HCRSE103-IPv6 基础

IPv6 知识点：

IPv6 地址长度，报文组成，基本报头分段和长度，IPv6 地址分类，单播地址分类，EUI-64 原理，ICMPv6 协议，NDP 邻居发现协议，IPv6 过渡技术

EUI-64 (64-bit Extended Unique Identifier) 64 位扩展唯一标识符

IPv6 地址包括 128 比特，由冒号分隔的 32 位十六进制数表示

IPv6 特点：

1 地址空间巨大

地址空间，IPv6 地址采用 128 比特标识。128 位的地址结构使 IPv6 理论上可以拥有（43 亿×43 亿×43 亿×43 亿）个地址。近乎无限的地址空间是 IPv6 的最大优势。

2　精简报文结构

报文结构，IPv6 使用了新的协议头格式，也就是说 IPv6 数据包有全新的报文头，而并不是仅仅简单地将 IPv4 报文头中的地址部分增加到 128bits 而已。在 IPv6 中，报文头包括固定头部和扩展头部，一些非根本性的和可选择的字段被移到了 IPv6 协议头之后的扩展协议头中。这使得网络中的中间路由器在处理 IPv6 协议头时，有更高的效率。

3 实现自动配置和重新编址

实现自动配置和重新编址，IPv6 协议内置支持通过地址自动配置方式使主机自动发现网络并获取 IPv6 地址，大大提高了内部网络的可管理性。

4 支持层次化网络编址

支持层次化网络结构，巨大的地址空间使得 IPv6 可以方便的

进行层次化网络部署。层次化的网络结构可以方便的进行路由聚合，提高了路由转发效率。

5 支持端对端安全

支持端对端安全，IPv6 中，网络层支持 IPSec 的认证和加密，支持端到端的安全。

6 更好的支持 QoS

更好的支持 QoS，IPv6 在包头中新定义了一个叫做流标签的特殊字段。IPv6 的流标签字段使得网络中的路由器可以对属于一个流的数据包进行识别并提供特殊处理。用这个标签，路由器可以不打开传送的内层数据包就可以识别流，这就使得即使数据包有效载荷已经进行了加密，仍然可以实现对 QoS 的支持。

7 支持移动特性

支持移动特性，由于采用了 Routing header 和 Destination option header 等扩展报头，使得 IPv6 提供了内置的移动性。

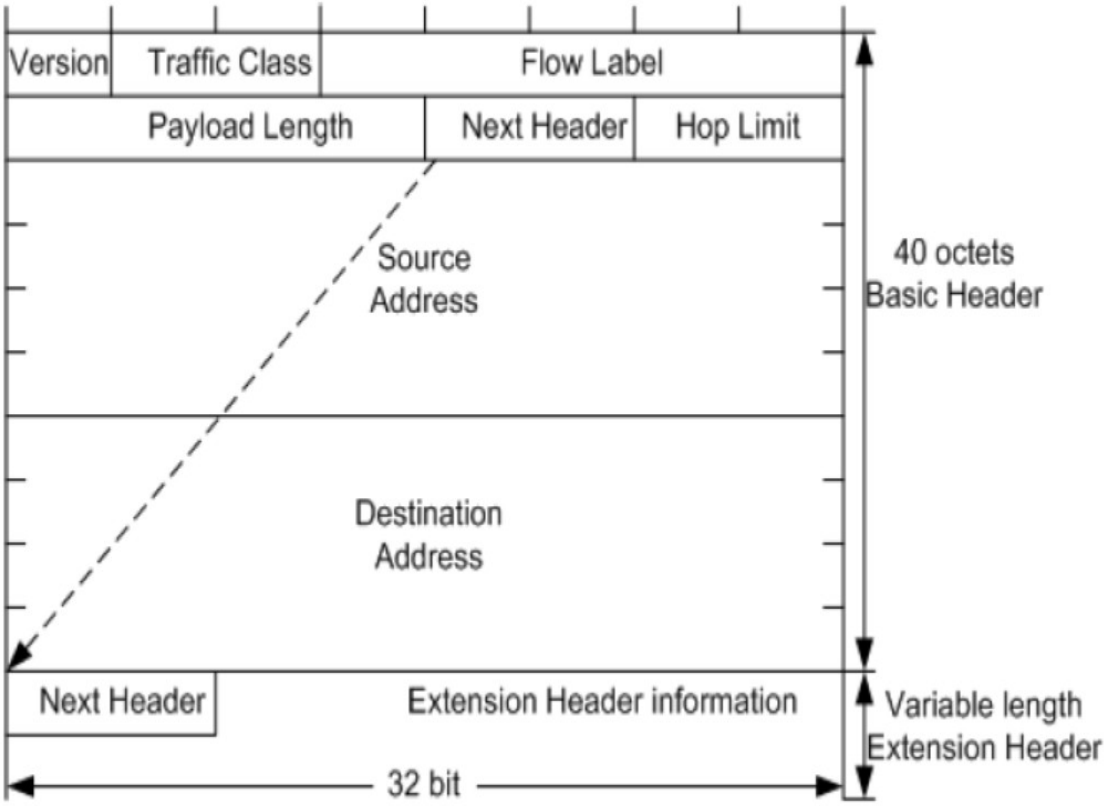
IPv6 报文格式

IPv6 报文由 IPv6 基本报头、IPv6 扩展报头以及上层协议数据单元三部分组成。

上层协议数据单元一般由上层协议报头和它的有效载荷构成，有效载荷可以是一个 ICMPv6 报文、一个 TCP 报文或一个 UDP 报文。

```
Internet Protocol Version 6, Src: ::, Dst: ff02::1:ff2a:15f7
  0110 .... = Version: 6
> .... 1100 0000 .... .... .... .... .... = Traffic Class: 0xc0 (DSCP: CS6, ECN: Not-ECT)
  .... .... .... 0000 0000 0000 0000 0000 = Flow Label: 0x00000
  Payload Length: 24
  Next Header: ICMPv6 (58)
  Hop Limit: 255
  Source: ::
  Destination: ff02::1:ff2a:15f7
Internet Control Message Protocol v6
```

IPv6 基本报头有 8 个字段，固定大小为 40 字节，每一个 IPv6 数据报都必须报含报头。



1 Version：版本号，长度为 4bit。对于 IPv6，该值为 6。

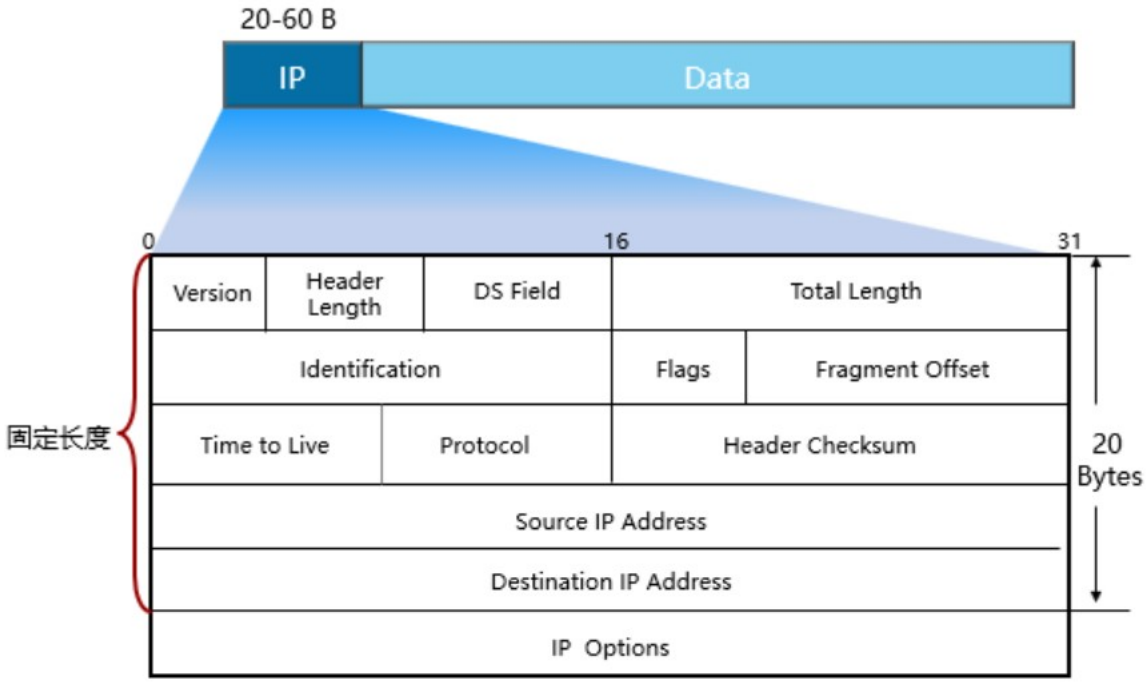
2 Traffic Class：流类别，长度为 8bit。等同于 IPv4 中的 TOS 字段，表示 IPv6 数据报的类或优先级，主要

应用于 QoS。

3 Flow Label：流标签，长度为 20bit。IPv6 中的新增字段，用于区分实时流量，不同的流标签+源地址
可以唯一确定一条数据流，中间网络设备可以根据这些信息更加高效率的区分数据流。
4 Payload Length：有效载荷长度，长度为 16bit。有效载荷是指紧跟 IPv6 报头的数据报的其它部分
（即扩展报头和上层协议数据单元）。
5 Next Header：下一个报头，长度为 8bit。该字段定义紧跟在 IPv6 报头后面的第一个扩展报头（如果
存在）的类型，或者上层协议数据单元中的协议类型。
6 Hop Limit：跳数限制，长度为 8bit。该字段类似于 IPv4 中的 Time to Live 字段，它定义了 IP 数据报所
能经过的最大跳数。
7 Source Address：源地址，长度为 128bit。表示发送方的地址。
8 Destination Address：目的地址，长度为 128bit。表示接收方的地址。

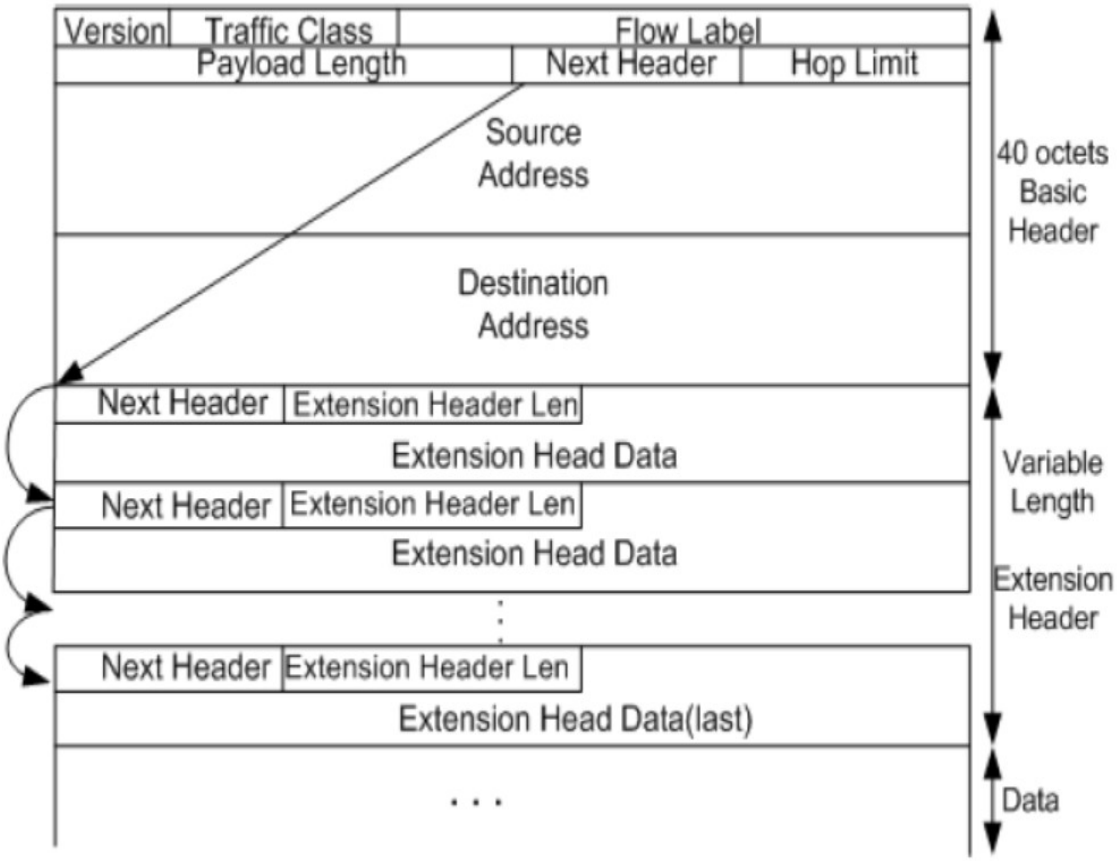
IPv6 和 IPv4 相比，去除了 IHL、identifiers、Flags、Fragment Offset、Header Checksum、 Options、Paddiing 域，只增了流标签域，因此 IPv6 报文头的处理较 IPv4 大大简化，提高了处理效率。另外，IPv6 为了更好支持各种选项处理，提出了扩展头的概念，新增选项时不必修改现有结构就能做到，理论上可以无限扩展，体现了优异的灵活性。

IPv4 头部



IPv6 将这些 Options 从 IPv6 基本报头中剥离，放到了扩展报头中，扩展报头被置于 IPv6 报头和上层协议数据单元之间。一个 IPv6 报文可以包含 0 个、1 个或多个扩展报头，仅当需要路由器或目的节点做某些特殊处理时，才由发送方添加一个或多个扩展头。与 IPv4 不同，IPv6 扩展头长度任意，不受 40 字节限制，这样便于日后扩充新增选项，这一特征加上选项的处理方式使得 IPv6 选项能得以真正的利用。但是为了提高处理选项头和传输层协议的性能，扩展报头总是 8 字节长度的整数倍。

当使用多个扩展报头时，前面报头的 Next Header 字段指明下一个扩展报头的类型，这样就形成了链状的报头列表。



1 Next Header：下一个报头，长度为 8bit。与基本报头的 Next Header 的作用相同。指明下一个扩展报头（如果存在）或上层协议的类型。
2 Extension Header Len：报头扩展长度，长度为 8bit。表示扩展报头的长度（不包含 Next Header 字段）。
3 Extension Head Data：扩展报头数据，长度可变。扩展报头的内容，为一系列选项字段和填充字段的组合。

==========

IPv6 地址的结构

一个 IPv6 地址可以分为如下两部分：
1 网络前缀：n 比特，相当于 IPv4 地址中的网络 ID
2 接口标识：128-n 比特，相当于 IPv4 地址中的主机 ID

IPv6 地址的表示方法

2031:0000:130F:0000:0000:09C0:876A:130B，这是 IPv6 地址的首选格式。
简写为：2031:0:130F::9C0:876A:130B

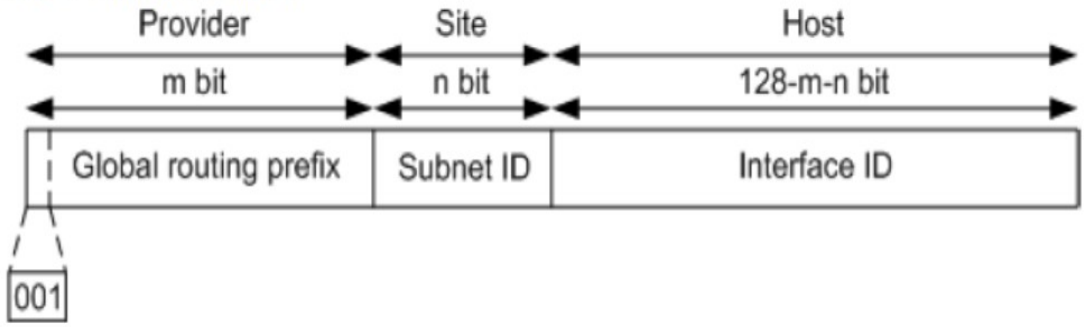
为了书写方便，IPv6 还提供了压缩格式，具体压缩规则为：
1 每组中的前导“0”都可以省略
2 地址中包含的连续两个或多个均为 0　的组，可以用双冒号“::”来代替，双冒号只能有一个

接口标识可通过三种方法生成：
手工配置、系统通过软件自动生成或 IEEE EUI-64 规范生成。其中，EUI-64 规范自动生成最为常用。
==========

IPv6 地址分为单播地址、任播地址（Anycast Address）、组播地址三种类型。

单播地址
IPv6 定义了多种单播地址，目前常用的单播地址有：未指定地址、环回地址、全球单播地址、链路本地地址、唯一本地地址 ULA（Unique Local Address）。
未指定地址:　即 0:0:0:0:0:0:0:0/128 或者::/128。
环回地址：即 0:0:0:0:0:0:0:1/128 或者::1/128。环回与 IPv4 中的 127.0.0.1 作用相同，主要用于设备给自己发送报文。
全球单播地址：是带有全球单播前缀的 IPv6 地址，其作用类似于 IPv4 中的公网地址。
全球路由前缀（Global routing prefix）、子网 ID（subnet ID）和接口标识（Interface ID）组成，

图 2 全球单播地址格式



目前已经分配的全球路由前缀的前 3bit 均为 001。因此前缀为 2000::/3。

链路本地地址 ： link-local 前缀 FE80::/10
是 IPv6 中的应用范围受限制的地址类型，只能在连接到同一本地链路的节点之间使用。它使用了特定的本地链路前缀 FE80::/10（最高 10 位值为 1111111010），同时将接口标识添加在后面作为地址的低 64 比特。

当一个节点启动 IPv6 协议栈时，启动时节点的每个接口会自动配置一个链路本地地址（其固定的前缀+EUI-64 规则形成的接口标识）。这种机制使得两个连接到同一链路的 IPv6 节点不需要做任何配置就可以通信。所以链路本地地址广泛应用于邻居发现，无状态地址配置等应用。

类似于 IPv4 的 169.254.0.0/16 是一个本地链接地址段

唯一本地地址 ： site-local FC00::/7
是另一种应用范围受限的地址，它仅能在一个站点内使用。它的作用类似于 IPv4 中的私网地址，任何没有申请到提供商分配的全球单播地址的组织机构都可以使用唯一本地地址。唯一本地地址只能在本地网络内部被路由转发而不会在全球网络中被路由转发。

类似于 IPv4 的私有地址
A 类 10.0.0.0 - 10.255.255.255
B 类 172.16.0.0 - 172.31.255.255
C 类 192.168.0.0 - 192.168.255.255

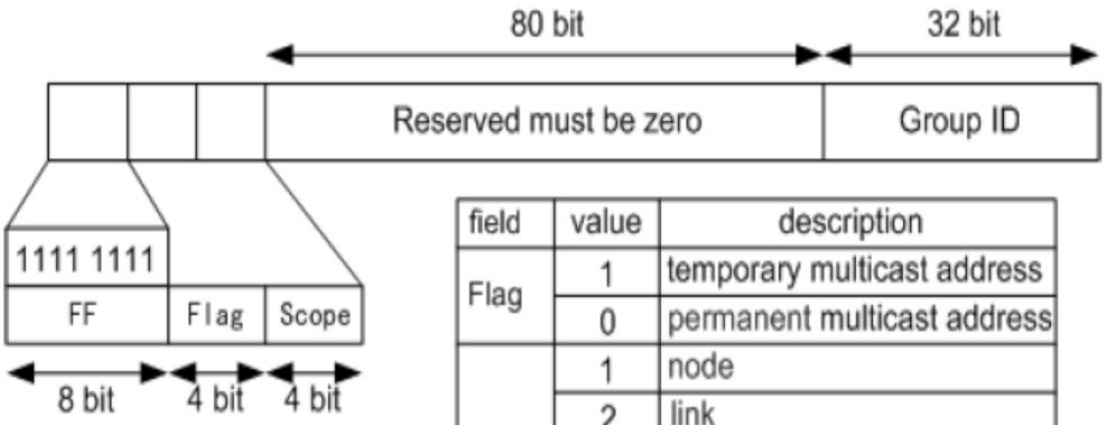
唯一本地地址具有如下特点：
具有全球唯一的前缀（虽然随机方式产生，但是冲突概率很低）。
可以进行网络之间的私有连接，而不必担心地址冲突等问题。
具有知名前缀（FC00::/7），方便边缘路由器进行路由过滤。
如果出现路由泄漏，该地址不会和其他地址冲突，不会造成 Internet 路由冲突。
应用中，上层应用程序将这些地址看作全球单播地址对待。
独立于互联网服务提供商 ISP（Internet Service Provider）。

组播地址

IPv6 的组播与 IPv4 相同，用来标识一组接口，一般这些接口属于不同的节点。一个节点可能属于 0 到多个组播组。发往组播地址的报文被组播地址标识的所有接口接收。
一个 IPv6 组播地址由前缀，标志（Flag）字段、范围（Scope）字段以及组播组 ID（Global ID）4 个部分组成：
前缀：IPv6 组播地址的前缀是 FF00::/8（1111 1111）类似于 224.0.0.0
标志字段（Flag）：长度 4bit，只用最后一比特（前三位必须置 0），
当该位值为 0 时，表示当前的组播地址是由 IANA 所分配的一个永久分配地址；
当该值为 1 时，表示当前的组播地址是一个临时组播地址（非永久分配地址）。

范围字段（Scop）：长度 4bit，用来限制组播数据流在网络中发送的范围
组播组 ID（Global ID）：长度 112bit，用以标识组播组。目前，并没有将所有的 112 位都定义成组标识，而是建议仅使用该 112 位的最低 32 位作为组播组 ID，将剩余的 80 位都置 0。这样每个组播组 ID 都映射到一个唯一的以太网组播 MAC 地址

0：预留
1：节点本地范围
2：链路本地范围，例如 FF02::1
5：站点本地范围
8：组织本地范围
E：全球范围
F：预留



field	value	description
Flag	1	temporary multicast address
	0	permanent multicast address
Scope	1	node
	2	link
	4	management
	5	site
	8	organization
	E	global
	the rest	unsigned or reserved

被请求节点组播地址

被请求节点组播地址（Solicited-Node Multicast Address）通

过节点的单播或任播地址生成。当一个节点具有了单播或任播地址，就会对应生成一个被请求节点组播地址，并且加入这个组播组。一个单播地址或任播地址对应一个被请求节点组播地址。该地址主要用于邻居发现机制和地址重复检测功能。

IPv6 中没有广播地址，也不使用 ARP。但是仍然需要从 IP 地址解析到 MAC 地址的功能。在 IPv6 中，这个功能通过邻居请求 NS（Neighbor Solicitation）报文完成。当一个节点需要解析某个 IPv6 地址对应的 MAC 地址时，会发送 NS 报文，该报文的目的 IP 就是需要解析的 IPv6 地址对应的被请求节点组播地址；只有具有该组播地址的节点会检查处理。
被请求节点组播地址由前缀 FF02::1:FF00:0/104 和单播地址的最后 24 位组成。

任播地址

任播地址标识一组网络接口（通常属于不同的节点）。目标地址是任播地址的数据包将发送给其中路由意义上最近的一个网络接口。适合于“One-to-One-of-Many”（一对组中的一个）的通讯场合。

IPv6 任播地址仅可用被分配给路由设备，不能应用于主机。
任播地址不能作为 IPv6 报文的源地址。

```
int g0/0/1
ipv6 enable
ipv6 address 2000::1 /3 anycast
```

任播地址设计用来在给多个主机或者节点提供相同服务时提供冗余功能和负载分担功能。目前，任播地址的使用通过共享单播地址方式来完成。将一个单播地址分配给多个节点或者主机，

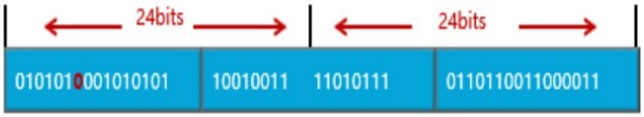
这样在网络中如果存在多条该地址路由，当发送者发送以任播地址为目的 IP 的数据报文时，发送者无法控制哪台设备能够收到，这取决于整个网络中路由协议计算的结果。这种方式可以适用于一些无状态的应用，例如 DNS 等。
IPv6 中没有为任播规定单独的地址空间，任播地址和单播地址使用相同的地址空间。目前 IPv6 中任播主要应用于移动 IPv6。在 6to4 中继中也使用了任播前缀（2002:c058:6301::）。
==========

接口 ID

关于接口 ID：接口 ID 为 64bit，用于标识链路上的接口，在每条链路上接口 ID 必须唯一。
接口 ID 可通过 3 种方法生成：

手工配置：建议在服务器和重要网络设备上配置。
系统通过软件自动生成：保护主机的私密性。
IEEE EUI-64 规范自动生成：最常用的方法。
EUI-64 (64-bit Extended Unique Identifier) 64 位扩展唯一标识符

48位以太网MAC地址



EUI-64生成的接口ID

将FFFE插入MAC地址的前24位与后24位之间，并将第7位的值取反（比如0改为1）即可生成接口ID。

将 48bit 的 MAC 地址对半劈开，然后插入“FFFE”，
再对从左数起的第 7 位，也就是 U/L 位取反，即可得到对应的接口 ID。

==========

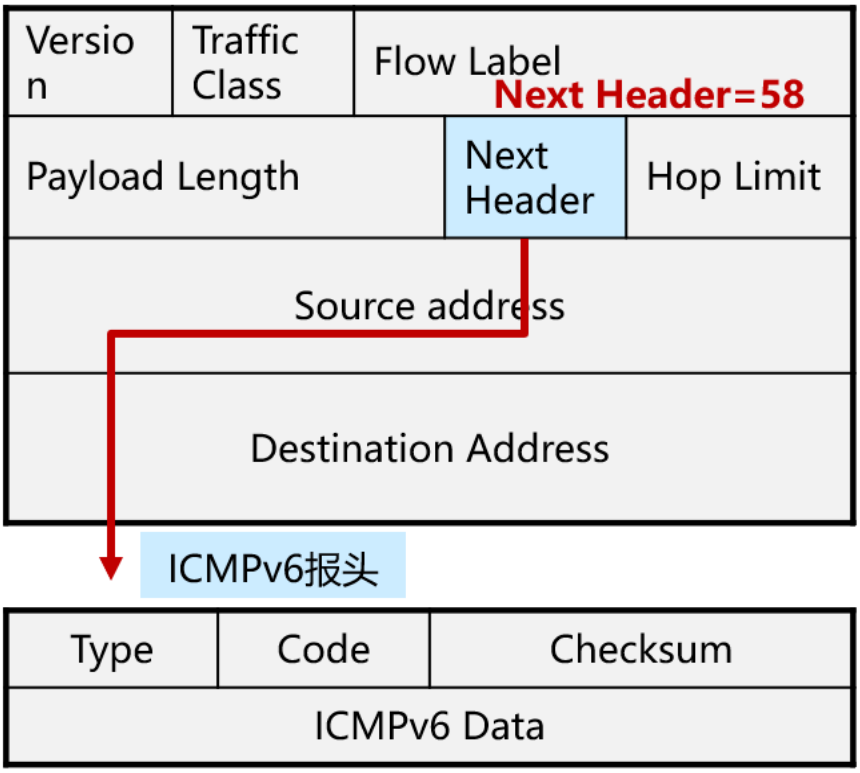
ICMPv6

ICMPv6（Internet Control Message Protocol for the IPv6）是 IPv6 的基础协议之一。
ICMPv6 的协议类型号（即 IPv6 报文中的 Next Header 字段的值）为 58 ，除了提供 ICMPv4 常用的功能之外，还是其它一些功能的基础，如邻接点发现、无状态地址配置（包括重复地址检测）、PMTU 发现等。

路径 MTU

路径 MTU 是指一条因特网传输路径中，从源地址到目的地址所经过的“路径”上的所有 IP 跳的最大传输单元的最小值。或

者从另外一个角度来看，就是无需进行分片处理就能穿过这条“路径”的最大传输单元的最大值。



ICMPv6 协议报文

（1）目的不可达错误报文：type=1
（2）数据包过大错误报文：type=2
（3）时间超时错误报文： type=3
（4）参数错误报文： type=4
（5）信息报文： type=128、type=129
（7）RS type=133
（8）RA type=134
（9）NS type=135
（10）NA type=136
（11）redirect type=137

地址解析：NS 组播请求 FF02::1:FFXX:XXXX，NA 单播响应
重复地址：NS 组播请求 FF02::1:FFXX.XXXX，NA 组播响应 FF02::1
路由发现：RS 组播请求 FF02::2，RA 组播响应 FF02::1

NDP

NDP（Neighbor Discovery Protocol，邻居发现协议）替代了 IPv4 的 ARP（Address Resolution Protocol）和 ICMP 路由器发现（Router Discovery），它定义了使用 ICMPv6 报文实现地址解析，跟踪邻居状态，重复地址检测，路由器发现以及重定向等功能。

NDP 的 7 个作用
1 路由器发现
2 无状态自动配置
3 重复地址检测
4 地址解析
5 邻居的状态跟踪
6 前缀重编址：
7 路由器重定向

RS（Router Solicitation）：路由器请求报文
RA（Router Advertisement）：路由器通告报文
NS（Neighbor Solicitation）：邻居请求报文
NA（Neighbor Advertisement）：邻居通告报文

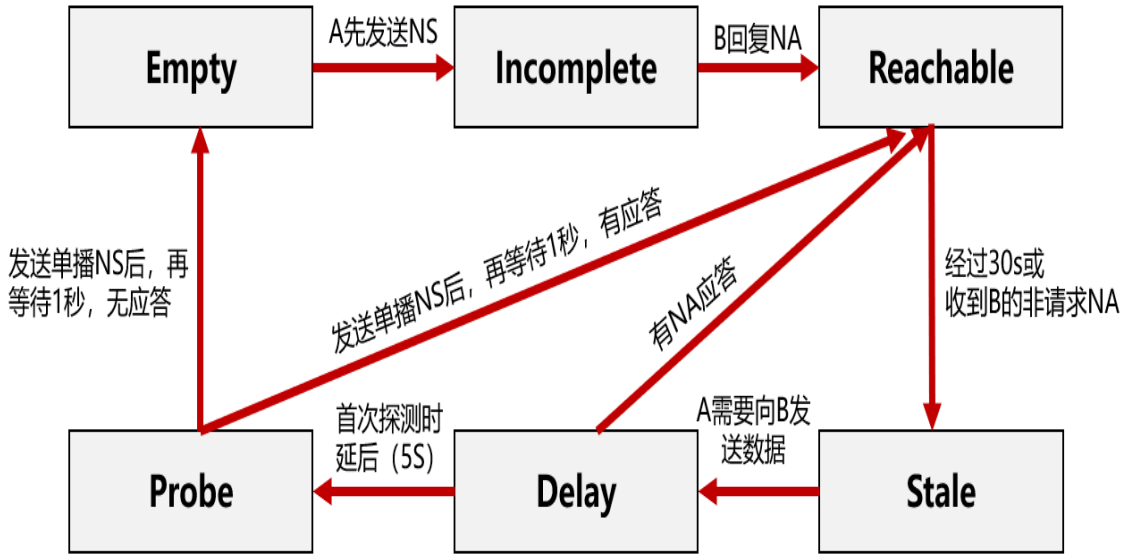
地址解析

IPv6 的地址解析不再使用 ARP，也不再使用广播方式。
地址解析在三层完成，针对不同的链路层协议可以采用相同的地址解析协议
通过 ICMPv6（类型 135 的 NS 及类型 136 的 NA 报文）来实现地址解析。
NS 报文发送使用组播的方式，报文的目的 IPv6 地址为被请求的 IPv6 地址对应的“被请求节点组播地址”，报文的目的 MAC 为组播 MAC。
采用组播的方式发送 NS 消息相比于广播的方式更加的高效，也减少了对其他节点的影响和对二层网络的性能压力。可以使用三层的安全机制（例如 IPSec）避免地址解析攻击。

地址解析过程中使用了两种 ICMPv6 报文：
邻居请求（Neighbor Solicitation）Type=135，Code=0
邻居通告（Neighbor Advertisement）Type=136，Code=0

display ipv6 neighbors
邻居状态种类
定义了 5 种邻居状态，分别是：未完成（INCOMPLETE）、可达（REACHABLE）、陈旧（STALE）、延迟（DELAY）、探查（PROBE）

INCOMPLETE 未完成，邻居请求已经发送到目标节点的请求组播地址，但没有收到邻居的通告；
REACHABLE 可达，收到确认，不续再发包确认；
STALE 陈旧，从收到上一次可达性确认后过了超过 30s；
DELAY 延迟，在 stale 状态后发送过一个报文，并且 5s 内没有可达性确认；
PROBE 探查，每隔 1s 重传邻居请求来主动请求可达性确认，直到收到确认。



重复地址检测

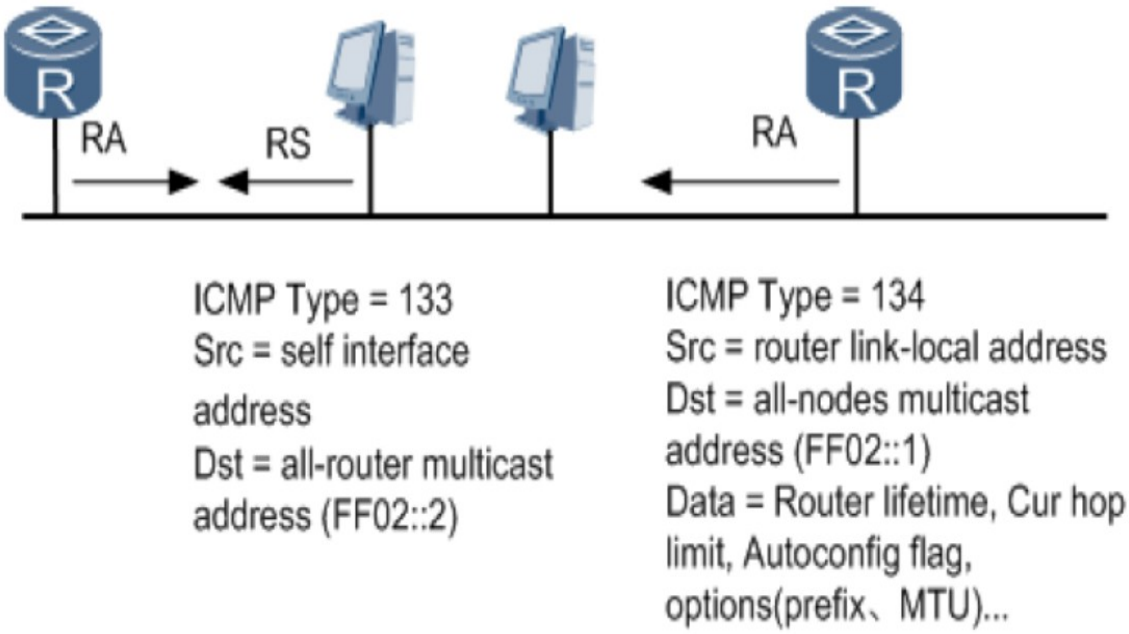
重复地址检测 DAD（Duplicate Address Detect）是在接口使用某个 IPv6 单播地址之前进行的，主要是为了探测是否有其它的节点使用了该地址。

IPv6 重复地址检测技术和 IPv4 中的免费 ARP 类似：节点向一个自己将使用的试验地址所在的 Solicited-Node 组播组发送一个以该实验地址为请求的目标地址的 NS 报文，如果收到某个其他站点回应的 NA 报文，就证明该地址已被网络上使用，节点将不能使用该实验地址通讯。

无状态自动配置

无状态自动配置即自动生成链路本地地址，主机根据 RA 报文的前缀信息，自动配置全球单播地址等，并获得其他相关信息。

RS 路由器请求（Router Solicitation） type=133
RA 路由器通告（Router Advertisement）type=134

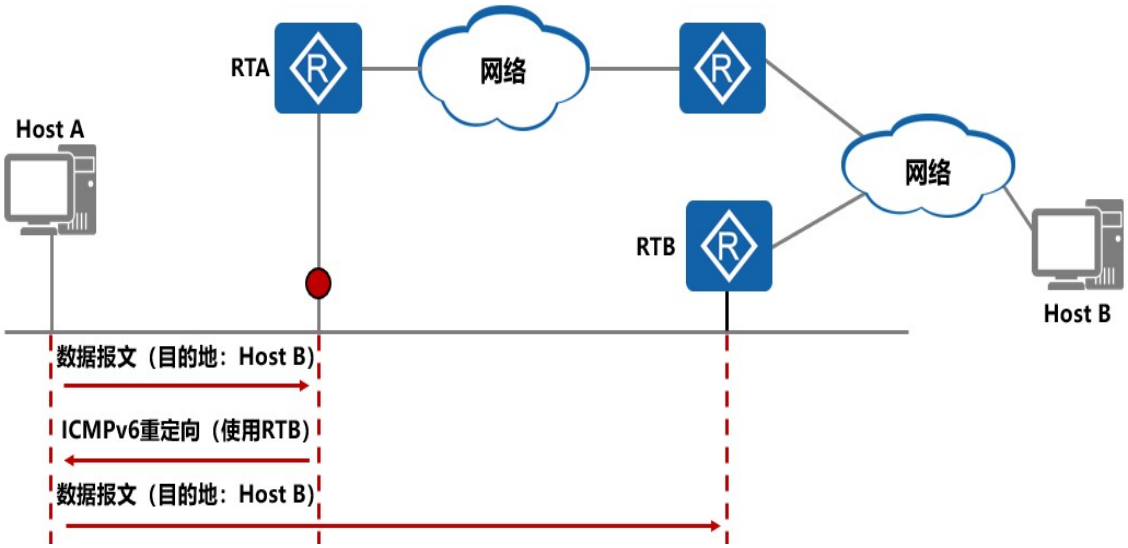


IPv6 主机无状态自动配置过程：

1. 根据接口标识产生链路本地地址。
2. 发出邻居请求，进行重复地址检测。
3. 如地址冲突，则停止自动配置，需要手工配置。
4. 如不冲突，链路本地地址生效，节点具备本地链路通信能力。
5. 主机会发送 RS 报文（或接收到路由器定期发送的 RA 报文）。
6. 根据 RA 报文中的前缀信息和通过 EUI-64 规范生成的接口标识得到 IPv6 地址。

重定向

当网关路由器发现报文从其它网关路由器转发更好，它就会发送重定向报文告知报文的发送者，让报文发送者选择另一个网关路由器。重定向报文也承载在 ICMPv6 报文中，其 Type 字段值为 137，报文中会携带更好的路径下一跳地址和需要重定向转发的报文的目的地址等信息。



RTA 接收到 A 发送的报文以后会发现实际上主机 A 直接发送给路由器 R2 更好，它将发送一个 ICMPv6 重定向报文给主机 A，其中 Target Address 为 RTB，Destination Address 为主机 B。

主机 A 接收到了重定向报文之后，会在默认路由表中添加一个主机路由，以后发往主机 B 的报文就直接给 RB。

有个问题：RTA 如何知道去往主机 B 的路径通过 RTB 更好呢？

其实这个很简单，因为 RTA 会发现报文进入的接口就是报文路由得出接口，也就是说发往主机 B 的路由实际上只是在 RTA 上转了一圈出来了，然后转发到 RTB，据此，RTA 能判断出直接给 RTB 是更好的路径。

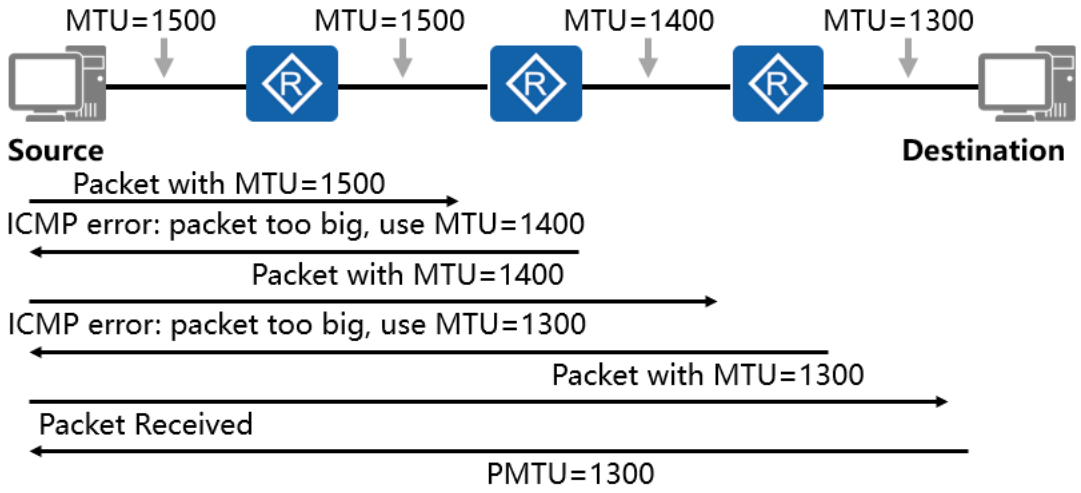
==========

Path MTU

PMTU 就是路径上的最小接口 MTU。

在 IPv4 中，报文如果过大，必须要分片进行发送，所以在每个节点发送报文之前，设备都会根据发送接口的最大传输单元 MTU（Maximum Transmission Unit）来对报文进行分片。但

是在 IPv6 中，为了减少中间转发设备的处理压力，中间转发设备不对 IPv6 报文进行分片，报文的分片将在源节点进行。



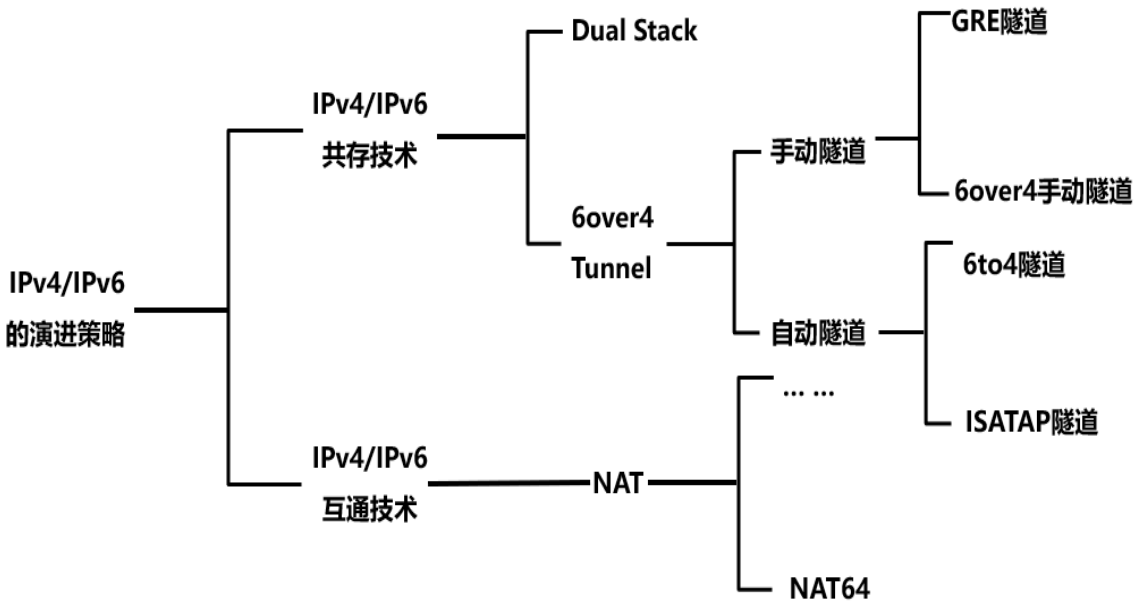
PMTU 协议是通过 ICMPv6 的 Packet Too Big 报文来完成的。首先源节点假设 PMTU 就是其出接口的 MTU，发出一个试探性的报文，当转发路径上存在一个小于当前假设的 PMTU 时，转发设备就会向源节点发送 Packet Too Big 报文，并且携带自己的 MTU 值，此后源节点将 PMTU 的假设值更改为新收到的 MTU 值继续发送报文。如此反复，直到报文到达目的地之后，源节点就能知道到达目的地的 PMTU 了。

==========

IPv6 过渡技术

现在是 IPv4 网络，IPv6 现在部署的话需要做些什么？

答：现在是 IPv4 网络，没有必要也不可能所有节点同时升级成 IPv6。在部署 IPv6 的时候可以逐步过渡，并且尽量使用一些技术使得网络升级变得比较平滑。



IPv6 与 IPv4 共存技术：
双协议栈： IPv6 节点同时支持 IPv6 和 IPv4 协议栈。
隧道： IPv6 报文作为 IPv4 的载荷，由 IPv4 Internet 中连接多个 IPv6 孤岛。

IPv6 与 IPv4 互通技术：
提供 IPv6 与 IPv4 互相访问。适用于 IPv6 Internet 与 IPv4 Internet 共存，而两者又有互相通讯的需求。

（1）双栈技术：即设备或者终端同时支持 IPv4 和 IPv6 的协议栈，使得设备能在 ipv4 和 ipv6 环境下都正常工作（所有设备都需要支持双栈）
特点：支持多种链路类型
缺点：当两个栈比较远时需要在通过的设备开启双栈技术

（2）隧道：解决 ipv6“孤岛”问题
1 手工隧道：需要手工指定隧道的终点
a)IPv6 over IPv4 手动隧道:
直接把 IPv6 报文封装在 IPv4 报文中。边界设备必须是支持双

栈的设备，中间的 IPv4 设备按照正常包转发进行处理
b)GRE 隧道:
不仅可传递 ipv6 的数据还可以传递 ipse 等其他数据。GRE 相对 ipv6 over ipv4 隧道多了 4byte 的 GRE 头部，封装和解封带来的开销相对来说比较大

2 自动隧道：可以自动获得隧道终点的 IPv4 地址

6to4 是使用 IPv4 地址做为网络前缀，而 ISATAP 用 IPv4 地址做为接口标识。
ISATAP（Intra-Site Automatic Tunnel Addressing Protocol）

a)ipv4 兼容 ipv6 自动隧道:ISATAP
IPv4 兼容 IPv6 地址的前 96 位全部为 0，后 32 位为 IPv4 地址。（ipv6 address ::2.2.2.2/96 ）
作用：只能实现两台主机之间的互访；
b)6to4:
把终点放在 IPv6 地址的第 2、3 块 tunnel 地址的格式：2002：32bit ipv4 address ：：x/64
写两条静态路由：2002：：/16 tunnel0/0/0（去往某个 ipv6 网段，只写出接口为 tunnel 接口，不写下一跳，因为下一跳是不确定的）引入到 site 的 IPV6 动态路由协议中；
作用：可以实现两个 ipv6 站点之间的互访；
6to4 自动隧道支持 Router 到 Router、Host 到 Router、Router 到 Host、 Host 到 Host。

（3）NAT-PT：实现 IPv6 网络的主机和 IPv4 主机的互访
1 静态映射的 NAT-PT 机制
2 动态映射的 NAT-PT 机制：需要建立地址池
3 NAPT-PT 机制：不同的 IPv6 地址转换时，可以对应同一个

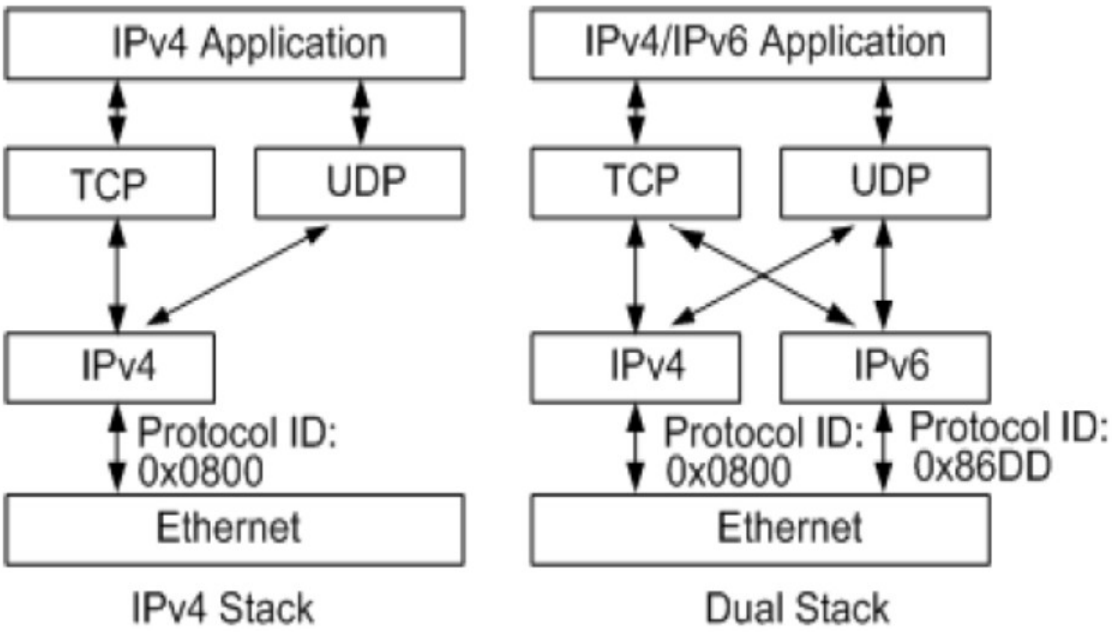
IPv4 地址，通过端口号来区分不同的 IPv6 主机，从而使多个 IPv6 主机能共享一个 IPv4 地址完成转换

表1 IPv6 over IPv4隧道的应用场景

隧道类型	隧道源/目的地址	隧道接口地址	应用场景
IPv6 over IPv4手动隧道	源/目的地址为手动配置的IPv4地址	IPv6地址	简单的IPv6网络或主机之间的点到点连接，隧道仅可以承载IPv6报文。
IPv6 over IPv4 GRE隧道	源/目的地址为手动配置的IPv4地址	IPv6地址	简单的IPv6网络或主机之间的点到点连接，隧道可以承载包括IPv6协议在内的多种上层协议。
IPv6 over IPv4自动隧道	源地址为手动配置的IPv4地址，目的地址不需配置	IPv4兼容IPv6地址，其格式为::IPv4-source-address/96	多用于IPv6主机之间的点到多点的连接。
6to4隧道	源地址为手动配置的IPv4地址，目的地址不需配置	6to4地址，其格式为2002:IPv4-source-address::/48	多用于IPv6网络之间的点到多点的连接。
ISATAP隧道	源地址为手动配置的IPv4地址，目的地址不需配置	ISATAP地址，其格式为Prefix:0	多用于在IPv4网络之内的IPv6节点之间的互联。

双栈 Dual Stack

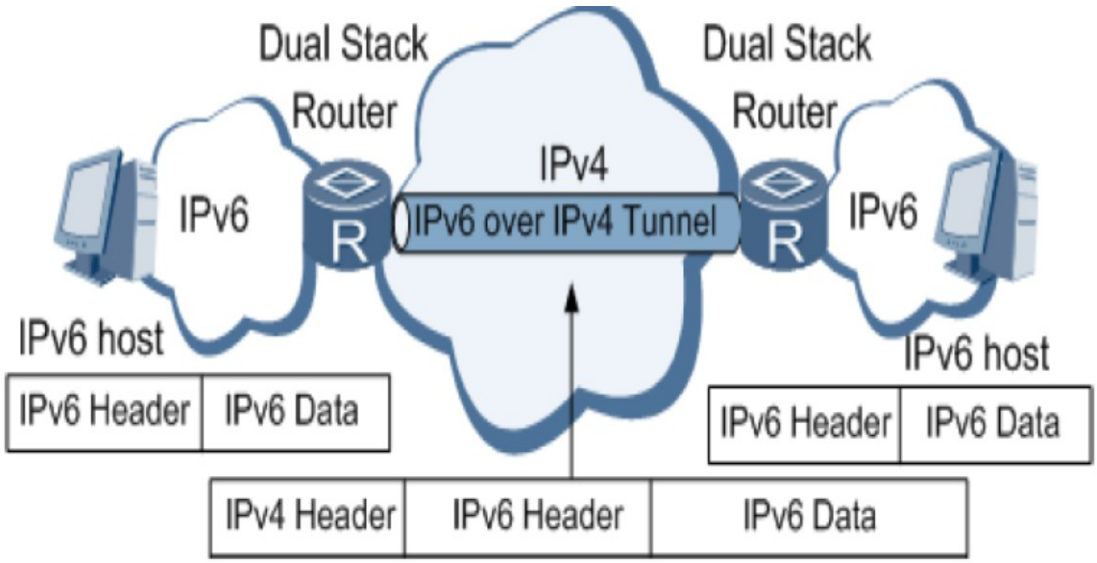
双栈技术是 IPv4 向 IPv6 过渡的一种有效的技术。网络中的节点同时支持 IPv4 和 IPv6 协议栈，源节点根据目的节点的不同选用不同的协议栈，而网络设备根据报文的协议类型选择不同的协议栈进行处理和转发。



6over4 手动隧道

隧道（Tunnel）是一种封装技术。

它利用一种网络协议来传输另一种网络协议，即利用一种网络传输协议，将其他协议产生的数据报文封装在自身的报文中，然后在网络中传输。隧道是一个虚拟的点对点的连接。一个Tunnel提供了一条使封装的数据报文能够传输的通路，并且在一个Tunnel的两端可以分别对数据报文进行封装及解封装。



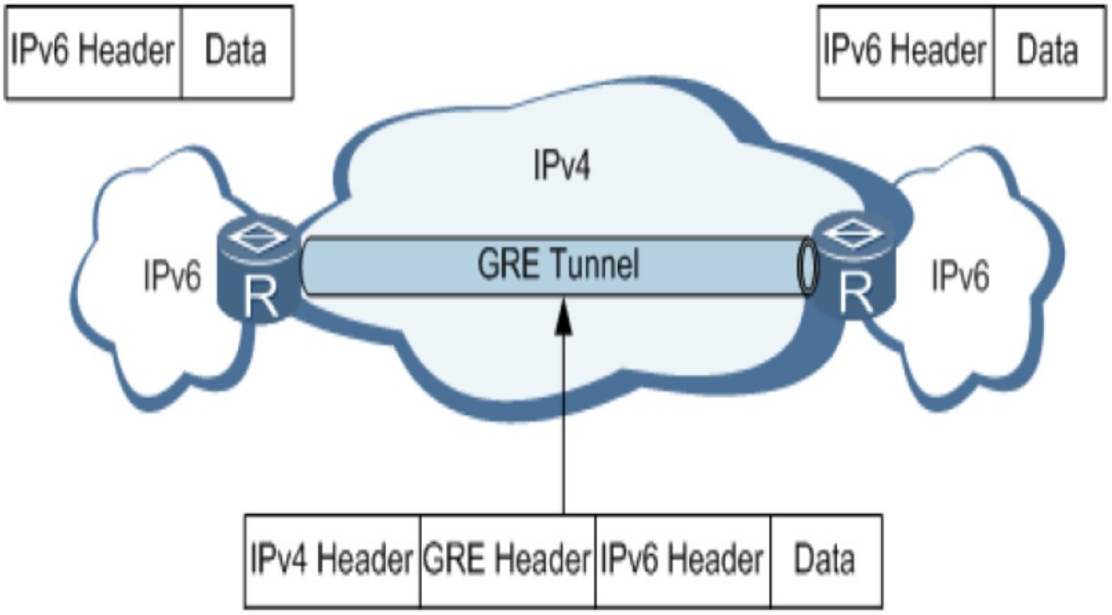
6over4 手动隧道优缺点

优点：可以用于任何IPv6穿越IPv4的环境，通用性好。

缺点：必须手工配置。

6over4 GRE 隧道

GRE（Generic Routing Encapsulation）通用路由封装协议



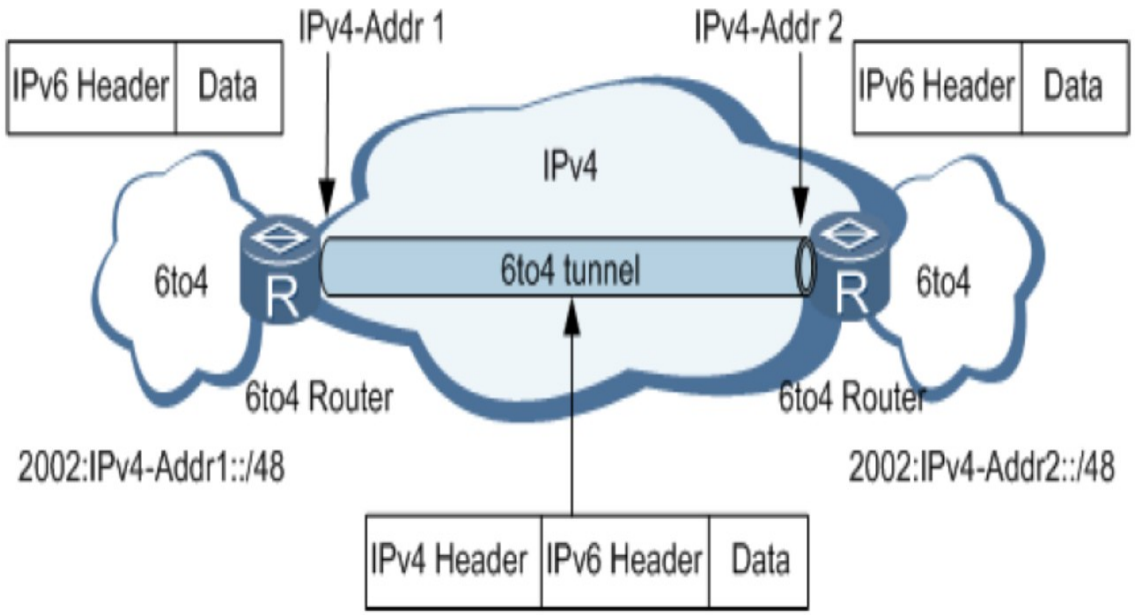
使用标准的 GRE 隧道技术提供了点到点连接服务，需要手工指定隧道的端点地址。GRE 隧道本身并不限制被封装的协议和传输协议，一个 GRE 隧道中被封装的协议可以是协议中允许的任意协议（可以是 IPv4、IPv6、OSI、MPLS 等）。

6to4 自动隧道

6over4 隧道 （一对一）

6over4 隧道也属于一种自动隧道，隧道也是使用内嵌在 IPv6 地址中的 IPv4 地址建立的。与 IPv4 兼容自动隧道不同，6to4 自动隧道支持 Router 到 Router、Host 到 Router、Router 到 Host、 Host 到 Host。

采用 6to4 专用地址，即 2002:IPv4::/48

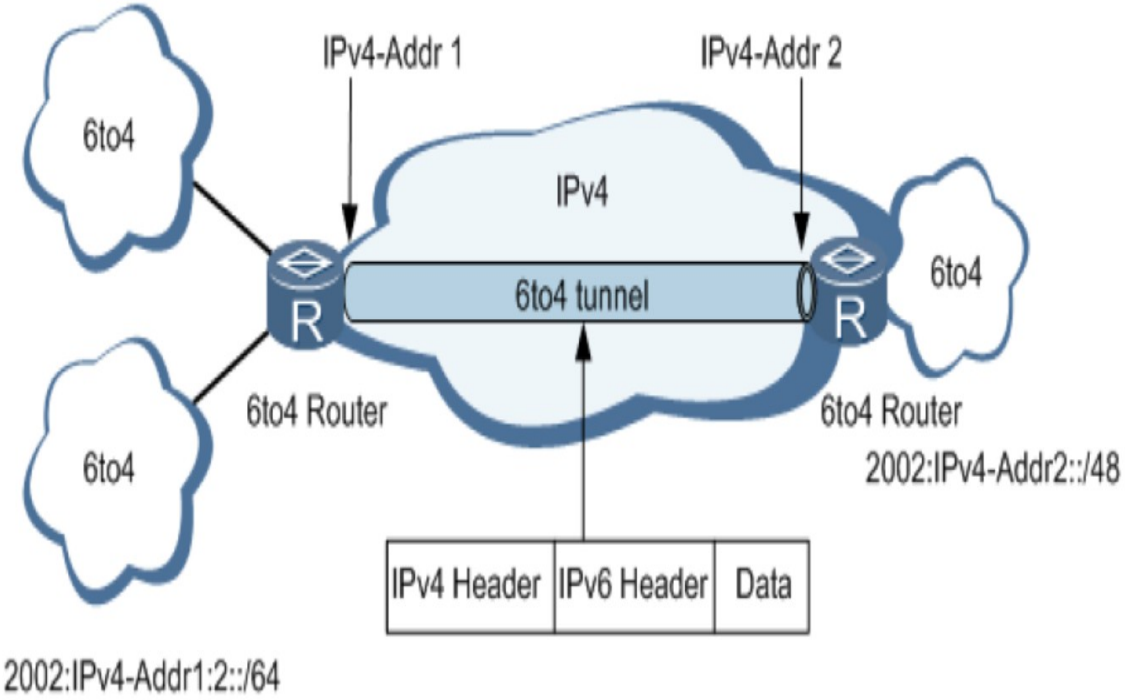


6to4 隧道（多对一）

可连接多个 6to4 网络。

通过 SLA ID 区分。

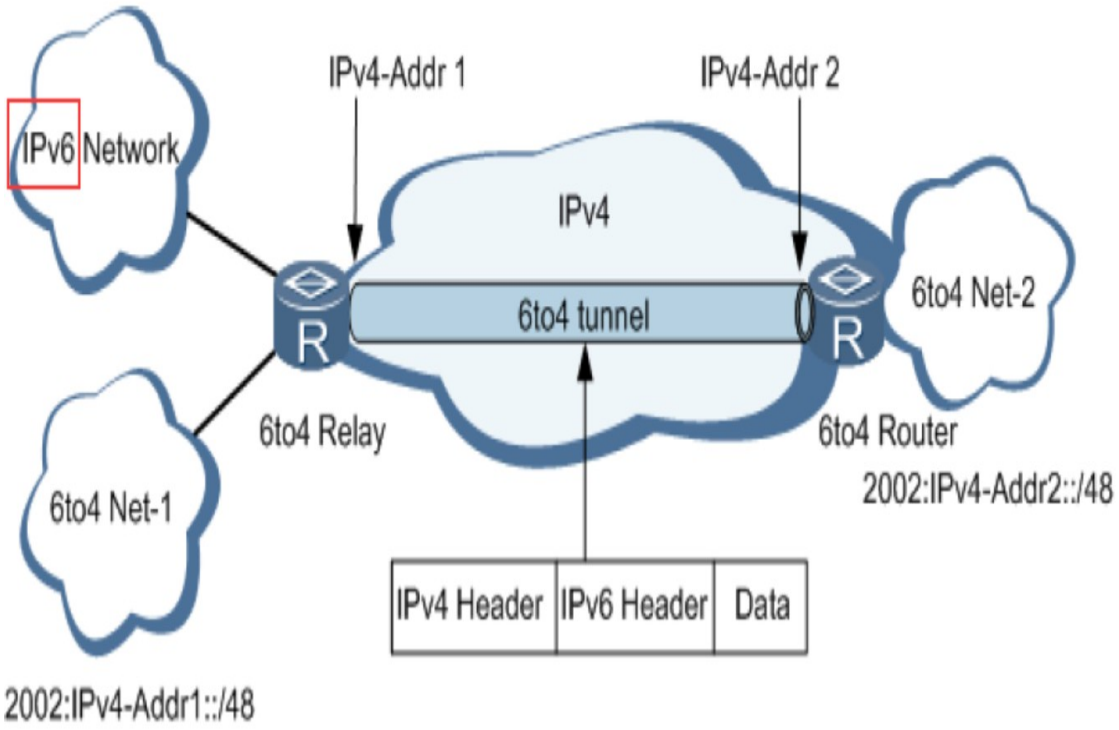
2002:IPv4-Addr1:1::/64



6to4 中继

实现 6to4 网络和 IPv6 普通网络互通。

普通 IPv6 网络需要与 6to4 网络通过 IPv4 网络互通，这可以通过 6to4 中继路由器方式实现。所谓 6to4 中继，就是通过 6to4 隧道转发的 IPv6 报文的目的地址不是 6to4 地址，但转发的下一跳是 6to4 地址，该下一跳为路由器我们称之为 6to4 中继。隧道的 IPv4 目的地址依然从下一跳的 6to4 地址中获得。



ISATAP 自动隧道

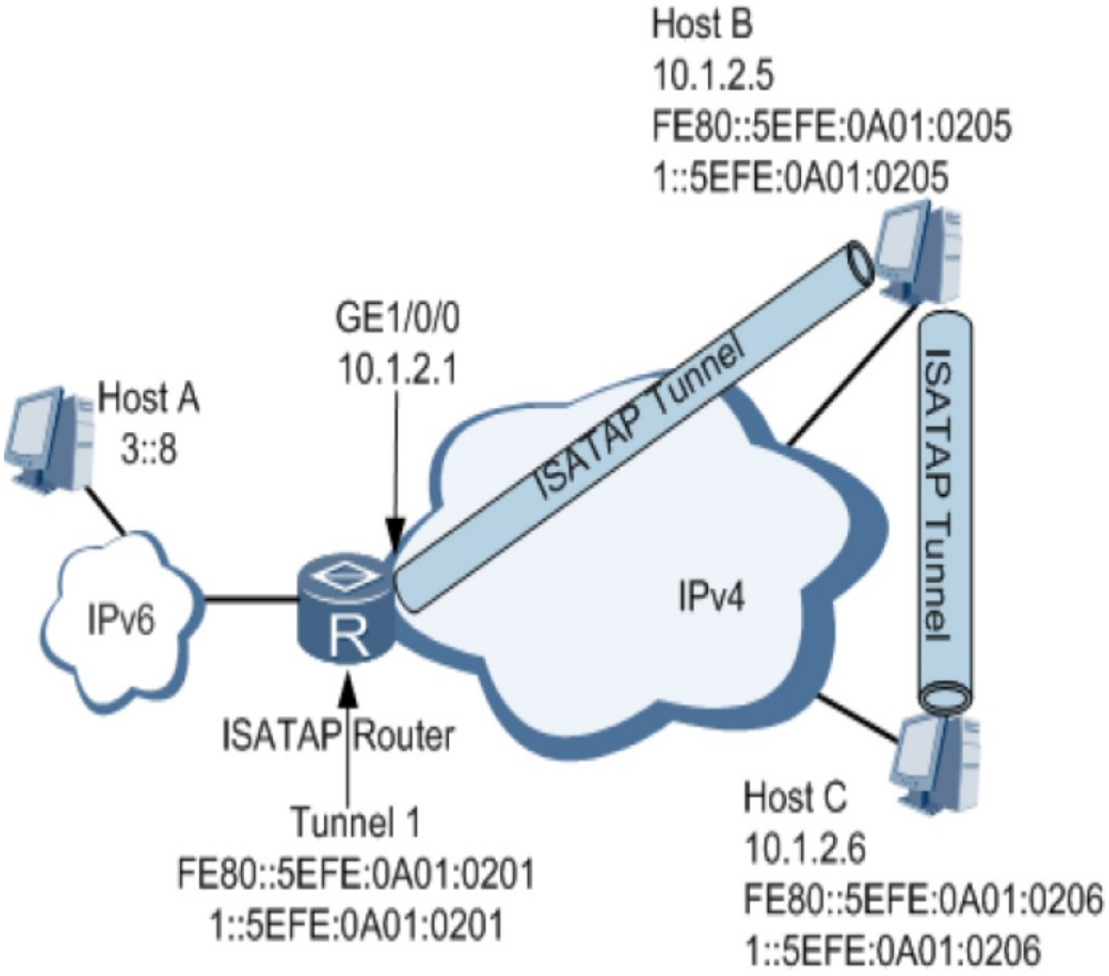
ISATAP（Intra-Site Automatic Tunnel Addressing Protocol）是另外一种自动隧道技术。ISATAP 隧道同样使用了内嵌 IPv4 地址的特殊 IPv6 地址形式，只是和 6to4 不同的是，6to4 是使用 IPv4 地址做为网络前缀，而 ISATAP 用 IPv4 地址做为接口标识。

6over4 自动隧道的一种

支持 Host 到 Router、Router 到 Host、 Host 到 Host

采用 ISATAP 隧道专用地址

ISATAP 过渡机制允许在现有的 IPv4 网络内部部署 IPv6，该技术简单而且扩展性很好，可以用于本地站点的过渡。ISATAP 支持 IPv6 站点本地路由和全局 IPv6 路由域，以及自动 IPv6 隧道。ISATAP 同时还可以与 NAT 结合，从而可以使用站点内部非全局唯一的 IPv4 地址。



NAT64

NAT64 技术实际上是一种协议转换技术，能够将分组在 V4 及 V6 格式之间灵活转换。

IPv6 过渡中的协议翻译技术就是将 IPv6 数据包的每个字段与 IPv4 数据包中的字段建立起一一映射的关系，从而在两个网

络的边缘实现数据报文的转换。

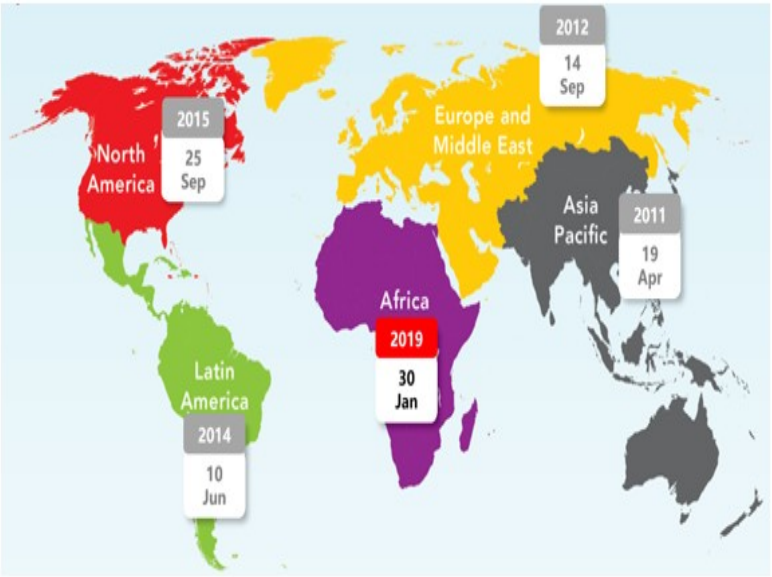
前言

- 众所周知，IPv4地址资源紧张限制了IP技术的进一步发展。我们迫切需要一种能够代替IPv4的技术，在满足IPv4功能的前提下，还能满足未来产业对于IP地址的需求。IPv6能从根本上解决这个问题，各行各业，从政府到市场对下一代互联网技术的迫切需求，推动了IPv6技术的出现与发展。
- 本课程将重点介绍IPv6的基础知识，包括IPv6出现的背景、报文格式、地址分类、基础协议以及过渡技术等内容。

- ICANN(the Internet Corporation for Assigned Names and Numbers)是负责全球互联网上的 IP 地址进行编号分配的机构。
- ICANN 将部分 IP 地址和 AS 号码分配给大洲级的互联网注册机构 RIR（Reginal Internet Registry）。
- 并不是所有的地址都会被分配。一些地址被预留，用于广播、测试、私有网络使用等。这些地址被称为专用地址(special-use address)。你可以查询 RFC5735 来了解哪些地址是专用地址。

国际IPv4地址分配现状

- 2011年2月3日，全球IP地址分配机构（IANA）宣布将其最后的468万个IP地址平均分到全球5个地区的互联网络信息中心，此后再没有可分配的IPv4地址。



- 实践证明 IPv4 是一个非常成功的协议，它本身也经受住了 Internet 从数目很少的计算机发展到目前上亿台计算机互联的考验。但该协议是几十年前基于当时的网络规模而设计的。在今天看来，IPv4 的设计者们对于 Internet 的估计和预想显得很不充分。随着 Internet 的扩张和新应用的不断推出，IPv4 越来越显示出它的局限性。
- Internet 规模的快速扩大是当时完全没有预料到的，特别是近十年来，更是爆炸式增长，已经走进了千家万户，人们的日常生活已经离不开它了。但也就是这种快速发展，出现了迫在眉睫的 IP 地址空间耗尽问题。

IPv4怎么了？

- IPv4公网地址耗尽。这应该是当前IPv6替代IPv4的最大原动力。
- Internet用户快速增长，随着科技行业的发展，有更多的用户、更多种类的设备接入公网。
- IPv4缺乏真正的端到端通信模型。NAT确实能解决私有地址空间与公网互访的问题，但是却破坏了端到端通信的完整性。
- IPv4无法适应新技术的发展，如物联网等。所有行业都是IPv6的潜在用户。
- 广播机制的存在，对ARP的依赖等，使得IPv4局域网的相关运作问题频发。
- IPv4对移动性的支持不够理想。

临时应对措施

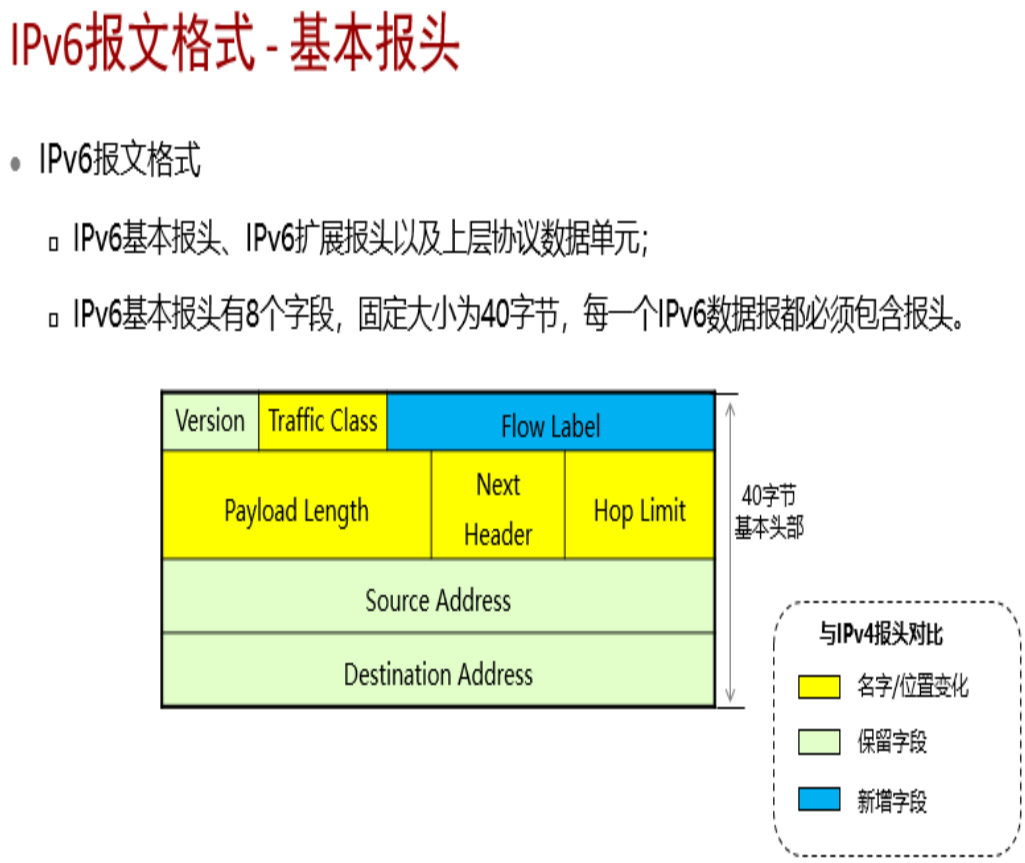
- 1991年，IETF为了推迟IPv4地址耗尽发生的时间点，推出分类网络方案；
- 1993年，推出网络地址转换（NAT）与无类别域间路由（CIDR）；
- 但是这些过渡方案皆无法阻止位址枯竭问题的发生，只能减缓它的发生速度，并不能从根本上解决问题。
- IETF在20世纪90年代提出下一代互联网协议——IPv6，目前IPv6成为公认的IPv4未来的升级版本。

IPv6技术特点

- 地址空间巨大。
- 精简报文结构。
- 实现自动配置和重新编址。
- 支持层次化网络编址。
- 支持端对端安全。
- 更好的支持QoS。
- 支持移动特性。

- IPv6 特点：
- 地址空间，IPv6 地址采用 128 比特标识。128 位的地址结构使 IPv6 理论上可以拥有（43 亿×43 亿×43 亿×43 亿）个地址。近乎无限的地址空间是 IPv6 的最大优势。
- 报文结构，IPv6 使用了新的协议头格式，也就是说 IPv6 数据包有全新的报文头，而并不是仅仅简单地将 IPv4 报文头中的地址部分增加到 128bits 而已。在 IPv6 中，报文头包括固定头部和扩展头部，一些非根本性的和可选择的字段被移到了 IPv6 协议头之后的扩展协议头中。这使得网络中的中间路由器在处理 IPv6 协议头时，有更高的效率。
- 实现自动配置和重新编址，IPv6 协议内置支持通过地址自动配置方式使主机自动发现网络并获取 IPv6 地址，大大提高了内部网络的可管理性。
- 支持层次化网络结构，巨大的地址空间使得 IPv6 可以方便的进行层次化网络部署。层次化的网络结构可以方便的进行路由聚合，提高了路由转发效率。

- 支持端对端安全，IPv6 中，网络层支持 IPSec 的认证和加密，支持端到端的安全。
- 更好的支持 QoS，IPv6 在包头中新定义了一个叫做流标签的特殊字段。IPv6 的流标签字段使得网络中的路由器可以对属于一个流的数据包进行识别并提供特殊处理。用这个标签，路由器可以不打开传送的内层数据包就可以识别流，这就使得即使数据包有效载荷已经进行了加密，仍然可以实现对 QoS 的支持。
- 支持移动特性，由于采用了 Routing header 和 Destination option header 等扩展报头，使得 IPv6 提供了内置的移动性。

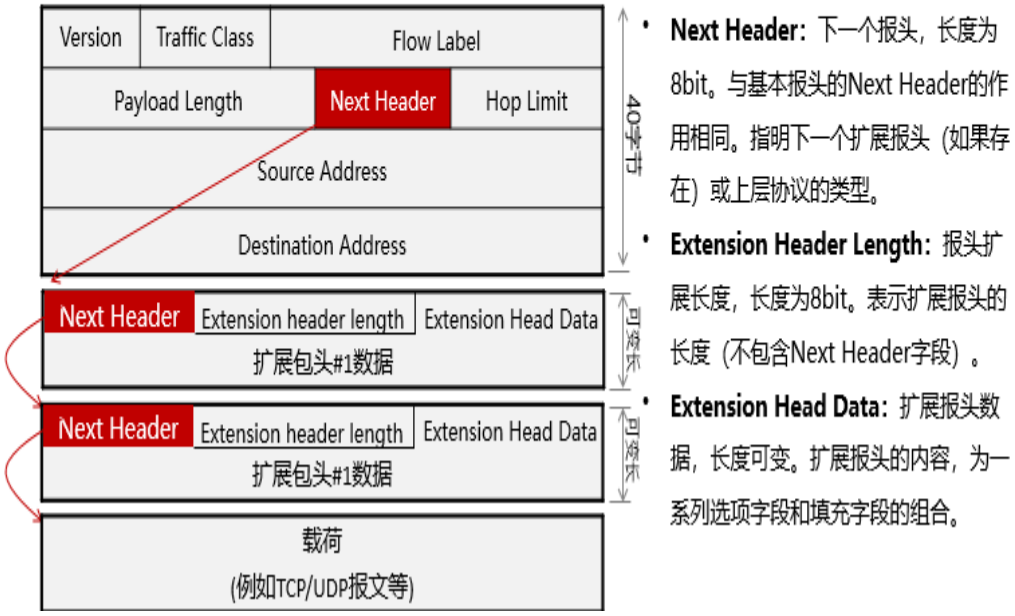


- IPv6 数据包由一个 IPv6 报头、多个扩展报头和一个上层协议数据单元组成。
- IPv6 基本报头（IPv6 Header）

- 每一个 IPv6 数据包都必须包含报头，其长度固定为 40bytes。
- 基本报头提供报文转发的基本信息，会被转发路径上面的所有路由器解析。
- 上层协议数据单元（Upper Layer Protocol Data Unit）
- 上层协议数据单元一般由上层协议包头和它的有效载荷构成，有效载荷可以是一个 ICMPv6 报文、一个 TCP 报文或一个 UDP 报文。
- IPv6 报头格式中主要字段解释如下：
- Version：版本号，长度为 4bit。对于 IPv6，该值为 6。
- Traffic Class：流类别，长度为 8bit。等同于 IPv4 中的 ToS 字段，表示 IPv6 数据报的类或优先级，主要应用于 QoS。
- Flow Label：流标签，长度为 20bit。IPv6 中的新增字段，用于区分实时流量，不同的流标签+源地址可以唯一确定一条数据流，中间网络设备可以根据这些信息更加高效率的区分数据流。
- Payload Length：有效载荷长度，长度为 16bit。有效载荷是指紧跟 IPv6 报头的数据报的其它部分（即扩展报头和上层协议数据单元）。
- Next Header：下一个报头，长度为 8bit。
- Hop Limit：跳数限制，长度为 8bit。该字段类似于 IPv4 中的 Time to Live 字段，它定义了 IP 数据报所能经过的最大跳数。每经过一个路由器，该数值减去 1，当该字段的值为 0 时，数据报将被丢弃。
- Source Address：源地址，长度为 128bit。表示发送方的地址。
- Destination Address：目的地址，长度为 128bit。表示接收方的地址。

IPv6报文格式 - 扩展报头

- 扩展报头是可选的，只有需要该扩展报头对应的功能时，数据的发送者才会添加相应扩展报头。



- 在 IPv4 中，IPv4 报头包含可选字段 Options，内容涉及 security、Timestamp、Record route 等，这些 Options 可以将 IPv4 报头长度从 20 字节扩充到 60 字节。在转发过程中，处理携带这些 Options 的 IPv4 报文会占用路由器很大的资源，因此实际中也很少使用。
- IPv6 将这些 Options 从 IPv6 基本报头中剥离，放到了扩展报头中，扩展报头被置于 IPv6 报头和上层协议数据单元之间。一个 IPv6 报文可以包含 0 个、1 个或多个扩展报头，仅当需要路由器或目的节点做某些特殊处理时，才由发送方添加一个或多个扩展头。与 IPv4 不同，IPv6 扩展头长度任意，不受 40 字节限制，这样便于日后扩充新增选项，这一特征加上选项的处理方式使得 IPv6 选项能得以真正的利用。但是为了提高处理选项头和传输层协议的性能，扩展报头总是 8 字节长度的整数倍。

- 当使用多个扩展报头时，前面报头的 Next Header 字段指明下一个扩展报头的类型，这样就形成了链状的报头列表。

IPv6报文格式 - 扩展报头种类

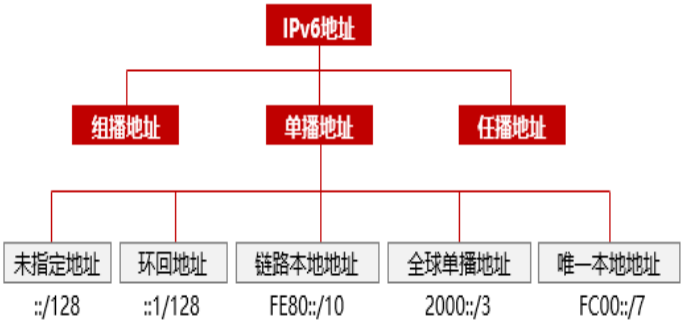
- 当超过一种扩展报头被用在同一个IPv6报文里时，报头必须按照下列顺序出现：

报头类型	Next Header 字段值	描述
逐跳选项报头	0	该选项主要用于为在传送路径上的每跳转发指定发送参数，传送路径上的每台中间节点都要读取并处理该字段，应用场景： 用于巨型载荷 用于路由器提示 用于资源预留
目的选项报头	60	目的选项报头携带了一些只有目的节点才会处理的信息。目前，目的选项报文头主要应用于移动IPv6。
路由报头	43	路由报头和IPv4的Loose Source and Record Route选项类似，该报头能够被IPv6源节点用来强制数据包经过特定的路由器。
分段报头	44	同IPv4一样，IPv6报文发送也受到MTU的限制。当报文长度超过MTU时就需要将报文分段发送，而在IPv6中，分段发送使用的是分段报头。
认证报头	51	该报头由IPSec使用，提供认证、数据完整性以及重放保护。它还对IPv6基本报头中的一些字段进行保护。
封装安全净载报头	50	该报头由IPSec使用，提供认证、数据完整性以及重放保护和IPv6数据报的保密，类似于认证报头。

- 说明：
- 路由设备转发时根据基本报头中 Next Header 值来决定是否要处理扩展头，并不是所有的扩展报头都需要被转发路由设备查看和处理的。
- 除了目的选项扩展报头可能在一个 IPv6 报文中出现一次或两次（一次在路由扩展报头之前，另一次在上层协议数据报文之前），其余扩展报头只能出现一次。

IPv6地址类型

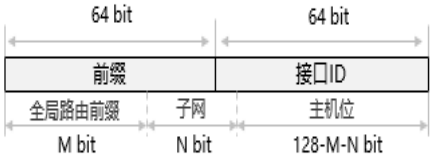
- 单播地址（Unicast Address）：标识一个接口，目的地址为单播地址的报文会被送到被标识的接口。在IPv6中，一个接口拥有多个IPv6地址是非常常见的现象。
- 组播地址（Multicast Address）：标识多个接口，目的地址为组播地址的报文会被送到被标识的所有接口。只有加入相应组播组的设备接口才会侦听发往该组播地址的报文。
- 任播地址（Anycast Address）：任播地址标识一组网络接口（通常属于不同的节点）。目标地址是任播地址的数据包将发送给其中路由意义上最近的一个网络接口。
- IPv6没有定义广播地址（Broadcast Address）。



- IPv4 地址分为：单播地址、组播地址、广播地址。而 IPv6 中没有广播地址，增加了任播地址。也就是说 IPv6 地址被分为：单播地址、组播地址、任播地址。
- 单播地址用于标识一个接口，发往该目的地址的报文会被送到被标识的接口；
- 组播地址用于标识多个接口，发往该目的地址的报文会被送到被标识的所有接口；
- 任播地址用于标识多个接口，发往该目的地址的报文会被送到被标识的所有接口中最近的一个接口上。实际上任播地址与单播地址使用同一个地址空间，也就是说，由路由器决定数据包是做任播转发还是单播转发。

IPv6单播地址 - 可聚合全球单播地址

- 全球单播地址定义用于IPv6 Internet。它们是全局唯一的和全局可路由的。
  - 类似IPv4公网地址。
  - 由前缀、子网ID和接口标识组成。

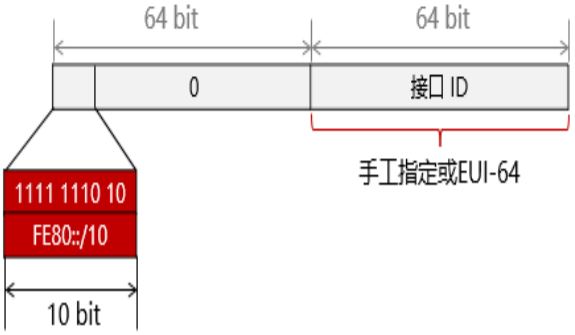


  - 全局路由前缀：由提供商指定给一个组织机构，一般至少为48bit。目前已经分配的全局路由前缀的前3bit均为001。因此前缀为2000::/3。
  - 子网：组织机构可以用子网ID来构建本地网络（Site），与IPv4中的子网号作用相似。子网ID通常最多分配到第64位。
  - 主机位：用来标识一个设备（Host），与IPv4中的主机ID作用相似。

- 全球单播地址是带有全球单播前缀的 IPv6 地址，其作用类似于 IPv4 中的公网地址。这种类型的地址允许路由前缀的聚合，从而限制了全球路由表项的数量。

IPv6单播地址 - 链路本地地址

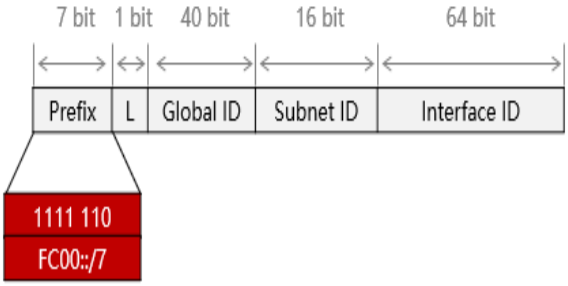
- 在一个节点启动IPv6协议栈时，节点的每个接口会自动配置一个链路本地地址。该地址专门用来和相同链路上的其他主机通信。
  - 只能在连接到同一本地链路的节点之间使用，广泛应用于邻居发现、无状态地址等。
  - 链路本地地址前缀FE80::/10，将接口ID添加在后面作为地址的低64位。
  - 每一个IPv6接口都必须具备一个链路本地地址。



- 链路本地地址是 IPv6 中的应用范围受限制的地址类型，只能在连接到同一本地链路的节点之间使用。它使用了特定的本地链路前缀 FE80::/10（最高 10 位值为 1111111010），同时将接口标识添加在后面作为地址的低 64 比特。
- 当一个节点启动 IPv6 协议栈时，启动时节点的每个接口会自动配置一个链路本地地址（其固定的前缀+EUI-64 规则形成的接口标识）。这种机制使得两个连接到同一链路的 IPv6 节点不需要做任何配置就可以通信。所以链路本地地址广泛应用于邻居发现，无状态地址配置等应用。
- 以链路本地地址为源地址或目的地址的 IPv6 报文不会被路由设备转发到其他链路。

IPv6单播地址 - 唯一本地地址

- 为了代替站点本地地址的功能，又使这样的地址具有唯一性，避免产生像IPv4的私有地址泄漏到公网而造成的问题，RFC4193定义了唯一本地地址。
  - 唯一本地地址，概念上类似于IPv4中的私网地址，仅能够在本地网络使用，在IPv6 Internet上不可被路由。
  - 唯一本地地址固定前缀FC00::/7。它被分为两块，其中FC00::/8暂未定义，另一块是FD00::/8，其格式如下：



- 唯一本地地址是另一种应用范围受限的地址，它仅能在一个站点内使用。由于本地站点地址的废除（RFC3879），唯一本地地址被用来代替本地站点地址（RFC4193）。
- 唯一本地地址的作用类似于 IPv4 中的私网地址，任何没有申请到提供商分配的全球单播地址的组织机构都可以使用唯一本地地址。唯一本地地址只能在本地网络内部被路由转发而不会在全球网络中被路由转发。
- 字段解释：
- Prefix：前缀；固定为 FC00::/7。
- L：L 标志位；值为 1 代表该地址为在本地网络范围内使用的地址；值为 0 被保留，用于以后扩展。
- Global ID：全球唯一前缀；通过伪随机方式产生（RFC 4193）。
- Subnet ID：子网 ID；划分子网使用。

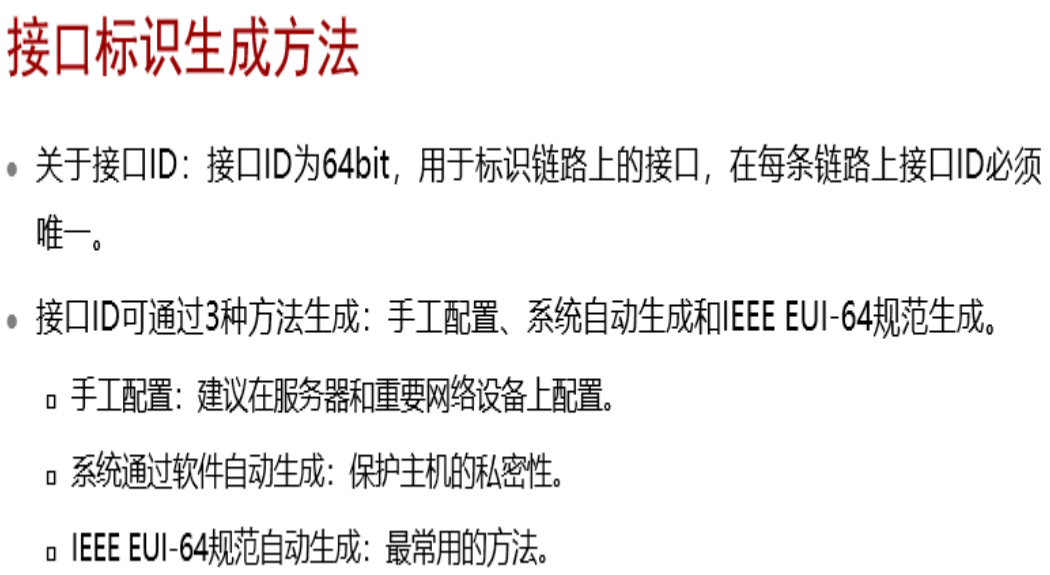
- Interface ID：接口标识。
- 唯一本地地址具有如下特点：
- 具有全球唯一的前缀（虽然随机方式产生，但是冲突概率很低）。
- 可以进行网络之间的私有连接，而不必担心地址冲突等问题。
- 具有知名前缀（FC00::/7），方便边缘路由器进行路由过滤。
- 如果出现路由泄漏，该地址不会和其他地址冲突，不会造成 Internet 路由冲突。
- 应用中，上层应用程序将这些地址看作全球单播地址对待。
- 独立于互联网服务提供商 ISP（Internet Service Provider）。

IPv6单播地址 - 特殊地址

- 未指定地址。
  - 0:0:0:0:0:0:0:0/128 或者::/128。
  - 该地址作为某些报文的源地址，比如作为重复地址检测时发送的邻居请求报文（NS）的源地址，或者DHCPv6初始化过程中客户端所发送的请求报文的源地址。
- 环回地址。
  - 0:0:0:0:0:0:0:1/128 或者::1/128。
  - 与IPv4中的127.0.0.1作用相同，用于本地回环，发往::/1的数据包实际上就是发给本地，可用于本地协议栈回环测试。
- IPv4兼容地址。
  - 在过渡技术中，为了让IPv4地址显得更加突出一些，定义了内嵌IPv4地址的IPv6地址格式。在这种表示方法中，IPv6地址的部分使用十六进制表示，IPv4地址部分可用十进制格式。
  - 该地址已经几乎不再使用。

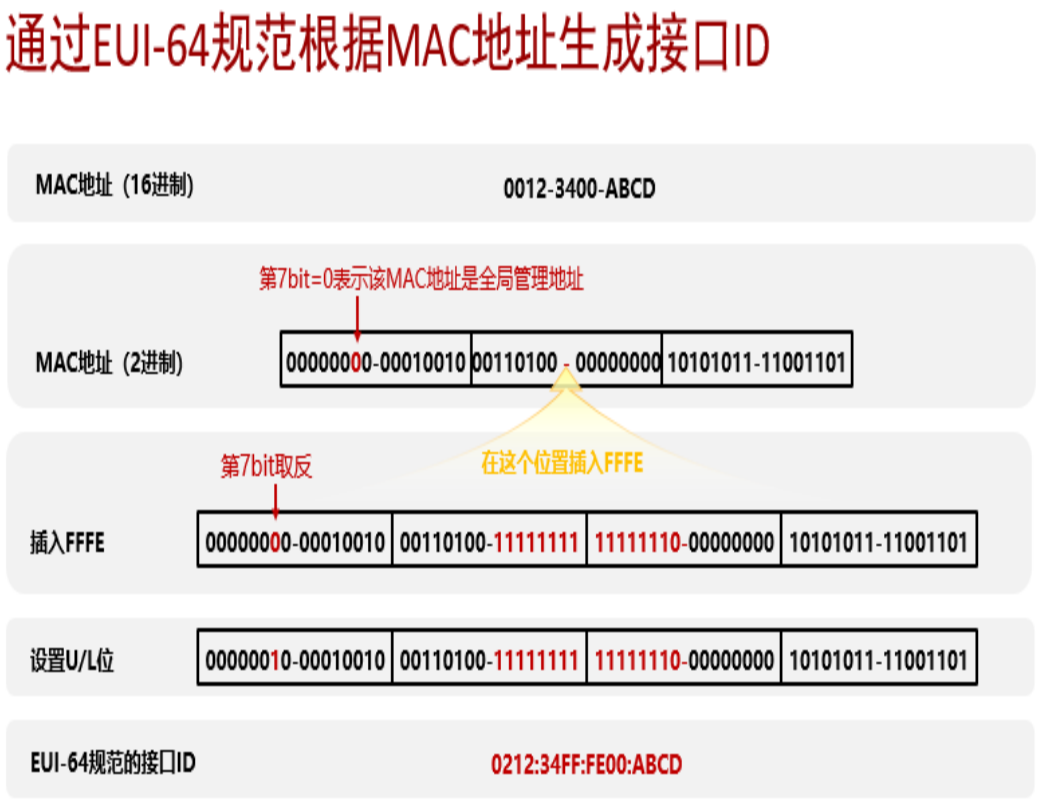
- 未指定地址

- IPv6 中的未指定地址即 0:0:0:0:0:0:0:0/128 或者::/128。该地址可以表示某个接口或者节点还没有 IP 地址，可以作为某些报文的源 IP 地址（例如在 NS 报文的重复地址检测中会出现）。源 IP 地址是::的报文不会被路由设备转发。
- 环回地址
- IPv6 中的环回地址即 0:0:0:0:0:0:0:1/128 或者::1/128。环回与 IPv4 中的 127.0.0.1 作用相同，主要用于设备给自己发送报文。该地址通常用来作为一个虚接口的地址（如 Loopback 接口）。实际发送的数据包中不能使用环回地址作为源 IP 地址或者目的 IP 地址。



- 对于 IPv6 单播地址来说，如果地址的前三 bit 不是 000，则接口标识必须为 64 位，如果地址的前三位是 000，则没有此限制。
- 接口 ID 的长度为 64bit，用于标识链路上的接口。在每条链路上，接口 ID 必须唯一。接口 ID 有许多用途，最常见的用于就是黏贴在链路本地地址前缀后面，形成接口的链路本地地址。或者在无状态自动配置中，黏贴在获取到的 IPv6 全局单播地址前缀后面，构成接口的全局单播地址。
- IEEE EUI-64 规范

- 这种由 MAC 地址产生 IPv6 地址接口 ID 的方法可以减少配置的工作量，尤其是当采用无状态地址自动配置时（后面会介绍），只需要获取一个 IPv6 前缀就可以与接口 ID 形成 IPv6 地址。
- 使用这种方式最大的缺点就是某些恶意者可以通过二层 MAC 推算出三层 IPv6 地址。



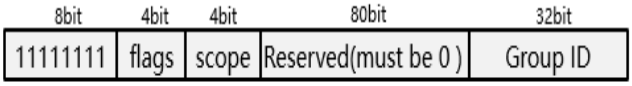
- 假设一个接口的 MAC 地址如上图所示，那么采用 EUI-64 规范，接口可根据该 MAC 地址计算得到接口 ID，由于 MAC 地址全局唯一，因此该接口 ID 也相应的具备全局唯一性。计算过程如下。
- 将 48bit 的 MAC 地址对半劈开，然后插入“FFFE”，再对从左数起的第 7 位，也就是 U/L 位取反，即可得到对应的接口 ID。
- 在单播 MAC 地址中，第 1 个 Byte 的第 7bit 是 U/L（Universal/Local，也称为 G/L，其中 G 表示 Global）位，用于表示

MAC 地址的唯一性。如果 U/L=0，则该 MAC 地址是全局管理地址，是由拥有 OUI 的厂商所分配的 MAC 地址；如果 U/L=1，则是本地管理地址，是网络管理员基于业务目的自定义的 MAC 地址。

- 而在在 EUI-64 接口 ID 中，第 7bit 的含义与 MAC 地址正好相反，0 表示本地管理，1 表示全球管理，所以使用 EUI-64 格式的接口 ID，U/L 位为 1，则地址是全球唯一的，如果为 0，则为本地唯一。这就是为什么要反转该位。

IPv6组播地址

- 用来标识一组接口，发往组播地址的数据将被转发给侦听该地址的多个设备。
- 地址范围：FF00::/8。



- **Flags**
  - 用来表示永久或临时组播组
  - 0000表示 永久分配或众所周知
  - 0001表示 临时的
- **Scope**
  - 表示组播组的范围
- **Group ID**
  - 组播组ID

0：预留
1：节点本地范围
2：链路本地范围，例如FF02::1
5：站点本地范围
8：组织本地范围
E：全球范围
F：预留

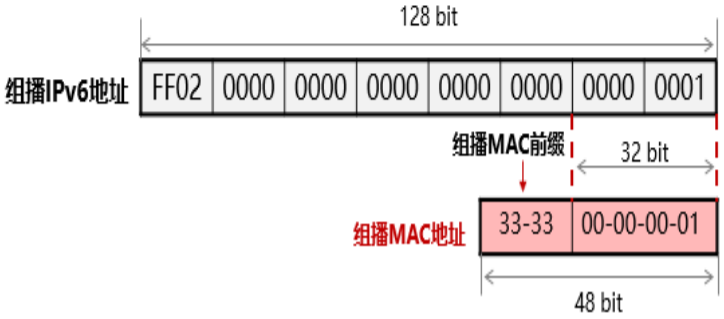
IPv6地址分类 - 预定义组播地址

- Node-local
  - FF01:0:0:0:0:0:0:1，所有节点的组播地址。
  - FF01:0:0:0:0:0:0:2，所有路由器的组播地址。
- Link-local
  - FF02:0:0:0:0:0:0:1，所有节点的组播地址。
  - FF02:0:0:0:0:0:0:2，所有路由器的组播地址。
  - FF02:0:0:0:0:1:FFXX:XXXX，Solicited-Node组播地址。
  - FF02:0:0:0:0:0:0:5，所有OSPF路由器组播地址。
  - FF02:0:0:0:0:0:0:6，所有OSPF的DR路由器组播地址。
  - FF02:0:0:0:0:0:0:D，所有PIM路由器组播地址。

- 类似于 IPv4，IPv6 同样有一些特殊的组播地址，这些地址由特别的含义，这里举几个例子（还有很多类似的特殊地址）：
- FF01::1（节点本地范围组播地址）
- FF02::1（链路本地范围所有节点组播地址）
- FF01::2（节点本地范围所有路由器组播地址）
- FF02::2（链路本地范围所有路由器组播地址）
- FF05::2（站点本地范围所有路由器组播地址）

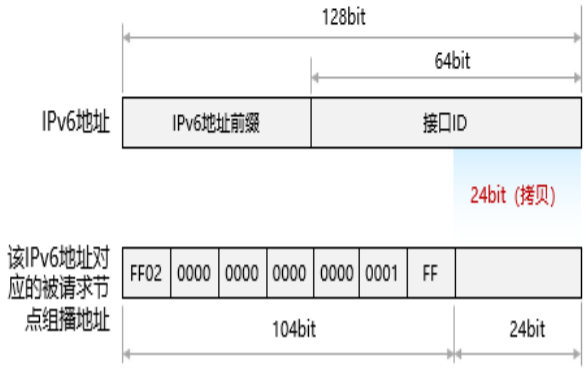
IPv6组播地址的MAC地址映射

- 在以太网环境中，一个组播IPv6报文必须执行以太网封装。
- 组播IPv6报文的目的IP地址是组播IPv6地址，而目的MAC地址则必须是组播MAC地址，并且该地址必须与组播IPv6地址对应。
- 33-33是专门为IPv6组播预留的MAC地址前缀，MAC地址的后32bit从对应的组播IPv6地址的后32bit拷贝而来。



被请求节点组播地址

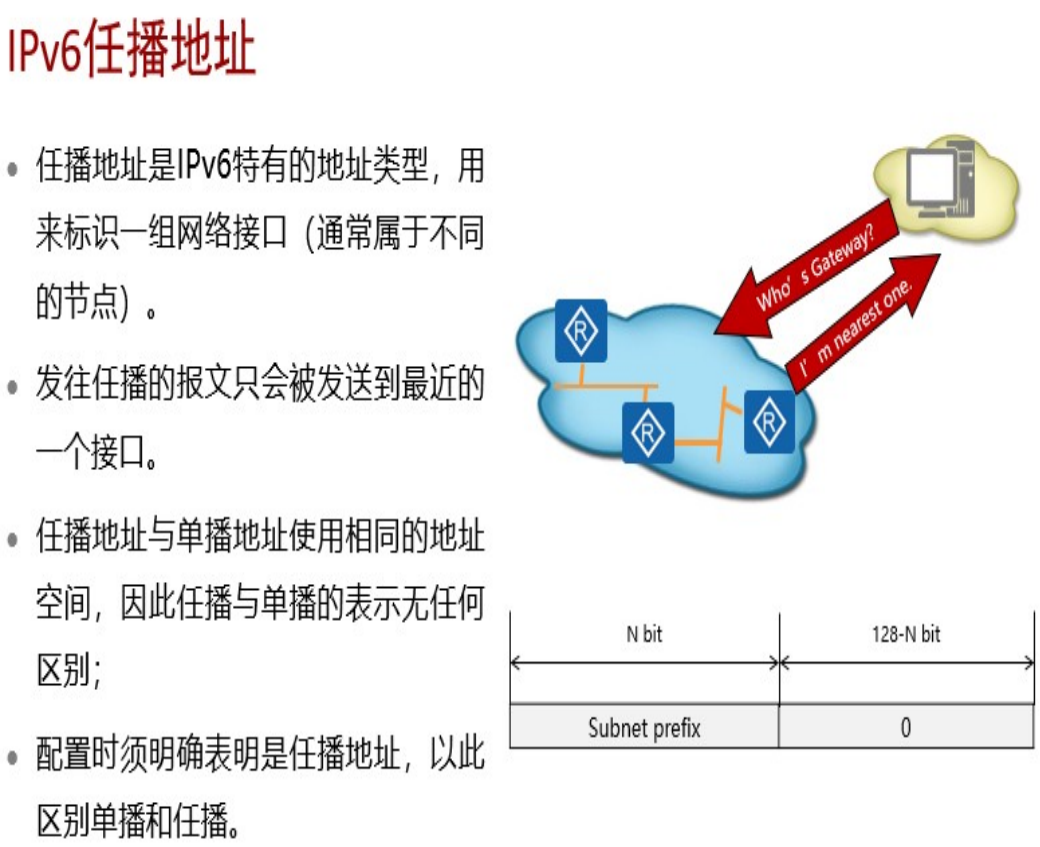
- 被请求节点组播地址（Solicited-Node Multicast Address）通过节点的单播或任播地址生成。当一个节点具有了单播或任播地址，就会对应生成一个被请求节点组播地址，并且加入这个组播组。
- 一个单播地址或任播地址对应一个被请求节点组播地址。该地址主要用于邻居发现机制和地址重复检测功能。
- 被请求节点组播地址由固定前缀FF02::1:FF00:0/104和对应IPv6地址的最后24bit组成。被请求节点组播地址的有效范围为本地链路范围。



- 当一个节点具有了单播或任播地址，就会对应生成一个与之相对应的被请求节点组播地址，并且加入这个组播组。一个单播地址或任播地址对应一个被请求节点组播地址。该地址主要用于地址解析、邻居发现机制和地址重复检测等功能。
- 被请求节点组播地址由固定前缀 FF02::1:FF00:0/104 和对应 IPv6 地址的最后 24bit 组成。被请求节点组播地址的有效范围为本地链路范围。
- 被请求节点组播地址的作用究竟是什么呢？举个非常简单的例子，回顾一下 IPv4 中的 ARP，这个协议主要用于地址解析，当设备需要解析某个 IP 地址对应的 MAC 地址时，就会发送一个广播 ARP Request 帧，之所以要发送广播帧，是因为它要确保广播域内所有节点都能收到。然而除了目标节点之外，该帧对于其他节点而言是个困扰，因为它们不得不去解析这个帧（一直解析到 ARP 载荷），这个动作将会浪费设备的

资源。

- 在 IPv6 中，ARP 及广播都被取消，当设备需要请求某个 IPv6 地址对应的 MAC 地址时，设备依然需要发送请求报文，但是该报文是一个组播报文，其目的 IPv6 地址是目标 IPv6 单播地址对应的被请求节点组播地址，而目的 MAC 地址则是该组播地址对应的组播 MAC 地址。由于只有目标节点才会侦听这个被请求节点组播地址，因此当其他设备收到该帧时，这些设备可以通过目的 MAC 地址、在网卡层面就判断出不需要处理它并将帧丢弃。



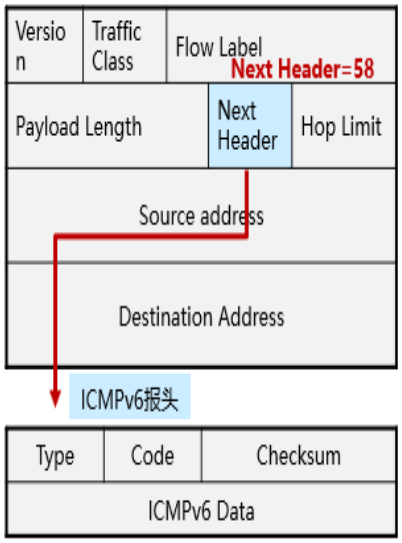
- 这是 IPv6 特有的地址类型，它用来标识一组网络接口（通常属于不同的节点）。目标地址是任播地址的数据包将发送给其中路由意义上最近的一个网络接口。适合于“One-to-One-of-Many”（一对组中的一个）的通讯场合。接收方只需要是一组接口中的一个即可，如移动用户上网就需要因地理位置

的不同，而接入离用户最近的一个接收站，这样才可以使移动用户在地理位置上不受太多的限制。

- 任播地址从单播地址空间中进行分配，使用单播地址的任何格式。因而，从语法上，任播地址与单播地址没有区别。被分配具有任播地址的节点必须得到明确的配置，从而知道它是一个任播地址。目前，任播地址仅被用做目标地址，且仅分配给路由器。
- 在 RFC3513 中定义了子网路由器任播地址（Subnet-Router anycast Address），其接口 ID 为全 0。
- 发往该任播地址的报文会被发送到任播地址所代表子网（子网路由器任播地址的前缀）内的某一台路由器，该路由器是离得最“近”的一台。所谓最近一般是路由的概念。

ICMPv6概述

- ICMPv6是IPv6的基础协议之一，用于向源节点传递报文转发的信息或者错误。
- 协议类型号（即IPv6 Next Header）为58。
- 在IPv6中，ICMPv6除了提供ICMPv4的对应功能之外，还有其它一些功能的基础，如邻居发现、无状态地址配置、重复地址检测、PMTU发现等。



- ICMPv6 的协议类型号（即 IPv6 报文中的 Next Header 字段的值）为 58。
- 在 IPv4 中，Internet 控制报文协议（ICMP）向源节点报

告关于向目的地传输 IP 数据包过程中的错误和信息。它为诊断、信息和管理目的定义了一些消息，如：目的不可达、数据包超长、超时、回应请求和回应应答等。在 IPv6 中，ICMPv6 除了提供 ICMPv4 常用的功能之外，还有其它的一些机制需要 ICMPv6 消息，诸如邻居发现、无状态地址配置（包括重复地址检测）、路径 MTU 发现等等。

- 所以 ICMPv6 是一个非常重要的协议。它是理解 IPv6 中其它机制的基础。

- 报文解释：
- Type：表明消息的类型，0 至 127 表示差错报文类型，128 至 255 表示消息报文类型。
- Code：表示此消息类型细分的类型。
- Checksum：表示 ICMPv6 报文的校验和。

ICMPv6

消息类型	TYPE	名称	CODE
差错消息	1	目的不可达	0 无路由
			1 因管理原因禁止访问
			2 未指定
			3 地址不可达
			4 端口不可达
	2	数据包过长	0
	3	超时	0 跳数到0
			1 分片重组超时
	4	参数错误	0 错误的包头字段
			1 无法识别的下一包头类型
			2 无法识别的ipv6选项
信息消息	128	Echo request	0
	129	Echo reply	0

- 目标不可达（Destination Unreachable）。
- 当数据包无法被转发到目标节点或上层协议时，路由器或目标节点发送 ICMPv6 目标不可达差错报文。在目标不可达报文中，类型（Type）字段值为 1，代码（Code）字段值为 0-4，每一个代码值都定义了具体含义（RFC2463）：
- 0：没有到达目标的路由。
- 1：与目标的通信被管理策略禁止。
- 2：未指定。
- 3：地址不可达。
- 4：端口不可达。
- 数据包超长（Packet Too Big）。
- 如果由于出口链路的 MTU 小于 IPv6 数据包的长度而导致数据包无法转发，路由器就会发送数据包超长报文。该报文被用于 IPv6 路径 MTU 发现的处理 。数据包超长报文的类型字段值为 2，代码字段值为 0。
- 超时（Time Exceeded）。
- 当路由器收到一个 IPv6 报头中的跳限制（Hop Limit）字段值为 0 的数据包时，会丢弃该数据包并向源发送 ICMPv6 超时报文。在超时报文中，类型字段的值为 3，代码字段的值为 0 或 1：
- 0：在传输中超越了跳限制。
- 1：分片重组超时。

- 参数问题（Parameter Problem）。
- 当 IPv6 报头或者扩展报头出现错误，导致数据包不能进一步处理时，IPv6 节点会丢弃该数据包并向源发送此报文，指明问题的位置和类型。参数问题报文中，类型字段值为 4，代码字段值为 0~2，32 位指针字段指出错误发生的位置。其中代码字段是这样定义的：
- 0：遇到错误的报头字段。

- 1：遇到无法识别的下一个报头（Next Header）类型。
- 2：遇到无法识别的 IPv6 选项。
- 在 RFC2463 中只定义了两种信息报文：回送请求 Echo Request 以及回送应答 Echo Reply。
- 回送请求报文。
- 回送请求报文用于发送到目标节点，以使目标节点立即发回一个回送应答报文。回送请求报文的类型字段值为 128，代码字段的值为 0。标志符（Identifier）和序列号（Sequence Number）字段有发送方主机设置，用于将即将收到的回送应答报文与发送的回送请求的报文进行匹配。
- 回送应答报文。
- 当收到一个回送请求报文时，ICMPv6 会用回送应答报文响应。回送应答报文的类型字段的值为 129，代码字段的值为 0。标志符（Identifier）和序列号（Sequence Number）字段的值被指为与回送请求报文中的相应字段一样的值。

IPv6邻居发现协议 - NDP概述

- NDP (Neighbor Discovery Protocol，邻居发现协议) 在RFC2462及RFC4861中定义。NDP实现了IPv6中诸多重要机制，如下图所示：

- 路由器发现：该功能帮助设备发现链路上的路由器，并获得路由器通告的信息。
- 无状态自动配置：无状态自动配置是 IPv6 的一个亮点功能，它使得 IPv6 主机能够非常便捷地连入到 IPv6 网络中，即插即用，无需手工配置繁冗的 IPv6 地址，无需部署应用服务器（例如 DHCP 服务器）为主机分发地址。无状态自动配置机制使用到了 ICMPv6 中的路由器请求报文（RS）及路由器通告报文（RA）。
- 重复地址检测：重复地址检测是一个非常重要的机制，一个 IPv6 地址必须经历重复地址检测并通过检测之后才能够启用。重复地址检测用于发现链路上是否存在 IPv6 地址冲突。
- 地址解析：在 IPv6 中，取消了 IPv4 中的 ARP 协议，使用 NDP 所定义的邻居请求报文（NS）及邻居通告报文（NA）来实现地址解析功能。

- 邻居的状态跟踪：IPv6 定义了节点之间邻居的状态机，同时还维护邻居 IPv6 地址与二层地址如 MAC 的映射关系，相应的表项存储于设备的 IPv6 邻居表中。
- 前缀重编址：IPv6 路由器能够通过 ICMPv6 的路由器通告报文（RA）向链路上通告 IPv6 前缀信息。通过这种方式，主机能够从 RA 中所包含的前缀信息自动构建自己的 IPv6 单播地址。当然这些自动获取的地址是有生存时间的。通过在 RA 中通告 IPv6 地址前缀，并且灵活地设定地址的生存时间，能够实现网络中 IPv6 新、老前缀的平滑过渡，而无需在主机终端上消耗大量的手工劳动重新配置地址。
- 路由器重定向：路由器向一个 IPv6 节点发送 ICMPv6 的重定向消息，通知它在相同的本地链路上有一个更好的、到达目的地的下一跳。IPv6 中的重定向功能与 IPv4 中的是一样的。

NDP使用ICMPv6的相关报文

ICMPv6报文 / 机制	RS 133	RA 134	NS 135	NA 136	重定向 137
地址解析			•	•	
前缀公告	•	•			
前缀重新编址	•	•			
DAD			•	•	
路由重定向					•

- RS（Router Solicitation）：路由器请求报文
- RA（Router Advertisement）：路由器通告报文
- NS（Neighbor Solicitation）：邻居请求报文
- NA（Neighbor Advertisement）：邻居通告报文

地址解析

- IPv6的地址解析不再使用ARP，也不再使用广播方式。
- 地址解析在三层完成，针对不同的链路层协议可以采用相同的地址解析协议
- 通过ICMPv6（类型135的NS及类型136的NA报文）来实现地址解析。
- NS报文发送使用组播的方式，报文的目的IPv6地址为被请求的IPv6地址对应的“被请求节点组播地址”，报文的目的MAC为组播MAC。
- 采用组播的方式发送NS消息相比于广播的方式更加的高效，也减少了对其他节点的影响和对二层网络的性能压力。
- 可以使用三层的安全机制（例如IPSec）避免地址解析攻击。

- 在 IPv4 中，可以通过 ARP 就可以由 IP 地址解析到链路层地址，ARP 协议是工作在第二层。在 IPv6 中在邻居发现协议（RFC2461）中定义地址解析的，其中使用了 ICMPv6 的报文，在三层完成地址解析，主要带来以下几个好处：
- 加强了介质独立性：这就意味着我们无需为每一个链路层定义一个新的地址解析协议，在每一个链路层都使用相同的地址解析协议；
- 可以利用三层安全机制：ARP 欺骗（如伪造 ARP 应答以盗窃数据流）是 IPv4 中的一个很大的安全问题，在第三层实现地址解析，可以利用三层标准的安全认证机制（例如 IPSEC）解决这个问题；
- ARP 请求报文使用广播，会泛滥到整个二层网络中每台主机是公认的一个 IPv4 性能问题。在第三层实现地址解析可以将地址解析请求仅仅发送到待解析地址所属的“Solicited-node”组播组即可。采用组播的传送方式，大大减轻了性能压力。

地址解析报文

- 地址解析过程中使用了两种ICMPv6报文：邻居请求（Neighbor Solicitation）和邻居通告（Neighbor Advertisement）。

Type	Code	checksum
Reserved		
Target Address		
Options...		

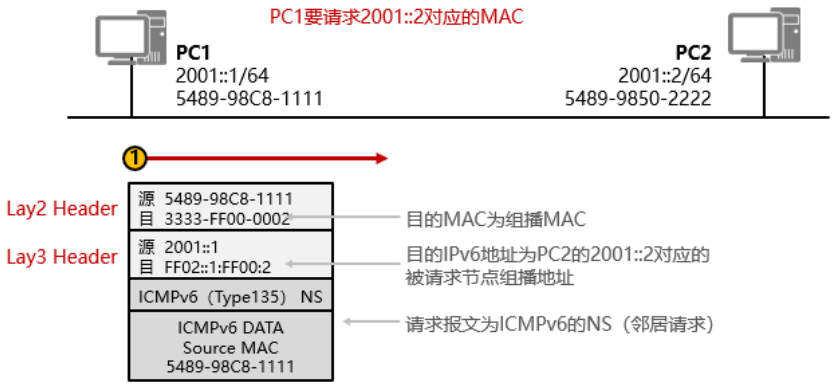
  - **邻居请求（Neighbor Solicitation，NS）。**
    - Type=135，Code=0。
    - Target Address是需要解析的IPv6地址，因此该处不准出现组播地址。
  - **邻居通告（Neighbor Advertisement，NA）。**
    - Type=136，Code=0
    - R标志（Router flag）表示发送者是否为路由器，如果1则表示是；
    - S标志（Solicited flag）表示发送邻居通告是否是响应某个邻居请求，如果1则表示是；
    - O标志（Overide flag）表示邻居通告中的消息是否覆盖已有的条目信息，如果1则表示是；
    - Target Address表示所携带的链路层地址对应的IPv6地址。

Type			Code	Checksum
R	S	O	Reserved	
Target Address				
Options...				

- 地址解析过程中使用了两种 ICMPv6 报文：邻居请求（Neighbor Solicitation）和邻居通告（Neighbor Advertisement）。
- 邻居请求 Neighbor Solicitation
- ICMP 的 Type 为 135，Code 为 0；
- Target Address 是需要解析的 IPv6 地址，因此该处不准出现组播地址。
- 邻居请求发送者的链路层地址会被放在 Options 字段中。
- 邻居通告 Neighbor Advertisement
- ICMP Type 为 136，Code 为 0；
- R 标志（Router flag）表示发送者是否为路由器，如果 1 则表示是；
- S 标志（Solicited flag）表示发送邻居通告是否是响应某个邻居请求，如果 1 则表示是；
- O 标志（Overide flag）表示邻居通告中的消息是否覆盖已有的条目信息，如果 1 则表示是；
- Traget Address 表示所携带的链路层地址对应的 IPv6 地址。

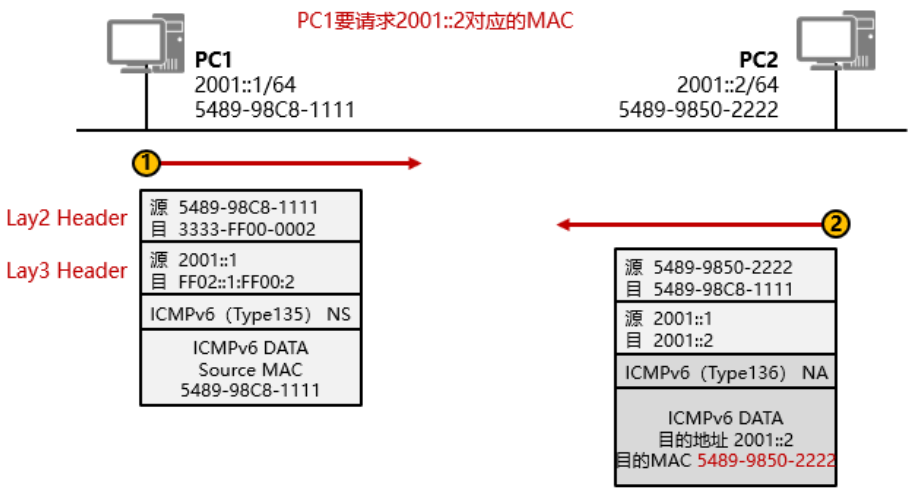
- 被请求的链路层地址被放在 Options 字段中，其格式仍然采用 TLV 格式，具体可以参考 RFC2463 。

地址解析 (1)



- 有了 NS 和 NA 两种报文，两台主机如何获取对方的链路层地址呢？
- 在上图所示的场景中，PC1 要请求 PC2 的 2001::2 这个地址对应的 MAC 地址，PC1 将发送一个 NS 报文达到这个目的。这个 NS 报文的源地址是 2001::1，目的地址则是 2001::2 对应的被请求节点组播地址。
- 然后 IPv6 数据包又被封装上数据帧的头部，其中源 MAC 地址是 PC1 的 MAC 地址，目的 MAC 地址则是 2001::2 这个目标地址对应的被请求节点组播地址映射得到的 MAC 地址，这是一个组播 MAC 地址。
- 这样就完成了一个双向交互链路层地址的过程。

地址解析 (2)



- 除 R2 外的其他节点也会收到这个数据帧，在读取数据帧头的时候发现目的 MAC 地址是一个组播 MAC 地址，而该组播 MAC 地址在本地并不侦听，因此在网卡层面就将数据帧丢弃而不再往报文里看了。
- PC2 收到这个数据帧后，由于本地网卡接收目的 MAC 地址为 3333-FF00-0002 的数据帧，因此在对数据帧做校验之后从帧头的类型字段得知里头是个 IPv6 报文，于是将帧头拆掉，把 IPv6 报文上送 IPv6 协议栈处理。IPv6 协议栈从报文的 IPv6 头部中的目的 IPv6 地址得知这个数据包是发往一个被请求节点组播地址 FF02::1:FF00:2，而本地网卡加入了这个组播组。接着，从 IPv6 包头的 NextHeader 字段得知 IPv6 包头后面封装着一个 ICMPv6 的报文，因此将 IPv6 包头拆除，将 ICMPv6 报文交给 ICMPv6 协议去处理。最后 ICMPv6 发现这是个 NS 报文，要请求自己 2001::2 对应的 MAC 地址，于是回送一个 NA 报文给 PC1，在该报文中就包含着 PC2 的 MAC 地址。

查看IPv6邻居路由表

- IPv6不像IPv4那样使用ARP表来缓存IP与MAC地址的映射，而是维护一个IPv6邻居表。在华为数通设备上则使用**display ipv6 neighbors**命令来查看IPv6邻居表。

```
[R1] display ipv6 neighbors
------------------------------------------------------------------------------
IPv6 Address    : 2012::2
Link-layer      : 00e0-fcc2-13b6              State : STALE
Interface       : GE0/0/0                     Age   : 0
VLAN            : -                           CEVLAN: -
VPN name        :                             Is Router: TRUE
Secure FLAG     : UN-SECURE

IPv6 Address    : FE80::2E0:FCFF:FEC2:13B6
Link-layer      : 00e0-fcc2-13b6              State : STALE
Interface       : GE0/0/0                     Age   : 0
VLAN            : -                           CEVLAN: -
VPN name        :                             Is Router: TRUE
Secure FLAG     : UN-SECURE
------------------------------------------------------------------------------
Total: 2      Dynamic: 2      Static: 0
```

- 在 windows7 操作系统里，可以使用 netsh interface ipv6 show neighbors 命令查看邻居缓存的内容。

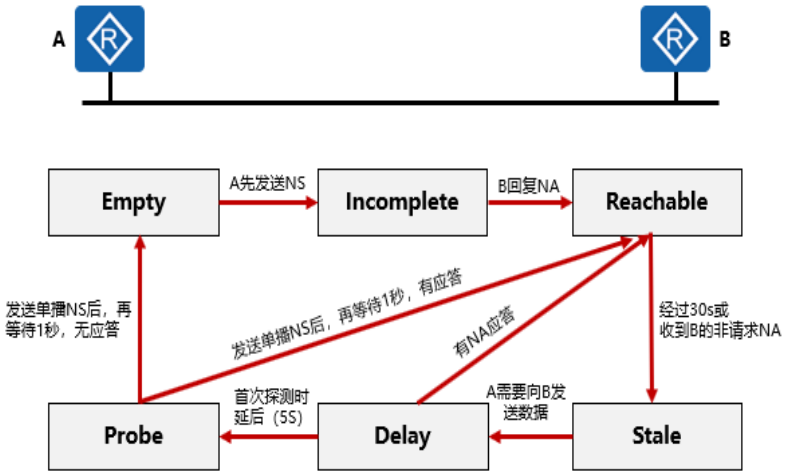
邻居状态种类

- 实际的通讯过程中不仅仅是地址解析这么简单，而是需要维护一张邻居表，每个邻居都有相应的状态，状态之间可以迁移。
- 邻居状态有5种：
  - INCOMPLETE 未完成，邻居请求已经发送到目标节点的请求组播地址，但没有收到邻居的通告；
  - REACHABLE 可达，收到确认，不续再发包确认；
  - STALE 陈旧，从收到上一次可达性确认后过了超过30s；
  - DELAY 延迟，在stale状态后发送过一个报文，并且5s内没有可达性确认；
  - PROBE 探查，每隔1s重传邻居请求来主动请求可达性确认，直到收到确认。

- 前面简单地讲述了如何进行地址解析的，但是实际的通讯过程中不仅仅是地址解析这么简单，而是需要维护一张邻居表，每个邻居都有相应的状态，状态之间可以迁移。
- RFC2461 中定义了 5 种状态：INCOMPLETE、REACHABLE、STALE、DELAY、PROBE。

邻居状态变化

- 一个例子：节点A要访问节点B，A的缓存中无B的条目，下图是邻居状态机的变化



- 邻居状态的迁移是比较复杂的，此处不会做详细地介绍，下面以 A、B 两个节点之间相互通讯过程的 A 节点的邻居状态变化，假设 A、B 两个节点之前没有任何通讯：
- A 先发送 NS，并生成邻居缓存条目，状态为 Incomplete；
- 若 B 回复 NA，则 Incomplete->Reachable，否则 10s 后 Incomplete->Empty，即删除条目；
- 经过 ReachableTime（默认 30s），条目状态 Reachable->Stale；
- 或者在 Reachable 状态，收到 B 的非请求 NA，且链路层地址不同，则马上->Stale；
- 在 Stale 状态若 A 需要向 B 发送数据，则 Stale->Delay，同时发送 NS 请求；
- 在 Delay_First_Probe_Time（默认 5 秒）后，Delay->Probe，其间若有 NA 应答，则 Delay->Reachable；
- 在 Probe 状态，每隔 RetransTimer（默认 1 秒）发送单播 NS，发送 MAX_UNICAST_SOLICIT 个后再等 RestransTimer，有应答则->Reachable，否则进入 Empty，即删除表项。
- 从以上的机制可以看出 IPv6 的邻居关系优于 IPv4 的 AR

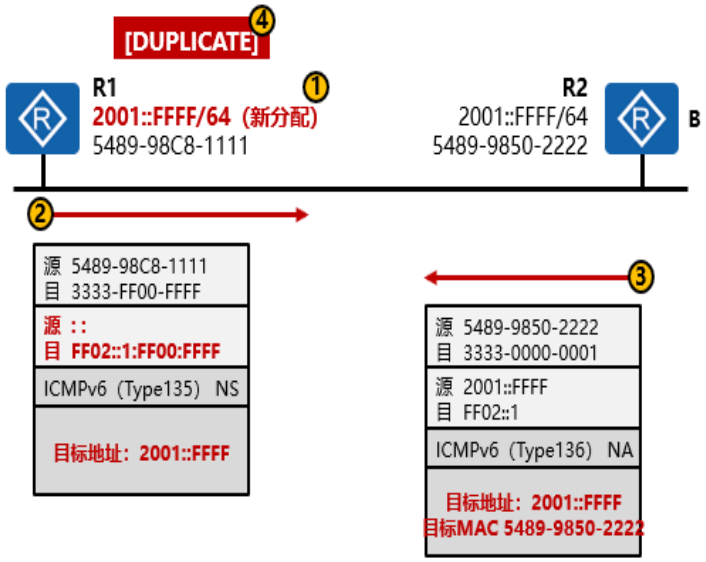
P，IPv6 的邻居关系维护机制确保通讯发起之前邻居是可达的，而 ARP 本身是做不到的，仅仅通过老化机制来实现。

- 关于邻居状态的维护以及状态迁移可以参考 RFC2461。

重复地址检测DAD

- 机制概述
  - 重复地址检测确保网络中无两个相同的单播地址。
  - 所有地址都需要做DAD。
  - 使用NS和NA完成DAD交互过程。
- 原理
  - 一个地址在通过DAD地址重复检测之前称为“tentative地址”也就是试验性地址。接口暂时还不能使用这个试验性地址进行正常的IPv6单播通讯，但是会加入和该地址所对应的Solicited-Node组播组。
  - DAD重复地址检测：节点向该tentative地址所在的Solicited-Node组播地址发送一个NS，如果收到某个其他站点回应的NA，就证明该地址已被网络上使用，节点将不能使用该tentative地址通讯。
  - 接口在启用任何一个单播IPv6地址前都需要先进行DAD，包括Link-Local地址。

DAD过程



- 在上图中，R2 已是在线的设备，该设备已经使用了如图所示的地址，现在我们为 R1 新配置 IPv6 的地址 2001::FFFF/64，观察一下会发生什么事情。R1 的接口配置 2001::FFFF/64 地址后，该地址立即进入 tentative 状态，此时仍然是不可用

的，除非该地址通过 DAD 检测。

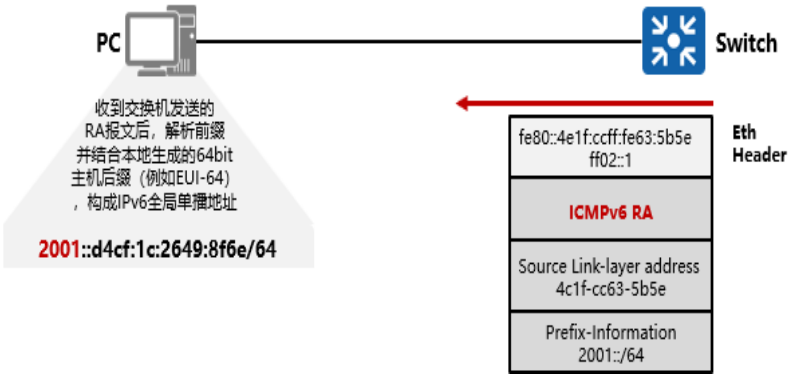
- R1 向链路上以组播的方式发送一个 NS 报文，该 NS 的源 IPv6 地址为“::”，目的 IPv6 地址为要进行 DAD 检测的 2001::FFFF 对应的被请求节点组播地址，也就是 FF02::1:FF00:FFFF。这个 NS 里包含着要做 DAD 检测的目标地址 2001::FFFF。
- 链路上的节点都会收到这个组播的 NS 报文，没有配置 2001::FFFF 的节点接口由于没有加入该地址对应的被请求节点组播组，因此在收到这个 NS 的时候默默丢弃。而 R2 在收到这个 NS 后，由于它的接口配置了 2001::FFFF 地址，因此接口会加入组播组 FF02::1:FF00:FFFF，而此刻所收到的报文又是以该地址为目的地址，因此它会解析该报文，它发现对方进行 DAD 的目标地址与自己本地接口地址相同，于是立即回送一个 NA 报文，该报文的目的地址是 FF02::1，也就是所有节点组播地址，同时在报文内写入目标地址 2001::FFFF，以及自己接口的 MAC 地址。
- 当 R1 收到这个 NA 后，它就知道 2001::FFFF 在链路上已经有人在用了，因此将该地址标记为 Duplicate（重复的），该地址将不能用于通信。

在R1上可以观察到的DAD过程


```
[R1] display ipv6 interface
GigabitEthernet0/0/0 current state : UP
IPv6 protocol current state : UP
IPv6 is enabled, link-local address is FE80::5489:98FF:FEC8:1111
  Global unicast address(es):
    2001::FFFF, subnet is 2001::/64 [DUPLICATE]
  Joined group address(es):
    FF02::1:FF00:FFFF
    FF02::1:FFC8:1111
    FF02::2
    FF02::1
  MTU is 1500 bytes
  ND DAD is enabled, number of DAD attempts: 1
  ......
```

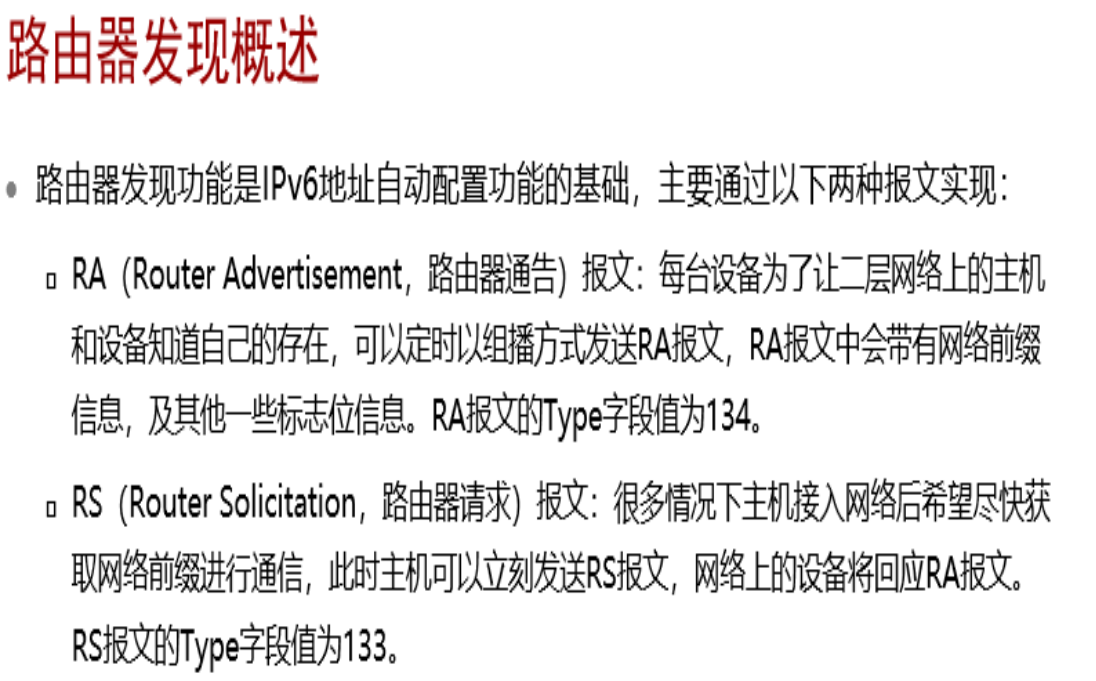
IPv6地址无状态自动配置概述

- IPv6地址无状态自动配置（StateLess Address AutoConfiguration，SLAAC）是IPv6的标准功能，在RFC2462中定义。
- 在IPv6中，设备可以通过手工或者动态的方式获取地址。在动态获取地址的方式中，存在DHCPv6及无状态地址自动配置两种方式。
- 相比于DHCPv6这种动态地址分配技术而言，SLAAC无需部署应用服务器，更加轻量。



- 使用 IPv6 地址无状态自动配置后，设备的 IPv6 地址无需进行手工配置，即插即用，减轻网络管理的负担。
- 大致的工作过程如下：
- 主机根据本地接口 ID 自动产生网卡的链路本地地址。

- 主机对链路本地地址进行 DAD 检测，如果该地址不存在冲突则可以启用。
- 主机发送 RS 报文尝试在链路上发现 IPv6 路由器，该报文的源地址为主机的链路本地地址。
- 路由器回复 RA 报文（携带 IPv6 前缀信息，路由器在未收到 RS 时也能够配置主动发出 RA 报文）。
- 主机根据路由器回应的 RA 报文，获得 IPv6 地址前缀信息，使用该地址前缀，加上本地产生的接口 ID，形成单播 IPv6 地址。
- 主机对生成的 IPv6 地址进行 DAD 检测，如果没有检测到冲突，那么该地址才能够启用。

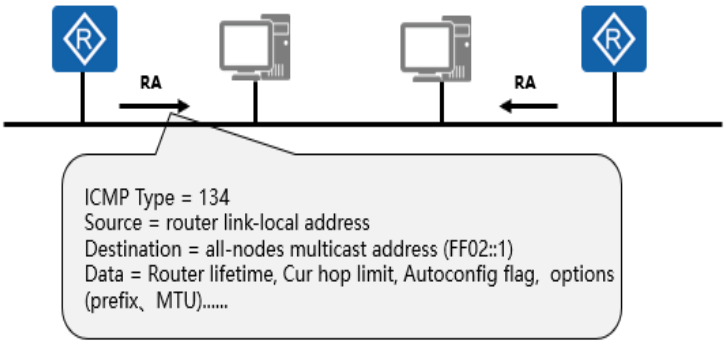


- 路由器发现功能用来发现与本地链路相连的设备，并获取与地址自动配置相关的前缀和其他配置参数。
- 经过前面的介绍，我们已经知道 IPv6 地址支持无状态自动配置，即主机通过路由器发送的 RA 报文获取网络前缀信息，然后主机自己生成地址的接口标识部分，并自动配置 IPv6 地

址。

路由器发现 - 路由器周期发送RA

- 链路上的路由器会定期的发送RA（Router Advertisement）消息。

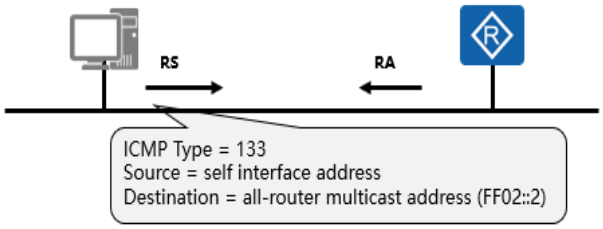


- 收到RA的主机将加入默认路由器列表中。
- 收到RA的路由器将检查RA内容的一致性。

- 主机如何获知网络的前缀（实际上不仅仅前缀前缀信息，还有其它的信息）呢？主要通过两个途径：被动接收到网络上路由器通告（Router Advertisement），从通告中获得；主动发送路由器请求（Router Solicitation），路由器回应路由器通告后，主机从通告中获得。

路由器发现 - 路由器回应RA

- 主机接口初始化时发RS（Router Solicitation）消息，路由器回应RA。



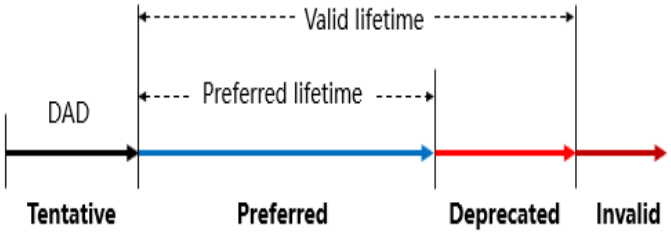
注：回复的RA可以直接单播给请求的主机，也可以选择多播到所有节点。

主机获得前缀及其它参数过程

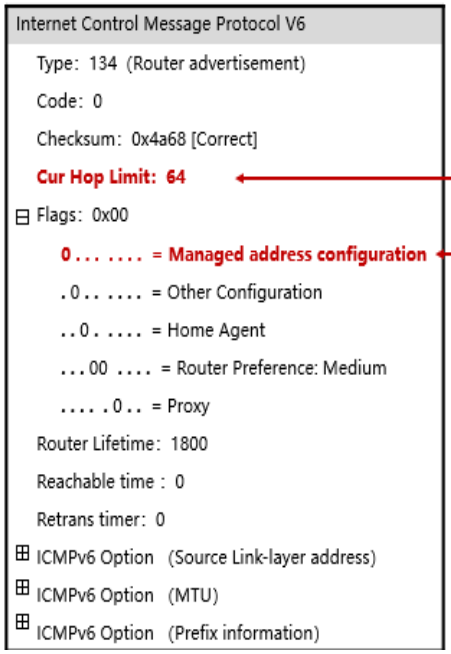
- 当存在以下情况时忽略RA发送的前缀：
  - RA报文选项中的“auto”未置位。
  - 前缀与已有地址前缀重复（包括link-local地址）。
  - RA报文选项中的“preferred lifetime”时间大于 “valid lifetime”。
  - 前缀长度与接口ID长度之和不等于128位。
- 除以上情况外，主机获得前缀同时也获得一些相关时间参数：
  - “preferred lifetime” =发起新通讯的有效时间。
  - “valid lifetime” =原有通讯的有效时间。
- 主机会周期性的收到RA报文，并据此报文来更新自己的时间参数。

几个生存时间

- 当地址处于Deprecated状态，地址不能主动的发起连接只能是被动的接受连接，这也是为了保证上层应用而设计的，但是过了valid lifetime时间地址就变为invalid，这时任何连接就会down掉。



ICMPv6 RA消息中的Flags字段 (1)

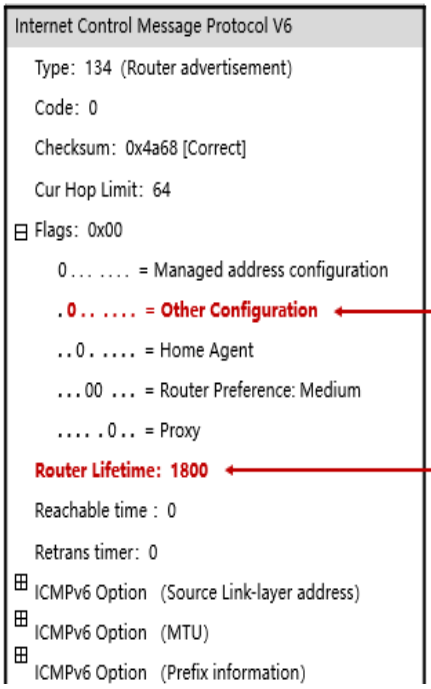


该字段用于帮助主机完成跳数限制。当PC使用该RA通告的前缀构建IPv6地址后，该PC发送的IPv6报文的跳数限制被设置为该值（64）。

M位默认为0，为0时，收到该RA的主机使用RA中包含的IPv6前缀用于无状态地址自动配置。

使用如下命令，可将该值设置为1。
ipv6 nd autoconfig managed-address-flag
当该值为1时，收到该RA的主机将采用有状态自动配置，也就是DHCPv6的方式来获取IPv6地址。

ICMPv6 RA消息中的Flags字段 (2)

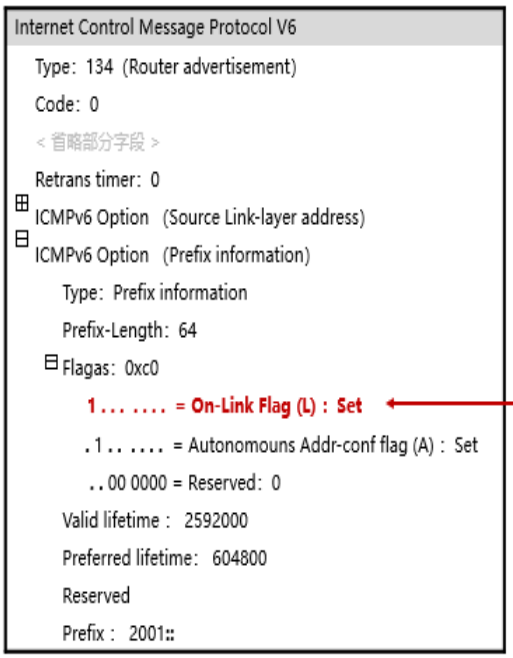


Other-Config-Flag，默认为0，表示主机不应该使用有状态自动配置机制来配置除了IPv6地址外的其他参数。

使用命令：
ipv6 nd autoconfig other-flag
将该值置1，则主机需使用DHCPv6来配置除了IPv6地址外的其他信息，如DNS，域名等。

单位是秒，主机将路由器视为缺省路由器的时间。该计时器到计数为0时，该路由器将不会出现在主机的缺省网关列表中。

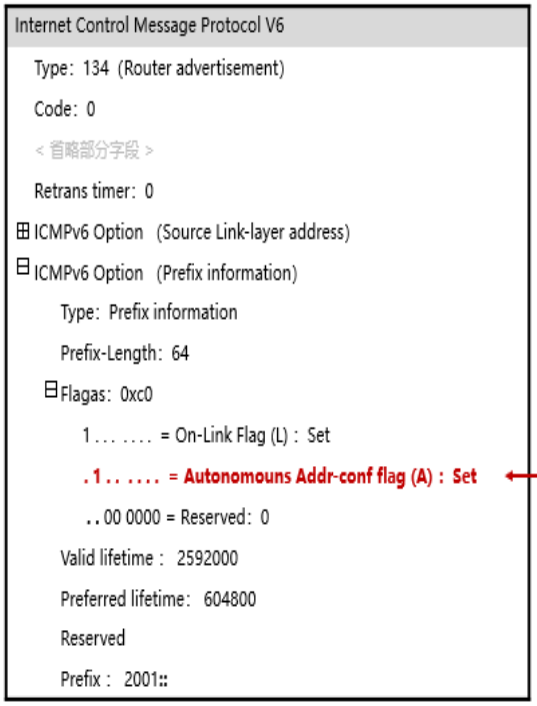
ICMPv6 RA消息中IPv6前缀信息的Flags字段 (1)



L比特位（RFC2461），默认为1。
表示在RA消息中的前缀是分配给本地链路的。因此，向包含这个指定前缀的地址发送数据的节点认为目的地是本地链路可达。

可以使用如下命令设置为0：
ipv6 nd ra prefix 2001:: 64 2592000 604800 off-link

ICMPv6 RA消息中IPv6前缀信息的Flags字段 (2)

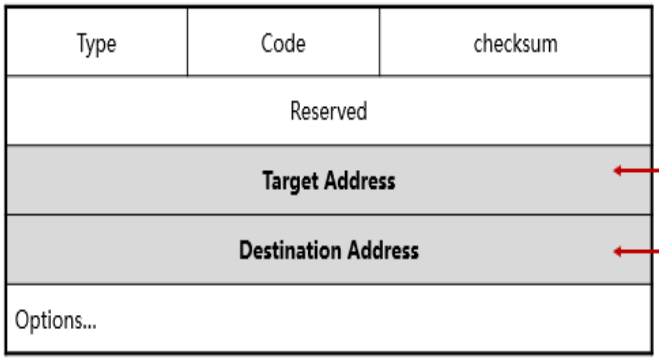


A比特位（RFC2461），默认为1，表示本地链路的主机可以使用该前缀进行无状态自动配置，如果为0，则不能用于无状态自动配置。

使用如下命令将该比特位设置为0：
ipv6 nd ra prefix 2001:: 64 2592000 604800 no-autoconfig

重定向报文

- 当网关路由器知道更好的转发路径时，会以重定向报文的方式告知主机
- 重定向报文的结构如下：



Target Address是更好的路径下一跳地址。

Destination Address是需要重定向转发的报文的目的地址。

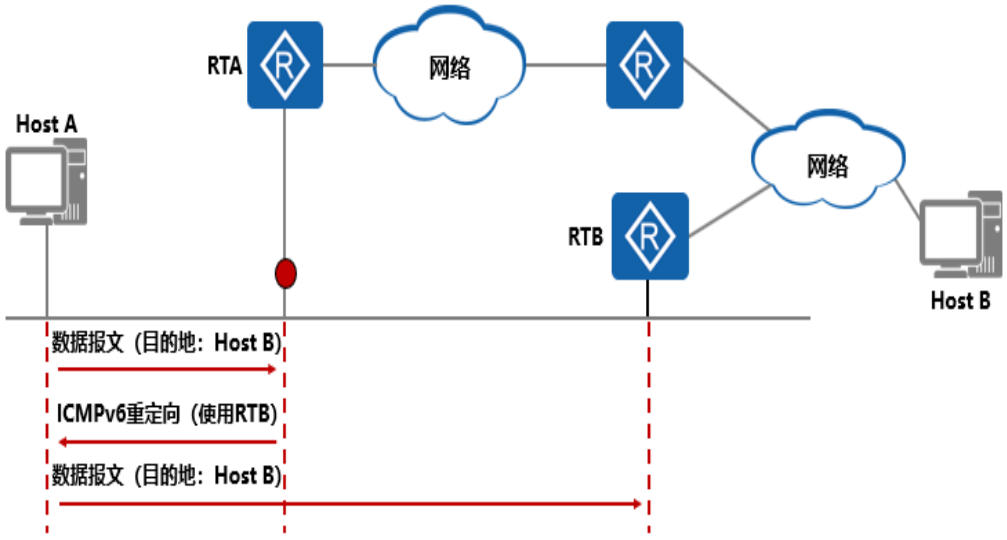
- 经常网关路由器发现报文从其它网关路由器转发更好，它就会发送重定向报文告知报文的发送者，让报文发送者选择

另一个网关路由器。

- 报文格式中 Type 为 137，Code 为 0；
- Target Address 是更好的路径下一跳地址；
- Destination Address 是需要重定向转发的报文的目的地址。

重定向过程

- 主机A的默认路由器为RTA，当主机A要给主机B发送数据时：

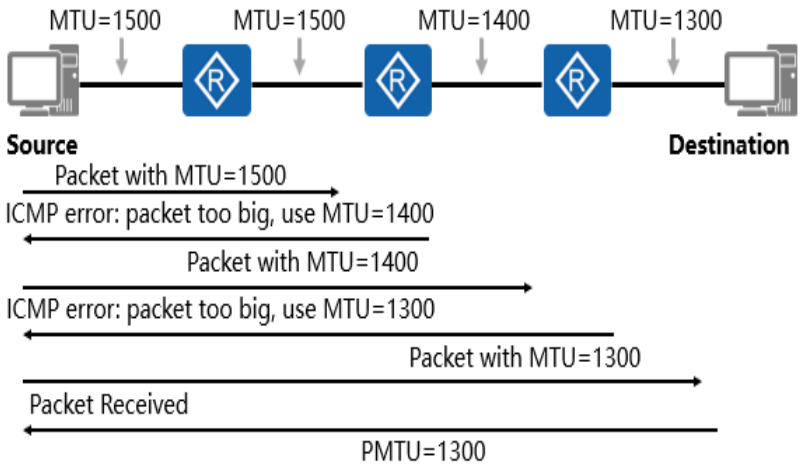


- 下面是一个具体的例子，假设主机 A 想与主机 B 通讯，主机 A 的默认网关路由器是 RTA，那么当 A 发送报文给 B 时报文会被送到 RTA。
- RTA 接收到 A 发送的报文以后会发现实际上主机 A 直接发送给路由器 R2 更好，它将发送一个 ICMPv6 重定向报文给主机 A，其中 Target Address 为 RTB，Destination Address 为主机 B。
- 主机 A 接收到了重定向报文之后，会在默认路由表中添加一个主机路由，以后发往主机 B 的报文就直接给 R2。
- 这就是重定向的一个简单过程，其中会有个问题：RTA

如何知道去往主机 B 的路径通过 RTB 更好呢？其实这个很简单，因为 RTA 会发现报文进入的接口就是报文路由得出接口，也就是说发往主机 B 的路由实际上只是在 RTA 上转了一圈出来了，然后转发到 RTB，据此，RTA 能判断出直接给 RTB 是更好的路径。

PMTU发现 (1)

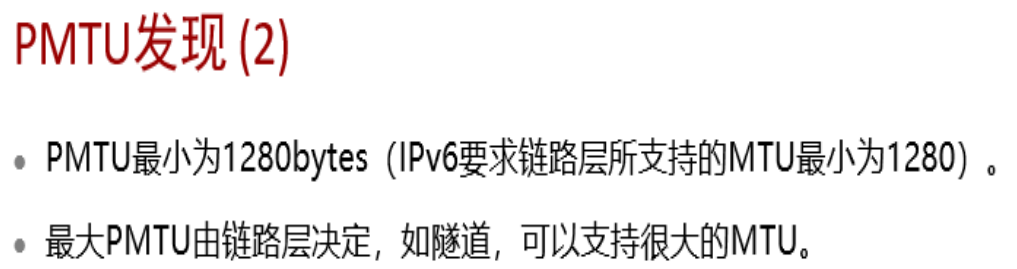
- PMTU就是路径上的最小接口MTU。
- 在RFC1981中定义了PMTU发现协议。



- 前面学习的关于 IPv6 报文转发相关知识的时候知道，IPv6 报文在转发的过程中是不进行分片操作的，当然也不进行分片报文的整合工作。IPv6 报文仅在源节点进行分片，在目的节点进行组装。那么这会产生一个问题，源节点将报文到底分成多大的呢？很简单，为了所有的报文都能在路径上畅通无阻，那么分片的报文大小不能超过路径上最小的 MTU，也就是 PMTU——路径 MTU。
- RFC1981 中定义了 PMTU 发现的机制，它是通过 ICMPv6 的 Packet Too Big 报文来完成的。首先源节点假设 PMTU 就是其出接口的 MTU，发出报文，当转发路径上存在一个小于当前假设的 PMTU 时，就会向源节点发送 Packet Too Big

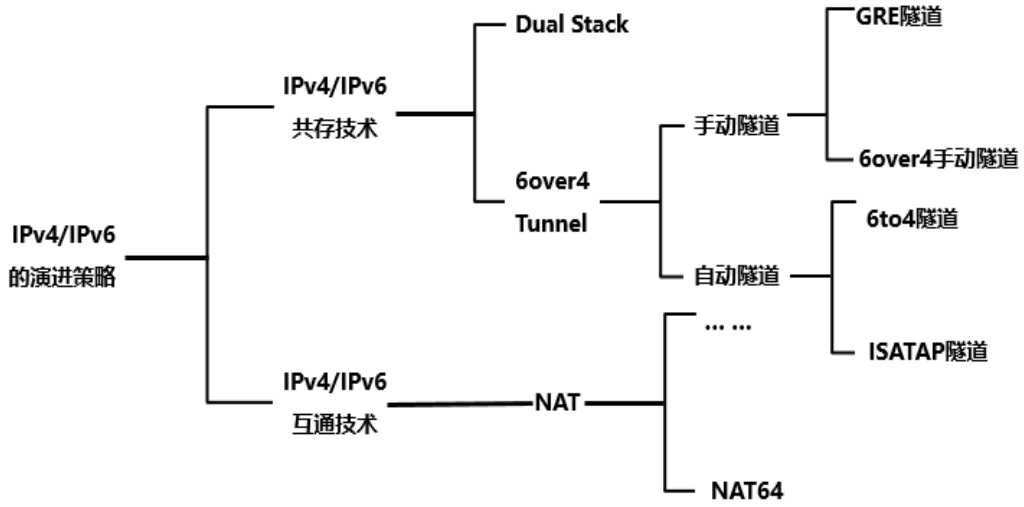
报文，并且携带自己的 MTU 值，此后源节点将 PMTU 的假设值更改为新收到的 MTU 值。如此反复，直到报文到达目的地之后，源节点就能知道到达某个目的地的 PMTU 了。

- 假设源到目的要先后经过 4 条链路，链路地 MTU 分别是 1500、1500、1400、1300，当源发送一个分片报文的时候，首先分成 1500 大小的片，当到达 1400 的出接口时，路由器就会返回 Packet Too Big 错误，同时携带 1400 的 MTU 值。源接收到之后就会重新分成 1400 大小的片，当到达 1300 的出接口时，同样返回 Packet Too Big 错误，携带 1300 的 MTU 值。之后源重新分成 1300 的报文，最终到达目的地，这样就找到了该路径的 PMTU。



- 值得注意的是，只有数据包超过路径上的最小 MTU 时，PMTU 发现机制才有意义，因为如果报文很小，小于路径最小的 MTU，就不可能产生 Packet Too Big 报文。
- 由于 IPv6 要求链路层所支持的最小 MTU 为 1280，所以 PMTU 的值不会小于 1280。而最大的 PMTU 一般由链路层决定，如果链路层是一个隧道，那么支持的 PMTU 可能很大。

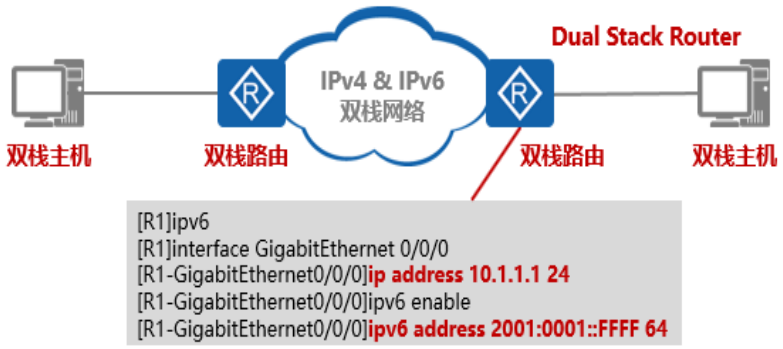
IPv6过渡技术简介



- IPv6 与 IPv4 共存技术：
- 双协议栈：
- IPv6 节点同时支持 IPv6 和 IPv4 协议栈。
- 隧道：
- IPv6 报文作为 IPv4 的载荷，由 IPv4 Internet 中连接多个 IPv6 孤岛。
- IPv6 与 IPv4 互通技术：
- 提供 IPv6 与 IPv4 互相访问的技术。
- 适用于 IPv6 Internet 与 IPv4 Internet 共存，而两者又有互相通讯的需求。

双栈Dual Stack

- 双栈协议
  - 设备必须支持IPv4/IPv6协议栈。
  - 连接双栈网络的接口必须同时配置IPv4地址和IPv6地址。



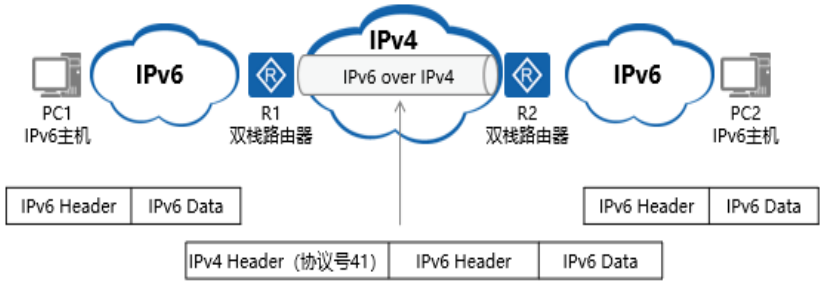
- 双栈技术是 IPv4 向 IPv6 过渡的一种有效的技术。网络中的节点同时支持 IPv4 和 IPv6 协议栈，源节点根据目的节点的不同选用不同的协议栈，而网络设备根据报文的协议类型选择不同的协议栈进行处理和转发。双栈可以在一个单一的设备上实现，也可以是一个双栈骨干网。对于双栈骨干网，其中的所有设备必须同时支持 IPv4/IPv6 协议栈，连接双栈网络的接口必须同时配置 IPv4 地址和 IPv6 地址。
- 所谓的双栈就是主机或者网络设备同时支持 IPv4 及 IPv6 双协议栈，如果节点支持双栈，那么它能够同时使用 V4 和 V6 的协议栈、同时处理 IPv4 及 IPv6 的数据。在双栈设备上，上层应用会优先选择 IPv6 协议栈，而不是 IPv4。 比如，一个同时支持 v4 和 v6 的应用请求通过 DNS 请求地址，会先请求 AAAA 记录，如果没有，则再请求 A 记录。双栈是 V4、V6 并存及 IPv6 过渡技术的基础。

就拿上图来说，路由器就是一个双栈设备，默认情况下路由器本身就已经支持 IPv4，接口上也配置了 IPv4 的地址，已经能够正常转发 IPv4 的报文，此刻在激活路由器的 IPv6 数据转发

能力，再为接口分配 IPv6 的单播地址，那么这个接口又有了 IPv6 数据转发能力。当然，此时对于路由器而言，IPv4 及 IPv6 协议栈互不干扰，独立工作。

6over4手动隧道

- 6over4手动隧道。
  - 6over4手动隧道的一种;
  - 源地址和目的地址均需手工指定;
  - 用于边界路由器与边界路由器，或者主机与边界路由器之间。

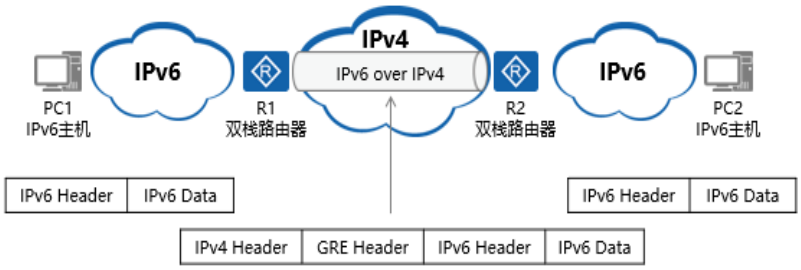


- 如果一个边界设备要与多个设备建立手动隧道，就需要在设备上配置多个隧道，配置比较麻烦。所以手动隧道通常用于两个边界路由器之间，为两个 IPv6 网络提供连接。
- 手动隧道优缺点
- 优点：可以用于任何 IPv6 穿越 IPv4 的环境，通用性好。
- 缺点：必须手工配置。
- 转发机制
- IPv6 over IPv4 手动隧道转发机制为：当隧道边界设备的 IPv6 侧收到一个 IPv6 报文后， 根据 IPv6 报文的目的地址查找 IPv6 路由转发表，如果该报文是从此虚拟隧道接口转发出去，则根据隧道接口配置的隧道源端和目的端的 IPv4 地址进行封装。封装后的报文变成一个 IPv4 报文，交给 IPv4 协议栈处理。报文通过 IPv4 网络转发到隧道的终点。隧道终点收到一个隧道协议报文后，进行隧道解封装。并将解封装后的报文

交给 IPv6 协议栈处理。

6over4 GRE隧道

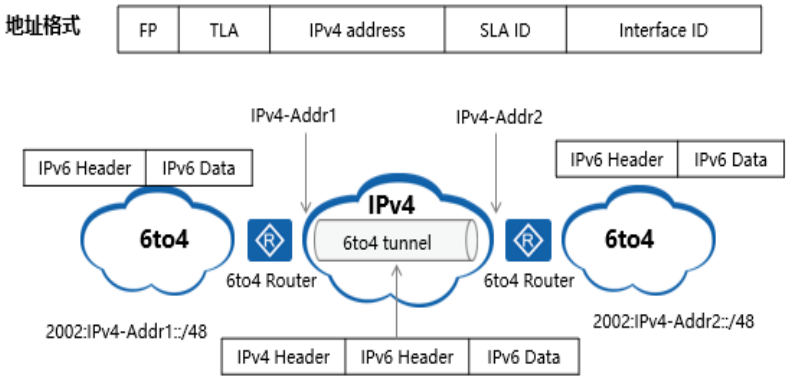
- 6over4 GRE隧道。
  - 6over4手动隧道的一种;
  - 手工指定隧道的端点地址;
  - GRE承载IPv6协议。



- IPv6 over IPv4 GRE 隧道使用标准的 GRE 隧道技术提供了点到点连接服务，需要手工指定隧道的端点地址。GRE 隧道本身并不限制被封装的协议和传输协议，一个 GRE 隧道中被封装的协议可以是协议中允许的任意协议（可以是 IPv4、IPv6、OSI、MPLS 等）。
- IPv6 over IPv4 GRE 隧道在边界路由器上的传输机制和 IPv6 over IPv4 手动隧道相同。

6to4隧道 (1)

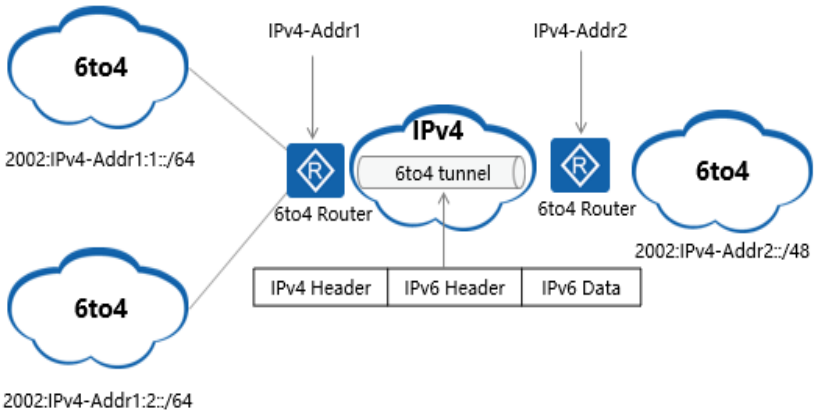
- 6to4隧道
  - 6over4自动隧道的一种。
  - 支持Router到Router、Host到Router、Router到Host、 Host到Host。
  - 采用6to4专用地址，即2002:IPv4::/48。



- 6to4 隧道也属于一种自动隧道，隧道也是使用内嵌在 IPv6 地址中的 IPv4 地址建立的。与 IPv4 兼容自动隧道不同，6to4 自动隧道支持 Router 到 Router、Host 到 Router、Router 到 Host、 Host 到 Host。
- 地址格式：
- FP：可聚合全球单播地址的格式前缀（Format Prefix），其值为 001。
- TLA：顶级聚合标识符（Top Level Aggregator），有 13 个比特位，其二进制值为 0 0000 0000 0010。
- SLA：站点级聚合标识符（Site Level Aggregator）。
- 6to4 地址可以表示为 2002::/16，而一个 6to4 网络可以表示为 2002:IPv4 地址::/48。6to4 地址的网络前缀长度为 64bit，其中前 48bit（2002: a.b.c.d）被分配给路由器上的 IPv4 地址决定了，用户不能改变，而后 16 位（SLA）是由用户自己定义的。

6to4隧道 (2)

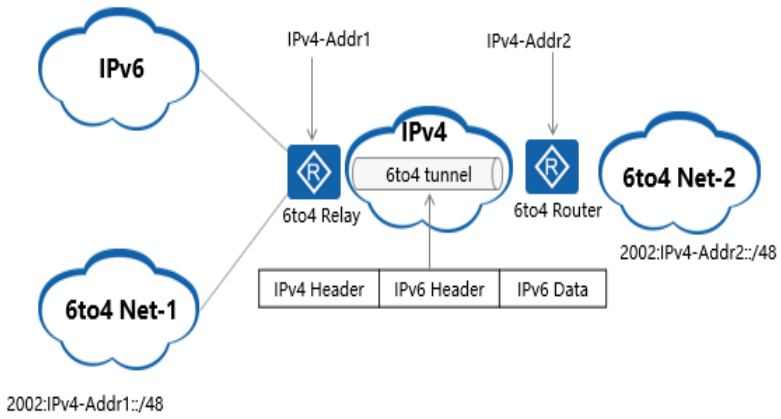
- 6to4隧道
  - 可连接多个6to4网络。
  - 通过SLA ID区分。



- 一个 IPv4 地址只能用于一个 6to4 隧道的源地址，如果一个边界路由器连接了多个 6to4 网络使用同样的 IPv4 地址做为隧道的源地址，则使用 6to 地址中的 SLA ID 来区分，但他们共用一个隧道。

6to4隧道 (3)

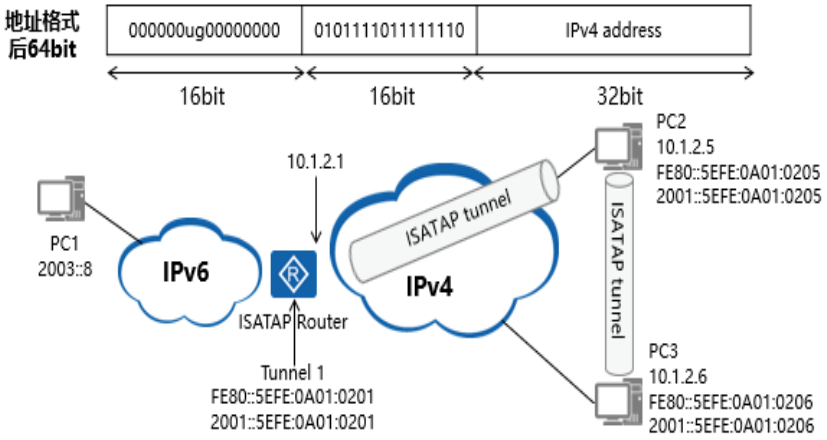
- 6to4中继
  - 实现6to4网络和IPv6普通网络互通。



- 普通 IPv6 网络需要与 6to4 网络通过 IPv4 网络互通，这可以通过 6to4 中继路由器方式实现。所谓 6to4 中继，就是通过 6to4 隧道转发的 IPv6 报文的目的地址不是 6to4 地址，但转发的下一跳是 6to4 地址，该下一跳为路由器我们称之为 6to4 中继。隧道的 IPv4 目的地址依然从下一跳的 6to4 地址中获得。
- 如果 6to4 网络 2 中的主机要与 IPv6 网络互通，在其边界路由器上配置路由指向的下一跳为 6to4 中继路由器的 6to4 地址，中继路由器的 6to4 地址是与中继路由器的 6to4 隧道的源地址相匹配的。6to4 网络 2 中去往普通 IPv6 网络的报文都会按照路由表指示的下一跳发送到 6to4 中继路由器。6to4 中继路由器再将此报文转发到纯 IPv6 网络中去。当报文返回时，6to4 中继路由器根据返回报文的目的地址（为 6to4 地址）进行 IPv4 报文头封装，数据就能够顺利到达 6to4 网络中了。

ISATAP隧道

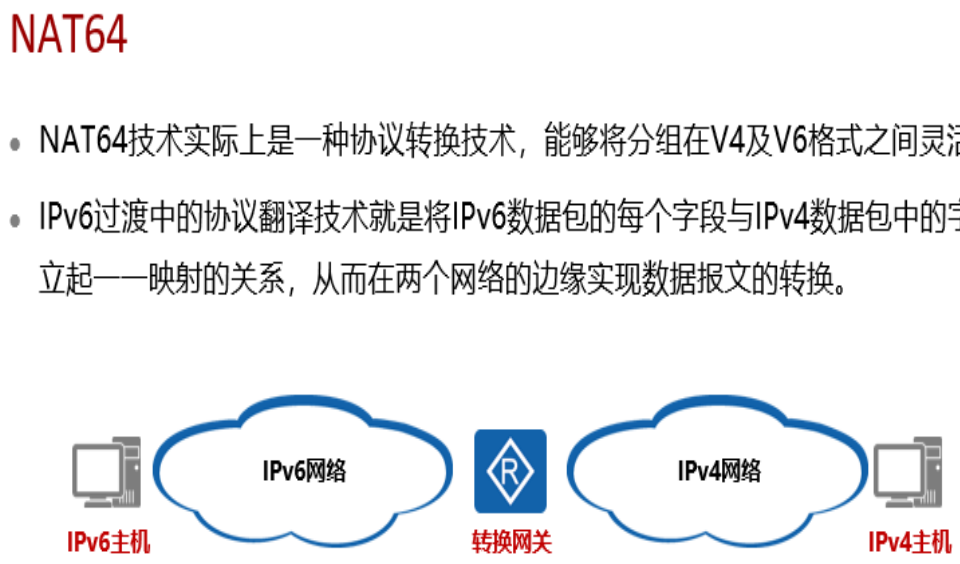
- ISATAP隧道
  - 6over4自动隧道的一种
  - 支持Host到Router、Router到Host、 Host到Host
  - 采用ISATAP隧道专用地址



- ISATAP（Intra-Site Automatic Tunnel Addressing Protocol）是另外一种自动隧道技术。ISATAP 隧道同样使用了内嵌 IPv4 地址的特殊 IPv6 地址形式，只是和 6to4 不同的是，6to4 是使用 IPv4 地址做为网络前缀，而 ISATAP 用 IPv4 地址做为接口标识。
- 地址描述
- 如果 IPv4 地址是全局唯一的，则 u 位为 1，否则 u 位为 0。g 位是 IEEE 群体/个体标志。由于 ISATAP 是通过接口标识来表现的，所以，ISATAP 地址有全局单播地址、链路本地地址、ULA 地址、组播地址等形式。ISATAP 地址的前 64 位是通过向 ISATAP 路由器发送请求来得到的，它可以进行地址自动配置。在 ISATAP 隧道的两端设备之间可以运行 ND 协议。ISATAP 隧道将 IPv4 网络看作一个非广播的点到多点的链路（NBMA）。
- 转发过程描述：
- 在 IPv4 网络内部有两个双栈主机 PC2 和 PC3，它们分

别有一个私网 IPv4 地址。要使其具有 ISATAP 功能，需要进行如下操作：

- 首先配置 ISATAP 隧道接口，这时会根据 IPv4 地址生成 ISATAP 类型的接口 ID。
- 根据接口 ID 生成一个 ISATAP 链路本地 IPv6 地址，生成链路本地地址以后，主机就有了在本地链路上进行 IPv6 通信的能力。
- 进行自动配置，主机获得 IPv6 全球单播地址、ULA 地址等。
- 当主机与其它 IPv6 主机进行通讯时，从隧道接口转发，将从报文的下一跳 IPv6 地址中取出 IPv4 地址作为 IPv4 封装的目的地址。如果目的主机在本站点内，则下一跳就是目的主机本身，如果目的主机不在本站点内，则下一跳为 ISATAP 路由器的地址。

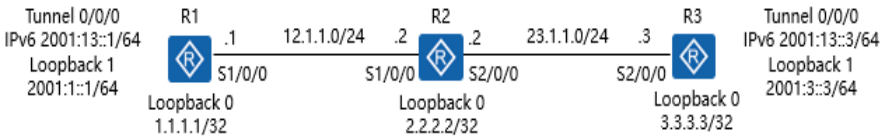


- 当 IPv4 网络的节点需要直接与 IPv6 网络的节点进行通信时，默认情况下当然是行不通的，因为两个协议栈无法兼容。但是借助一台设备，由该设备来实现 IPv6 与 IPv4 的互转，那

么上述通信需求就可以实现了。

配置GRE隧道 (1)

- 公司A网络拓扑如下所示，现根据需求完成如下配置：
  - R1、R2和R3的IPv4地址如图所示，部署在OSPFv2的区域0中，该部分配置已经完成；
  - 所需的IPv6地址已经标出；
  - 采用IPv6 over IPv4 GRE隧道的形式，实现R1与R3的Loopback1之间的互通。



- 案例描述：
- IPv6 和 IPv4 所需地址已经给出。

GRE隧道 - 基本配置命令

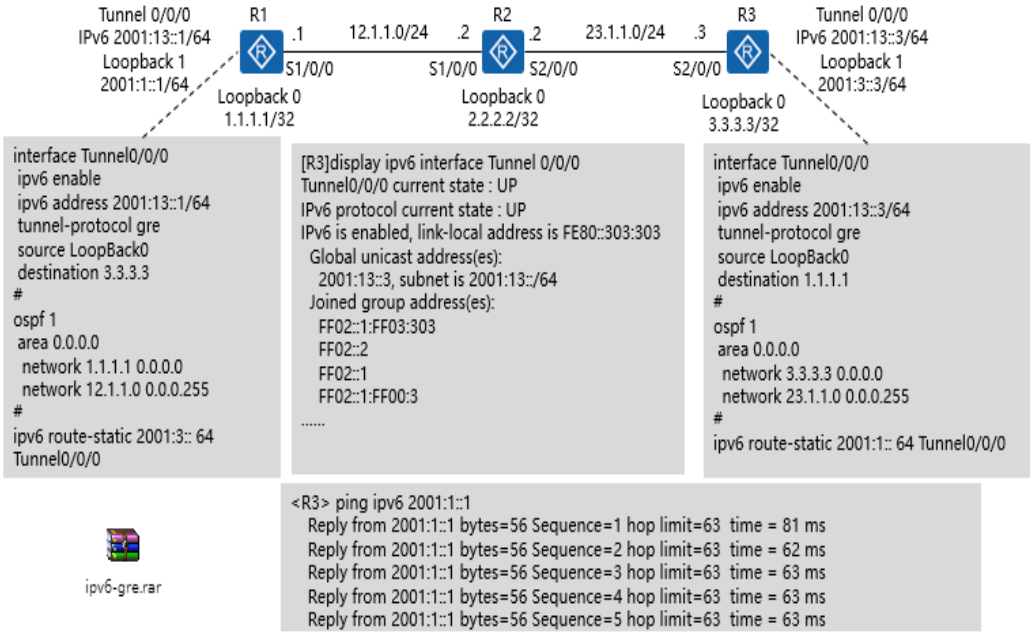
步骤	操作	命令
1	进入系统视图	**system-view**
2	创建Tunnel接口	**interface tunnel** *interface-number*
3	**指定Tunnel为GRE隧道模式**	**tunnel-protocol gre**
4	指定Tunnel的源接口	**source** { *ipv4-address* | *interface-type interface-number*}
5	指定Tunnel的目的接口	**destination ipv4-address**
6	**设置对GRE报文头进行校验**	**gre checksum**
7	**设置GRE报文头的关键字**	**gre key key-number**
8	设置Tunnel接口的IPv6地址	**ipv6 address** *ipv6-address prefix-length*

- 设置对 GRE 报文头进行校验是一个可选的操作步骤。如果设置了对 GRE 报文头进行校验，则发送端根据 GRE 报文头和净荷信息计算校验和，然后将包含校验和的报文转发到对

端。接收端收到报文后，计算接收报文的校验和，并将该校验和与报文中的校验和进行比较。如果结果一致，那么它将会继续处理此报文，否则将其丢弃。如果本端配置了校验和，但是对端没有配置校验和，那么本端不会对接收的报文进行校验和验证。

- 设置 GRE 报文头的关键字也是一个可选的操作步骤。如果设置了 GRE 报文头中的 KEY 字段，接收端将会检查接收的 GRE 报文头的关键字，如果与本端配置的关键字完全相同，表明验证成功，接受该报文，否则丢弃该报文。

配置GRE隧道 (2)



- 命令含义：
- interface tunnel 命令用来创建一个 Tunnel 接口，并进入该 Tunnel 接口视图。
- tunnel-protocol gre 指定 Tunnel 为手动隧道模式。
- source { ipv4-address | interface-type interface-number }指定 Tunnel 的源接口。
- destination { ipv4-address }指定 Tunnel 的目的接

口。

- ipv6 address { ipv6-address prefix-length }设置 Tunnel 接口的 IPv6 地址。

IPv6备考建议

- 掌握IPv6各协议相关命令；
- 熟悉IPv6等原理和应用；
- 熟悉IPv6等相关策略工具的使用；
- 熟读HedEx文档；
  - 包括HedEx涵盖的案例；
- 熟练掌握理解课程中设计的案例场景。