

IPv6

RFC-2460

RFC-4291

Отличия от IPv4

Расширение адресации

- длина адреса расширена до 128 бит
позволяет обеспечить больше уровней иерархии
адресации, увеличить число адресуемых узлов,
упростить авто-конфигурацию
- добавлено субполе "scope" (группа адресов)
- новый тип адреса "anycast"

Отличия от IPv4

Спецификация формата заголовков

- Некоторые поля заголовка IPv4 отбрасываются или делаются опциональными, уменьшая издержки, связанные с обработкой заголовков.

Отличия от IPv4

Улучшенная поддержка расширений и опций

- Изменение кодирования опций IP-заголовков позволяет облегчить переадресацию пакетов, ослабляет ограничения на длину опций, и делает более доступным введение дополнительных опций в будущем

Отличия от IPv4

Возможность пометки потоков данных

- Введена возможность пометчать пакеты, принадлежащие определенным транспортным потокам

Отличия от IPv4

Идентификация и защита частных обменов

- введена спецификация идентификации сетевых объектов или субъектов, для обеспечения целостности данных и, при желании, защиты частной информации

Формат заголовка IPv6

0	4	12	16	24	31
Version (6)	Traffic Class	Flow Label			
Payload Length			Next Header	Hop Limit	
Source Address (IP - адрес отправителя)					
Destination Address (IP - адрес получателя)					

Поля заголовка IPv6

Traffic class (8 бит).
Класс трафика.

Используется для идентификации и различения различных классов либо приоритетов.

- 0 по-умолчанию
- Промежуточные узлы могут изменять значение

Поля заголовка IPv6

Flow Label (20 бит).

Метка потока.

Это поле может использоваться отправителем для того, чтобы пометить пакеты, которые требуют специальной обработки сетевыми модулями IPv6.

Все пакеты, принадлежащие к одному потоку, должны отправляться по одному и тому же адресу назначения и с одним и тем же приоритетом.

Поля заголовка IPv6

Payload Length (16 бит).
Длина данных.

Длина данных пакета (в байтах), которые
следуют за заголовком.

Если величина этого поля равна 0, то длина
данных дейтаграммы более 65535 и хранится в
поле Jumbo Payload

Поля заголовка IPv6

Next Header (8 бит).

Поле следующего заголовка.

Это поле содержит информацию типа заголовка, который следует за заголовком IPv6.

Поля заголовка IPv6

Hop Limit (8 бит).

Поле ограничения пересылок.

Величина этого поля уменьшается на 1 при прохождении дейтаграммой шлюза или хоста. Если величина этого поля равна 0, дейтаграмма уничтожается.

Поля заголовка IPv6

Source Address (128 бит).

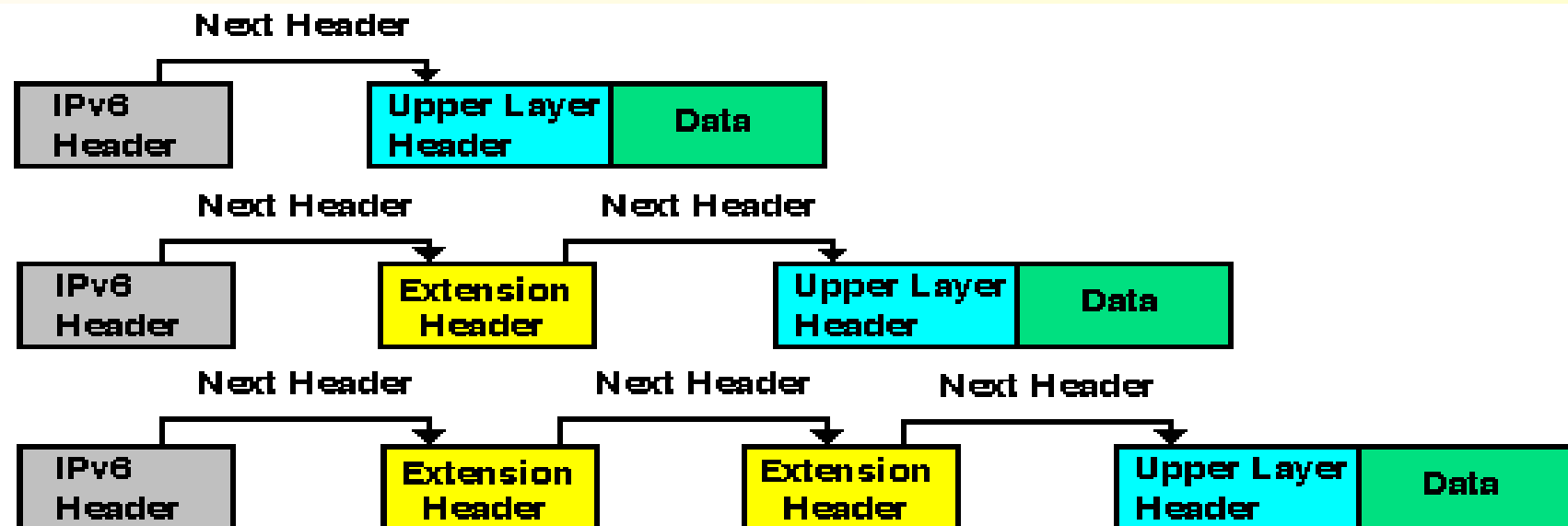
Адрес отправителя.

Destination Address (128 бит).

Адрес получателя.

Если в заголовке присутствует вложенный заголовок маршрутизации (Routing header), то поле **адреса получателя** может и не быть адресом назначения.

Инкапсуляция заголовков



Заголовки расширений

1. IPv6.
2. Hop-by-Hop Options
Тут может быть опция “Jumbo Frames”.
3. Destination Options (опции получателя 1).
4. Routing (маршрутизация).
5. Fragment (фрагментация).
6. Authentication (аутентификация).
7. Encapsulating Security Payload (дополнительная аутентификация).
8. Destination Options (опции получателя 2).
9. Заголовок верхнего уровня (например, TCP).

Адресация в IPv6

Типы адресов:

Unicast

Anycast

Multicast

Unicast

Идентификатор одиночного интерфейса.

Пакет, посланный по уникастному адресу, доставляется интерфейсу, указанному в адресе.

Anycast

Идентификатор набора интерфейсов
(принадлежащих разным узлам).

Пакет, посланный по эникастному адресу,
доставляется одному из интерфейсов,
указанному в адресе (ближайший, в
соответствии с мерой, определенной протоколом
маршрутизации).

Multicast

Идентификатор набора интерфейсов (обычно принадлежащих разным узлам).

Пакет, посланный по мультикастинг-адресу, доставляется всем интерфейсам, заданным ЭТИМ адресом.

В IPv6 не существует широковещательных адресов, их функции переданы мультикастинг-адресам.

Представление записи адресов

Основная форма имеет вид

$x:x:x:x:x:x:x:x$

где 'x' шестнадцатеричные 16-битовые числа.

fedc:ba98:7654:3210:FEDC:BA98:7654:3210
1080:0:0:0:8:800:200C:417A

Представление записи адресов

Синтаксис для удаления лишних нулей в адресе.

Использование записи "::" указывает на наличие групп из 16 нулевых бит. Комбинация "::" может появляться только при записи адреса.

Последовательность "::" может также использоваться для удаления из записи начальных или завершающих нулей в адресе.

$1080:0:0:0:8:800:200c:417a = 1080::8:800:200c:417a$

$Ff01:0:0:0:0:0:0:43 = ff01::43$

$0:0:0:0:0:0:0:1 = ::1$

Представление записи адресов

Альтернативная форма для работы с ipv4 и Ipv6

$x:x:x:x:x:x:d.d.d.d$

где 'x' шестнадцатеричные 16-битовые коды адреса,
а 'd' десятичные 8-битовые, составляющие
младшую часть адреса (стандартное IPv4
представление).

$0:0:0:0:0:FFFF:129.144.52.38 = ::FFFF:129.144.52.38$

Идентификация типа адреса

Идентификация происходит по первым битам адреса.

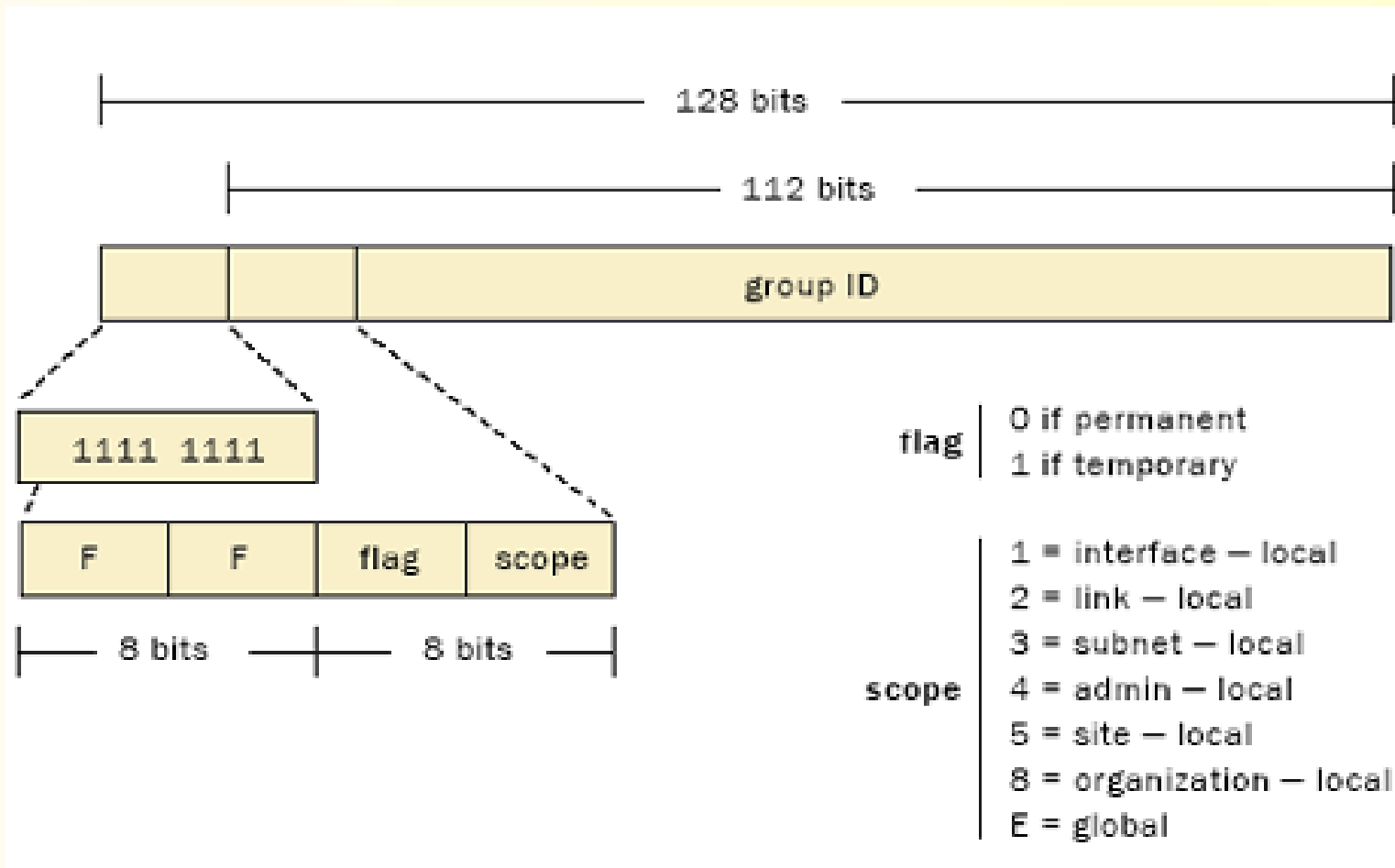
Тип адреса	Префикс	Нотация IPv6
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Адрес IPv6, отображенный на IPv4	(96 bits)	::ffff:xx.xx.xx.xx
Global Unicast	Все остальные	

0:0:0:0:0:0:0:0 – Специальный тип адреса,
который не должен присваиваться ни одному
узлу

0:0:0:0:0:0:0:1 – Loopback (адрес обратной связи)

Link-local unicast:
1111111010 :: interface ID (64 бита)

Multicast Address



Сокеты IPv6

Структура адреса сокета IPv6

```
struct in6_addr {  
    unsigned char s6_addr[16];  
}  
  
struct sockaddr_in6 {  
    sa_family_t sin6_family; //AF_INET6  
    in_port_t sin6_port;  
  
    // Приоритет и метка потока  
    uint32_t sin6_flowinfo;  
  
    struct in6_addr sin6_addr;  
  
    // Набор интерфейсов  
    uint32_t sin6_scope_id;  
}
```

sockaddr_storage

Новая универсальная структура адреса сокета —
часть API сокетов IPv6

```
struct sockaddr_storage {  
    uint8_t ss_len;  
    sa_family_t ss_family;  
    // Данные, достаточные для хранения адреса  
    // любого сокета  
}
```

Пример создания универсального клиента, работающего по IPv4 и IPv6

```
struct in6_addr serveraddr;  
struct addrinfo hints, *res=NULL;  
  
    memset(&hints, 0x00, sizeof(hints));  
    hints.ai_flags      = AI_NUMERICSERV;  
    hints.ai_family     = AF_UNSPEC;  
    hints.ai_socktype   = SOCK_STREAM;
```

```
rc = inet_pton(AF_INET, server, &serveraddr);
if (rc == 1)      /* IPv4 */
{
    hints.ai_family = AF_INET;
    hints.ai_flags |= AI_NUMERICHOST;
}
else
{
    rc = inet_pton(AF_INET6, server,
&serveraddr);
    if (rc == 1) /* IPv6 */
    {
        hints.ai_family = AF_INET6;
        hints.ai_flags |= AI_NUMERICHOST;
    }
}
```

```
getaddrinfo(server, servport, &hints, &res);
```

```
sd = socket(res->ai_family, res->ai_socktype,  
res->ai_protocol);
```

```
connect(sd, res->ai_addr, res->ai_addrlen);
```


Совместимость IPv4 и IPv6

	Сервер (узел)	Сервер (узел)	Сервер (узел)	Сервер (узел)
Клиент (узел)	ipv4 (ipv4)	ipv4 (ipv6)	ipv4 (both)	ipv6 (both)
ipv4 (ipv4)	ipv4	-	ipv4	ipv4
ipv6 (ipv6)	-	ipv6	-	ipv6
ipv4 (both)	ipv4	-	ipv4	ipv4
ipv6 (both)	ipv4	ipv6	-	ipv6