



# IPv6技术概览

— mospan 2019-08-01

# 目录

- 1. 为什么要用IPv6
- 2. IPv6的优势
- 3. IPv6技术
- 4. IPv6现状
- 5. IPv6过渡技术
- 6. IPv6展望

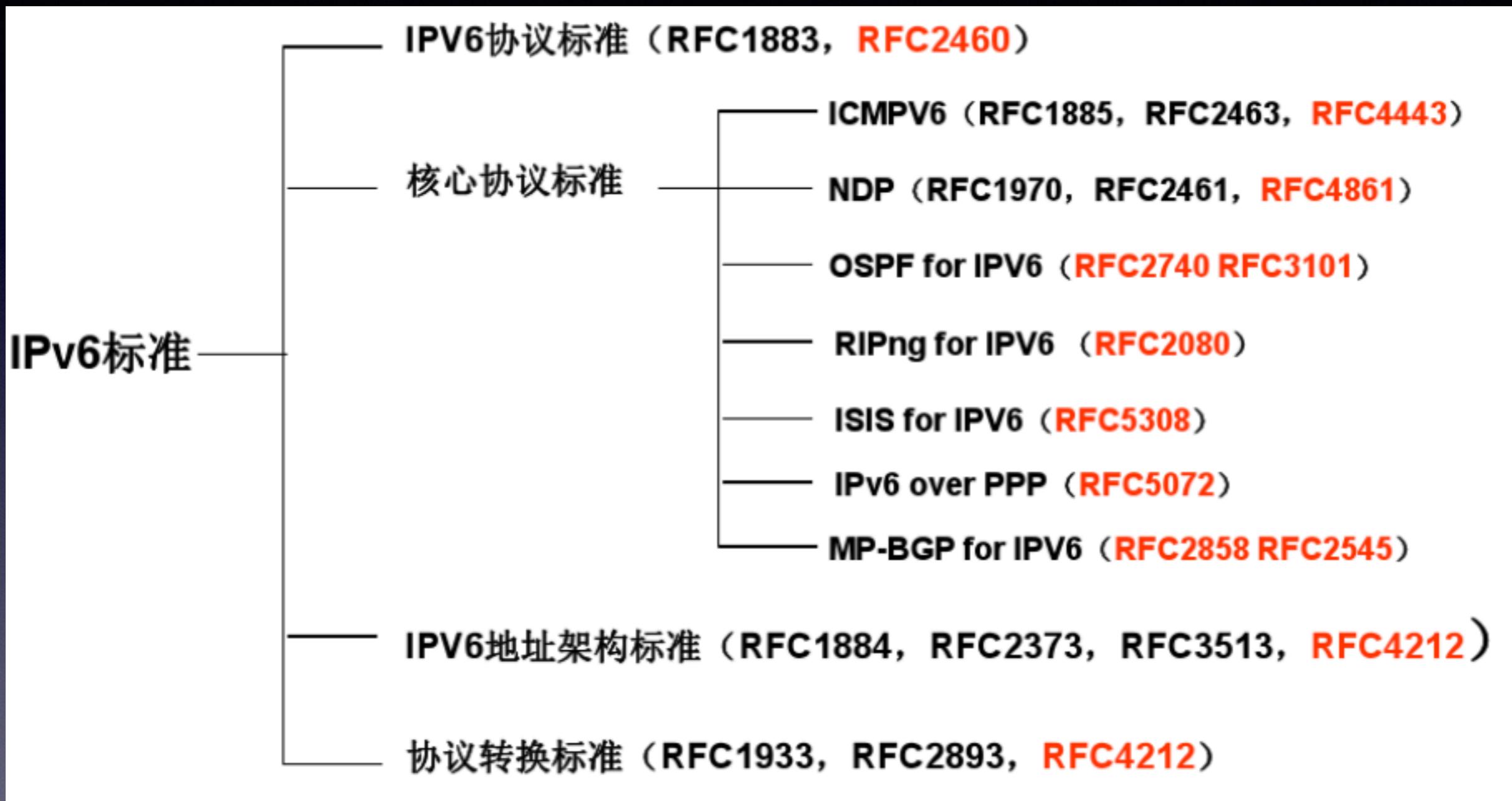
## 1. 为什么要用IPv6

- IPv4地址匮乏
- NAT非根本解决办法，破坏了端到端模型
- 新技术对IP协议催生了更多的需求
- 政策驱动：解读

## 2. IPv6优势

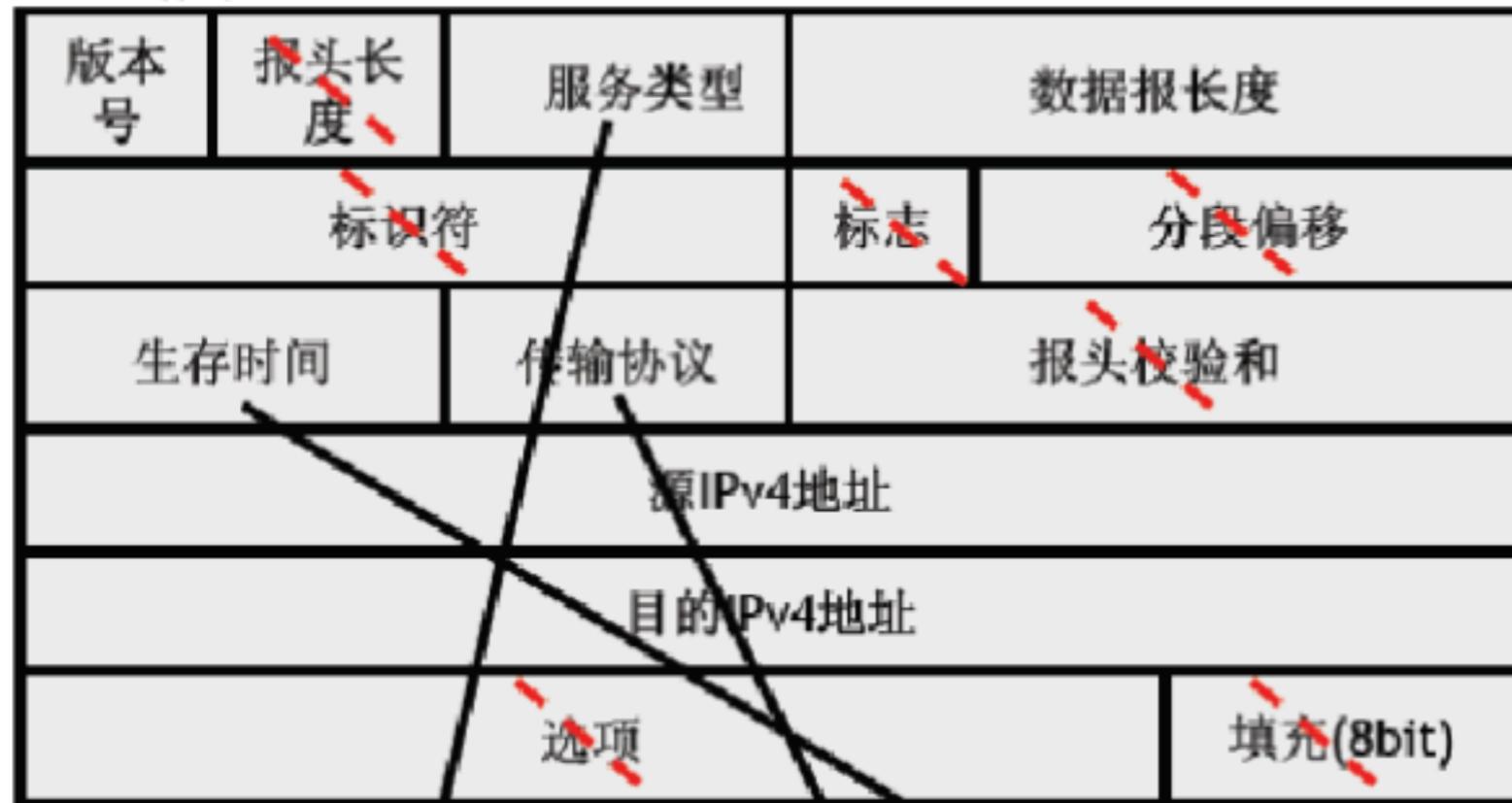
- 近乎无限的地址空间
- 更简洁的报文头部
- 内置的安全性
- 更好的QoS支持
- 更好的移动性
- 编址层次等级

### 3. IPv6技术—标准(RFC)



### 3. IPv6技术—报文格式

IPv4报头



IPv6报头



- 将所有可选字段移出IPv6报头，或删除或置于扩展报头中
- 服务类型、传输协议和生存时间3个域的名称或部分功能被改变
- 新增加了1个域，即流标签
- 固定的基本报头长度（40字节），故不需要消耗过多的内存容量

### 3. IPv6技术—地址

## 巨大的地址空间

- ◆ IPv4

- 地址长度：32位

- 地址空间： $2^{32}$  约42亿(世界上平均3个人有2个IP地址)

- ◆ IPv6

- 地址长度：128位

- 地址空间： $2^{128}$  约 $3.4 \times 10^{38}$ 个(地球上每一粒沙子都有一个IP地址)

### 3. IPv6技术－地址表示

- 非压缩: 2001:0da8:0207:0000:0000:0000:8207
- 压缩前导0: 2001:0da8:0207:0:0:0:8207
- 压缩: 2001:0d08:0207::8207      注: 只能有一个::
- 内嵌IPv4地址: ::192.168.1.1

### 3. IPv6技术 – 地址表示

□ IPv6地址 = 前缀 + 接口标识

- 前缀：相当于IPv4地址中的网络ID
- 接口标识：相当于IPv4地址中的主机ID
- 前缀长度用“/xx”来表示

2001: da8: 207: : 8207/64



### 3. IPv6技术 – 地址分类

类型	用途	ipv6地址	与ipv4对比
单播 (Unicast)	特殊地址 - 未指定地址 (Unspecified address)	::/128	0.0.0.0/32
	特殊地址 - 环回地址 (Loopback Address)	::1/128	127.0.0.1/8
	链路本地 (Link-local)	fe80::/10	169.254.0.0/16
	全球单播 (Global unicast)	2000::/3	公网地址
	唯一本地 (Unique-local)	fc00::/7, 其中 - fc00::/8 【未定义】 - fd00::/8 【可使用】	10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
	站点本地 (Site-local) 【已弃用】	fec0::/10	
	过渡 - ipv4兼容地址 (IPv4-mapped) 【已弃用】	::192.168.1.1	
	过渡 - ipv4映射地址 (IPv4-compatible)	::ffff:192.168.1.1	
多播 (Multicast)	地址范围	ff00::/8	224.0.0.0/4

### 3. IPv6技术－地址配置(手工)

添加IPV6地址

```
ip -6 addr add <ipv6address>/<prefixlength> dev <interface>
```

```
ip -6 addr add 2001:0db8:0:f101::1/64 dev eth0
```

```
ifconfig <interface> inet6 add <ipv6address>/<prefixlength>
```

```
ifconfig eth0 inet6 add 2001:0db8:0:f101::1/64
```

添加默认路由

```
ip -6 route add <ipv6network>/<prefixlength> via <ipv6address>
```

```
ip -6 route add default via 2001:0db8:0:f101::1
```

```
route -A inet6 add <ipv6network>/<prefixlength> gw
```

```
route -A inet6 add default gw 2001:0db8:0:f101::1
```

查看路由

```
ip -6 route show
```

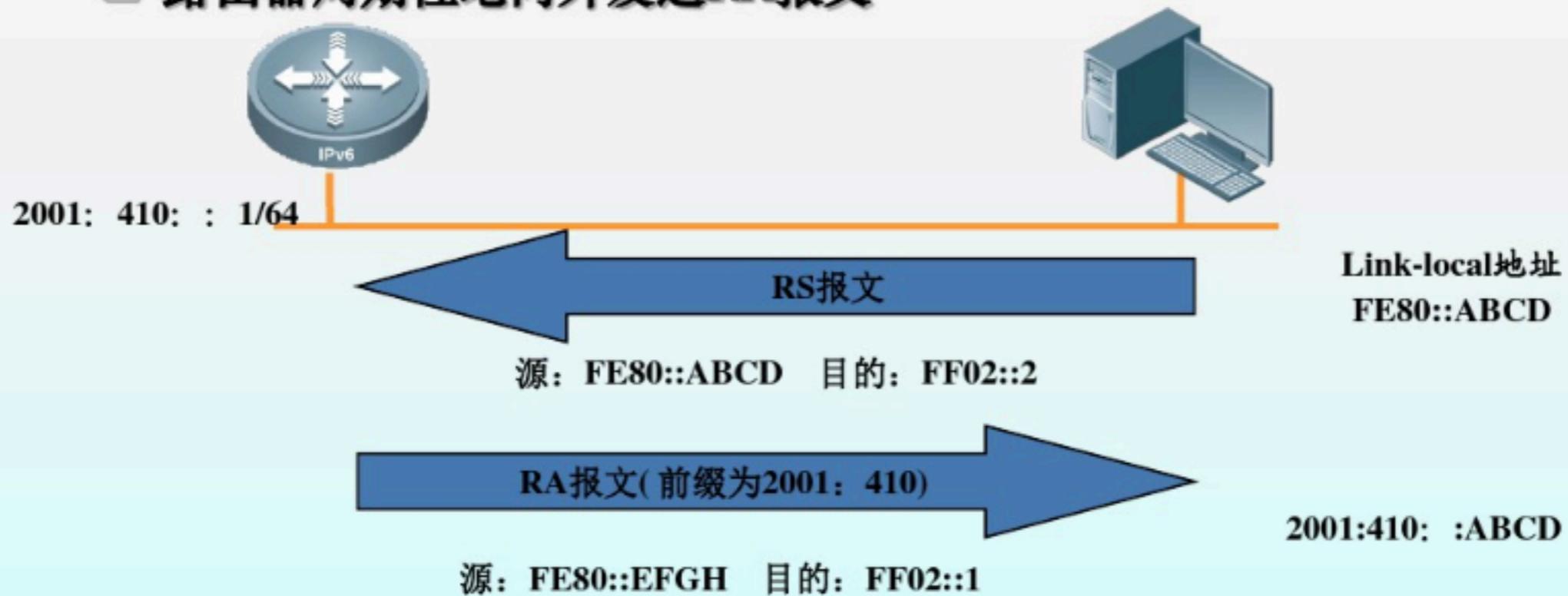
```
route -A 'inet6'
```

```
route -6
```

### 3. IPv6技术—地址配置(自动)

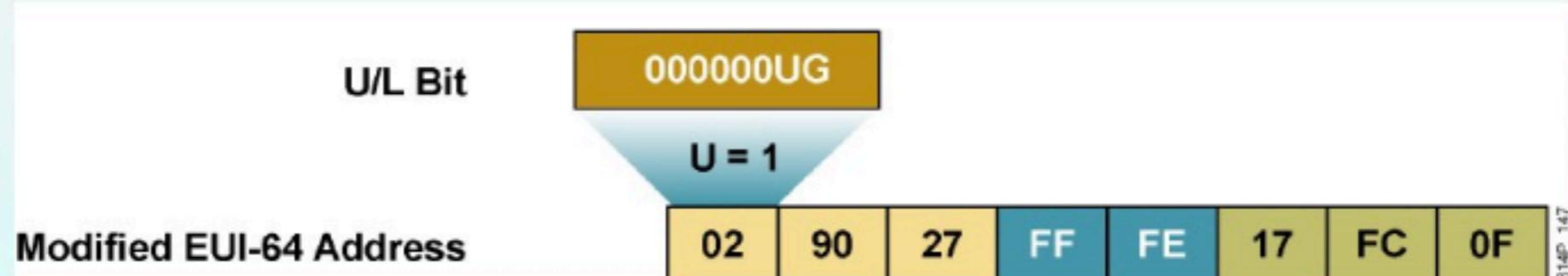
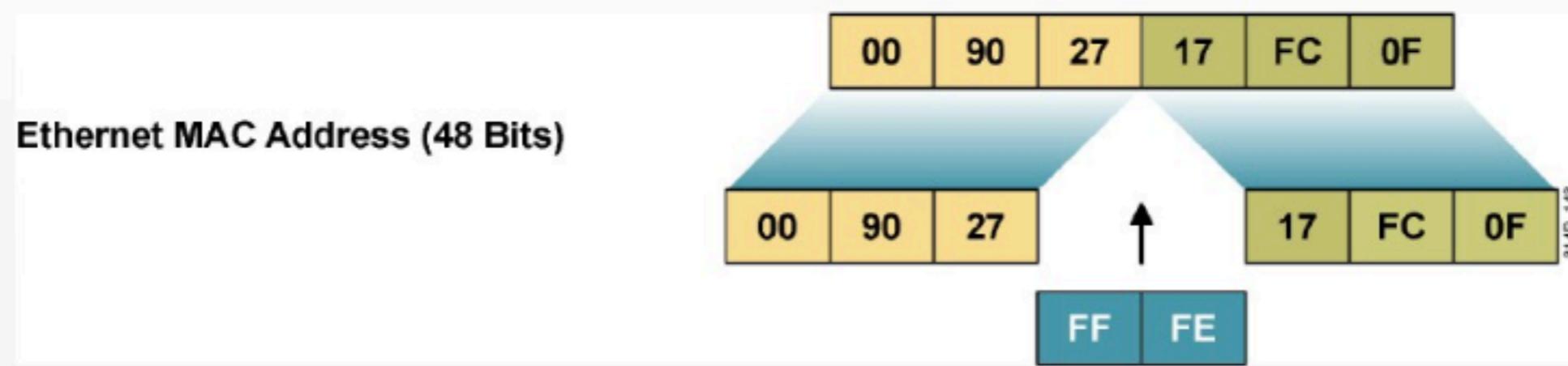
#### 无状态自动配置—前缀获得

- 主机发送Router Solicitation报文
- 路由器回应Router Advertisement报文
- 主机获得前缀及其它参数
- 路由器周期性地向外发送RA报文



### 3. IPv6技术—地址配置(自动)

#### 无状态地址自动配置——接口ID生成



### 3. IPv6技术—常用命令

```
# 查看ip  
ip -6 addr  
  
# 查看路由  
ip -6 route  
  
# 查看多播地址  
ip -6 maddr  
netstat -gn  
  
# 查dns AAAA记录 (走ipv4线路)  
dig -t AAAA fzxiaomange.com  
nslookup -query=AAAA fzxiaomange.com  
  
# 查dns AAAA记录 (走ipv6线路)  
dig @2620:0:ccc::2 -6 -t AAAA fzxiaomange.com  
nslookup -query=AAAA fzxiaomange.com 2620:0:ccc::2  
  
# 查dns AAAA记录 (走ipv6线路的最纯粹、最完整、最靠谱方法)  
dig @2620:0:ccc::2 -t AAAA -6 fzxiaomange.com +trace
```

```
# curl  
curl -6 https://fzxicmange.com  
  
# ping  
ping6 fzxicmange.com  
  
# trace route  
mtr -6 fzxicmange.com  
traceroute6 fzxicmange.com  
tracert6 fzxicmange.com  
tracepath6 fzxicmange.com  
  
# 邻居地址解析, 类似ipv4的arping  
ndisc6 -n 2002::102 eth0  
  
# iptables查看  
ip6tables-save  
  
# tcpdump  
tcpdump -nnn -i eth0 ip6
```

### 3. IPv6技术－在线工具

- 公共dns列表：

<http://dns.icoa.cn/ipv6/>

- 查询网站是否提供IPv6：

<https://ready.chair6.net/>

<http://ip6tools.com/#checkWebServer>

- 查询我自己是否拥有公网IPv6地址：

<http://www.test-ipv6.com>

<http://ip6tools.com/#WhatisMyIPv6Address>

# 3. IPv6技术—协议栈(linux)

```
# mosp @ mospdeMacBook-Pro in ~/osrc/kernel/linux-3.10.1/net/ipv6 [9:49:21]      |IPv6
$ ls
Kconfig          ah6.c          exthdrs_offload.c    ip6_fib.c        ip6_offload.h      mcast.c        proc.c          syncookies.c    udp_impl.h       xfrm6_mode_transport.c
Makefile         anycast.c      fib6_rules.c       ip6_flowlabel.c   ip6_output.c      mip6.c         protocol.c     sysctl_net_ipv6.c  udp_offload.c   xfrm6_mode_tunnel.c
addrconf.c       datagram.c     icmp.c           ip6_gre.c       ip6_tunnel.c     ndisc.c        raw.c          tcp_ipv6.c      udplite.c       xfrm6_output.c
addrconf_core.c esp6.c          inet6_connection_sock.c ip6_icmp.c     ip6mr.c        netfilter      reassembly.c   tcpv6_offload.c xfrm6_input.c   xfrm6_policy.c
addrlabel.c      exthdrs.c     inet6_hashtables.c ip6_input.c     ipcomp6.c      netfilter_c   route.c        tunnel6.c      xfrm6_beet.c   xfrm6_state.c
af_inet6.c       exthdrs_core.c ip6_checksum.c     ip6_offload.c   ipv6_sockglue.c output_core.c  sit.c          udp.c          xfrm6_beet.c   xfrm6_state.c
                                         ip6_checksum.c

# mosp @ mospdeMacBook-Pro in ~/osrc/kernel/linux-3.10.1/net/ipv6 [9:49:28]      |IPv4
$ ls .../ipv4
Kconfig          esp4.c          igmp.c           ip_forward.c    ip_vti.c        proc.c          tcp_cong.c     tcp_input.c    udp.c          xfrm4_output.c
Makefile         fib_frontend.c  inet_connection_sock.c ip_fragment.c  ipcomp.c       protocol.c    tcp_cubic.c   tcp_ipv4.c    udp_diag.c    xfrm4_policy.c
af_inet.c        fib_lookup.h    inet_diag.c       ip_gre.c       ipconfig.c    raw.c          tcp_diag.c    tcp_lp.c      udp_impl.h    xfrm4_state.c
ah4.c            fib_rules.c    inet_fragment.c  ip_input.c    ipip.c        route.c       tcp_fastopen.c  tcp_vegas.c  udp_implement.c
arp.c            fib_semantics.c  inet_hashtables.c ip_options.c  ipmr.c        syncookies.c  tcp_memcontrol.c  tcp_vegas.h  udplite.c    xfrm4_tunnel.c
cipso_ipv4.c    fib_trie.c     inet_lro.c       ip_output.c   netfilter      sysctl_net_ipv4.c  tcp_metrics.c  tcp_veno.c    xfrm4_input.c
datagram.c       gre.c          inet_timewait_sock.c ip_sockglue.c netfilter_c   tcp_htcp.c    tcp_minisocks.c  tcp_westwood.c xfrm4_beet.c
devinet.c       icmp.c          inetpeer.c      ip_tunnel.c   ping.c        tcp.c          tcp_hybla.c   tcp_output.c  tcp_yeah.c    xfrm4_mode_transport.c
                                         inetpeer.c

# mosp @ mospdeMacBook-Pro in ~/osrc/kernel/linux-3.10.1/net/ipv6 [9:49:32]
$ cat * | wc -l
cat: netfilter: Is a directory
44559 |IPv6

# mosp @ mospdeMacBook-Pro in ~/osrc/kernel/linux-3.10.1/net/ipv6 [9:49:41]
$ cat .../ipv4/* | wc -l
cat: ../ipv4/netfilter: Is a directory
71125 |IPv4
```

## 4. IPv6现状

全球IPv6用户数（估计）排名：

排名	国家/地区	IPv6 用户普及率	IPv6 用户数（估计）	互联网用户数
1	印度(IN)	48.77	227271142	466034684
2	美国(US)	38.29	109943488	287146624
3	德国(DE)	42.99	31061760	72260517
4	日本(JP)	23.77	27604647	116138333
5	巴西(BR)	19.02	26430375	138967416
6	英国(GB)	26.04	15961177	61284147
7	法国(FR)	20.53	11526757	56142328
8	加拿大(CA)	19.75	6399932	32412416
9	比利时(BE)	57.61	5827451	10114962
10	马来西亚(MY)	18.16	3938742	21694245
11	越南(VN)	7.72	3834094	49681216
12	澳大利亚(AU)	15.46	3216155	20807427
13	韩国(KR)	6.88	3006751	43691755
14	中国(CN)	0.39	2836765	735768081
15	希腊(GR)	36.22	2619480	7231532

2017

数据来源：<https://labs.apnic.net/dists/v6dcc.html>

全球IPv6用户数普及率排名：

排名	国家/地区	IPv6 用户普及率	IPv6 用户数（估计）	互联网用户数
1	比利时(BE)	57.61%	5827451	10114962
2	印度(IN)	48.77%	227271142	466034684
3	德国(DE)	42.99%	31061760	72260517
4	美国(US)	38.29%	109943488	287146624
5	希腊(GR)	36.22%	2619480	7231532
6	瑞士(CH)	35.16%	2598896	7391076
7	卢森堡(LU)	33.43%	185712	555449
8	英国(GB)	26.04%	15961177	61284147
9	乌拉圭(UY)	25.70%	577342	2246887
10	葡萄牙(PT)	25.49%	1771982	6951757
11	日本(JP)	23.77%	27604647	116138333
12	法国(FR)	20.53%	11526757	56142328
.....				
67	中国(CN)	0.39%	2836765	735768081
68	冰岛(IS)	0.30%	1000	335025
69	阿曼(OM)	0.26%	8564	3296382
.....				

## 5. IPv6过渡技术—方式

- IPv4到IPv6过渡从网络边缘向核心演进

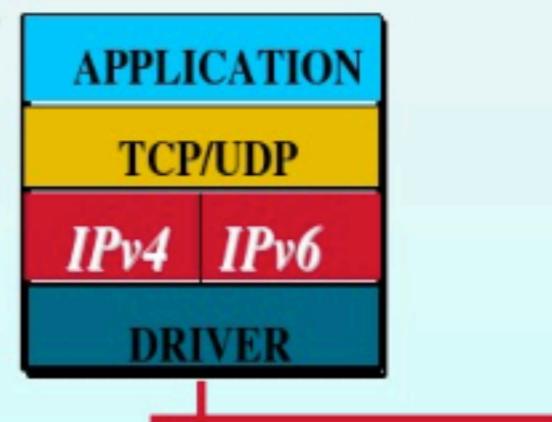


## 5. IPv6过渡技术—简介

- 过渡技术主要解决IPV4和IPV6网络共存状况下的互联互通，主要有三种策略：
  - 双栈协议（Dual Stack）：设备升级到IPv6的同时保留IPv4支持；应用程序可以选择使用IPv6或IPv4；协议允许应用逐渐从IPv4过渡到IPv6
  - 隧道封装（Tunneling）：IPv6报文作为IPv4的载荷，或由MPLS承载在IPv4 Internet海洋中连接多个IPv6孤岛
  - 协议转换：提供IPV4和IPV6的互通动作。如NAT-PT等提供IPv6与IPv4互相访问地技术，适用于IPv6 Internet与IPv4 Internet共存，而两者又需要互相通讯的要求。

## 5. IPv6过渡技术—双栈协议

- 设备升级到IPv6的同时保留IPv4支持，应用程序可以选择使用IPv6或IPv4协议。
- 所有的过渡技术都是基于双协栈实现的
- 优点：互通性好，易于理解，实现简单
- 缺点：
  - 对每个IPv4节点都要升级，成本较大，没有解决IPv4地址紧缺问题；
  - 部署复杂，所有设备要同时配置IPV4和IPV6；
  - 对设备性能要求高，即要运行IPV6，又要运行IPV4。
- **interface ethernet 0**  
**ipv6 address 3ffe:b00:c18:1::3/127**  
**ip address 202.38.16.1 255.255.0.0**



## 5. IPv6过渡技术—隧道封装

- **IPv6 over IPv4手动隧道（配置隧道）**：隧道两端的边界路由器上通过静态指定隧道的源IPv4地址和目的IPv4地址而创建的隧道。
- **IPv6 over IPv4 GRE隧道（简称GRE隧道）**：使用IPv4的GRE隧道上承载IPv6流而实现IPV6互通的隧道。
- **IPv6 over IPv4自动隧道（简称自动隧道）**：使用兼容IPv4的IPv6地址（使用兼容地址）。
- **6to4隧道**：每个站点使用一个可聚合的全球单播IPV6地址前缀，并且至少使用一个全球单播IPV4地址。也是一种自动隧道。
- **ISATAP隧道**：站点间自动隧道寻址协议（Intrasite Automatic Tunnel Addressing Protocol）。
- **6PE**：使用已有MPLS网络承载IPV6。

## 5. IPv6过渡技术—隧道封装

- **NAT-PT：**静态模式提供一对一的IPV6地址和IPV4地址的映射。
- **动态NAT-PT：**动态的提供一对一的映射，但使用一个IPV4地址池，类似IPV4 NAT的公网地址池。
- **NAPT-PT：**网络地址端口转换协议转换。提供多个IPV6地址到一个IPV4地址多对一动态映射。
- **NAT-PT DNS ALG：**
- **TCP-UDP中继；Bump in the Stack；DSTM；基于SOCLS的IPv6/IPv4网关**

## 6. IPv6展望



*Thanks!*

