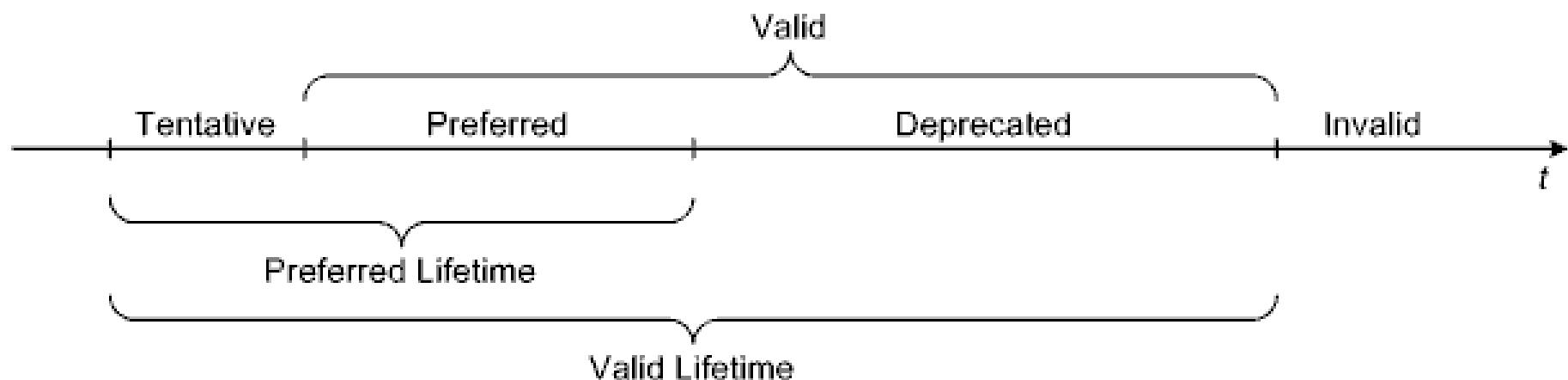
4.0.19.11d

Состояния адреса, полученного в результате автоконфигурирования:

1. Tentative -- уникальность адреса еще не проверена (протоколы вышестоящих уровней не могут использовать этот адрес).
2. Preferred -- адрес является предпочтительным (протоколы вышестоящих уровней могут использовать этот адрес без ограничений).
3. Deprecated -- использование адреса нежелательно (протоколы вышестоящих уровней не могут использовать этот адрес для создания новых соединений).
4. Valid -- адрес находится в состоянии Preferred либо Deprecated.
5. Invalid -- время жизни адреса истекло (протоколы вышестоящих уровней не могут использовать этот адрес).

4.0.19.11e

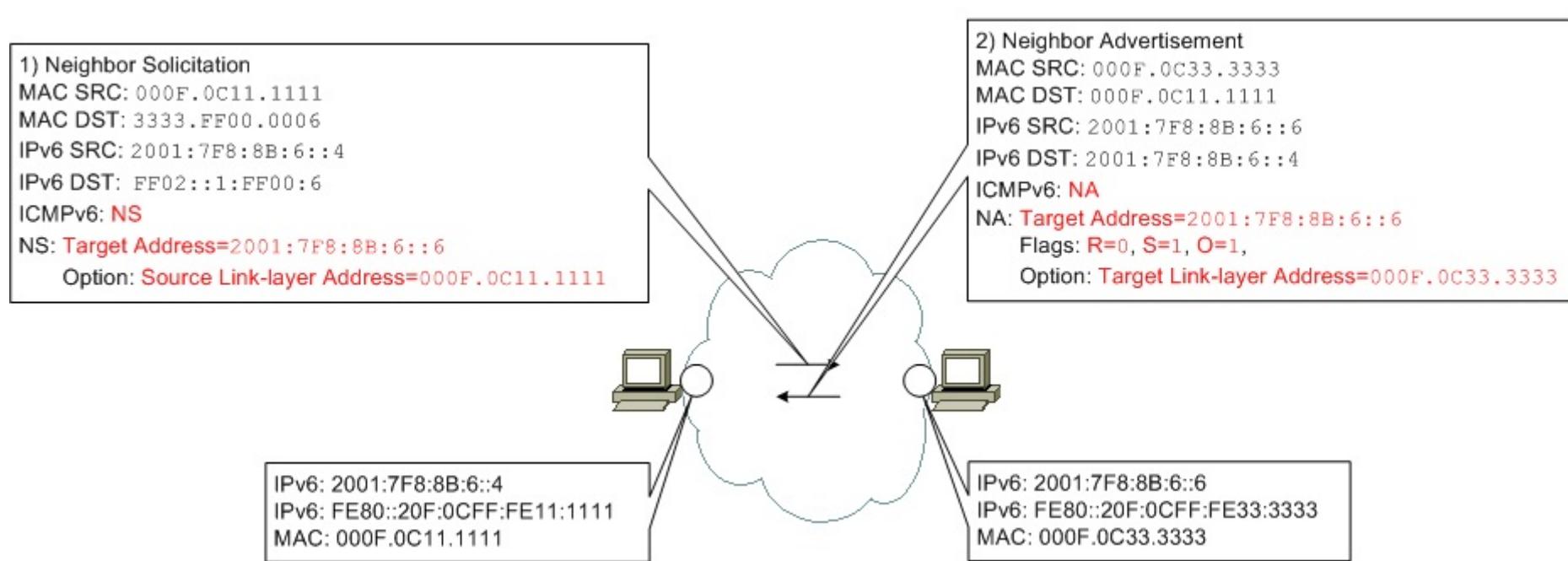
Валидность сгенерированных адресов контролируется двумя таймерами: Preferred Lifetime -- интервал времени, в течение которого адрес является предпочтительным (с охватом состояния Tentative); и Valid Lifetime -- интервал времени, равный собственно времени жизни адреса. Таймеры инициализируются исходя из значений соответствующих полей в сообщениях ND либо DHCPv6.



Жизненный цикл адреса при автоконфигурировании

4.0.19.12а

Для решения пятой задачи используется связка NS и NA.



NA содержит три флага: R (Router) -- данное NA передано маршрутизатором (не хостом), S (Solicited) -- данное NA передано в ответ на NS, O (Override) -- данное NA содержит новый MAC-адрес.

Опять же, стандарт ND не запрещает передавать NA «на упреждение» (IPv6-адресом назначения FF02::1).

4.0.19.12b

Почему возникла необходимость в адресе вида Solicited-node Multicast?

4.0.19.12с

Адрес вида **Solicited-node Multicast** используется только при решении пятой (восьмой) задачи -- чтобы уменьшить количество станций, которым необходимо обработать NS.

4.0.19.13а

Любой маршрутизатор, сам по себе, уже является потенциальным маршрутизатором следующего звена для соседних хостов (маршрутизаторов) -- если требуется послать пакет за пределы соответствующего линка.

Но ND нельзя рассматривать как альтернативу динамической маршрутизации. ND работает в рамках линка и, по понятным причинам, на ND не возлагают обязанности автоматического нахождения маршрутов к внешним подсетям.

А вот автоматически назначать маршрут по умолчанию ND может, и это является очень важной составляющей автоконфигурирования.

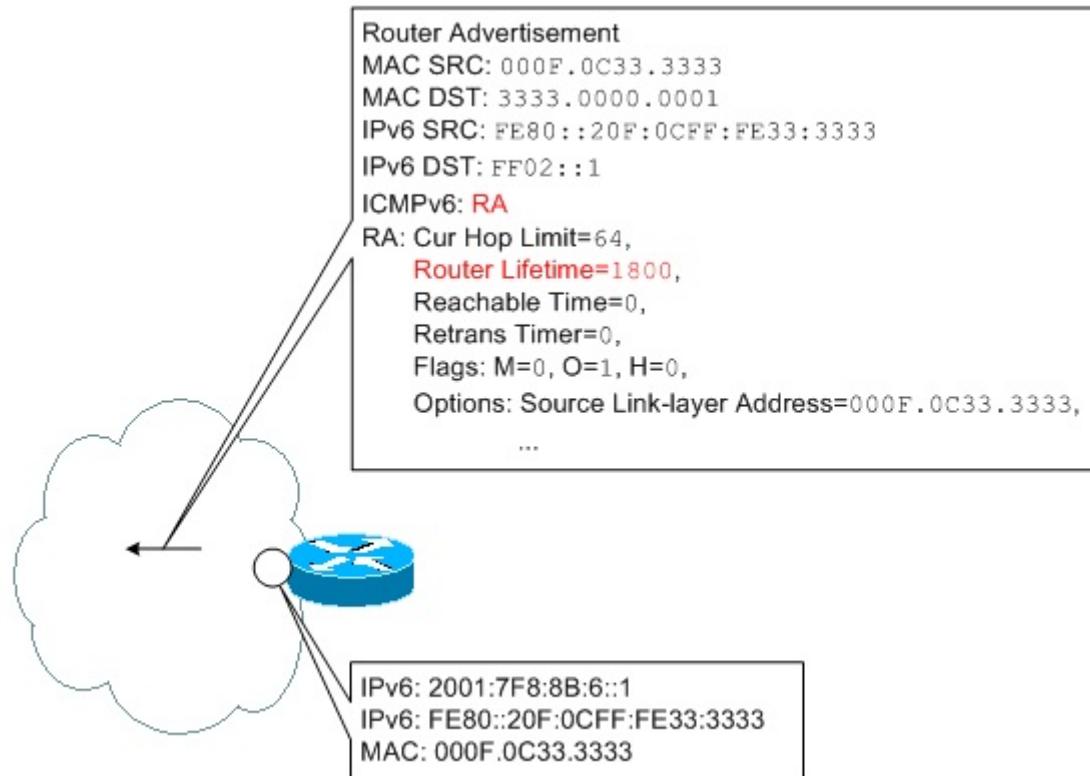
Более того, при автоконфигурировании все соседние маршрутизаторы автоматически рассматриваются как кандидаты в маршрутизаторы по умолчанию -- создается специальный список (*default router list*).

4.0.19.13b

Текущий маршрутизатор по умолчанию рекомендуется выбирать исходя из состояния связей (ND-кэша). А также исходя из значения специального поля в RA под названием Router Lifetime -- время жизни маршрутизатора (нулевое значение запрещает использовать маршрутизатор как маршрутизатор по умолчанию).

Если же в линке оказывается несколько равноценных маршрутизаторов, то очевидно возникает проблема выбора. В базовой редакции стандарта ND четко не оговорено поведение в таких случаях. В реализациях, как правило, выбирается первый «попавшийся» маршрутизатор.

4.0.19.13c



4.0.19.13d

В опциональном расширении ND (RFC 4191) предложено новое поле в RA под названием Default Router Preference -- приоритет маршрутизатора по умолчанию -- с тремя уровнями приоритетов: high, medium (по умолчанию) и low.

Также formalизованы три типа хостов:

- A. Игнорируют Default Router Preference.
- B. Учитывают только Default Router Preference (игнорируют другие параметры).

C. Учитывают как Default Router Preference, так и другие параметры.

Наконец, разрешено анонсировать посредством ND любые требующиеся маршруты (more-specific routes), для чего предусмотрена новая ND-опция под названием Route Information со следующими ключевыми полями под уже знакомыми названиями: Prefix Length, Route Preference (так же три уровня приоритета, 10b -- игнорировать приоритет), Route Lifetime, Prefix. Таковые маршруты вносятся в таблицу маршрутизации как маршруты к внешним подсетям через анонсировавший их маршрутизатор.

4.0.19.13e

Еще в одном optionalном расширении (RFC 4311) разрешена балансировка нагрузки.

4.0.19.13f

При нормальном завершении работы ОС или при административном выключении сетевого интерфейса маршрутизатор передает RA с нулевым значением поля Router Lifetime.

4.0.19.14а

Задача NUD (Neighbor Unreachability Detection) является закономерным «продолжением» задачи восстановления MAC-адресов и так же решается использованием связи NS (но не с мультикаст-, а с юникаст-адресом назначения) и NA.

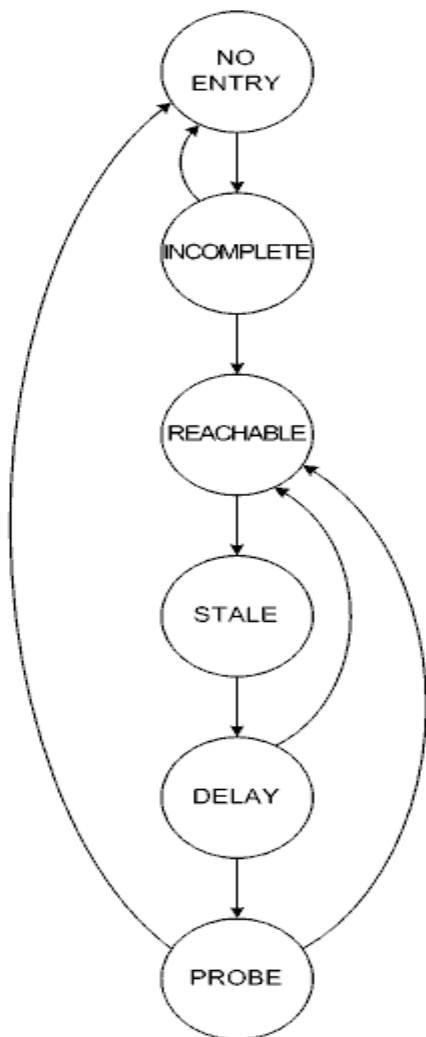
4.0.19.14b

Каждый сетевой интерфейс IPv6 должен иметь свой ND-кэш. ND-кэш напоминает ARP-таблицу.

Каждому из соседей в ND-кэше соответствует строка и одно из состояний:

1. INCOMPLETE -- сосед неизвестен, возникла необходимость передать ему пакет, идет восстановление его MAC-адреса.
2. REACHABLE -- сосед известен и считается достижимым.
3. STALE -- сосед известен, уже считается недостижимым, но нет необходимости передать ему пакет.
4. DELAY -- сосед известен, считается недостижимым, возникла необходимость передать ему пакет, пакет передан, ожидается подтверждение от протоколов вышестоящих уровней (именно так).
5. PROBE -- идет собственно проверка достижимости соседа.

4.0.19.14c



Тонкости условных переходов между состояниями зависят от реализации.

Диаграмма переходов между состояниями строки ND-кэша

4.0.19.14d

Заметно, что, в отличие от ARP, проверка достижимости соседа проводится, причем по мере надобности -- упор сделан на то, что сетевые интерфейсы способны сообщать о своем состоянии (исключая административное выключение).

Одна проверка достижимости соседа, как и одно восстановление МАС-адреса подразумевает несколько попыток (согласно стандарту по умолчанию три и три попытки соответственно).

4.0.19.14e

Алгоритм проверки достижимости опирается на два основных таймера: Reachable Time -- интервал времени после приема последнего сообщения NA от соседа, в течение которого этот сосед считается достижимым (согласно стандарту по умолчанию генерируется случайно в диапазоне от 15 до 45 s -- чтобы не порождать «штормы» NSes при большом количестве сетевых интерфейсов в линке); Retrans Timer -- интервал между передачей NSes при переходе к следующей попытке (согласно стандарту по умолчанию 1 s).

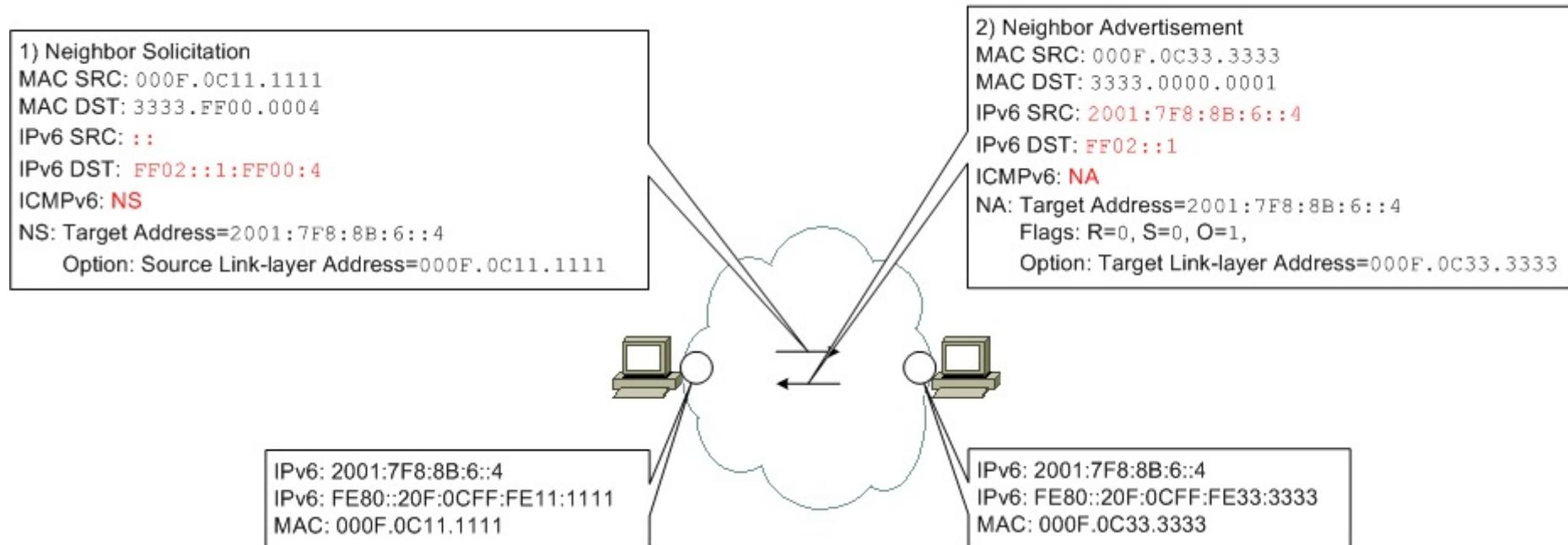
Маршрутизаторы могут предлагать значения этих таймеров в отношении линка анонсируя RAs с ненулевыми значениями одноименных полей (нулевые значения говорят о неопределенности со стороны маршрутизатора).

Reachable Time задает длительность состояния REACHABLE. Есть еще таймер, который задает длительность состояния DELAY (согласно стандарту по умолчанию 5 s).

4.0.19.15

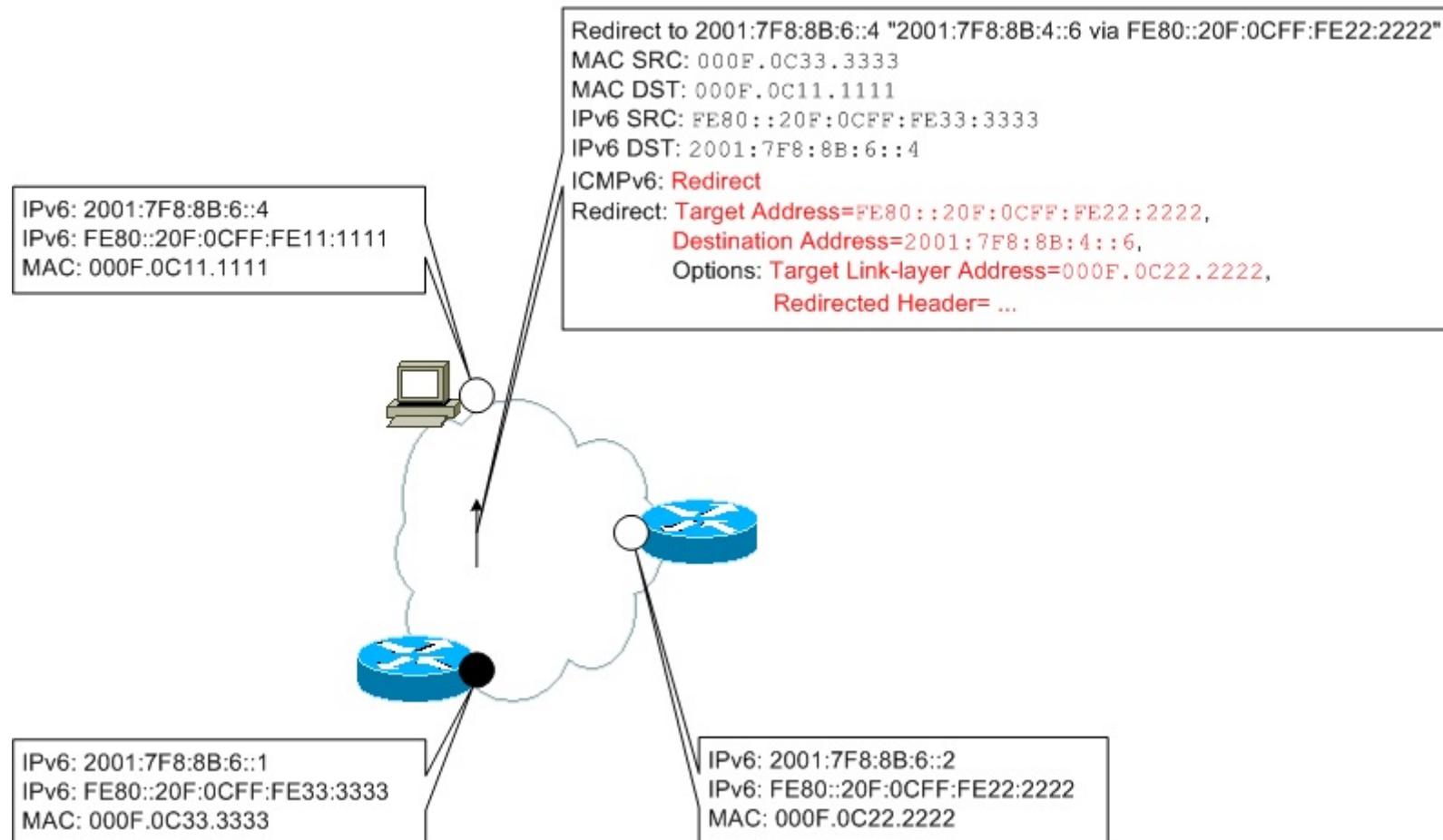
Не смотря на всю гибкость IPv6, проверку конфликта адресов никто не отменял -- исключая эникаст-адреса.

Задача DAD (Duplicate Address Detection) решается передачей специальным образом наполненного NS (с нулевым IPv6-адресом источника) и проверкой есть ли ответ.



4.0.19.16а

Для решения последней задачи используется специальное сообщение Redirect.



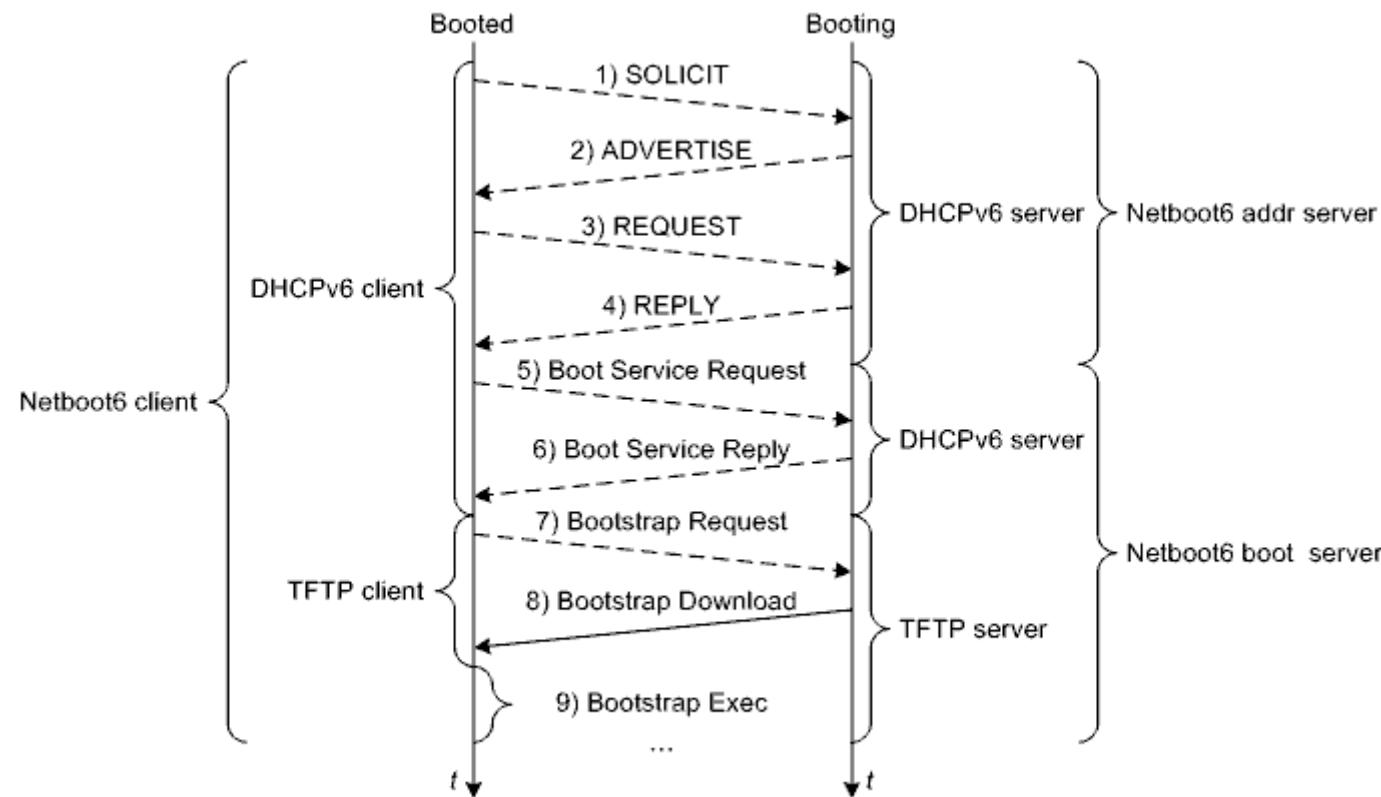
4.0.19.16b

Сообщение Redirect содержит следующие ключевые поля: Target Address -- адрес Link-local Unicast соседа, которому в дальнейшем нужно напрямую передавать пакеты с указанным адресом назначения; Destination Address -- адрес назначения (информации о подсети нет, аналогичного явного поля в сообщении ICMPv4 нет); и две ND-опции: Target Link-layer Address -- MAC-адрес соседа, которому в дальнейшем нужно напрямую передавать пакеты; Redirected Header -- фрагмент предварительно принятого пакета, который послужил причиной создания данного сообщения (начиная с первичного заголовка IPv6 и «сколько влезет» без превышения MTU, аналогичное поле в сообщении ICMPv4 позволяет узнать адрес назначения).

4.0.19.17

Логика DHCPv6 совпадает с логикой DHCPv4. Но пожалуй единственное разумное применение DHCPv6 (RFC 3315) -- это получение параметров при удаленной загрузке.

4.0.19.18a



Клиент-серверное взаимодействие по протоколу Netboot6

4.0.19.18b

Специально для обращения к DHCPv6-серверам и DHCPv6 relays стандартизированы два вида мультикаст-адресов: Link-local All DHCP Relay Agents and Servers ($\text{FF02}::1:2$) и Site-local All DHCP Servers ($\text{FF05}::1:3$).

Стандартный порт на стороне DHCPv6-сервера: UDP 547.

Стандартный порт на стороне DHCPv6-клиента: UDP 546.

Современные реализации TFTP поддерживают IPv6.

4.0.19.19

| 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 | | | |
| + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + | | | |
| msg-type | transaction-id | | |
| + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + | | | |
| | | | |
| . | options | . | . |
| . | (variable) | . | . |
| | | | |
| + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + | | | |

DHCPv6 message [RFC]

4.0.19.20

Обмен между DHCPv6-клиентом и DHCPv6-сервером рекомендуется защищать с помощью случайно генерируемого значения поля `transaction-id` (идентификатор в ответе должен совпадать с идентификатором в запросе).

DHCPv6 может использовать транспорт TCP.

Одно из расширений DHCPv6 позволяет создавать топологии с резервными DHCPv6-серверами (в DHCPv4 поддержка резервирования не заложена).

4.0.19.21а

| Тип | Название | Направление | Описание |
|-----|---------------------|-------------|------------------------------------------------------------------------------------------------------------|
| 1 | SOLICIT | C->S | Аналог DHCPv4 DHCPDISCOVER |
| 2 | ADVERTISE | C-<S | Аналог DHCPv4 DHCPOFFER |
| 3 | REQUEST | C->S | Аналог DHCPv4 DHCPREQUEST |
| 4 | CONFIRM | C->S | Запрос о валидности предоставленных до этого опций при изменении состояния DHCPv6-клиента |
| 5 | RENEW | C->S | Запрос о продлении времени жизни предоставленных опций и скорректированных или новых опциях, если они есть |
| 6 | REBIND | C->S | Запрос о новом сервисе при отсутствии ответа на RENEW в течение отведенного интервала времени |
| 7 | REPLY | C-<S | Аналог DHCPv4 DHCPACK, а также ответ на CONFIRM, RENEW, REBIND |
| 8 | RELEASE | C->S | Аналог DHCPv4 DHCPRELEASE |
| 9 | DECLINE | C->S | Аналог DHCPv4 DHCPDECLINE |
| 10 | RECONFIGURE | C-<S | Уведомление о наличии скорректированных или новых опций (затем RENEW либо INFORMATION-REQUEST) |
| 11 | INFORMATION-REQUEST | C->S | Аналог DHCPv4 DHCPINFORM |
| 12 | RELAY-FORW | Relay->S | Перенаправляемое от DHCPv6-клиента сообщение |
| 13 | RELAY-REPL | Relay-<S | Перенаправляемое DHCPv6-клиенту сообщение |

DHCPv6-сообщения

4.0.19.21b

| | | | |
|----|---------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14 | LEASEQUERY | Any->S | Запрос об указанных опциях, относящихся к указанным DHCPv6-клиентам (RFC 5007, +RFC 5460, +RFC 7653) |
| 15 | LEASEQUERY-REPLY | Any<-S | Ответ либо первый фрагмент ответа на LEASEQUERY (RFC 5007, +RFC 5460, +RFC 7653) |
| 16 | LEASEQUERY-DONE | Any<-S | Последний фрагмент ответа на LEASEQUERY (RFC 5460, +RFC 7653) |
| 17 | LEASEQUERY-DATA | Any<-S | Промежуточный фрагмент ответа на LEASEQUERY (RFC 5460, +RFC 7653) |
| 18 | RECONFIGURE-REQUEST | Relay->S | Запрос о RECONFIGURE при обнаружении изменения состояния DHCPv6-клиента (RFC 6977) |
| 19 | RECONFIGURE-REPLY | Relay<-S | Подтверждение о RECONFIGURE (RFC 6977) |
| 20 | DHCPV4-QUERY | C->S | Запрос DHCPv4 over DHCPv6 (RFC 7341) |
| 21 | DHCPV4-RESPONSE | C<-S | Ответ DHCPv4 over DHCPv6 (RFC 7341) |
| 22 | ACTIVELEASEQUERY | Any->S | Запрос об автоматической передаче сообщений с указанными опциями, относящимися к указанным DHCPv6-клиентам, после каждой коррекции этих опций (RFC 7653) |
| 23 | STARTTLS | Any->S | Запрос о создании защищенного TCP-соединения (TLS) (RFC 7653) |
| 24 | BNDUPD | S1<->S2 | Обновление с параметрами при резервировании (RFC 8156) |
| 25 | BNDREPLY | S1<->S2 | Подтверждение BNDUPD (RFC 8156) |
| 26 | POOLREQ | S1<->S2 | Запрос о пуле адресов при резервировании (RFC 8156) |
| 27 | POOLRESP | S1->S2 | Подтверждение POOLREQ (RFC 8156) |
| 28 | UPDREQ | S1<->S2 | Запрос новых (еще неподтвержденных) обновлений при резервировании (RFC 8156) |
| 29 | UPDREQALL | S1<->S2 | Запрос всех обновлений при резервировании (RFC 8156) |
| 30 | UPDDONE | S1<->S2 | Уведомление о том, что при резервировании все обновления переданы и подтверждены (RFC 8156) |
| 31 | CONNECT | S1->S2 | Запрос о создании соединения при резервировании (RFC 8156) |
| 32 | CONNECTREPLY | S2<->S1 | Подтверждение CONNECT (RFC 8156) |
| 33 | DISCONNECT | S1<->S2 | Запрос о закрытии соединения при резервировании (RFC 8156) |
| 34 | STATE | S1<->S2 | Уведомление об изменении состояния при резервировании (RFC 8156) |
| 35 | CONTACT | S1<->S2 | Периодическое уведомление о продолжении использования соединения при резервировании (RFC 8156) |

DHCPv6-сообщения

4.0.19.22

| Value | Description | Reference |
|-------|-------------------------|-----------|
| 0 | Reserved | |
| 1 | OPTION_CLIENTID | [RFC3315] |
| 2 | OPTION_SERVERID | [RFC3315] |
| 3 | OPTION_IA_NA | [RFC3315] |
| 4 | OPTION_IA_TA | [RFC3315] |
| 5 | OPTION_IAADDR | [RFC3315] |
| 6 | OPTION_ORO | [RFC3315] |
| 7 | OPTION_PREFERENCE | [RFC3315] |
| 8 | OPTION_ELAPSED_TIME | [RFC3315] |
| 9 | OPTION_RELAY_MSG | [RFC3315] |
| 10 | Unassigned | |
| 11 | OPTION_AUTH | [RFC3315] |
| 12 | OPTION_UNICAST | [RFC3315] |
| 13 | OPTION_STATUS_CODE | [RFC3315] |
| 14 | OPTION_RAPID_COMMIT | [RFC3315] |
| 15 | OPTION_USER_CLASS | [RFC3315] |
| 16 | OPTION_VENDOR_CLASS | [RFC3315] |
| 17 | OPTION_VENDOR_OPTS | [RFC3315] |
| 18 | OPTION_INTERFACE_ID | [RFC3315] |
| 19 | OPTION_RECONF_MSG | [RFC3315] |
| 20 | OPTION_RECONF_ACCEPT | [RFC3315] |
| 21 | OPTION_SIP_SERVER_D | [RFC3319] |
| 22 | OPTION_SIP_SERVER_A | [RFC3319] |
| 23 | OPTION_DNS_SERVERS | [RFC3646] |
| 24 | OPTION_DOMAIN_LIST | [RFC3646] |
| 25 | OPTION_IA_PD | [RFC3633] |
| 26 | OPTION_IAPREFIX | [RFC3633] |
| 27 | OPTION_NIS_SERVERS | [RFC3898] |
| 28 | OPTION_NISP_SERVERS | [RFC3898] |
| 29 | OPTION_NIS_DOMAIN_NAME | [RFC3898] |
| 30 | OPTION_NISP_DOMAIN_NAME | [RFC3898] |
| ... | | |
| 65535 | Unassigned | |

DHCPv6 options [IANA]

4.0.19.23а

Одни и те же DHCPv6-опции могут передаваться в обоих направлениях.

В отличие от DHCPv4-опций, DHCPv6-опции имеют сложный формат с вариативным количеством полей и подопций.

DHCPv6-клиент не обязан выполнять «предписания» DHCPv6-сервера, даже сам может «высказывать пожелания» о значениях некоторых параметров DHCPv6-серверу.

DHCPv6-клиент и DHCPv6-сервер должны иметь уникальные идентификаторы DUIDs (DHCP Unique IDentifiers), по которым они однозначно опознают друг друга. DHCPv6 поддерживает аутентификацию сообщений.

4.0.19.23b

DHCPv6-сервер способен выдавать как постоянные (по аналогии с динамическими адресами IPv4), так и временные адреса.

Постоянные адреса имеют Valid Lifetime и Preferred Lifetime (по аналогии с SLAAC, с теми же зарезервированными значениями).

Для обеспечения выдачи и последующего сопровождения адресов, между DHCPv6-клиентом и DHCPv6-сервером создается ассоциация с уникальным идентификатором IAID (Identity Association IDentifier).

Параметра Lease Time как такового нет.

Валидность выданных адресов контролируется двумя таймерами: T1 -- интервал времени, начиная с приема REPLY, по истечении которого необходимо передать RENEW (рекомендуется $T1 = 0,5 * \text{Preferred Lifetime}$) и T2 -- интервал времени, начиная с приема REPLY, по истечении которого необходимо передать REBIND, если не поступило ответа на RENEW (рекомендуется $T1 = 0,8 * \text{Preferred Lifetime}$). Если не поступило ответа на REBIND, то по истечении Valid Lifetime адрес становится недействительным.

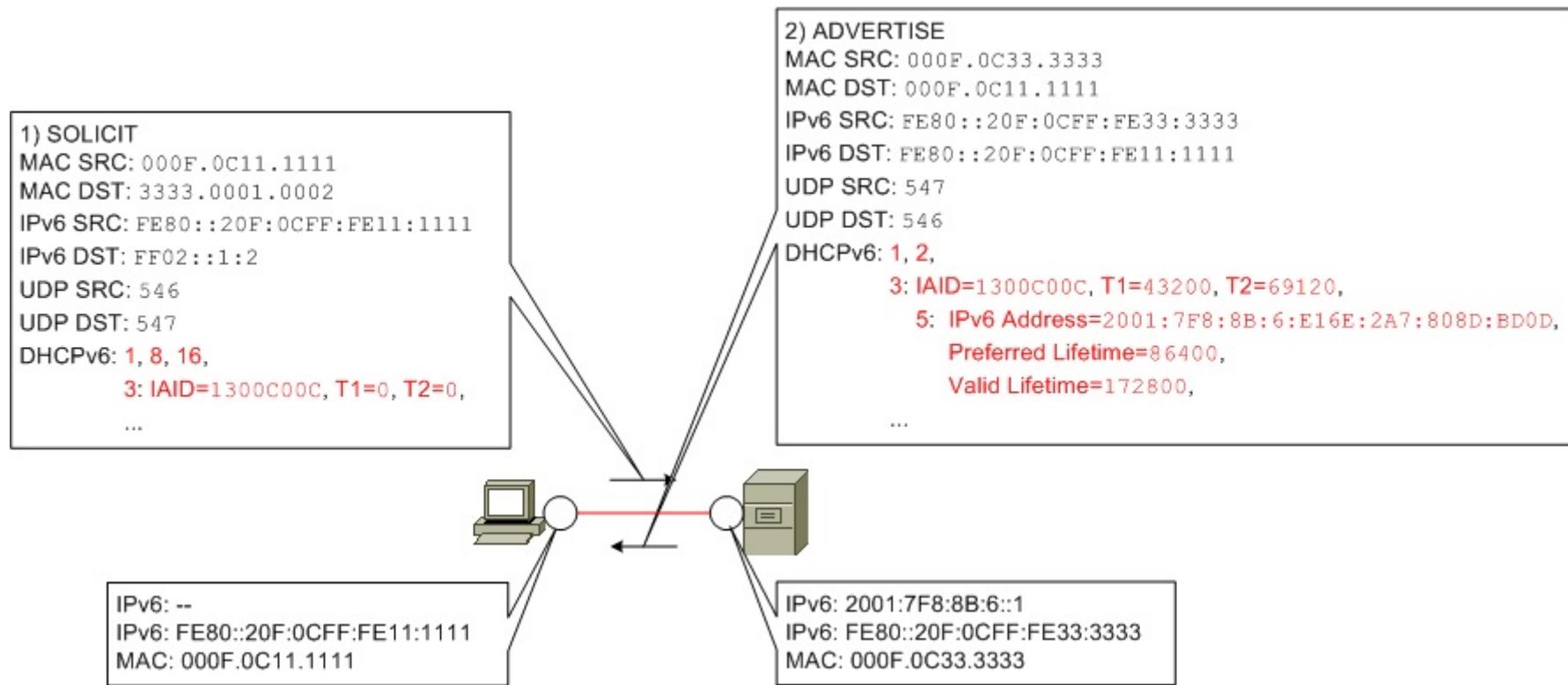
Кроме адресов, посредством DHCPv6 можно передавать префиксы подсетей -- могут использоваться DHCPv6-клиентом по своему усмотрению (например, их можно подставлять при вводе адресов).

4.0.19.24

| Опция (тип) | Название | Описание |
|----------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Client Identifier | Идентификатор DHCPv6-клиента. Содержит поле DUID. Аналог DHCPv4-опции 97 |
| 2 | Server Identifier | Идентификатор DHCPv6-сервера. Содержит поле DUID |
| 3 | Identity Association for Non-temporary Addresses | Ассоциация по предоставлению постоянных динамических адресов. Содержит поля: IAID, T1 и T2 (0 – не определен, FFFFFFFFh – бесконечность). Далее в качестве подопции может следовать по крайней мере одна опция 5 |
| 4 | Identity Association for Temporary Addresses | Ассоциация по предоставлению временных адресов. Содержит поле IAID. Далее в качестве подопции может следовать по крайней мере одна опция 5 |
| 5 | Identity Association Address | Собственно адрес и метаданные о нем. Содержит поля: IPv6 Address, Preferred Lifetime, Valid Lifetime, IAaddr Options – дополнительные параметры |
| 6 | Option Request | Аналог DHCPv4-опции 55. Содержит список запрашиваемых либо предоставляемых DHCPv6-опций |
| 7 | Preference | Приоритет DHCPv6-сервера (чем больше значение, тем выше) |
| 8 | Elapsed Time | Интервал времени, который прошел с начала обмена (в десятках миллисекунд) |
| 12 | Server Unicast | Юникаст-адрес DHCPv6-сервера |
| 14 | Rapid Commit | Ускоренное взаимодействие (связка SOLICIT и REPLY) |
| 16 | Vendor Class | Аналог DHCPv4-опции 60 |
| 23 | DNS Recursive Name Server | Список DNS-серверов. (RFC 3646) |
| 24 | Domain Search List | Список доменных названий для использования в DNS-запросах. (RFC 3646) |
| 25 | Identity Association for Prefix Delegation | Ассоциация по предоставлению префиксов подсетей. Содержит поля: IAID, T1 и T2. Далее в качестве подопции может следовать по крайней мере одна опция 26. (RFC 3633) |
| 26 | Identity Association for Prefix Delegation Prefix | Префикс подсети и метаданные о нем. Содержит поля: Preferred Lifetime, Valid Lifetime, Prefix Length, IPv6 Prefix, IAprefix Options – дополнительные параметры. (RFC 3633) |
| 47 | DHCPv6 Client FQDN | Доменное название DHCPv6-клиента. (RFC 4704) |
| 59 | Boot File URL | Аналог DHCPv4-опции 67. (RFC 5970) |
| 60 | Boot File Parameters | Аналог DHCPv4-опции 13. (RFC 5970) |
| 61 | Client System Architecture Type | Аналог DHCPv4-опции 93. (RFC 5970) |
| 62 | Client Network Interface Identifier | Аналог DHCPv4-опции 94. (RFC 5970) |

Некоторые ключевые DHCPv6-опции

4.0.19.25

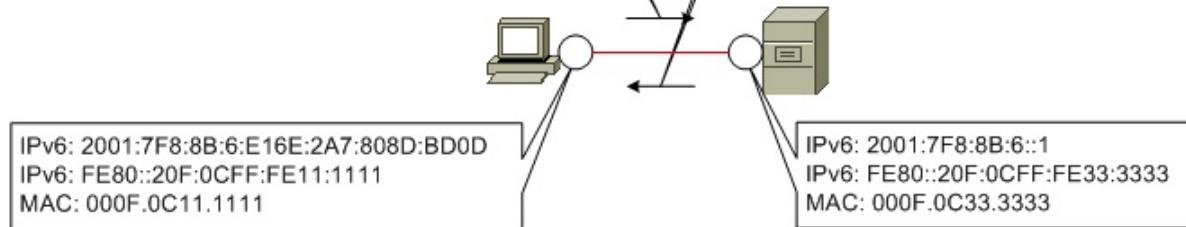


Пример SOLICIT и ADVERTISE

4.0.19.26

3) REQUEST
MAC SRC: 000F.0C11.1111
MAC DST: 3333.0001.0002
IPv6 SRC: FE80::20F:0CFF:FE11:1111
IPv6 DST: FF02::1:2
UDP SRC: 546
UDP DST: 547
DHCPv6: 1, 2, 8, 16,
3: IAID=1300C00C, T1=43200, T2=69120,
5: IPv6 Address=2001:7F8:8B:6:E16E:2A7:808D:BD0D,
Preferred Lifetime=86400,
Valid Lifetime=172800,
6: 23, 24, 39,
...

4) REPLY
MAC SRC: 000F.0C33.3333
MAC DST: 000F.0C11.1111
IPv6 SRC: FE80::20F:0CFF:FE33:3333
IPv6 DST: FE80::20F:0CFF:FE11:1111
UDP SRC: 547
UDP DST: 546
DHCPv6: 1, 2,
3: IAID=1300C00C, T1=43200, T2=69120,
5: IPv6 Address=2001:7F8:8B:6:E16E:2A7:808D:BD0D,
Preferred Lifetime=86400,
Valid Lifetime=172800,
23: DNS Recursive Name Server=2001:7F8:8B:1::53,
24: Domain Name=evm.bsuir.by,
...



Пример REQUEST и REPLY

4.0.20.1

Изменения в маршрутизации.

Специальные соглашения:

1. ::/0 -- маршрут по умолчанию.
2. x . . . x/<64 -- маршрут к большей чем линк подсети.
3. x:x:x:X/64 -- маршрут к подсети (в том числе и оконечной) размером с линк.
4. x:x:x:X:x:x:X/128 -- маршрут к одному сетевому интерфейсу.

4.0.20.2

С целью поддержки IPv6 разработаны новые версии протоколов маршрутизации и дополнения к существующим версиям:

1. RIPng.
2. EIGRP for IPv6.
3. OSPFv3.
4. IS-IS for IPv6.
5. BGPv4+.

4.0.20.3а

IPv6-маршрутизация претерпела серьезные количественные, но, в отличие от IPv6-адресации, не качественные изменения. Тем не менее, нужно сделать несколько замечаний.

Общее правило IP-адресации (касается и IPv6, и IPv4) гласит, что подсети, к которым относятся разные сетевые интерфейсы маршрутизатора (шлюза), не должны перекрываться.

Соблюдение правила позволяет маршрутизатору однозначно соотносить подсети с сетевыми интерфейсами -- значит правильно выбирать сетевой интерфейс для передачи пакета (конечно, выходной сетевой интерфейс в маршруте может быть задан принудительно, но это не касается своих подсетей).

4.0.20.3b

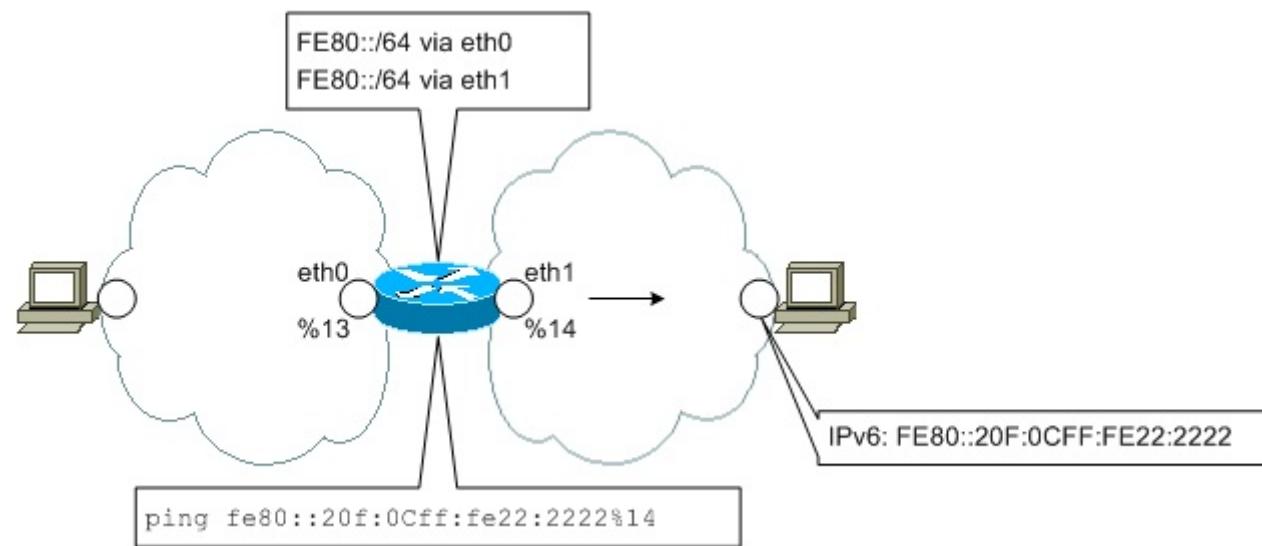
Формат адресов LLU напрямую нарушает приведенное правило (в этом смысле является исключением, разные сетевые интерфейсы могут иметь даже одинаковые адреса LLU) и порождает проблему выбора выходного сетевого интерфейса (source interface selection) при передаче соответствующего пакета, созданного на маршрутизаторе вне рамок ND.

В таблице маршрутизации возникают несколько (согласно числу административно включенных сетевых интерфейсов) абсолютно равноправных маршрутов к подсети FE80::/64.

Балансировка нагрузки (которую поддерживают далеко не все реализации) в отношении своих подсетей просто неуместна.

Проблему решают явным указанием выходного сетевого интерфейса (с помощью идентификатора зоны).

4.0.20.3c



Source interface selection

4.0.20.4а

Согласно идеологии IPv4 в качестве адреса источника подставляется адрес выходного интерфейса. Наличие у одного сетевого интерфейса множества адресов разных видов создает проблему выбора адреса источника (source address selection) при инкапсуляции, когда пакет создан на самом маршрутизаторе и адрес источника явно не задан.

4.0.20.4b

В RFC 6724 сформулированы восемь единых правил для всех реализаций:

1. Приоритетнее адрес, совпадающий с адресом назначения.
2. Приоритетнее адрес из подсети, вид которой более приближен к виду подсети назначения.
3. Preferred-адрес приоритетнее deprecated-адреса.
4. Домашний адрес приоритетнее дорожного адреса (мобильность).
5. Приоритетнее адрес сетевого интерфейса, обращенного в сторону адреса назначения.
6. Приоритетнее адрес, чья метка равна метке адреса назначения (гибридные технологии L2 -- L3).
7. Временный адрес приоритетнее постоянного.
8. Приоритетнее адрес из подсети, которая имеет наиболее длинный общий префикс с подсетью назначения.

4.0.20.4c

Адреса сравниваются попарно (порядок не важен).

Если текущее правило не выявило победителя, то выполняется переход к следующему правилу.

Если в результате выявить одного победителя не удалось, то дальнейший выбор зависит от реализации.

В общем случае, адрес источника и адрес назначения в пакете вполне могут быть разных видов.

4.0.20.5

Возникает и еще один закономерный вопрос -- о том, адреса каких видов использовать для указания маршрутизаторов следующего звена при вводе статических маршрутов.

Согласно рекомендациям о применении IPv6, при настройке статической маршрутизации между маршрутизаторами, для ссылки на маршрутизаторы следующего звена (в том числе на маршрутизаторы по умолчанию) рекомендуется использовать адреса Link-local Unicast, как это и делают протоколы динамической маршрутизации (для удобства часто заменяют на FE80::1, с указанием выходных сетевых интерфейсов).

А маршрутизатор по умолчанию для хостов рекомендуется назначать автоматически -- посредством ND.

Имеет право на существование альтернативный подход, заключающийся в независимой настройке статической маршрутизации в отношении подсетей различных видов.

4.0.20.6

Таблица маршрутизации фактически содержит список префиксов.
ND может корректировать таблицу маршрутизации.

4.0.20.7

С целью ускорения обработки таблицы маршрутизации (которая громоздка даже на обычном хосте) предусмотрен специальный маршрутизационный кэш (destination cache). В этом нет ничего удивительного (если упомянуть широко используемые гибридные технологии L2 -- L3), но маршрутизационный кэш изначально описан в стандарте.

Маршрутизационный кэш состоит из строк со следующими полями:

1. Destination -- IPv6-адрес хоста либо маршрутизатора назначения.
2. Next-hop -- IPv6-адрес соседа, которому нужно передать пакет (если в поле Destination записан IPv6-адрес соседа, то совпадает с полем Destination).
3. Options -- специфические опции, например PMTU (Path MTU).

Маршрутизационный кэш просматривается в первую очередь. Обращение к таблице маршрутизации происходит только в следствие промаха. После обращения к таблице маршрутизации маршрутизационный кэш обновляется.

4.0.20.8

Часто в реализациях IPv6, как и в реализациях IPv4, часть маршрутов скрыта от просмотра.

4.0.20.9

При нестандартной маршрутизации (дополнение к PBR -- Policy-Based Routing) в пакеты могут вставляться маршрутизационные заголовки.

Пока такие заголовки нашли применение при маршрутизации на основе IPv6-адресов источников и при мобильной маршрутизации.

4.0.21.1

Замечание о DNS.

Поддержка IPv6-адресации службой DNS (RFC 3596) выражена в записях нового типа (AAAA) в базах DNS-серверов.

4.0.22.1

Безопасность.

Серьезной (не новой) возможностью IPv6 является интегрированная поддержка защиты передаваемой информации с помощью IPsec-шифрования.

В пакет могут вставляться дополнительные заголовки Authentication Header (AH) и Encapsulation Security Payload (ESP) Header.

Возможна работа в туннельном режиме.

4.0.23.1

Мобильность.

Новой возможностью IPv6 является заложенная целенаправленная поддержка адресации мобильных станций (RFC 6275) (мобильными как правило являются хосты).

4.0.23.2а

Структура мобильной IPv6-системы состоит из нескольких компонентов. Мобильный хост изначально «приписан» к своему *домашнему линку* (*home link*).

В домашнем линкециальному хосту, как правило автоматически, назначается *домашний адрес* (*home address*).

В домашнем линке определен *домашний префикс подсети* (*home subnet prefix*).

Любой доступный линк, в который мобильный хост может быть перемещен из домашнего, является для этого хоста *чужим линком* (*foreign link*).

В чужом линкециальному хосту также назначается адрес -- *дорожный адрес* (*care-of address*).

В чужом линке определен *чужой префикс подсети* (*foreign subnet prefix*).

4.0.23.2b

Если мобильный хост находится в чужом линке, то он регистрируется у своего домашнего агента (*home agent*) (маршрутизатор в домашнем линке), который затем перенаправляет трафик с домашнего адреса на дорожный адрес через специально создаваемый туннель. Таким образом, мобильный хост всегда доступен по домашнему адресу, вне зависимости от места фактического подключения.

Мобильный хост может взаимодействовать с любым хостом (в том числе мобильным) либо маршрутизатором -- *станцией-корреспондентом* (*correspondent node*). Причем может зарегистрироваться напрямую у станции-корреспондента, если та поддерживает мобильность.

4.0.23.3

Домашние агенты и станции-корреспонденты хранят регистрационные данные в виде специального кэша привязки (binding cache). Строки кэша привязки содержат соответствия между домашними и дорожными адресами и сопутствующие данные.

Мобильные хосты, со своей стороны, «ведут» списки привязки (binding update list).

В домашнем линке могут находиться сразу несколько домашних агентов (имеют приоритеты), поэтому каждый из них должен «вести» список домашних агентов (home agent list).

4.0.23.4а

Поддержка мобильности реализуется посредством следующих составляющих:

1. Специальный заголовок Mobility header -- заголовок для обеспечения мобильности.

Этот заголовок используется для пересылки восьми типов mobility-сообщений: Home Test Init, Home Test, Care-of Test Init, Care-of Test, Binding Update, Binding Acknowledgement, Binding Refresh Request, Binding Error. Все mobility-сообщения обеспечивают привязку мобильного хоста.

Mobility-сообщения могут включать в себя различные mobility-опции.

2. Дополнительная опция для пересылки с помощью заголовка предназначенных станции назначения опций: Home Address.

С помощью этой опции мобильный хост указывает свой домашний адрес.

3. Специальный тип маршрутизационного заголовка (тип 2, маршрутизационный заголовок имеет несколько типов).

Используется для пересылки пакета от станции-корреспондента напрямую к мобильной станции и содержит домашний адрес.

4.0.23.4b

4. Четыре вида ICMPv6-сообщений: Home Agent Address Discovery Request, Home Agent Address Discovery Reply, Mobile Prefix Solicitation, Mobile Prefix Advertisement (кодируют как типы 144 -- 147).

Используются при взаимодействии мобильного хоста и домашнего агента.

5. Дополнения ND.

Еще один флаг в RA: H (Home Agent) -- данный маршрутизатор является домашним агентом.

Уменьшены нижние граничные значения MaxRtrAdvInterval и MinRtrAdvInterval -- при мобильности частота RAs должна быть более высокой.

Еще один флаг в ND-опции Prefix Information: R (Router Address) -- данная ND-опция содержит полный адрес маршрутизатора (все 128 битов).

Еще две ND-опции: Advertisement Interval -- максимальный интервал времени между RAs (MaxRtrAdvInterval), и Home Agent Information -- информация о домашнем агенте.

4.0.24.1

IPv6-адресация в Windows и Linux.

Linux- и Windows-станции с текущими реализациями IPv6 по умолчанию относятся к типу IPv4/IPv6 (можно скорректировать).

Как и в случае с другими реализациями IPv6, нужно соблюдать правила назначения адресов сетевым интерфейсам.

Типовой хост имеет следующие адреса:

1. Адрес Link-local Unicast.
2. Дополнительные адреса Unicast (Unique Local Unicast и Global Unicast).
3. Адрес сетевого интерфейса -- заглушки.
4. Адрес Link-local All Nodes Multicast.
5. Адреса Solicited-node Multicast для каждого из адресов Unicast.
6. Дополнительные групповые адреса Multicast.
7. Адреса туннелей IPv6-over-IPv4.

4.0.24.2

Типовой маршрутизатор, в дополнение к указанным адресам (применительно к каждому из сетевых интерфейсов), имеет следующие:

8. Адреса Link-local All Routers Multicast каждого из сетевых интерфейсов.
9. Адреса Site-local All Routers Multicast соответствующих сетевых интерфейсов.
10. Адреса Subnet-router Anycast для каждой из подсетей.

4.0.24.3

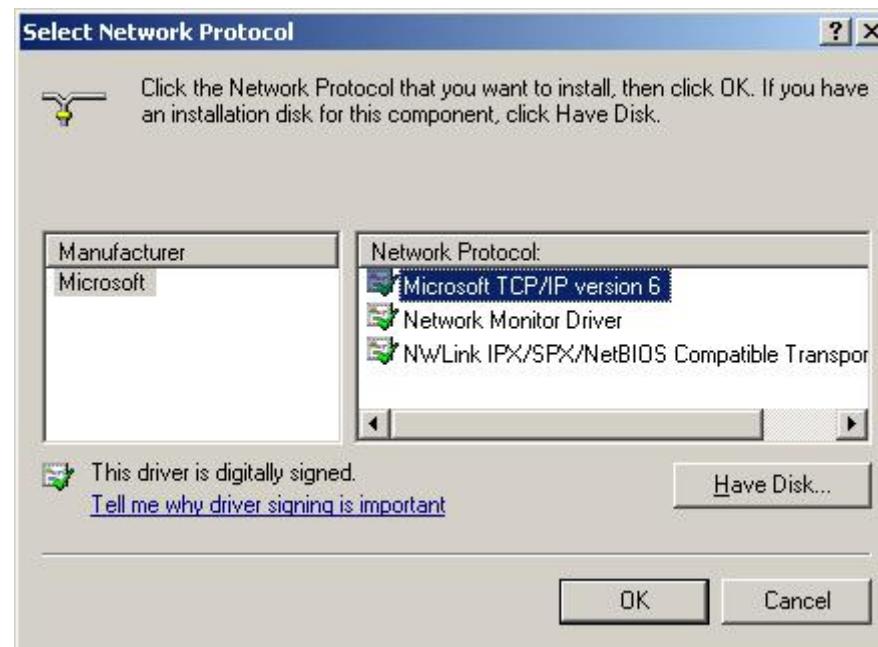
В Windows XP SP2 и Server 2003 поддержка IPv6 уже была интегрирована в составе Advanced Networking Pack и устанавливалась как опциональный компонент с помощью графического интерфейса (свойства сетевых интерфейсов) либо командой netsh interface ipv6 install. Для работы с адресами использовались только расширения команды netsh interface ipv6 (вместо отмененной команды iprv6).

Полноценная поддержка IPv6 доступна начиная с Windows Vista и Server 2008. Может быть задействован как графический интерфейс, так и различные варианты команды netsh interface ipv6.

Начиная с Windows 10 1607 по умолчанию запрещен туннельный интерфейс 6to4, Windows 10 1703 – ISATAP, Windows 10 1803 -- Teredo.

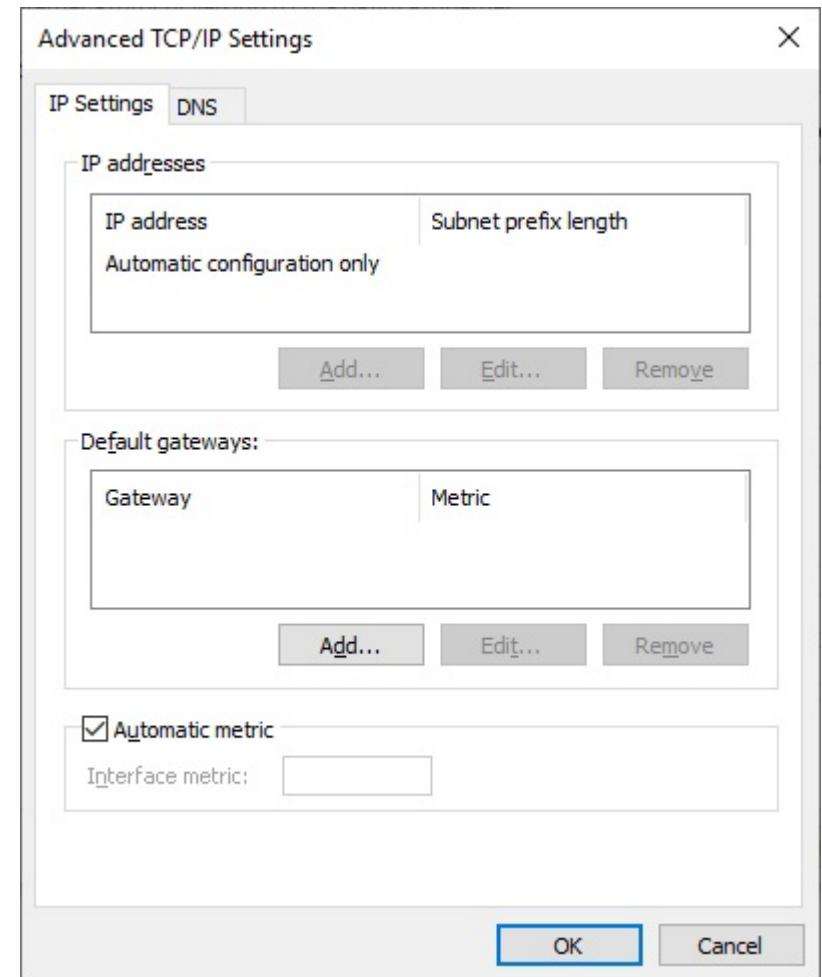
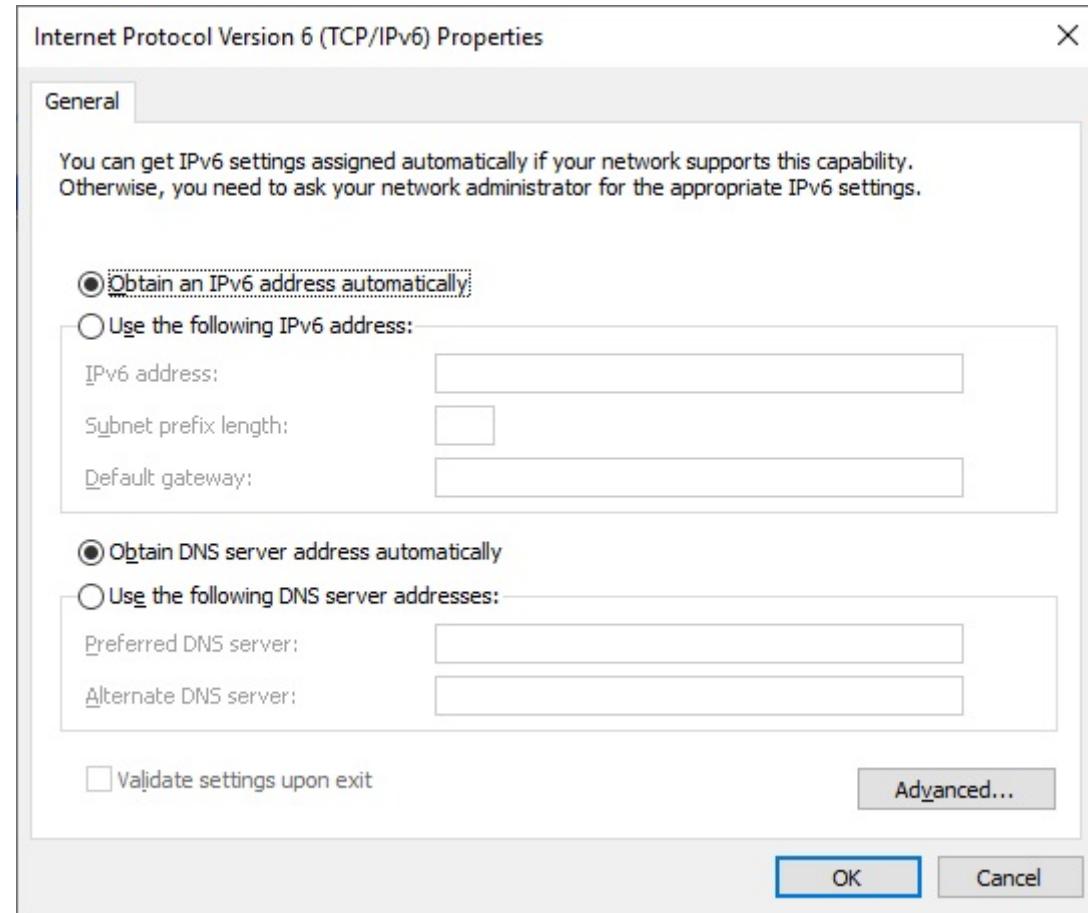
Следует обратить внимание на то, что по умолчанию автоконфигурирование работает даже при статическом конфигурировании адресов.

4.0.24.4a



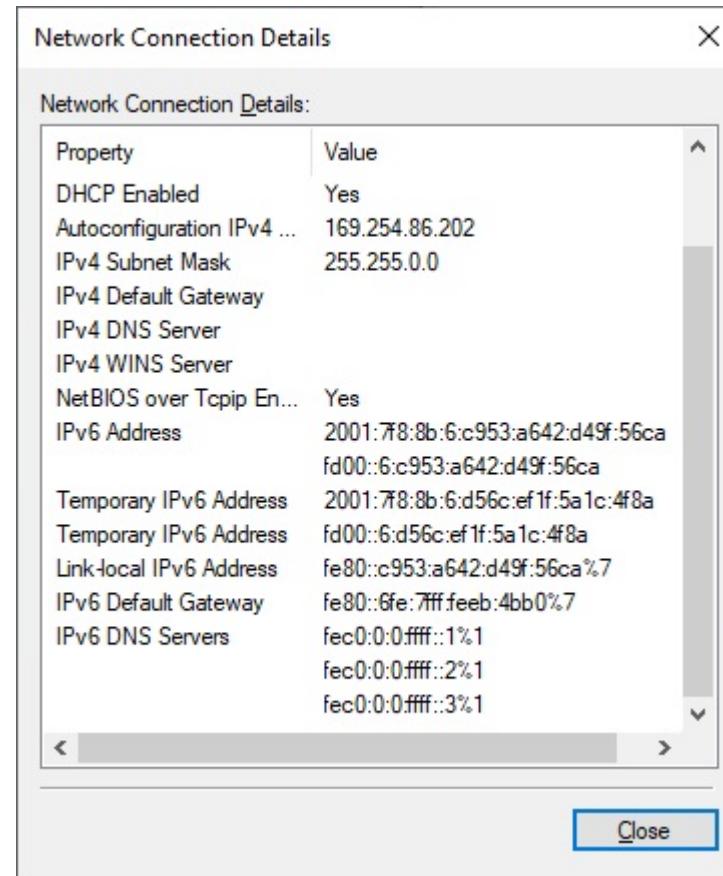
Добавление протокола IPv6 к сетевому интерфейсу в Windows XP

4.0.24.4b



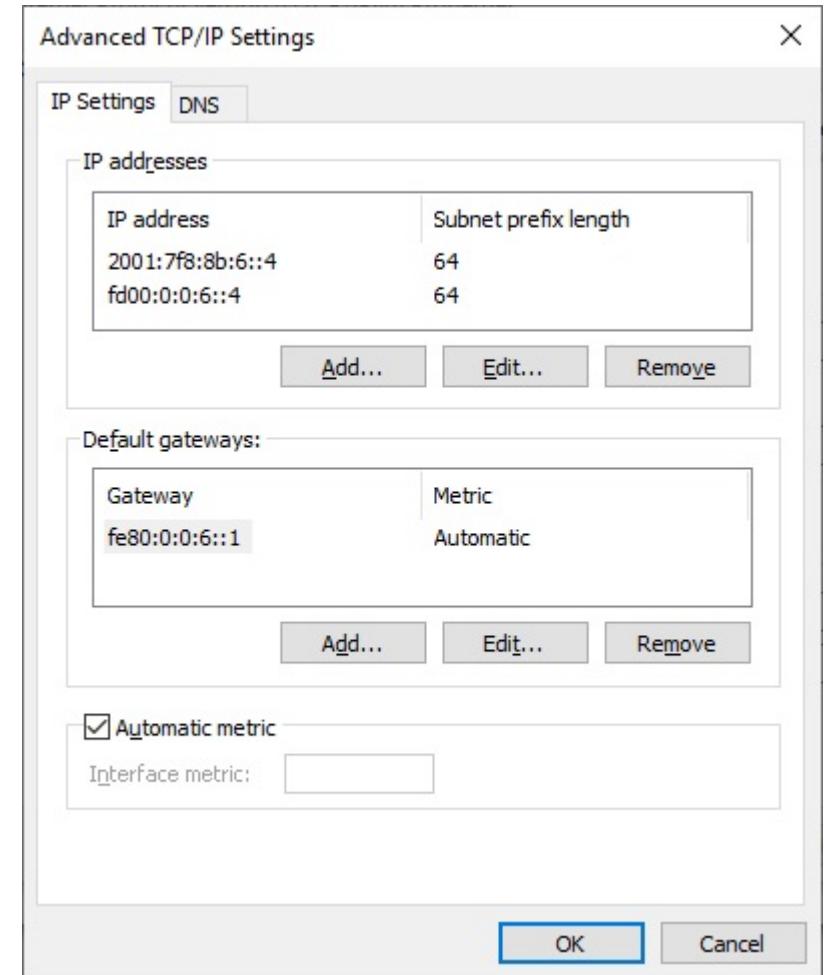
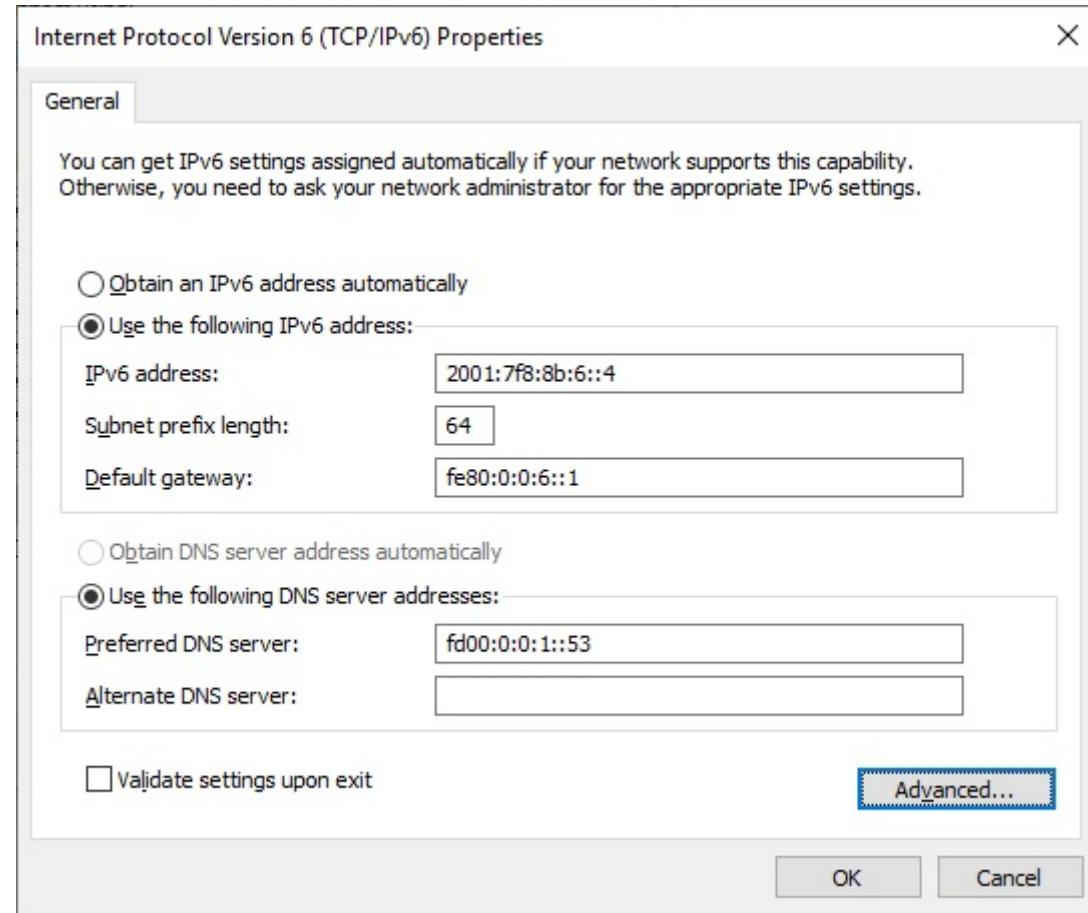
IPv6 в Windows 10 (автоконфигурирование)

4.0.24.4c



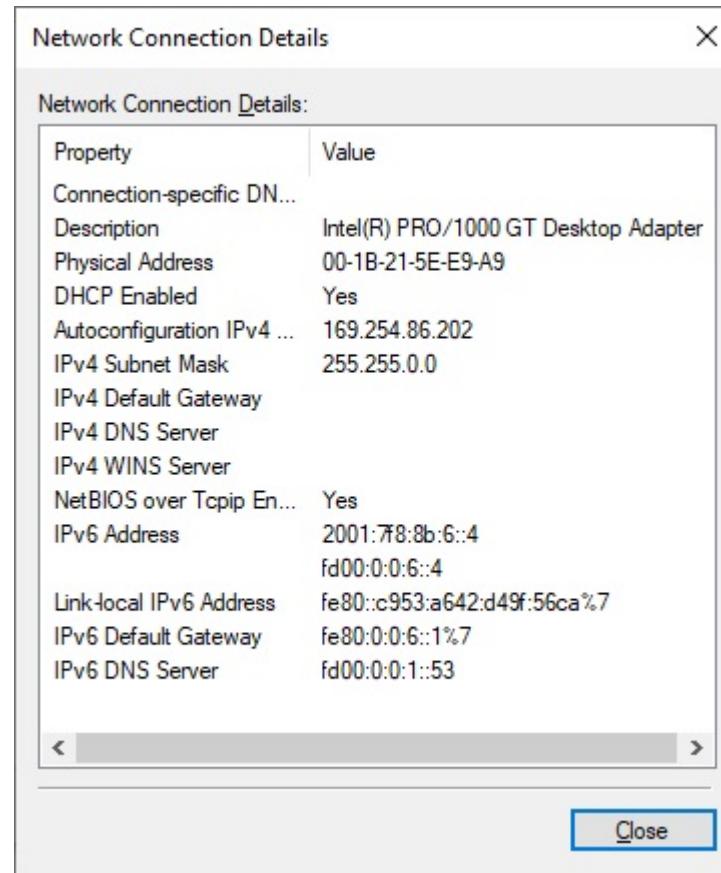
IPv6 в Windows 10 (автоконфигурирование)

4.0.24.4d



IPv6 в Windows 10

4.0.24.4e



IPv6 в Windows 10

4.0.24.5

В Linux поддержка IPv6 имеется в дистрибутивах с ядрами 2.2.x и последующими.

Присвоение адресов IPv6 сводится к работе с соответствующими конфигурационными файлами.

4.0.24.6

/etc/sysconfig/network:

```
...
NETWORKING_IPV6=yes
...
```

/etc/sysconfig/network-scripts/ifcfg-eth1 (**ветви Red Hat и SUSE**):

```
...
IPV6INIT=yes
IPV6ADDR=2001:7f8:8b:6::4/64
IPV6ADDR_SECONDARIES(fd00:0:0:6::4
IPV6_DEFAULTGW=fe80::1
...
```

/etc/network/interfaces (**ветвь Debian**):

```
...
iface eth1 inet6 static
    address 2001:7f8:8b:6::4
    netmask 64
    gateway 2001:7f8:8b:6::1
iface eth1 inet6 static
    address fd00:0:0:6::4
    netmask 64
    gateway fd00:0:0:6::1
...
```

Примеры IPv6-дополнений в конфигурационных файлах Linux

4.0.24.7

Генерирование временных адресов:

```
>netsh interface ipv6 set privacy=enabled|disabled
```

Генерирование случайных значений интерфейсных частей постоянных адресов
(в отличие от других основных ОС, включено по умолчанию):

```
>netsh interface ipv6 set global randomizeidentifiers=enabled|disabled
```

Автоконфигурирование адресов:

```
>netsh interface ipv6 set interface Interface_Name_Or_Index routerdiscovery=enabled|disabled|dhcp
```

Примеры управления IPv6-автоконфигурированием в Windows

4.0.24.8

Генерирование временных адресов:

```
#sysctl net.ipv6.conf.default.use_tempaddr=integer
```

либо

```
#echo "integer" > /proc/sys/net/ipv6/conf/default/use_tempaddr
```

где integer:

<= 0 -- запрет

= 1 -- разрешение, причем временные адреса менее приоритетны

> 1 -- разрешение, причем временные адреса более приоритетны

Автоконфигурирование, включая ND:

конфигурационный файл /etc/sysconfig/network-scripts/ifcfg-<interface-name>:

...

IPV6_AUTOCONF=yes | no

IPV6_ROUTER=yes | no

...

демон radvd со стандартным конфигурационным файлом /etc/radvd.conf

Примеры управления IPv6-автоконфигурированием в Linux

4.0.24.9

Ключ реестра:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters\DisabledComponents

где DisabledComponents (DWORD) формируется из битов:

бит 0 = 1 -- запрет всех туннельных интерфейсов IPv6-over-IPv4

бит 1 = 1 -- запрет туннельного интерфейса 6to4

бит 2 = 1 -- запрет туннельного интерфейса ISATAP

бит 3 = 1 -- запрет туннельного интерфейса Teredo

бит 4 = 1 -- разрешение IPv6 только посредством туннельных интерфейсов IPv6-over-IPv4

бит 5 = 1 -- IPv4 предпочтительнее IPv6

Варианты команды netsh interface ipv6:

```
>netsh interface ipv6 6to4 set state state=enabled|disabled|default
```

```
>netsh interface ipv6 isatap set state state=enabled|disabled|default
```

```
>netsh interface ipv6 set teredo type=disabled|client|enterpriseclient|server|default
```

Примеры управления совместимостью с IPv4 в Windows

4.0.24.10

Возможности radvd

6to4:

конфигурационный файл /etc/sysconfig/network-scripts/ifcfg-<interface-name>:

```
IPV6TO4INIT=yes  
IPV6TO4_ROUTING="eth0-:1::1/64"  
IPV6_CONTROL_RADVD=yes
```

ISATAP:

```
#ip tunnel add is0 mode isatap local 192.168.11.216  
#ip addr add fd00::6:0:5efe:192.168.11.216/64 dev is0  
#ip link set is0 up
```

Teredo:

пакет Miredo, предоставляющий одноименный сервис, со стандартным конфигурационным файлом /etc/miredo.conf

4.0.24.11

Некоторые новые и обновленные команды:

ipconfig (Windows);

netsh interface ipv6 show interface (Windows);

ifconfig (Linux);

ping (Windows);

ping6 (Linux);

netsh interface ipv6 show neighbors (Windows);

ip -6 neigh show (Linux).

4.0.25.1

IPv6-маршрутизация в Windows и Linux.

Основными отличиями являются увеличение количества строк таблицы маршрутизации и изменение набора полей, что вполне адекватно ситуации.

В типовую таблицу маршрутизации включаются следующие маршруты:

1. К своим подсетям размером с линк (для всех адресов Link-local Unicast, Unique Local Unicast, Global Unicast).
2. К своим сетевым интерфейсам (для всех адресов Link-local Unicast, Unique Local Unicast, Global Unicast).
3. Маршрут по умолчанию.
4. Маршрут к сетевому интерфейсу -- заглушке.
5. Маршруты, связанные с адресами Multicast.
6. Дополнительные статические и динамические маршруты.
7. Маршруты к туннелям IPv6-over-IPv4.

Как и в случае с IPv4, при выборе маршрута применяется правило наиболее точного соответствия. В первую очередь выбирается маршрут к сетевому интерфейсу, в последнюю -- маршрут по умолчанию.

4.0.25.2

Некоторые новые и обновленные команды:

`route print -6 (Windows);`

`netsh interface ipv6 show route (Windows);`

`netstat -nr -A inet6 (Linux);`

`netsh interface ipv6 add route (Windows);`

`route -A inet6 add (Linux);`

`tracert (Windows);`

`traceroute6 (Linux).`

4.0.25.3

```
C:\Users\Administrator>route print -6

=====
Interface List
 13...00 27 0e 1f a0 b9 ....Intel(R) 82567LF-2 Gigabit Network Connection
    1.....Software Loopback Interface 1
 11...00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
 12...00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
=====

IPv6 Route Table
=====
Active Routes:
 If Metric Network Destination      Gateway
 13    266 ::/0                      2001:7f8:8b:6::1
 13    266 ::/0                      fd00:0:0:6::1
  1    306 ::1/128                  On-link
 13    266 2001:7f8:8b:6::/64      On-link
 13    266 2001:7f8:8b:6::4/128    On-link
 13    266 fd00:0:0:6::/64        On-link
 13    266 fd00:0:0:6::4/128     On-link
 13    266 fe80::/64              On-link
 11    266 fe80::5efe:192.168.11.216/128
                                         On-link
 13    266 fe80::2978:fe81:4c15:df82/128
                                         On-link
   1    306 ff00::/8                On-link
 13    266 ff00::/8                On-link
=====

Persistent Routes:
 If Metric Network Destination      Gateway
  0 4294967295 ::/0                2001:7f8:8b:6::1
  0 4294967295 ::/0                fd00:0:0:6::1
=====
```

4.0.25.4

```
#netstat -nr -A inet6
Kernel IPv6 routing table
Destination          Next Hop            Flags Metric Ref    Use Iface
2001:7f8:8b:1::/64   ::                  U      256    14        0 eth1
2001:7f8:8b:6::/64   ::                  U      256     0        0 eth0
fd00:0:0:1::/64      ::                  U      256     1        0 eth1
fd00:0:0:6::/64      ::                  U      256     0        0 eth0
fe80::/64             ::                  U      256     0        0 eth1
fe80::/64             ::                  U      256     0        0 eth0
::/0                2001:7f8:8b:1::1  UG     1      32        0 eth1
::/0                fd00:0:0:1::1  UG     1      0        0 eth1
::1/128             ::                  U      0      3        1 lo
2001:7f8:8b:1::11/128 ::                  U      0      36       1 lo
2001:7f8:8b:6::1/128  ::                  U      0      0        1 lo
fd00:0:0:1::11/128   ::                  U      0      0        1 lo
fd00:0:0:6::1/128   ::                  U      0      0        1 lo
fe80::227:eff:fe1f:a0e2/128 ::                  U      0      6        1 lo
fe80::2c0:cff:fe72:6867/128 ::                  U      0      3        1 lo
ff00::/8              ::                  U      256    72        0 eth1
ff00::/8              ::                  U      256    33        0 eth0
#
#Ref -- количество ссылок (ядром не используется)
#Use -- количество попаданий
#
```

4.0.25.5

Windows:

```
>netsh interface ipv6 set interface Interface_Name_Or_Index forwarding=enabled
```

либо

сервис Routing and Remote Access

Linux:

конфигурационный файл /etc/sysconfig/network:

...

```
IPV6FORWARDING=yes
```

...

либо

```
#echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

Следует обратить внимание на возможность «привязки» не ко всем сетевым интерфейсам, а к конкретным.

Включение IPv6 forwarding в Windows и Linux

4.0.26.1а

IPv6 в Cisco IOS.

На маршрутизаторах и коммутаторах Cisco IPv6-возможности по умолчанию находятся в административно выключенном состоянии.

Для административного включения на сетевом интерфейсе IPv6 и автоматической генерации адреса Link-local Unicast используют команду `ipv6 enable`. Как альтернатива, позволяющая в добавок задействовать возможности ND, выступает команда `ipv6 address autoconfig`.

Для присвоения сетевому интерфейсу адреса Unique Local Unicast либо Global Unicast, и тем самым активации на нем IPv6, используют команду `ipv6 address`. После ввода первого такого адреса автоматически генерируется и адрес Link-local Unicast (если до этого адреса Link-local Unicast не было).

Вариант с аргументом `eui-64` позволяет автоматически сгенерировать соответствующее значение интерфейсной части адреса.

Вариант с аргументом `link-local` позволяет заменить автоматически сгенерированный адрес Link-local Unicast (множество адресов Link-local Unicast одного сетевого интерфейса не поддерживается).

4.0.26.1b

Вариант с аргументом `anycast` позволяет добавить соответственно эникаст-адрес.

При вводе адресов можно использовать заранее подготовленные именованные префиксы, которые создаются с помощью команды `ipv6 general-prefix`.

Для работы с мультикаст-группами используют различные варианты команды `ipv6 mld`, например `ipv6 mld join-group`.

Шестнадцатеричные цифры в IPv6-адресах при выводе на экран и при переносе в конфигурационные файлы приводятся к верхнему регистру.

4.0.26.2

```
Router(config)#interface fa0/0
Router(config-if)#ipv6 address 2001:7f8:8b::1/64
Router(config-if)#ipv6 address fd5f:4cf8:7fcd:6::/64 eui-64
Router(config-if)#ipv6 address fe80::1 link-local
Router(config-if)#ipv6 address 2001:7f8:8b:6::/64 anycast
Router(config-if)#ipv6 mld join-group ff04::10
Router(config-if)#exit

Router(config)#ipv6 general-prefix EXAMPLE-PREFIX 2001:7f8:8b::/48
...
Router(config-if) ipv6 address EXAMPLE-PREFIX 0:0:0:8::1/64
```

4.0.26.3

Для вывода на экран IPv6-информации о сетевом интерфейсе (в том числе информации о ND) используют команду `show ipv6 interface`.

4.0.26.4

```
Router#show ipv6 interface brief
FastEthernet0/0                  [up/up]
    FE80::6FE:7FFF:FEEB:4BB0
    2001:7F8:8B:6::1
    FD00:0:0:6::1
FastEthernet0/1                  [up/up]
    FE80::6FE:7FFF:FEEB:4BB1
    2001:7F8:8B:8::1
Serial0/0/0                      [administratively down/down]
    unassigned
Serial0/0/1                      [administratively down/down]
    unassigned
```

4.0.26.5

Маршрутизатор Cisco по умолчанию функционирует в режиме IPv6-хоста -- применительно к каждому сетевому интерфейсу. При этом, исходя из соображений безопасности, в более новых версиях IOS (начиная с 15.0(2)SE) стандартные возможности автоконфигурирования по умолчанию запрещены (loose, or nonconformant, host mode) -- разрешают командой `ipv6 nd host mode strict` (strict, or conformant, host mode).

В режим IPv6-маршрутизатора переключают командой `ipv6 unicast-routing` (в **глобальном конфигурационном режиме**) -- не «просто» включают IPv6 forwarding.

4.0.26.6а

Для управления ND в основном используют различные варианты команды `ipv6 nd` (в режиме конфигурирования интерфейса):

`ipv6 address autoconfig` -- включить автоконфигурирование сетевого интерфейса (совместима с режимом IPv6-маршрутизатора -- без назначения маршрутизатора по умолчанию) (в **глобальном конфигурационном режиме доступна команда** `ipv6 address autoconfig default`);

`ipv6 nd autoconfig default-route` -- незамедлительно сгенерировать и передать RS с целью определения маршрутизатора по умолчанию без ожидания очередного RA (при автоконфигурировании);

`ipv6 nd autoconfig prefix` -- незамедлительно сгенерировать и передать RS с целью определения префиксов подсетей без ожидания очередного RA (при автоконфигурировании);

`ipv6 nd cache expire` -- установить указанное время валидности строки ND-кэша (по умолчанию 4 часа);

`ipv6 nd cache interface-limit` -- установить указанное максимальное количество строк ND-кэша (по умолчанию не определено);

`ipv6 nd dad attempts` -- установить указанное количество попыток определения конфликта адресов (по умолчанию одна);

4.0.26.6b

ipv6 nd dad time -- установить указанный интервал между попытками определения конфликта адресов (по умолчанию 1 s);

ipv6 nd managed-config-flag -- устанавливать флаг M в RAs (по умолчанию не устанавливается);

ipv6 nd na glean -- обрабатывать незапрошенные NAs (по умолчанию игнорируются);

ipv6 nd ns-interval -- установить указанное значение Retrans Timer в RAs (при восстановлении MAC-адресов и при определении конфликтов адресов) (по умолчанию 0 -- RAs, 1 s -- сам сетевой интерфейс);

ipv6 nd nud retry -- установить указанное количество попыток проверки достижимости соседей (по умолчанию 3 попытки с интервалом 1 s);

ipv6 nd other-config-flag -- устанавливать флаг O в RAs (по умолчанию не устанавливается);

ipv6 nd prefix -- передавать указанный префикс подсети как ND-опцию Prefix Information в RAs;

ipv6 nd ra hop-limit -- установить указанное значение Cur Hop Limit в RAs (по умолчанию 64);

4.0.26.6c

ipv6 nd ra interval -- установить указанный интервал между RAs (по умолчанию 200 s);

ipv6 nd ra lifetime -- установить указанное значение Router Lifetime в RAs (по умолчанию 1800 s);

ipv6 nd ra mtu -- передавать указанное значение как ND-опцию MTU в RAs (применительно к сетевым интерфейсам Ethernet по умолчанию 1500);

ipv6 nd ra suppress -- не передавать RAs (по умолчанию передаются сетевыми интерфейсами Ethernet и FDDI -- если включен IPv6 forwarding);

ipv6 nd reachable-time -- установить указанное значение Reachable Time (по умолчанию 0 -- RAs, по умолчанию 30 s -- сам сетевой интерфейс);

ipv6 nd router-preference -- установить указанное значение Default Router Preference в RAs (по умолчанию Medium).

4.0.26.7

```
Router(config)#interface fa0/0
Router(config-if)#ipv6 address autoconfig
Router(config-if)#exit
```

```
Router#show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::6FE:7FFF:FEED:4BB0
  No Virtual link-local address(es):
    Stateless address autoconfig enabled
  Global unicast address(es):
    2001:7F8:8B:6:6FE:7FFF:FEED:4BB0, subnet is 2001:7F8:8B:6::/64 [EUI/CAL/PRE]
      valid lifetime 2591989 preferred lifetime 604789
    FD00::6:6FE:7FFF:FEED:4BB0, subnet is FD00:0:0:6::/64 [EUI/CAL/PRE]
      valid lifetime 2591989 preferred lifetime 604789
  Joined group address(es):
    FF02::1
    FF02::1:FEED:4BB0
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 44711)
Default router is FE80::6FE:7FFF:FE37:A448 on FastEthernet0/0
```

4.0.26.8

```
Router(config)#ipv6 unicast-routing
```

```
Router#show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::6FE:7FFF:FEED:4BB0
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:7F8:8B:6::1, subnet is 2001:7F8:8B:6::/64
    FD00:0:0:6::1, subnet is FD00:0:0:6::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FFEB:4BB0
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 28220)
ND advertised reachable time is 0 (unspecified)
ND advertised retransmit interval is 0 (unspecified)
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
ND advertised default router preference is Medium
Hosts use stateless autoconfig for addresses.
```

4.0.26.9

Запуск DHCPv6-клиента происходит командой `ipv6 address dhcp` (в связке с `ipv6 enable`).

Поддерживается и сервис DHCPv6.

В общем случае сетевой интерфейс может быть DHCPv6-клиентом, либо DHCPv6-сервером, либо DHCPv6 relay.

4.0.26.10

```
Router(config)#interface fa0/0
Router(config-if)#ipv6 address dhcp
Router(config-if)#ipv6 enable
Router(config-if)#exit
```

```
Router#show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::6FE:7FFF:FEEB:4BB0
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:7F8:8B:6:85EB:5537:D7F5:A61D, subnet is 2001:7F8:8B:6:85EB:5537:D7F5:A61D/128
  Joined group address(es):
    FF02::1
    FF02::1:FFEB:4BB0
    FF02::1:FFF5:A61D
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachable are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 22851)
  Default router is FE80::6FE:7FFF:FEEB:4B68 on FastEthernet0/0
```

4.0.26.11

```
Router(config)#ipv6 unicast-routing

Router(config)#ipv6 dhcp pool EXAMPLE-DHCPV6-POOL
Router(config-dhcpv6)#address prefix 2001:7f8:8b:6::/64
Router(config-dhcpv6)#domain-name evm.bsuir.by
Router(config-dhcpv6)#dns-server fd00:0:0:1::53
Router(config-dhcpv6)#prefix-delegation 2001:7f8:8b:6::/64 0003000104fe7feb4bb0
Router(config-dhcpv6)#exit

Router(config)#interface fa0/0
Router(config-if)#ipv6 address 2001:7F8:8B:6::1/64
Router(config-if)#ipv6 nd managed-config-flag
Router(config-if)#ipv6 nd other-config-flag
Router(config-if)#ipv6 dhcp server EXAMPLE-DHCPV6-POOL
Router(config-if)#exit

Router#show ipv6 dhcp pool
DHCPv6 pool: EXAMPLE-DHCPV6-POOL
  Static bindings:
    Binding for client 0003000104FE7FEB4BB0
      IA PD: IA ID not specified
      Prefix: 2001:7F8:8B::/48
        preferred lifetime 604800, valid lifetime 2592000
      Address allocation prefix: 2001:7F8:8B:6::/64 valid 172800 preferred 86400 (1 in use, 0 conflicts)
      DNS server: FD00:0:0:1::53
      Domain name: evm.bsuir.by
    Active clients: 1

Router#show ipv6 dhcp binding
Client: FE80::6FE:7FFF:FE8B:4BB0
  DUID: 0003000104FE7FEB4BB0
  Username : unassigned
  IA NA: IA ID 0x00040001, T1 43200, T2 69120
    Address: 2001:7F8:8B:6:85EB:5537:D7F5:A61D
      preferred lifetime 86400, valid lifetime 172800
      expires at Apr 15 2019 05:55 PM (172355 seconds)
```

4.0.26.12

```
Router(config)#interface fa0/0
Router(config-if)#ipv6 dhcp client pd DHCPV6-PREFIX
Router(config-if)#ipv6 address DHCPV6-PREFIX 0:0:0:6::4/64
Router(config-if)#exit
```

```
Router#show ipv6 dhcp
This device's DHCPv6 unique identifier(DUID): 0003000104FE7FEB4BB0

Router#show ipv6 dhcp interface fa0/0
FastEthernet0/0 is in client mode
  Prefix State is OPEN
  Renew will be sent in 3d11h
  Address State is IDLE
  List of known servers:
    Reachable via address: FE80::6FE:7FFF:FEED:4B68
    DUID: 0003000104FE7FEB4B68
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00040001, T1 302400, T2 483840
      Prefix: 2001:7F8:8B:6::/64
        preferred lifetime 604800, valid lifetime 2592000
        expires at May 14 2019 09:44 AM (2591906 seconds)
      DNS server: FD00:0:0:1::53
      Domain name: evm.bsuir.by
      Information refresh time: 0
    Prefix name: DHCPV6-PREFIX
    Prefix Rapid-Commit: disabled
    Address Rapid-Commit: disabled
```

4.0.26.13

Поддерживаются следующие основные режимы туннелирования IPv6-over-IPv4:

ipv6ip -- manual;

ipv6ip 6to4 -- 6to4;

ipv6ip isatap -- ISATAP;

плюс gre ipv6 -- GRE (Generic Routing Encapsulation).

4.0.26.14

Для просмотра информации о соседях используют команду `show ipv6 neighbors`.

4.0.26.15

```
Router#show ipv6 neighbors
```

IPv6 Address

FE80::50B1:D597:7C33:4442

2001:7FC:8B:6:6C70:AE09:B5AC:A84F

| Age | Link-layer Addr | <u>State</u> | Interface |
|-----|-----------------|--------------|-----------|
| 4 | 00c0.0c72.6846 | <u>STALE</u> | Fa0/0 |
| 12 | 00c0.0c72.6846 | STALE | Fa0/0 |

4.0.26.16

Команды ping и traceroute совместимы с IPv6.

4.0.26.17

Для просмотра таблицы IPv6-маршрутизации используют команду `show ipv6 route`.

4.0.26.18

```
Router#show ipv6 route
IPv6 Routing Table - Default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       D - EIGRP, EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S  ::/0 [1/0]
    via FE80::1, FastEthernet0/1
C  2001:7F8:8B:1::/64 [0/0]
    via FastEthernet0/1, directly connected
L  2001:7F8:8B:1::11/128 [0/0]
    via FastEthernet0/1, receive
C  2001:7F8:8B:6::/64 [0/0]
    via FastEthernet0/0, directly connected
L  2001:7F8:8B:6::1/128 [0/0]
    via FastEthernet0/0, receive
B  2001:ACAD:ACAD:A::/64 [20/0]
    via FE80::6FE:7FFF:FE8B:4B68, FastEthernet0/1
C  FD00:0:0:1::/64 [0/0]
    via FastEthernet0/1, directly connected
L  FD00:0:0:1::11/128 [0/0]
    via FastEthernet0/1, receive
C  FD00:0:0:6::/64 [0/0]
    via FastEthernet0/0, directly connected
L  FD00:0:0:6::1/128 [0/0]
    via FastEthernet0/0, receive
L  FF00::/8 [0/0]
    via Null0, receive
```

4.0.26.19

Для внесения статического маршрута в таблицу маршрутизации используют команду `ipv6 route`.

Для включения IPv6 forwarding (в отличие от IPv4 forwarding по умолчанию выключен) используют команду `ipv6 unicast-routing`.

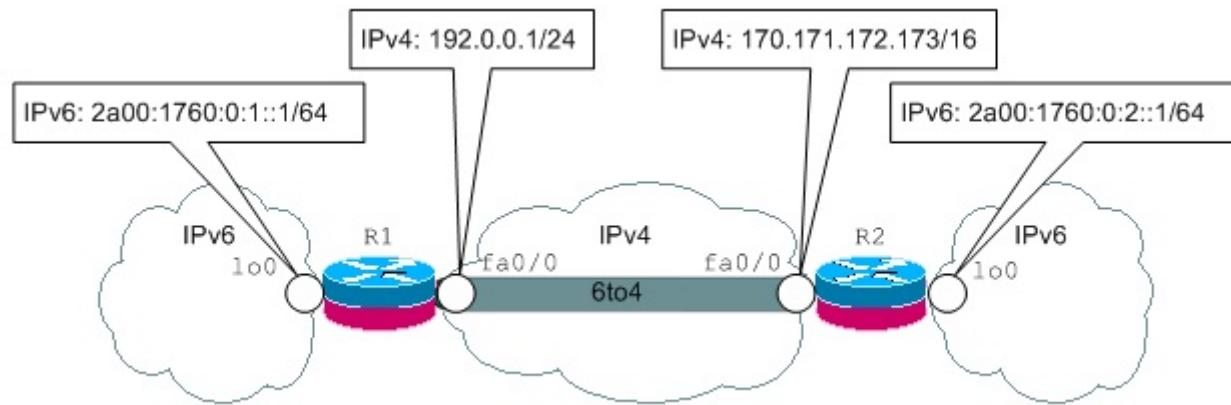
4.0.26.20

```
Router(config)#ipv6 route 2001:7f8:8b:10::/64 2001:7f8:8b:8::2
```

```
Router(config)#ipv6 route ::/0 fa0/1 fe80::1
```

Команды IOS

4.0.26.21a



192.0.0.1 -> c000:1

```
R1(config)#interface tunnel 0
R1(config-if)#ipv6 address 2002:c000:1:1::1/16
R1(config-if)#tunnel source 192.0.0.1
R1(config-if)#tunnel mode ipv6ip 6to4
R1(config-if)#exit
R1(config)#ipv6 route 2a00:1760:0:2::/64 2002:aaab:acad:1::1
```

170.171.172.173 -> aaab:acad

```
R2(config)#interface tunnel 0
R2(config-if)#ipv6 address 2002:aaab:acad:1::1/16
R2(config-if)#tunnel source fa0/0
R2(config-if)#tunnel mode ipv6ip 6to4
R2(config-if)#exit
R2(config)#ipv6 route 2a00:1760:0:1::/64 2002:c000:1:1::1
```

Пример туннеля 6to4

4.0.26.21b

Последовательность действий при передаче через туннель пакета (сформированного либо транзитного), предназначенного соседу по туннелю.

После обращения к таблице IPv6-маршрутизации будет установлено что следующий в звене маршрутизатор не требуется. Будет определен выходной интерфейс -- в данном случае туннельный интерфейс 6to4 (router-to-router, топологически point-to-multipoint).

При туннелировании для выполнения инкапсуляции вместо привлечения ARPчитываются параметры туннеля.

Если пакет не транзитный, то в качестве IPv6-адреса источника будет подставлен IPv6-адрес туннельного интерфейса.

IPv6-адрес назначения задан прикладным процессом либо, если пакет транзитный, уже имеется в пакете.

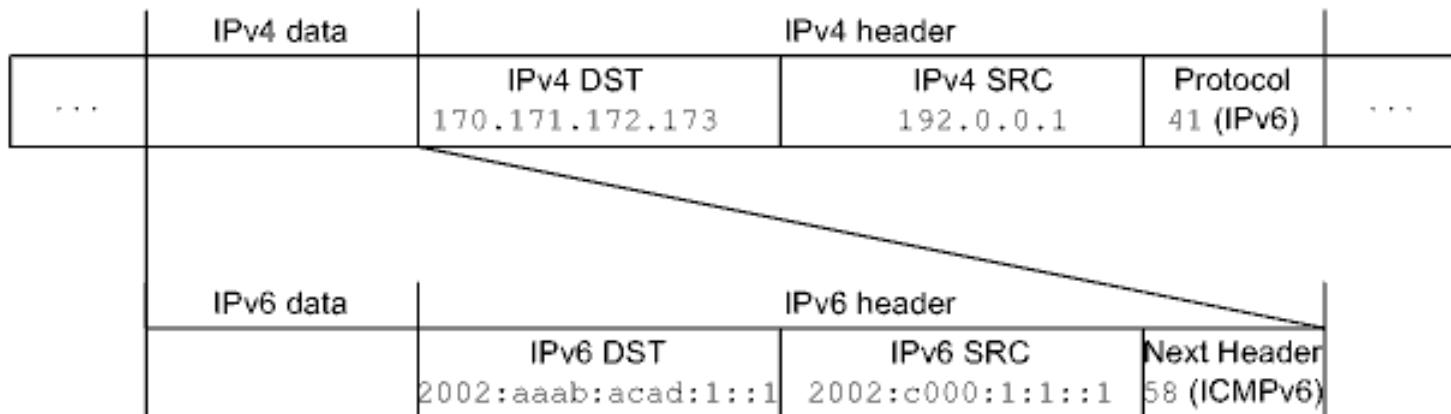
В качестве IPv4-адреса источника будет подставлен IPv4-адрес граничной точки источника туннельного интерфейса.

IPv4-адрес назначения будет выделен автоматически из IPv6-адреса назначения, так как IPv6-адрес назначения является 6to4-адресом (граничная точка назначения не задана и вычисляется автоматически).

Пример туннеля 6to4

4.0.26.21с

```
R1(config)#interface tunnel 0
R1(config-if)#ipv6 address 2002:c000:1:1::1/16
R1(config-if)#tunnel source 192.0.0.1
R1(config-if)#tunnel mode ipv6ip 6to4
R1(config-if)#exit
...
R1#ping 2002:aaab:acad:1::1
```



Дальнейшая пересылка сформированного IPv4-пакета по СПД будет основываться на IPv4-маршрутизации (сначала будет задействована текущая таблица IPv4-маршрутизации).

Пример туннеля 6to4

4.0.26.21d

Отличия при передаче через туннель пакета в случае, когда сосед по туннелю выступает в роли маршрутизатора следующего звена.

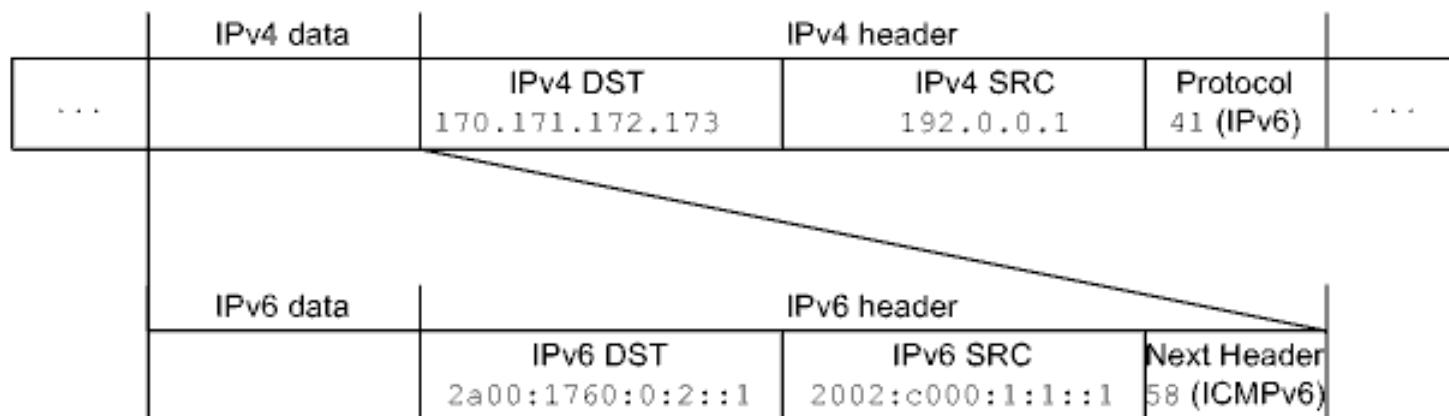
После обращения к таблице IPv6-маршрутизации будет установлено что маршрутизатор следующего звена требуется.

IPv4-адрес назначения будет выделен из указанного в маршруте 6to4-адреса маршрутизатора следующего звена (если в маршруте указать выходной интерфейс, то для вычисления граничной точки назначения «зацепиться» будет не за что).

Пример туннеля 6to4

4.0.26.21e

```
R1(config)#interface tunnel 0
R1(config-if)#ipv6 address 2002:c000:1:1::1/16
R1(config-if)#tunnel source 192.0.0.1
R1(config-if)#tunnel mode ipv6ip 6to4
R1(config-if)#exit
R1(config)#ipv6 route 2a00:1760:0:2::/64 2002:aaab:acad:1::1
...
R1#ping 2a00:1760:0:2::1
```



Пример туннеля 6to4

