

# Мрежов протокол IPv6

# Какво ще научим?

- Защо се налага преход към IPv6
- Предимства на IPv6 пред IPv4 освен дължината
- Формат на IPv6 адрес. Префикси.
- Типове IPv6 адреси. Защо няма бродкаст.
- ICMPv6 вместо ARP. Автоконфигуриране.
- Преход от IPv4 към IPv6

# Предпоставки за прехода

Преходът към IPv6 е **неизбежен**.

**IPv4** адресите са **изчерпани**.

IPv6 **не е обратно съвместим** с IPv4, необходими са промени в мрежови устройства и услуги.

Трудности при **едновременна работа на IPv4 и IPv6**, която е наложителна в дългия преходен период.

# Подобрения в IPv6

Според RFC 4291: адресното пространство от 32-битово става 128-битово:  $2^{32}$  ( $4.3 \times 10^9$ ) с/у  $2^{128}$  ( $3.4 \times 10^{38}$ ).

Автоконфигуриране. RFC 4862 дефинира автоматично (**plug-and-play**) присвояване на адрес без помощта на DHCP сървър като в IPv4.

# Header

В IPv6 е по-опростено от IPv4; с фиксирана дължина 40 байта (RFC 2460):

- 2 \* 16-byte IPv6 адреса;
- 8 байта друга информация.

По-бързо и лесно обработване на пакетите.

# Пример на IPv6 Packet Header

## Internet Protocol Version 6

Version: 6

Traffic class: 0x00

Flowlabel: 0x00000

Payload length: 24

Next header: TCP (0x06)

Hop limit: 64

Source address: 2001:0:53aa:64c:0:7fff:b85c:4985

Destination address: 2001:200:0:8002:203:47ff:fea5:3085

Transmission Control Protocol, Src Port: 51001 (51001), Dst Port: http (80), Seq: 0, Len: 0

# Структура на заглавието

Version (4)	Traffic Class (8)	Flow Label (20 bits)	
Payload length (16)		Next Header (8)	Hop Limit (8)
Source Address (128 bits)			
Destination Address (128 bits)			

traffic class (заменя IPv4 ToS);

flow label (ново QoS management);

payload length (до 64KB);

next header (заменя IPv4 protocol);

hop limit (заменя IPv4 TTL).

# Поддържа IPsec

IP security (IPsec) съдържа протоколи за аутентикация на изпращача и гарантиране на данните в IP комуникациите:

- Encapsulating Security Payload (ESP);
- Authentication Header (AH);
- Internet Key Exchange (IKE).

IPsec е част от IPv6.

Задължителен е IPsec за защита на Mobile IPv6 и OSPFv3.

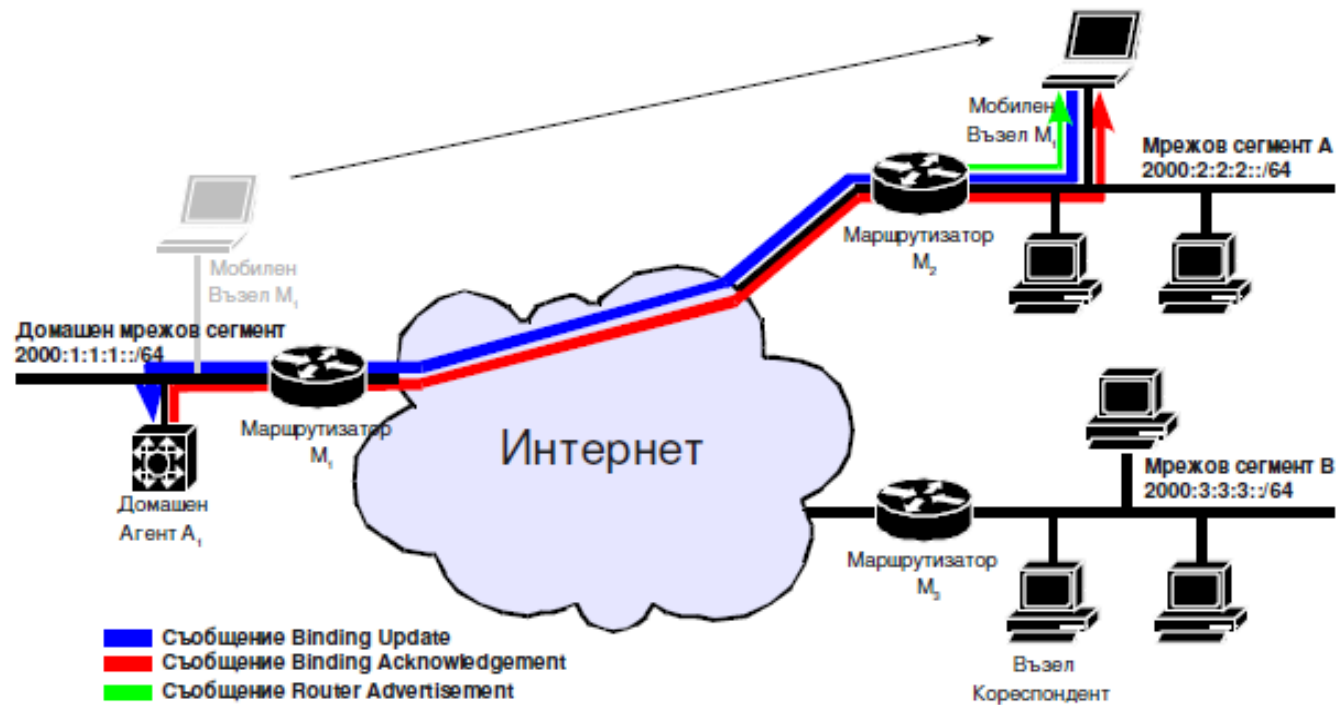


# Mobile IPv6

**MIPv6** поддържа **roaming** за мобилни възли (RFC 3775).

MIPv6 използва **Neighbor Discovery** (RFC 4861), за да реши проблема с прехвърлянето (**handover**) на мрежов слой и оптимизация на маршрута (RFC 4449).

# Mobile IPv6



Фигура 2.3: Домашният агент A<sub>1</sub>, получава CoA адреса на преместилия се в сегмент А мобилния възел M<sub>1</sub>, чрез съобщение BU и в отговор изпраща съобщение BACK

# Quality of Service (QoS)

IP третира всички пакети еднакво – **best effort**.

**TCP** (Transmission Control Protocol) гарантира доставянето, но не контролира закъснение, честотна лента и т.п.

**QoS** – опции за въвеждане на политики и приоритети на трафика.

IPv4 и IPv6 сходни QoS възможности:

**Differentiated Services** и **Integrated Services**.

# QoS

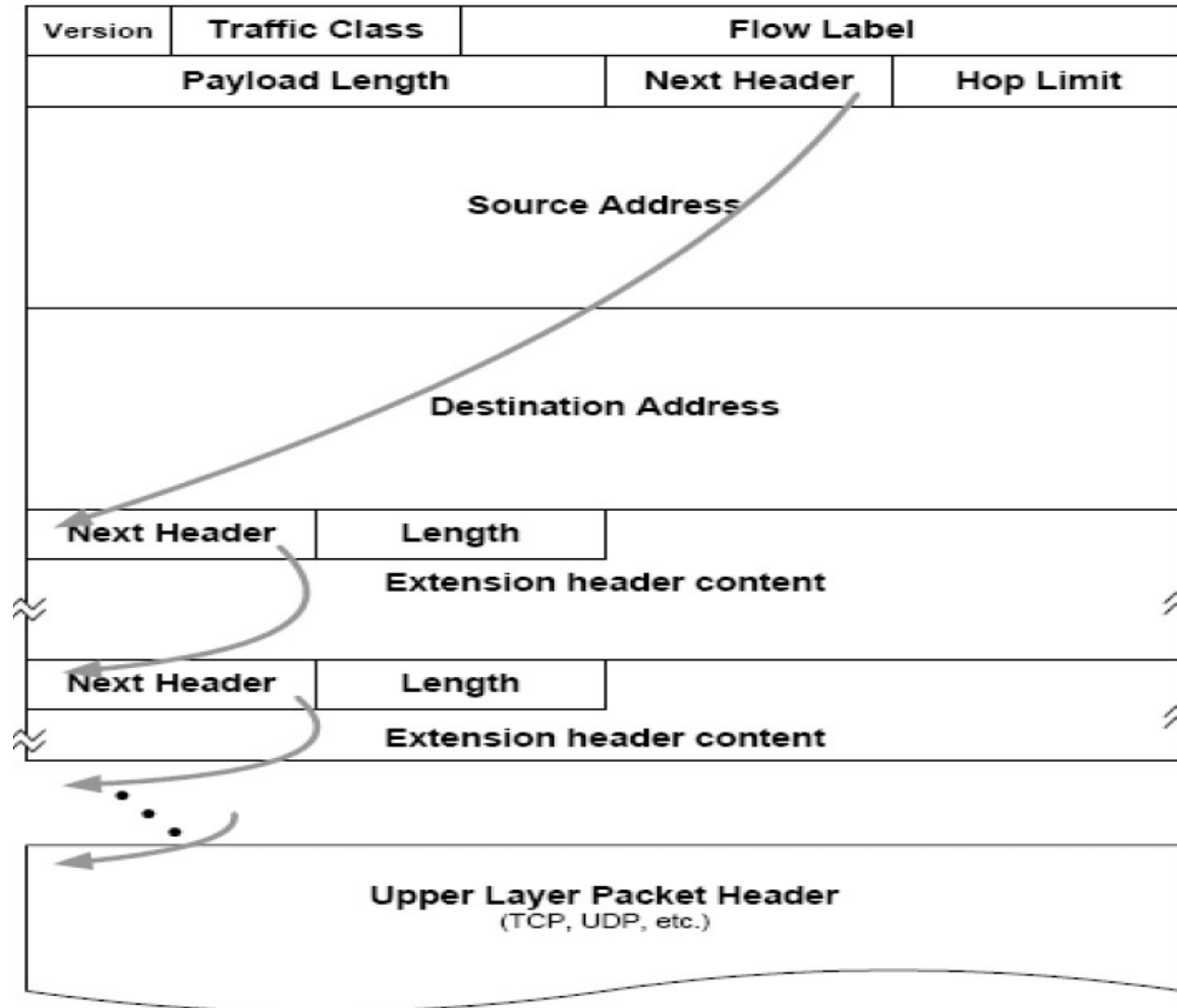
В IPv6 header има **две полета** за QoS:

- Traffic Class и
- Flow Label.

**Traffic Class** е разширено по-прецизно диференциране на различните типове трафик.

Новото **Flow Label** поле - съдържа етикет за идентифициране или приоритетизиране на определен поток от пакети като **VoIP** или видеоконференции, чувствителни към времето на доставяне.

# IPv6 Extension Header



# IPv6 Extension Header

Extension Header	Type	Remarks
Hop-by-hop Options	0	used for options that apply to intermediate routers
Routing	43	used for source routing
Fragment	44	processed only by the final recipient
Destination Options	60	used for options that apply only for the final recipient
Authentication header (AH)	51	used for IPsec integrity protection
Encapsulating Security Payload (ESP)	50	used for IPsec integrity and confidentiality protection
Mobility	135	used for managing mobile IPv6 bindings

# Jumbograms

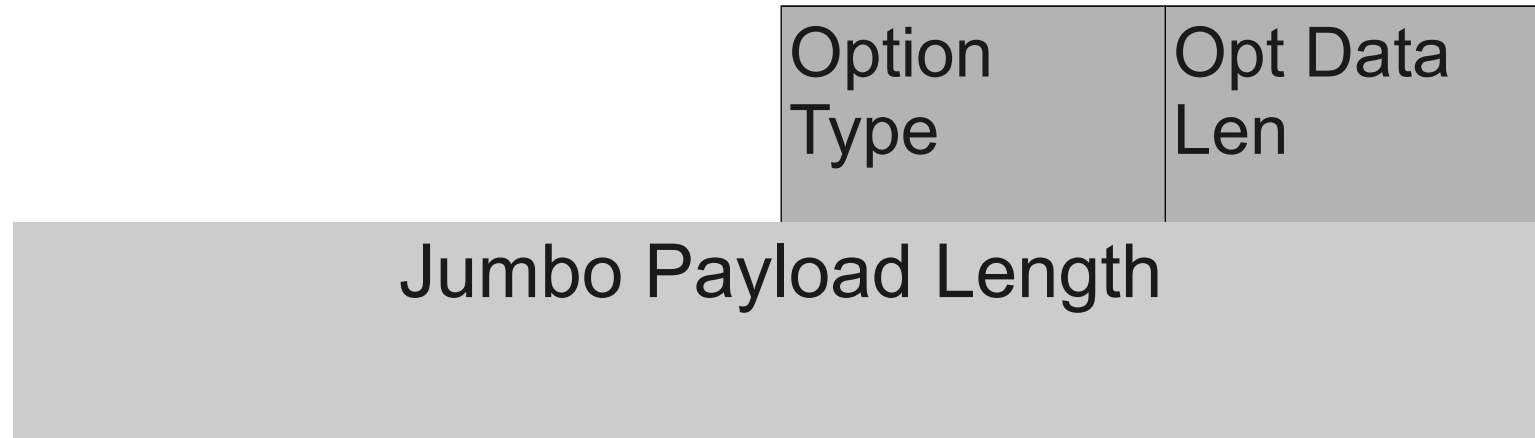
RFC 2675 дефинира IPv6 Hop-by-Hop Option - **jumbograms**, IPv6 пакет с поле за данни (payload) > 65 535 октета.

Важи за IPv6 интерфейси, които могат да поемат кадри с такива дължини ( $\geq 1$  gbps).

16-бит поле **Payload Length** (в IPv6 Header) = 0

След това:

# Jumbograms



Option Type (8-bit) = C2 (1100 0010)

Opt Data Len (8-bit) = 4

Jumbo Payload Length (32-bit) цяло число.  
Jumbograms  $\geq 65\,536$  bytes и да не са  
фрагментирани.



# IPv6 Fragment Extension Header

В IPv6 фрагментирането на пакетите става още при източника.

В IPv4 рутерът фрагментира пакета, когато MTU на следващия канал е по-малък. Ако отсреща не се възстанови оригиналният пакет, сесията се разваля.

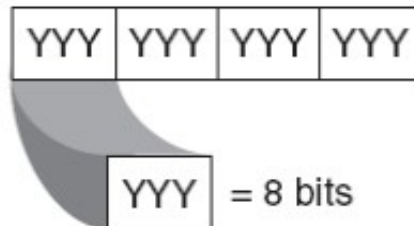
В IPv6 всеки хост използва **Path Maximum Transmission Unit (PMTU) Discovery**, за да научи размера на MTU по пътя, за да не се налага фрагментиране.

# IPv4 vs. IPv6

Property	IPv4	IPv6
Address size and network size	32 bits, network size 8-30 bits	128 bits, network size 64 bits
Packet header size	20-60 bytes	40 bytes
Header-level extension	limited number of small IP options	unlimited number of IPv6 extension headers
Fragmentation	sender or any intermediate router allowed to fragment	only sender may fragment
Control protocols	mixture of non-IP (ARP), ICMP, and other protocols	all control protocols based on ICMPv6
Minimum allowed MTU	576 bytes	1280 bytes
Path MTU discovery	optional, not widely used	strongly recommended
Address assignment	usually one address per host	usually multiple addresses per interface
Address types	use of unicast, multicast, and broadcast address types	broadcast addressing no longer used, use of unicast, multicast and anycast address types
Address configuration	devices configured manually or with host configuration protocols like DHCP	devices configure themselves independently using stateless address autoconfiguration (SLAAC) or use DHCP

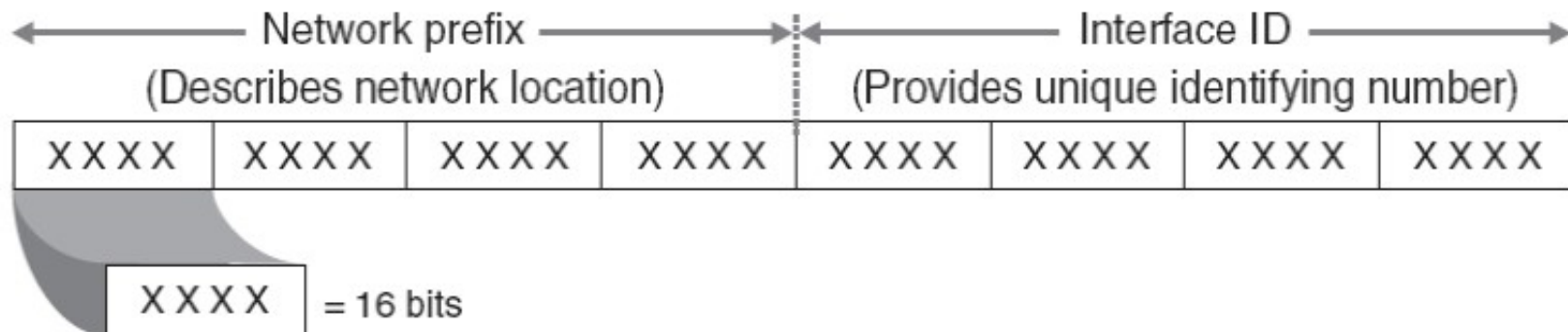
# IPv4 vs. IPv6

## 32-bit IPv4 address



(Resulting in 4,294,967,296 unique IP addresses)

## 128-bit IPv6 address



(Resulting in 340,282,366,920,938,463,374,607,432,768,211,456 unique IP addresses)

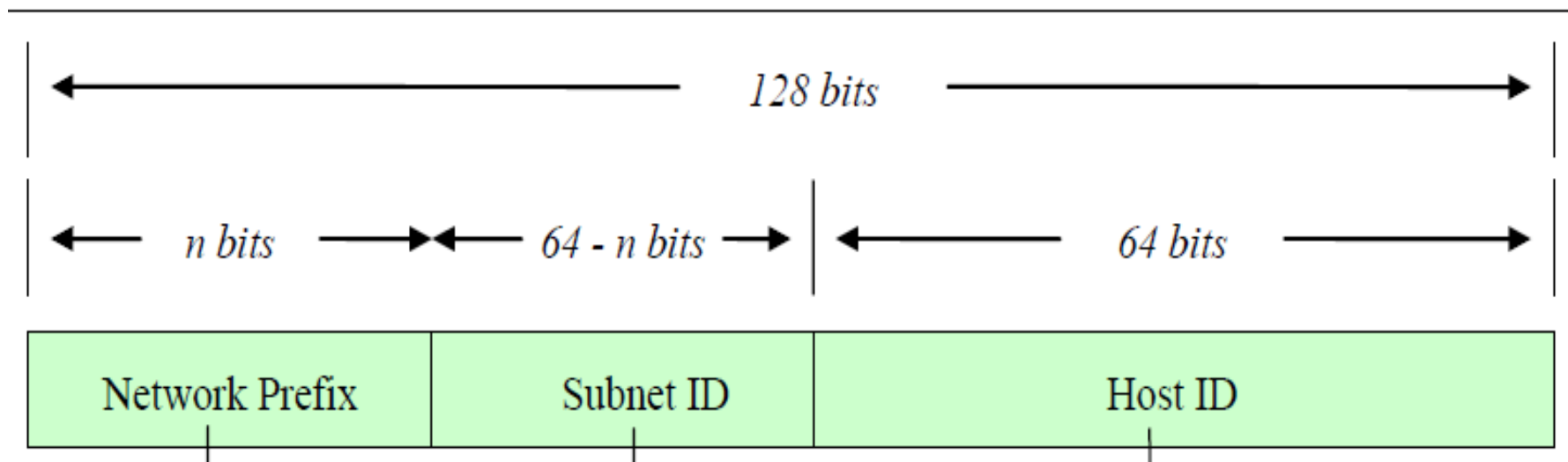
# IPv6 адресиране

IPv6 адрес (пример):

**2001:0db8:9095:02e5:0216:cbff:feb2:7474**

**8** групи с по **4** шестнадесетични числа

$$8 * 4 * 4 = 128$$



# Формат на IPv6 адрес

Мрежовият префикс (**network prefix**) – идентифицира дадена мрежа или специален адрес. Присвоява се от ISP (PA) или RIR (PI).

Идентификаторът на подмрежата (**subnet ID**) – връзка **вътре в мрежов обект**. Присвоява се от администратора на обекта. Един обект  $\geq 1$  subnet IDs. Определя на кой мрежов сегмент принадлежи даден хост.

**host ID** идентифицира конкретен възел в мрежата – конкретен негов интерфейс.

# Префикси в IPv6

Мрежовият префикс (RFC 4291) е аналогичен на означението с “/” на SM в IPv4:

IPv6 address/prefix length

**Например адрес с 32-bit мрежов префикс:**

2001:0db8:9095:02e5:0216:cbff:feb2:7474/32

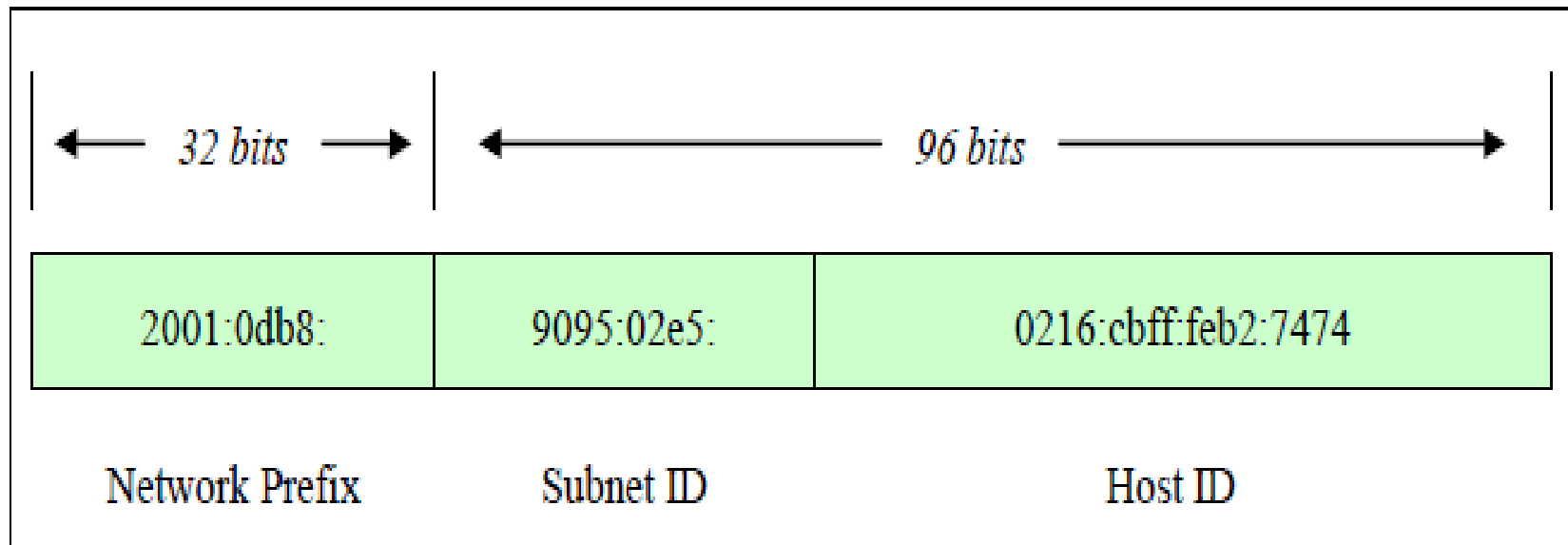
# Алокация на IPv6 префикси

IPv6 (подобно на IPv4) се присвояват от RIRs и ISP.

Големите провайдери (LIRs) могат да получат префикс с минимална дължина 32 бита:

- най-старшите 32 бита са мрежовия префикс;
- останалите 96 бита са на разположение на администратора за раздаване на subnet ID-та и за host ID.

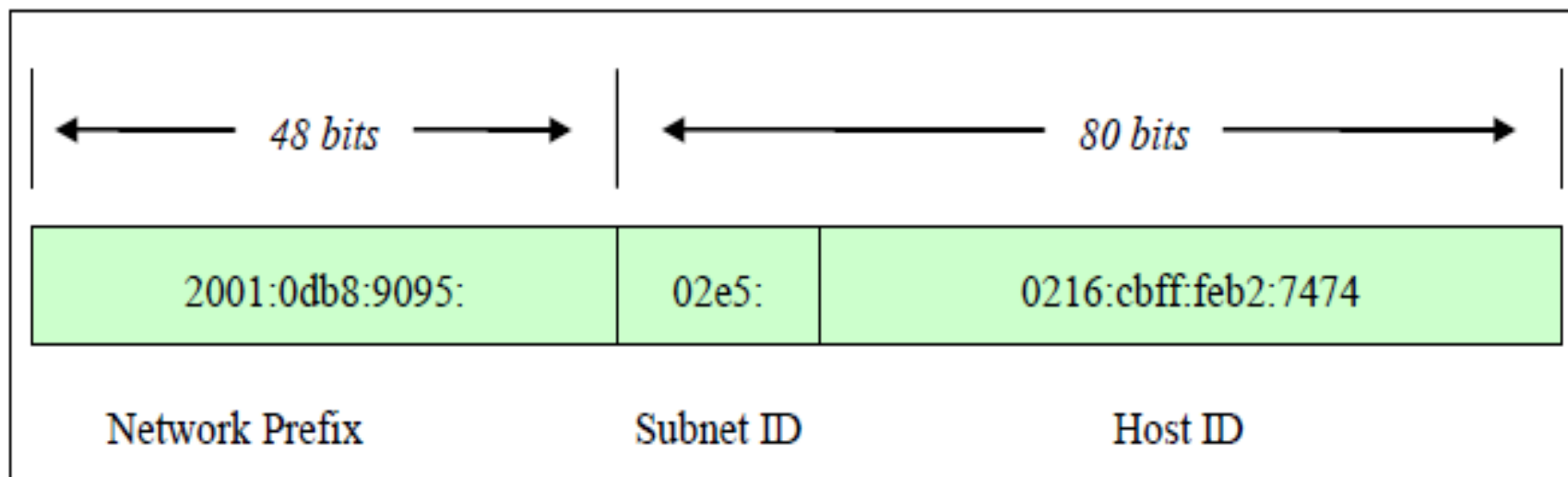
# 32-битов мрежов префикс



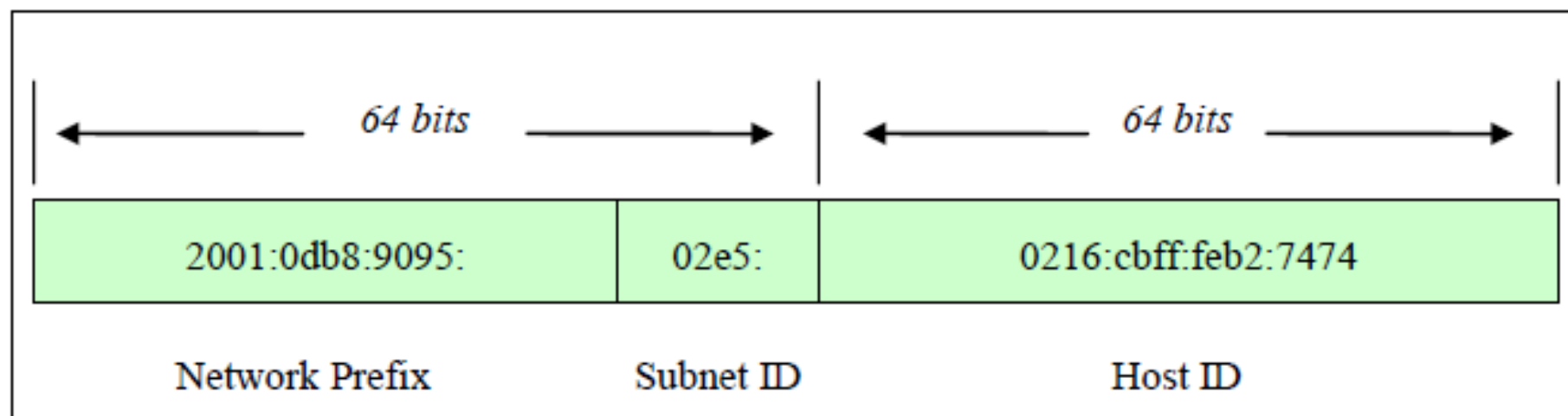


# 48-битов мрежов префикс

Правителствени, образователни, търговски и др. организации обикновено получават от големите ISPs (PA) или от RIRs (PI) 48-битови алокации (/48), оставяйки 80 бита за subnet ID и host ID.



# 64-битов мрежов префикс



Подмрежите в рамките на организация обикновено са 64 бита (**/64**)

64 бита остават за **host ID** - 64-bit идентификатор на интерфейса в **EUI-64** формат.

# Запис на IPv6 адреси

За да се улесни записването на адреси, съдържащи нули, те се компресират по определени правила.

"::" - една или повече 16-битови групи от нули; за компресиране на водещи или завършващи нули.

"::" може да се появи само веднъж в адреса.

Например:

# Запис на IPv6 адреси

2001:DB8:**0:0**:8:800:200C:417A unicast

FF01:**0:0:0:0:0:0:0**:101 multicast

**0:0:0:0:0:0:0**:1 loopback

**0:0:0:0:0:0:0:0** unspecified

Се представят:

2001:DB8::**:**8:800:200C:417A

FF01::**:**101

::**:**1

::**:**

# Запис на IPv6 адреси и префикси

**ПРАВИЛНО** представяне на 60-bit префикс:

2001:0DB8:0000:CD30:**0000:0000:0000:0000**/60

2001:0DB8::**CD30:0:0:0:0**/60

2001:0DB8:0:CD30::**:**/60

# Запис на IPv6 адреси и префикси

**НЕПРАВИЛНО** представяне на 60-bit префикс:

2001:0DB8:0:**CD3**/60 в 16-bit число от адреса се пропускат **водещи (незначещи) нули**, но не и **крайни (значещи) нули**

2001:0DB8::**CD30**/60 адресът вляво от "/" по-скоро е:

2001:0DB8:**0000:0000:0000:0000:0000:0000**:CD30

2001:0DB8::**CD3**/60 адресът вляво от "/" по-скоро е:

2001:0DB8:0000:0000:0000:0000:0000:0**CD3**

# Типове IPv6 адреси

Тип	Двоичен формат	IPv6 означение
Unspecified (неопределен)	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	Всиички останали	<b>Anycast</b> са част от unicast пространството

# Няма Broadcast адреси

**Broadcast** адреси **не са дефинирани** в IPv6.

Multicast адресирането в IPv6 поема функциите и на broadcast.

Разпределение на IPv6 адресното пространство:

<http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml>

Алокации между RIRs:

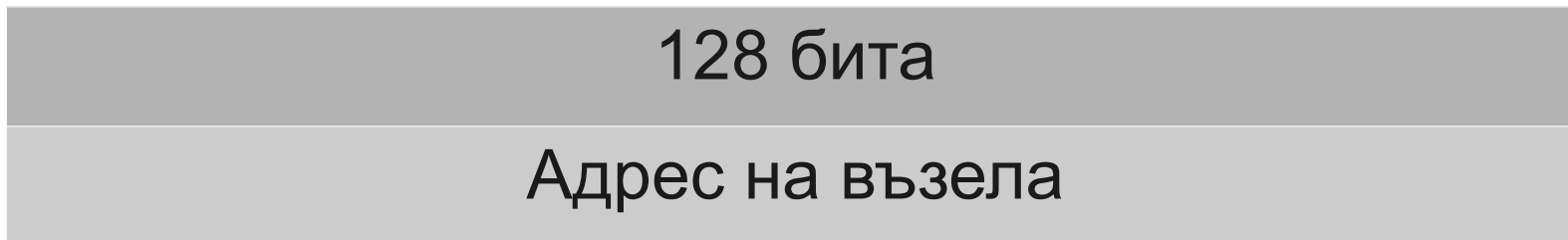
<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml>



# Unicast адреси

IPv6 unicast адресите, подобно на IPv4 CIDR, имат префикси с произволни дължини.

Един възел в IPv6 мрежа може да няма знание за вътрешната структура на адреса:



# Unicast адреси

Възелът може да е наясно с дължината на префикса  $n$ :

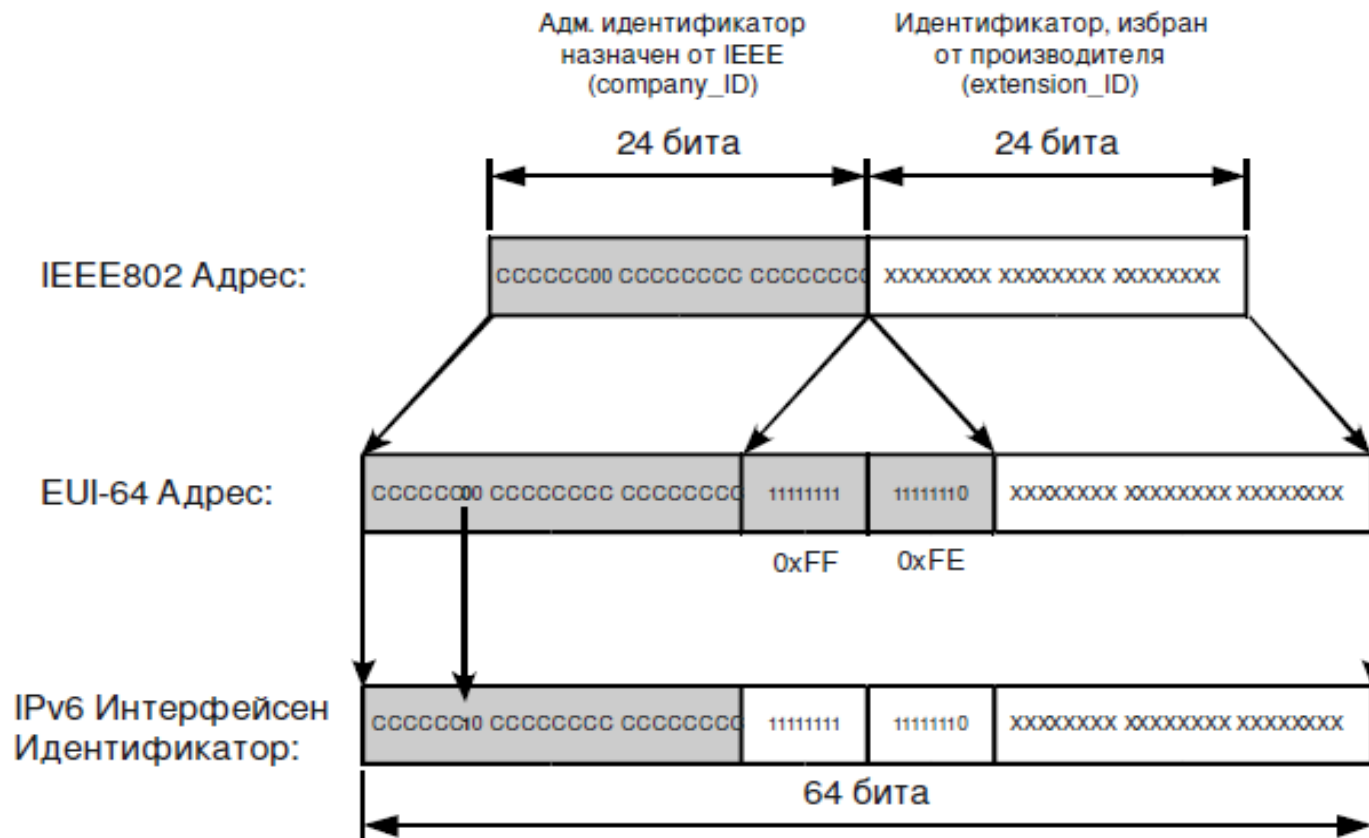
n bits	128-n bits
subnet prefix	interface ID

# Interface ID

Идентификаторите на интерфейси в IPv6 трябва да са **уникални** в рамките на subnet prefix.

Всички unicast адреси **с изключение** на започващите с двоична стойност **000** Interface IDs са 64 бита в Modified **EUI-64** формат.

# EUI-64 формат



# Link-local адреси

10 bits	54 bits	64 bits
1111 1110 10	0000.....0000	Interface ID
FE80/10	0000.....0000	Interface ID

Отнасят се до конкретна LAN или мрежов канал.

Всеки IPv6 интерфейс в LAN трябва да има такъв адрес.

# Пример

```
[root@shuttle ~]# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc  
noqueue
```

```
    link/loopback 00:00:00:00:00:00 brd  
00:00:00:00:00:00
```

```
    inet 127.0.0.1/8 scope host lo
```

```
    inet6 ::1/128 scope host
```

```
        valid_lft forever preferred_lft  
forever
```

# Пример (прод.)

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc pfifo_fast qlen 1000

    link/ether 00:16:17:b2:0e:96 brd
ff:ff:ff:ff:ff:ff

    inet 62.44.109.11/26 brd 62.44.109.63 scope
global eth0

    inet6 2a01:288:8003:0:216:17ff:feb2:e96/64
scope global dynamic

        valid_lft 2591981sec preferred_lft
604781sec

    inet6 fe80::216:17ff:feb2:e96/64 scope link

        valid_lft forever preferred_lft forever
```

# Пример (статично зададен адрес)

```
[root@shuttle ~]# ip a
```

```
...
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu  
1500 qdisc pfifo_fast qlen 1000
```

```
...
```

```
    inet6 2001:67c:20d0:10::11/64 scope global  
        valid_lft forever preferred_lft forever  
    inet6 fe80::216:17ff:feb2:e96/64 scope link  
        valid_lft forever preferred_lft forever
```



# Вградени IPv4 Unicast

Подпомагат плавния преход от IPv4 към IPv6, осигуряват обратна съвместимост

Използват се само **IPv4-mapped IPv6 адреси**.

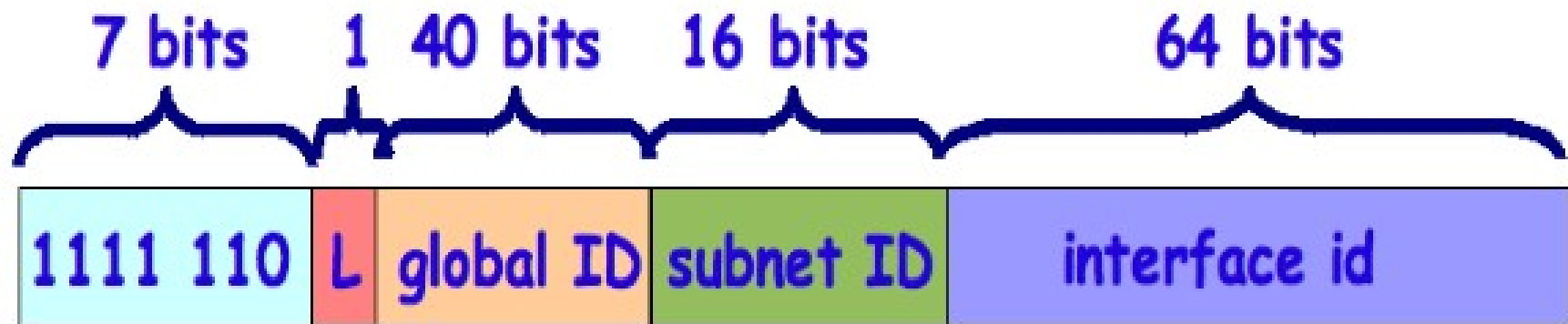
IPv4 адресите се вграждат в IPv6. Представя адреса на IPv4 възлите като IPv6 адрес.

Така IPv6 възел ще изпраща пакети към IPv4 възел.

# IPv4-mapped IPv6 адреси

80 bits	16 bits	32 bits
0000.....0000	FFFF	IPv4 address

# Unique Local IPv6 Unicast Addresses (ULA)



# ULA (прод.)

Prefix **FC00::/7**

L **=1** ако префиксът е локално присвоен.  
**=0** предстои да се дефинира.

Global ID **40-bit** глобално уникален префикс,  
генериран псевдо случайно.

Subnet ID **16-bit**, идентифицира подмрежата в  
сайта.

Interface ID **64-bit** (генериран от MAC адреса)

# ULA (прод.)

ULA адресите (**RFC 4193**) - Unique Local IPv6 Unicast Addresses (уникални локални адреси) или

локални IPv6 адреси (**Local IPv6 addresses**)

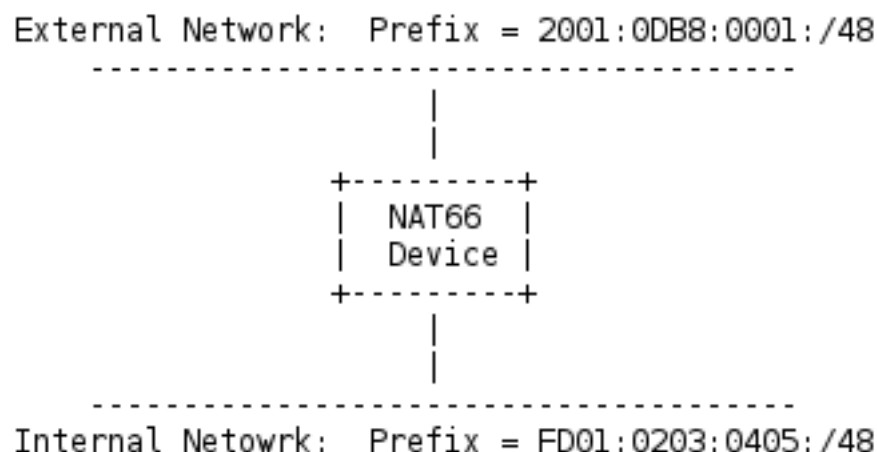
Не се маршрутизират в глобалния Internet.

Маршрутизират се само в рамките на сайт или група от сайтове.

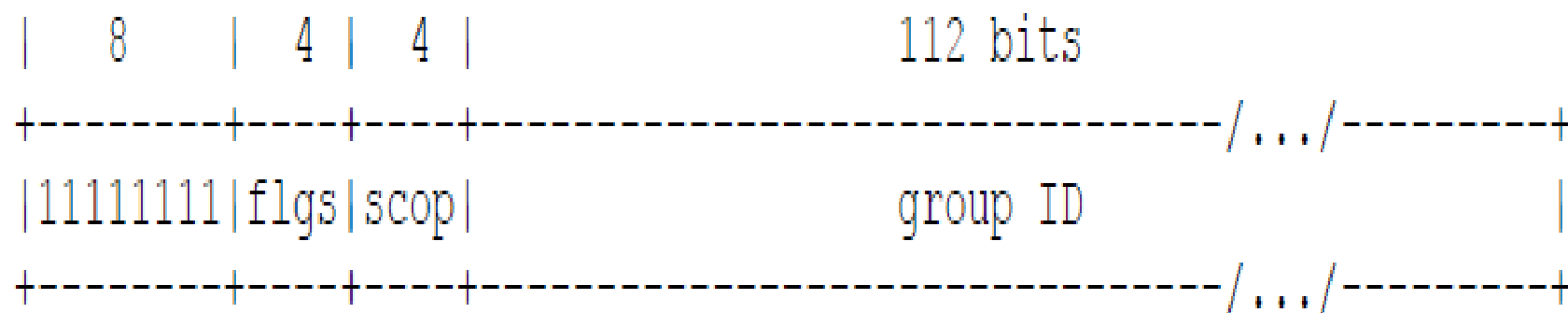
# ULA (прод.)

Прилагат се при ограничаване на достъп (вътре в сайт или автономна система) до защитени ресурси и

**NAT66:**



# IPv6 Multicast. Формат.



Въвеждането на обseg (**scope**) в IPv6 multicast ограничава разпространението на пакети само до необходимите части от мрежата: интерфейси, мрежови сегменти и префикси.

# IPv6 Multicast. Scope.

Value	Scope
1	Interface Local
2	Link Local
4	Admin. Local
5	Site Local
8	Organization Local
E	Global

Някои **добре известни** multicast Group IDs се дефинират с различни обсежи (scopes).

Например адресът на **"All NTP [Network Time Protocol] Servers"**:



# All NTP Servers

FF02::101	All NTP Servers	Link Local
FF04::101	All NTP Servers	Admin Local
FF05::101	All NTP Servers	Site Local
FF08::101	All NTP Servers	Organization Local
FF0E::101	All NTP Servers	Global

RFC 2375 съдържа списък с добре известни (well-known) IPv6 multicast адреси, категоризирани по обseg.

Актуален списък с адресите:

<http://www.iana.org/assignments/ipv6-multicast-addresses>

# Защо няма broadcast

FF02::1	All Nodes
FF02::2	All Routers
FF02::1:2	All DHCP Agents

IPv6 няма broadcast адреси, а Solicited Node multicast групи и all routers multicast адреси.

Така по-оптимално се използват мрежовите ресурси.

# ICMPv6 vs. ICMPv4.

Next Header (NH) = 58; (=1 в IPv4)

**Neighbor Discovery (ND)** вместо ARP. Нямаме broadcast трафика на ARP. Локализира link-local съседни. За разлика от IPv4 открива и достижимостта на съседа, локализира link-local рутери, открива дублирани IPv6 адреси.

**Удължен PMTU** (Path MTU). Минималният MTU в IPv4 е 576 байта, в IPv6 - 1280 bytes, препоръчителен минимален MTU е 1500 байта.

# ping6

```
[root@shuttle ~]# ping6 google.com
```

```
PING google.com(2a00:1450:8004::63) 56 data  
bytes
```

```
64 bytes from 2a00:1450:8004::63: icmp_seq=0  
ttl=54 time=47.4 ms
```

```
64 bytes from 2a00:1450:8004::63: icmp_seq=1  
ttl=54 time=46.3 ms
```

```
64 bytes from 2a00:1450:8004::63: icmp_seq=2  
ttl=54 time=46.6 ms
```

```
64 bytes from 2a00:1450:8004::63: icmp_seq=3  
ttl=54 time=46.2 ms
```

# traceroute6

```
[root@shuttle ~]# traceroute6 google.com
```

```
traceroute to google.com (2a00:1450:8004::63), 30 hops  
max, 40 byte packets
```

```
1    * * *
```

```
2    border-lozenets.uni-sofia.bg (2a01:288:8000::a)  
1.962 ms  1.956 ms  1.946 ms
```

```
3    core-su.lines.acad.bg (2001:4b58:acad:252::25)  
69.643 ms  69.663 ms  69.655 ms
```

```
...
```

```
14   2a00:1450:8004::63 (2a00:1450:8004::63)  46.596 ms  
47.212 ms  46.264 ms
```

# Neighbor Discovery (ND)

ND (RFC 4861) е процес, чрез който IPv6 възел може да научи адреси на 2 слой на интерфейси, свързани към локалния мрежов сегмент.

ND замества ARP в IPv4.

Работи в комбинация с ICMP Router Discovery и Redirect.

# Функции на ND

Функциите на ND се осъществяват чрез:

**Router Solicitation (RS)**. При активиране на интерфейс хостът изпраща RSs, със заявка рутерите веднага да генерират **RA**s.

**Router Advertisement (RA)**. Рутерите рекламират присъствието си и някои параметри периодически или веднага след RS. RA съдържа **префикси** на връзката, конфигурации на **адреси**, брой **hop-ве**, **MTU** и др.

# ND. Neighbor Solicitation.

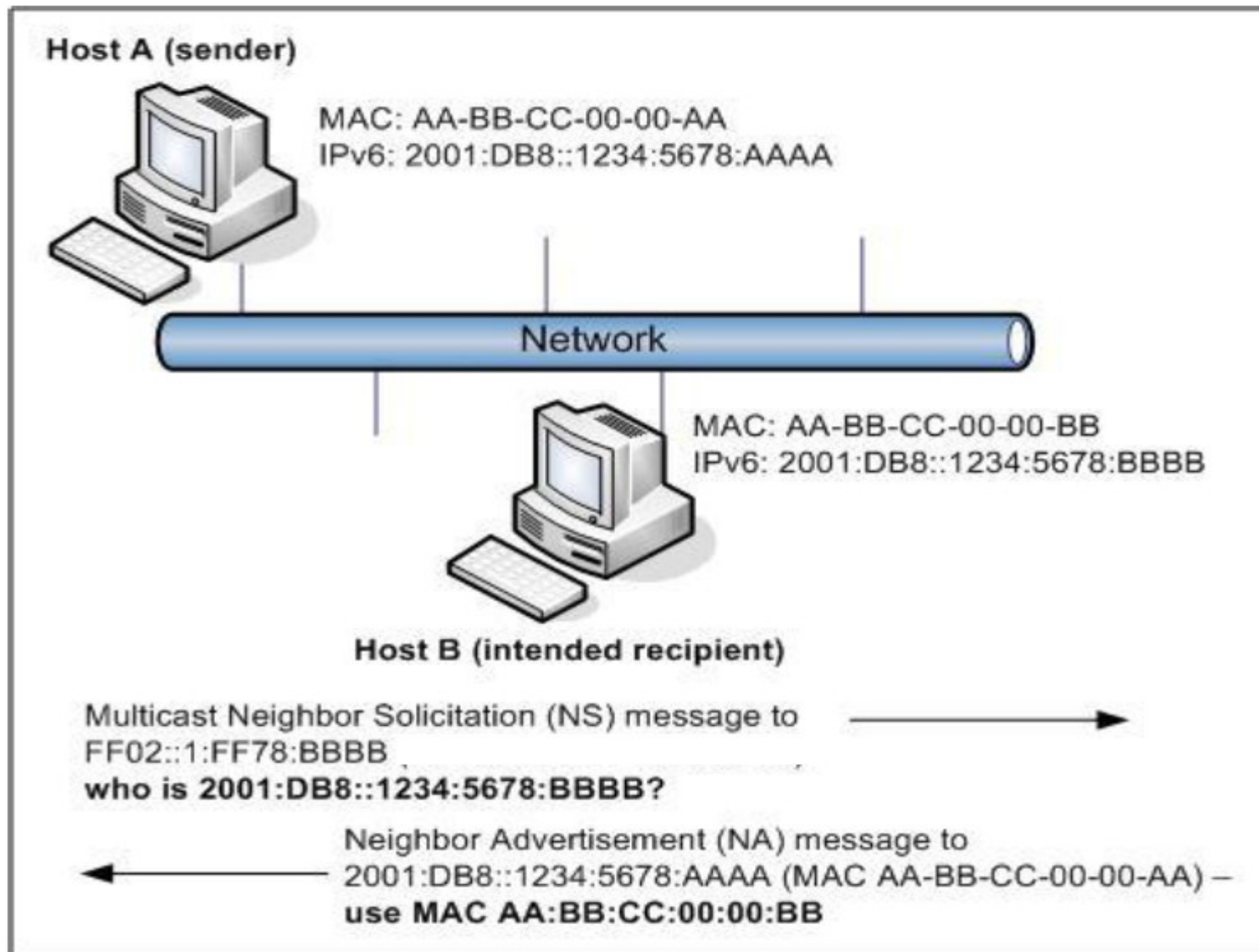
**Neighbor Solicitation (NS).** Възлите изпращат NSs, за да определят адреса на 2 слой на съседа или да се уверят, че съседът е все още достижим. NSs разпознават и дублирани адреси (**Duplicate Address Detection – DAD**).

**Neighbor Advertisement (NA).** Отговор на NS. Възел може да изпраща самостоятелно NAs, за да съобщи за промяна на адрес.

**Redirect Message.** Рутерите информират хостовете за по-добър първи хоп до дестинацията.



# Пример на ND



# Пример на ND

```
[root@shuttle ~]# ip neighbor
```

```
2001:67c:20d0:10::5 dev eth0 lladdr  
00:0d:56:b9:75:6d router STALE
```

```
62.44.109.5 dev eth0 lladdr  
00:0d:56:b9:75:6d DELAY
```

# Autoconfiguration

Дефинира се в RFC 4862.

В IPv6 има и **Stateful** (с определено състояние), и **Stateless** (неопределено) автоконфигуриране на адреси - **SLAAC**.

SLAAC не изисква ръчно конфигуриране на хостове, минимално на рутери, сървъри не са необходими.

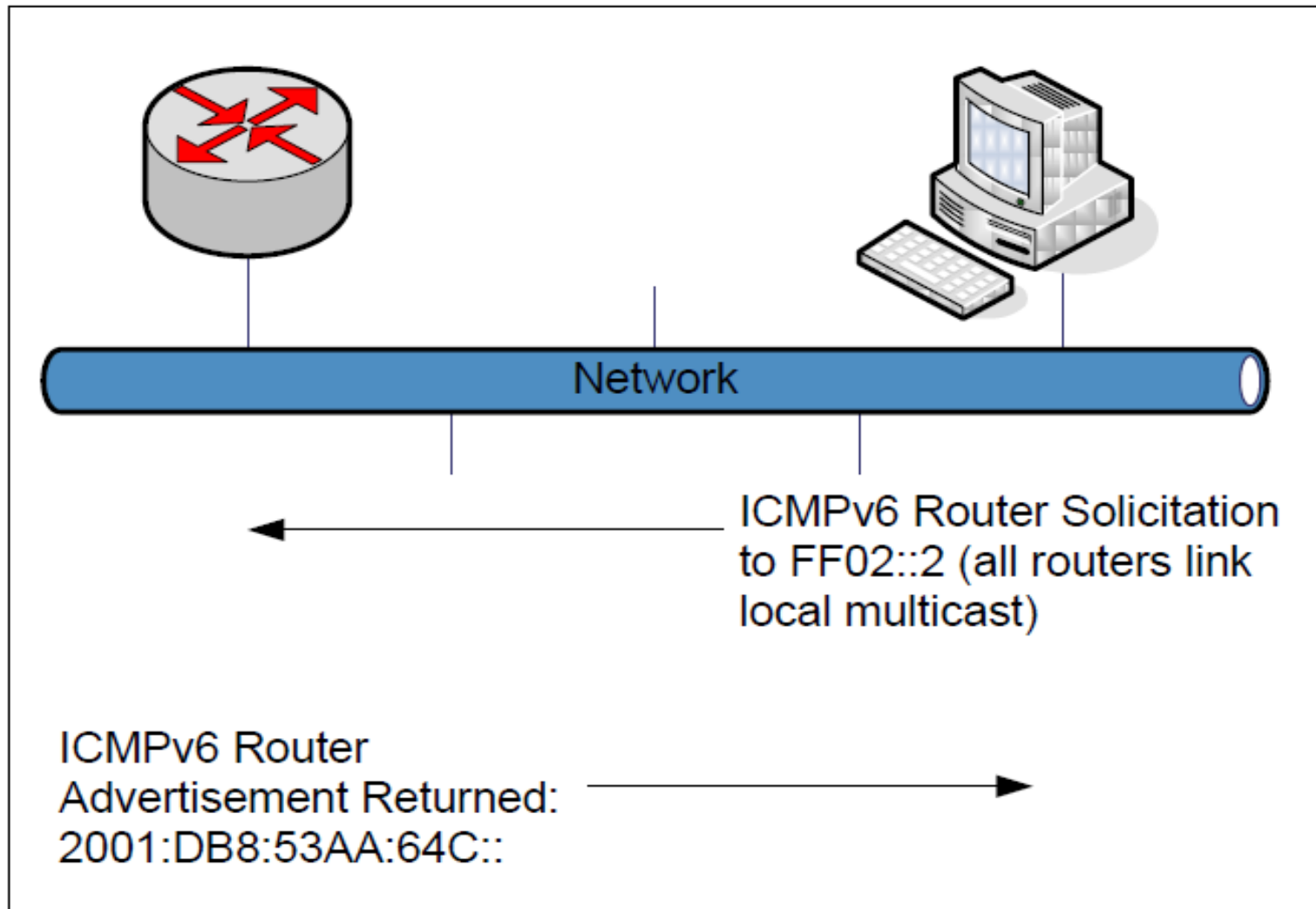
**Рутерите** рекламират мрежов **префикс**, а **хостът** генерира **interface ID**.

# Autoconfiguration

Ако в мрежовия сегмент няма рутер, хостът генерира само адрес на 2 слой, с който може да комуникира само в мрежовия си сегмент.

**Stateful** автоконфигуриране в **IPv4** е **DHCP**. За IPv6 версията е DHCPv6, който е много различен от DHCPv4.

# Autoconfiguration



# Router Advertisement Daemon (radvd)

```
[root@loz-gw ~]# less /etc/radvd.conf
```

```
...
```

```
interface eth1.109
```

```
{
```

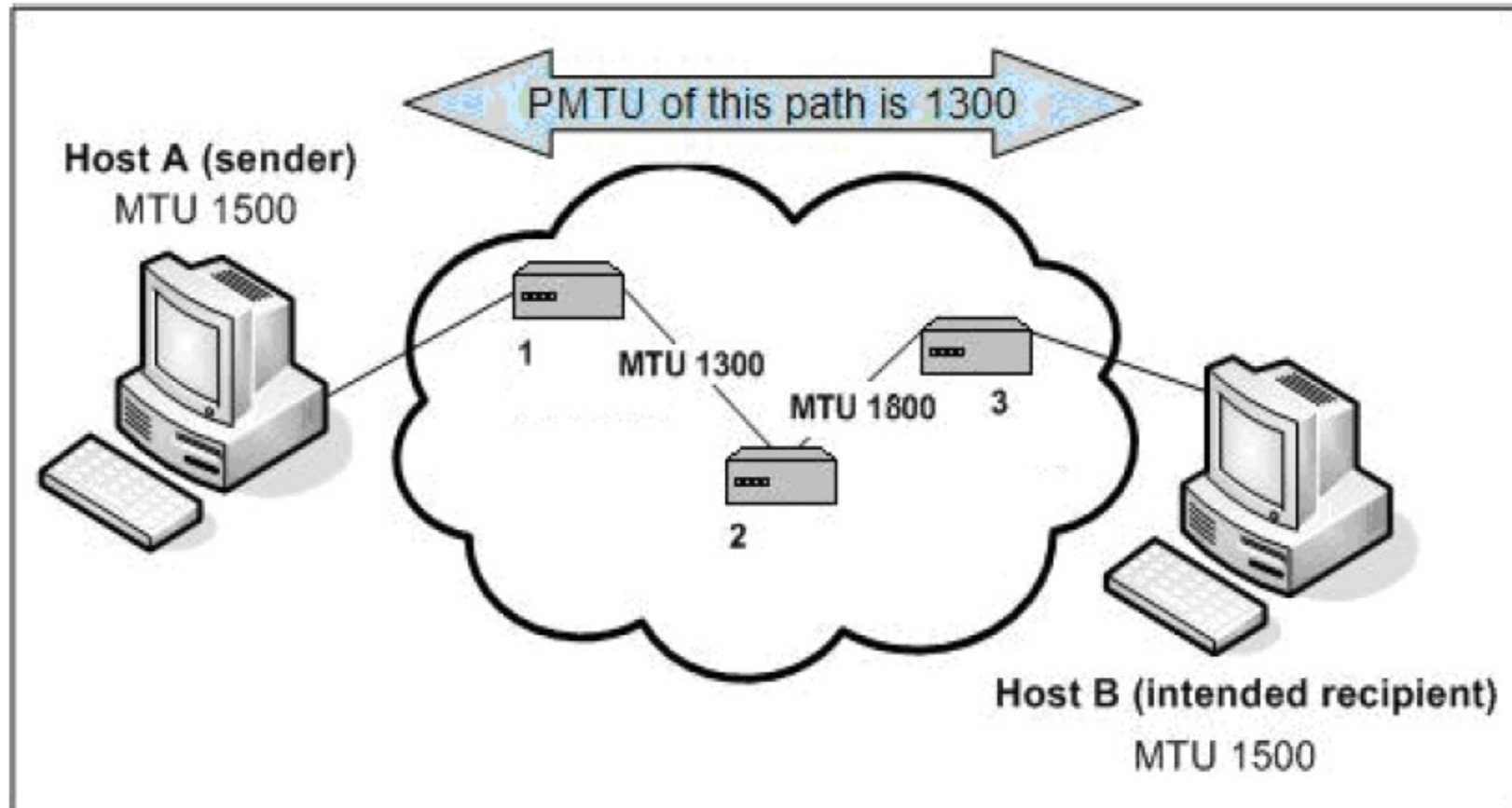
```
    AdvSendAdvert on;
```

```
    MinRtrAdvInterval 30;
```

```
    MaxRtrAdvInterval 100;
```

```
    prefix 2001:67c:20d0:10::/64
```

# Path Maximum Transmission Unit (PMTU) Discovery



# Механизми за преход от IPv4 към IPv6

IPv6 не е обратно съвместим с IPv4.

Механизмите за преход трябва да осигуряват взаимодействието.

Определят се и от вида на хостовете:

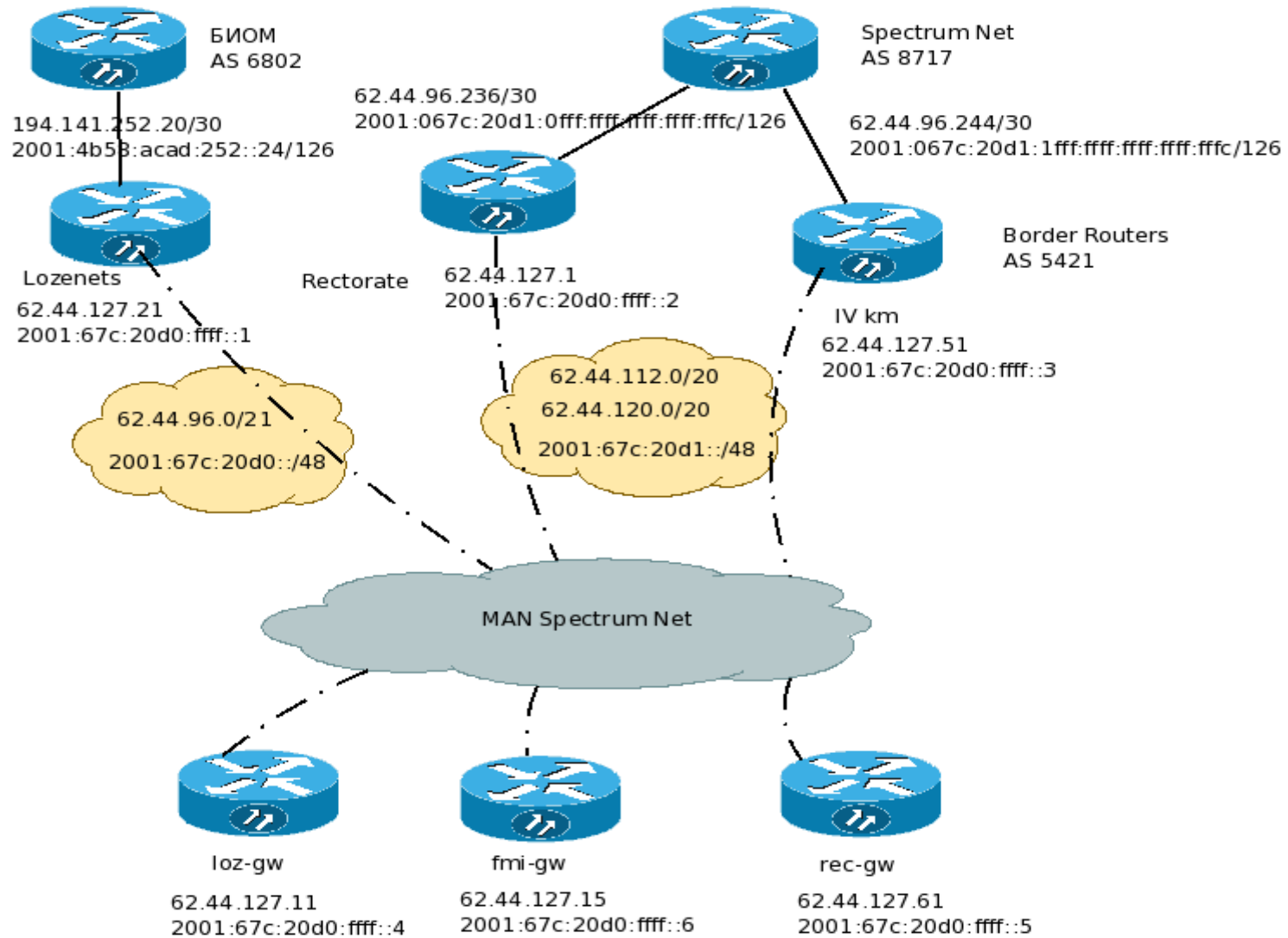
- само IPv4;
- само IPv6 и
- dual stack IPv4/IPv6.



# Видове механизми

- Dual stack
- Tunneling
- Translation (NAT)

# Dual Stack IPv4/IPv6



# Dual Stack IPv4/IPv6

За потребителите е прозрачно дали за дадена услуга ползват IPv4 или IPv6.

Постига се с оборудване, което поддържа и двата протокола:

- втора ръка сървъри за маршрутизатори, работещи под Linux с пакет Zebra Quagga;
- DNS е един и същ за IPv4 и IPv6;
- присвояване на адреси по IPv4 – статично или DHCP, IPv6 – автоматично;
- Web (Apache) – “слуша” по IPv4 и IPv6.

# Сървър като маршрутизатор



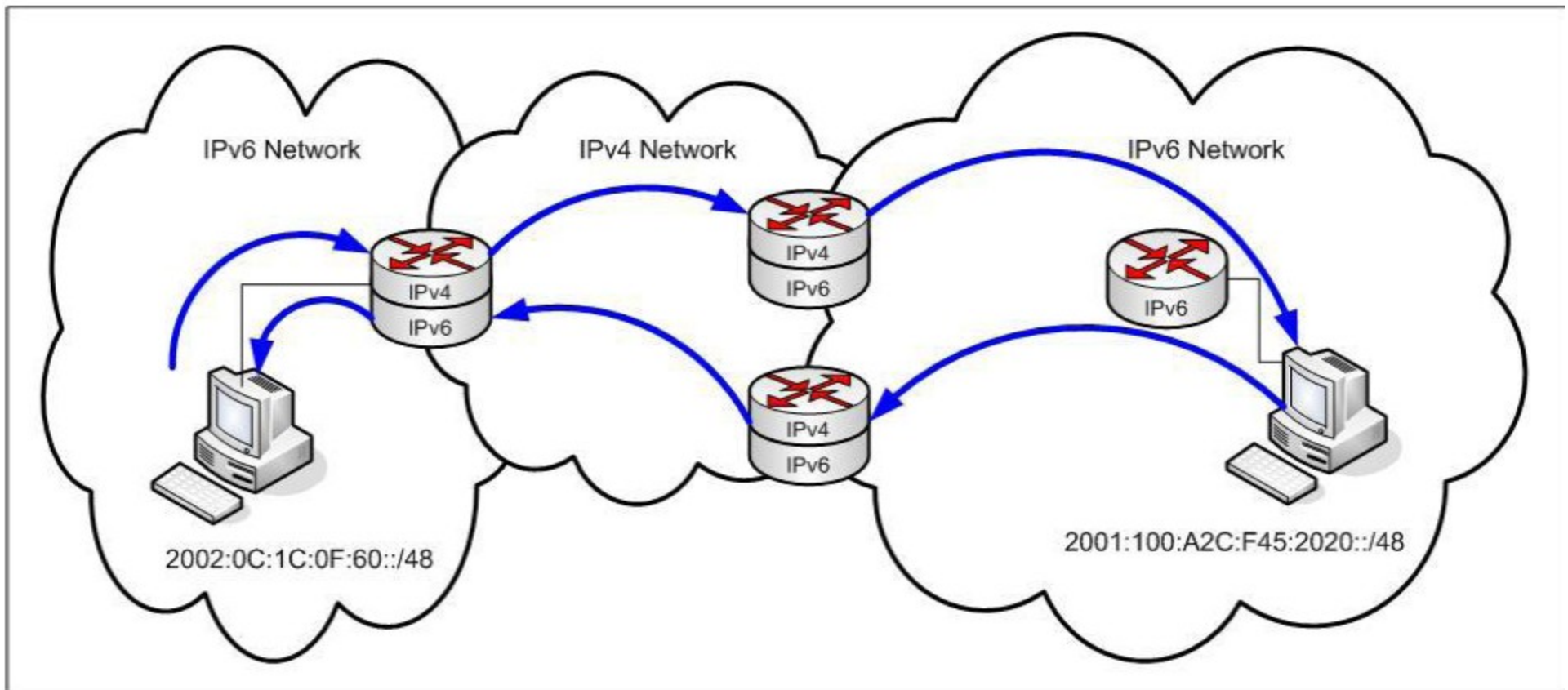
# Dual stack

Работни станции:

- Linux – IPv4/IPv6 – автоматично;
- Windows7/Vista – IPv4/IPv6 – автоматично;
- Windows XP – IPv6 се стартира ръчно.



# Tunneling IPv6 over IPv4



# NAT64

