

Многоадресная передача

Multicast

специальная форма широковещания, при которой копии пакетов направляются определённому подмножеству адресатов.

Дейтаграмму многоадресной передачи должны получать только заинтересованные в ней интерфейсы – т.е. на этих интерфейсах работают приложения, желающие принять участие в сеансе многоадресной передачи данных.

Multicast и канальный уровень

Современные сетевые карты умеют принимать кадры с установленным битом режима многоадресной передачи.

Для беспроводных сетей 802.11, передача кадра может задержаться в зависимости от режима сохранения энергии (power-save mode).

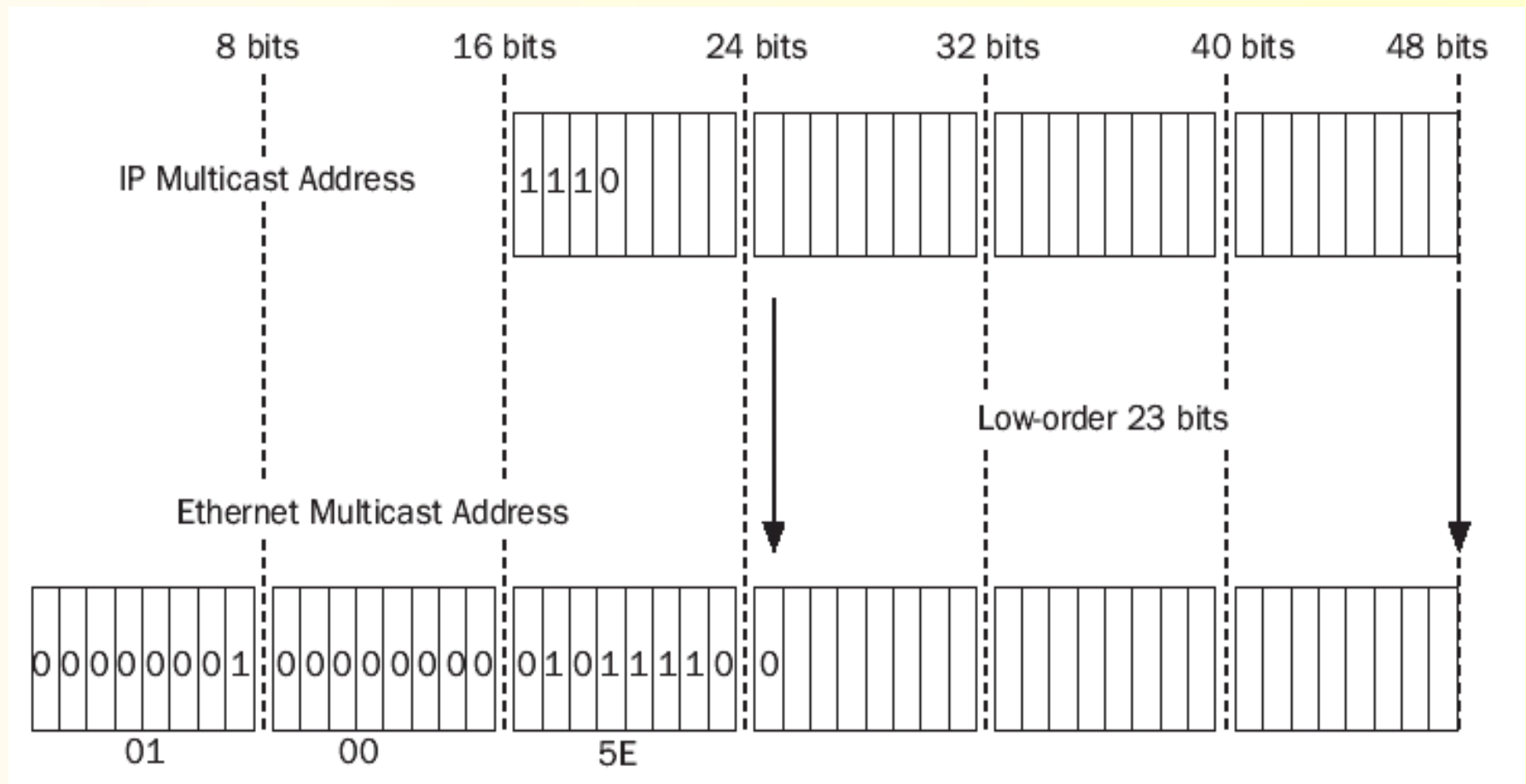
IPv4 multicast

Используются адреса класса D:

224.0.0.0 – 239.255.255.255

Младшие 28 бит – идентификатор группы многоадресной передачи (multicast group ID)

Multicast: Ethernet & IPv4



Используется универсально управляемый адрес Ethernet

Специальные адреса IPv4

224.0.0.1

группа всех узлов (all-hosts group)

К этой группе должны присоединиться все узлы в сети, имеющие возможность многоадресной передачи.

Специальные адреса IPv4

224.0.0.2

группа всех маршрутизаторов (all-routers group)

К этой группе должны присоединиться все маршрутизаторы сети, поддерживающие многоадресную передачу.

Специальные адреса IPv4

224.0.0.0 -224.0.0.255

Локальные адреса на канальном уровне
(link local)

Предназначены для низкоуровневого
определения топологии и служебных
протоколов.

Дейтаграммы с такими адресами никогда не
передаются маршрутизатором дальше.

Ограничение области действия

Исторически сложилось, что поле TTL в IPv4 выполняло роль поля области действия многоадресной передачи:

- 0: локальный в пределах узла
- 1: локальный в пределах сети
- <32: локальный в пределах сайта
- <64: локальный в пределах региона
- <128: локальный в пределах континента
- <255: глобальный

Ограничение области действия

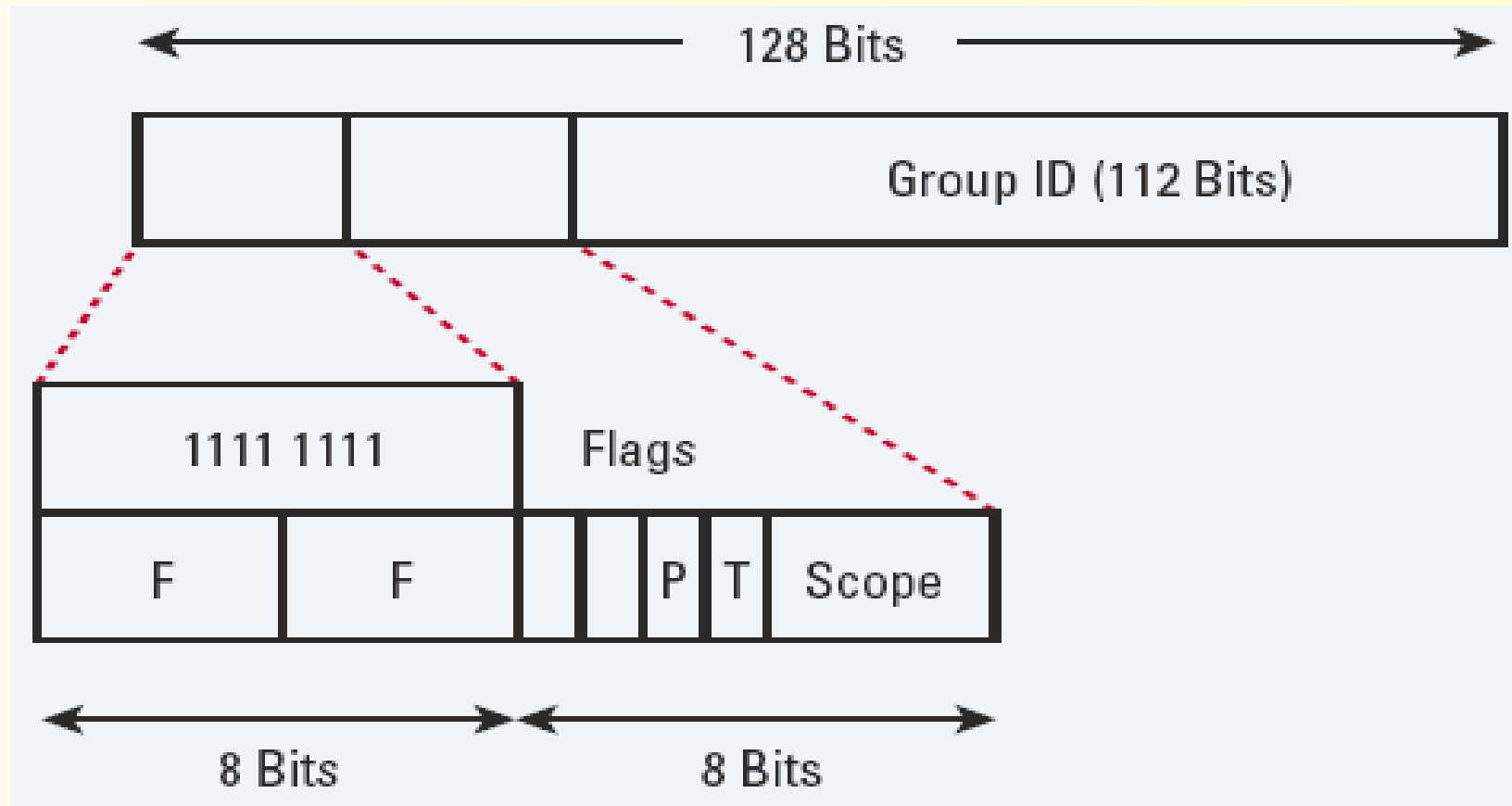
Более предпочтительно использовать
административное управление областями
действия

Адреса 239.0.0.0 - 239.255.255.255 –
пространство адресов с административным
ограничением областью действия

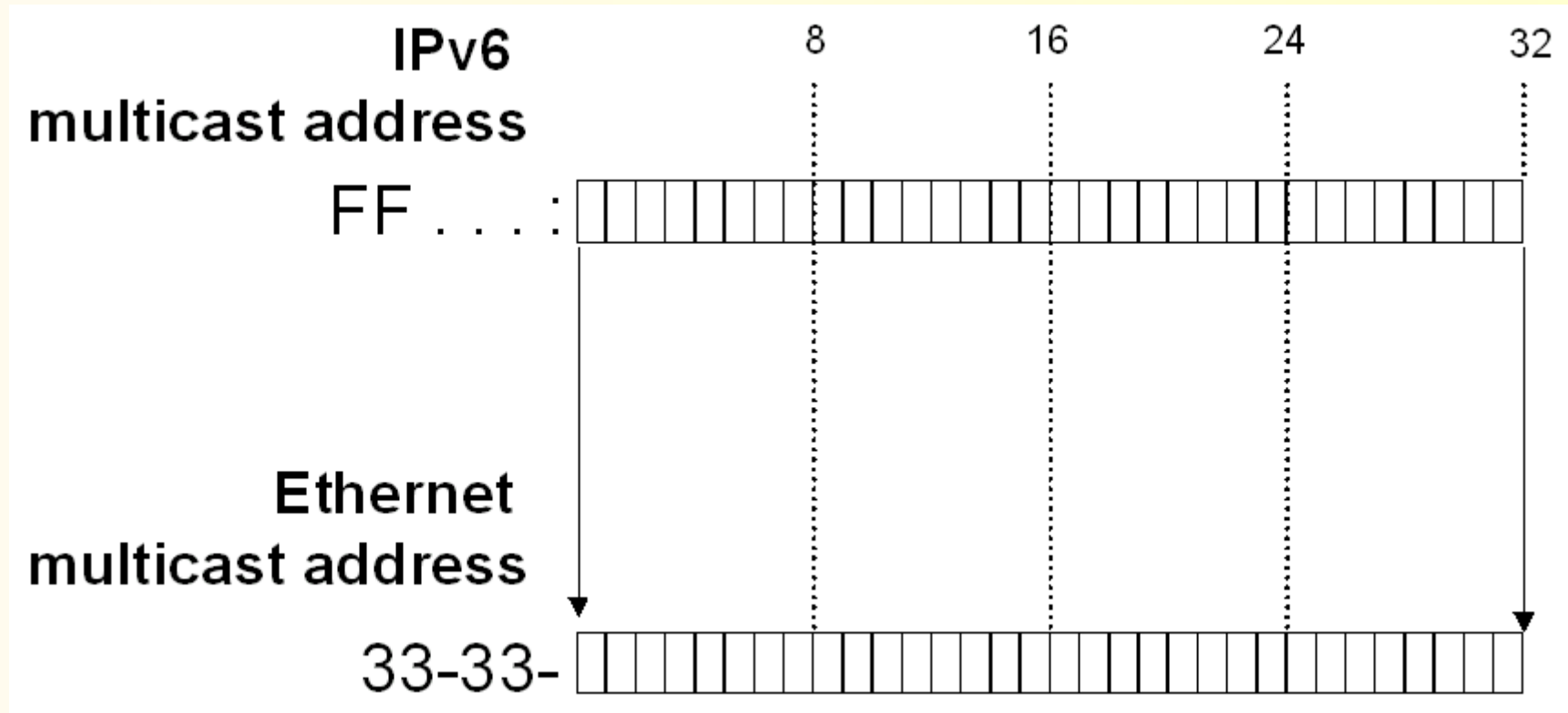
239.255.0.0 – 239.255.255.255 – локальная
область действия (~site-local)

239.192.0.0 – 239.195.255.255 – локальная
область действия в пределах организации

Multicast IPv6 Address



Multicast: Ethernet & IPv6



Используется локально управляемый групповой адрес Ethernet

Группы адресов multicast IPv6

Флаги $P=0$, $T=0$: группа заранее известных адресов (well-known)

Флаги $P=0$, $T=1$: группа временных адресов (transient)

Флаги $P=1$, $T=1$: адрес считается назначенным на основе одноадресного префикса

Специальные адреса IPv6

FF02::1

группа всех узлов (all-nodes group)

К этой группе должны присоединиться все узлы в сети, имеющие возможность многоадресной передачи

(включая маршрутизаторы, принтеры, тостеры и холодильники, если они имеют IPv6 адрес).

В отличие от IPv4 присоединение является обязательным!

Специальные адреса IPv6

FF02::2

группа всех маршрутизаторов (all-routers group)

К этой группе должны присоединиться все маршрутизаторы сети, поддерживающие многоадресную передачу.

Аналогичен адресу 224.0.0.2.

Область действия: scope

- 1: Node-local – локальная в пределах узла
- 2: Link-local – локальная в пределах физической сети
- 4: Admin-local – локальная в пределах области администрирования
- 5: Site-local – локальная в пределах сайта
- 8: Organization-local – локальная в пределах организации
- 14: глобальная

4,5,8 определяются администратором

Сеансы многоадресной передачи

Сочетание адреса многоадресной передачи и порта транспортного уровня часто называют сеансом (session).

Пример: потоковое мультимедиа делится на 2 сеанса – аудио и видео. При этом иногда выгодно использовать не только разные порты, но и разные группы.

Многоадресная передача в глобальной сети

Маршрутизаторы общаются между собой при помощи какого-либо протокола маршрутизации многоадресной передачи (например DVMRP).

Когда узел присоединяется к группе, он отправляет всем маршрутизаторам в своей сети сообщение IGMP.

Маршрутизаторы обмениваются этой информацией по протоколу маршрутизации многоадресной передачи.

Программирование

Управление режимом, присоединение к группам, отсоединение и др. осуществляется через вызовы **getsockopt** и **setsockopt**.

Структуры ip_mreq и ipv6_mreq

```
struct ip_mreqn{
    // IP адрес группы для группового вызова
    struct in_addr imr_multiaddr;
    // IP адрес локального интерфейса
    struct in_addr imr_interface;
    // указатель на интерфейс
    int      imr_ifindex;
}
```

```
struct ipv6_mreq{
    // адрес многоадресной передачи
    struct in6_addr imr_multiaddr;
    // индекс интерфейса
    unsigned int ipv6mr_interface;
}
```

Сструктура group_req

```
struct group_req{  
    // индекс интерфейса  
    uint32_t gr_interface;  
    // адрес многоадресной передачи ipv4 или ipv6  
    struct sockaddr_storage gr_group;  
}
```

Структуры

ip_mreq_source и group_source_req

```
struct ip_mreq_source{  
    struct in_addr imr_multiaddr;  
    // адрес источника ipv4  
    struct in_addr imr_sourceaddr;  
    struct in_addr imr_interface;  
}
```

```
struct group_source_req{  
    uint32_t gr_interface;  
    //адрес группы  
    struct sockaddr_storage gsr_group;  
    // адрес источника ipv4 или ipv6  
    struct sockaddr_storage gsr_source;  
}
```

Параметры сокетов многоадресной передачи влияющие на отправку

IP_MULTICAST_IF	struct ip_mreq	интерфейс по умолчанию для исходящих пакетов
IP_MULTICAST_TTL	u_char	TTL по умолчанию
IP_MULTICAST_LOOP	u_char	закольцовка
IPV6_MULTICAST_IF	u_int	интерфейс по умолчанию для исходящих пакетов
IPV6_MULTICAST_HOPS	int	TTL по умолчанию
IPV6_MULTICAST_LOOP	u_int	закольцовка

Присоединение к группе

IP_ADD_MEMBERSHIP (struct ip_mreq)
IPV6_JOIN_GROUP (struct ipv6_mreq)
MCAST_JOIN_GROUP (struct group_req)

Если необходимо присоединиться к определенной области видимости – то надо использовать соответствующий адрес (поле score для ipv6). Пример: ff01:101 и ff0e:101.

Выход из группы

IP_DROP_MEMBERSHIP (struct ip_mreq)
IPV6_LEAVE_GROUP (struct ipv6_mreq)
MCAST_LEAVE_GROUP (struct group_req)

Если локальный интерфейс не задан, то
удаляется первое совпадающее с искомым
вхождение в группу.

Блокирование получения трафика от конкретного источника

IP_BLOCK_SOURCE (struct ip_mreq_source)
MCAST_BLOCK_SOURCE
(struct group_source_req)

Если все сокеты, присоединенные к группе, заблокировали один и тот же источник, система может проинформировать маршрутизаторы о нежелательности трафика.

Разблокирование получения трафика от конкретного источника

IP_UNBLOCK_SOURCE (struct ip_mreq_source)
MCAST_UNBLOCK_SOURCE
(struct group_source_req)

Присоединение и отключение от группы конкретного источника

IP_ADD_SOURCE_MEMBERSHIP

(struct ip_mreq_source)

MCAST_JOIN_SOURCE_GROUP

(struct group_source_req)

IP_DROP_SOURCE_MEMBERSHIP

(struct ip_mreq_source)

MCAST_LEAVE_SOURCE_GROUP

(struct group_source_req)

Пример 1.

```
struct sockaddr_in sin;  
int sd;  
struct ip_mreq group;  
  
sd = socket(AF_INET, SOCK_DGRAM, 0);  
  
sin.sin_family = AF_INET;  
sin.sin_port = htons(4321);  
sin.sin_addr.s_addr = INADDR_ANY;
```

```
int reuse = 1;
setsockopt(sd, SOL_SOCKET, SO_REUSEADDR,
           (char *)&reuse, sizeof(reuse));

bind(sd, (struct sockaddr*)&sin, sizeof(sin))

group.imr_multiaddr.s_addr =
    inet_addr("226.1.1.1");
group.imr_interface.s_addr =
    inet_addr("127.0.0.1");

setsockopt(sd, IPPROTO_IP,
           IP_ADD_MEMBERSHIP,
           (char *)&group,
           sizeof(group));
```

Пример 2.

```
struct in_addr localInterface;  
struct sockaddr_in sin;  
int sd;  
sd = socket(AF_INET, SOCK_DGRAM, 0);  
  
groupSock.sin_family = AF_INET;  
groupSock.sin_addr.s_addr =  
    inet_addr("226.1.1.1");  
groupSock.sin_port = htons(4321);  
  
localInterface.s_addr = inet_addr("127.0.0.1");  
setsockopt(sd, IPPROTO_IP, IP_MULTICAST_IF,  
    (char *)&localInterface,  
    sizeof(localInterface));
```