

实验一 IPV6 地址基础

目录

概述.....	1
IPv6 地址格式.....	1
IPv6 地址表示方法.....	2
IPv6 地址类型.....	10
配置 IPv6 地址.....	14

概述

在我们现有的网络中,几乎所有网络都使用 IP 协议作为通信的地址协议,我们的网络使用 IP 来表示地址信息,每一个节点都应该分配一个唯一的地址,才能保证通信正常。现在正常使用的 IP 协议为版本 4,用 32 位来表示,地址空间为 65536×65536 ,结果约为 42.9 亿,需要说明的是,虽然地址共有 42.9 亿之多,但并不表示这些地址可以供 42.9 亿个节点使用,因为我们的地址是分网段的,也就是说即使在一个节点的情况下,分配地址时,也是分配一个网段而不是一个地址,所以这样就使得版本 4 的 IP 地址一下子变得空间狭小,再加之有相当一部分地址是不可用的,那么随着网络的迅速膨胀,IP ver4 的地址空间变得几乎快耗尽了。在这样的情况下,出现了一些如 VLSM 子网技术,NAT 网络地址翻译技术,试图来缓和地址空间的快速消耗。与此同时,人们也开发出了一个地址空间更为庞大的 IP 协议,这个协议拥有比 IP ver4 多出数倍的地址空间,来解决网络地址匮乏的问题,这个 IP 协议就是 IP 版本 6,即 IPv6。

IPv6 地址格式

IPv6 拥有更为庞大的地址空间,是因为 IPv4 只是采用 32 位来表示,而 IPv6 采用 128 位来表示,这样大的一个地址空间,几乎可以容纳无数个节点。正因为 IPv6 使用了 128 位来表示地址,在表示和书写上面具有相当的困难,原来的 IPv4 使用 10 进制来表示,而 IPv6 由于地址太长,则采用 16 进制来表示,但无论我们如何表示,计算机都是处理二进制。因为 10 进制表示时,使用 0 到 9 共十个数字来表示,而 16 进制需要在 10

进制原有的基础上多出 6 个数字，即需要多出 11,12,13,14,15，这 6 个数字则采用字母的形式来表示，分别为

A(表示 10)，B(表示 11)，C(表示 12)，D(表示 13)，E(表示 14)，F(表示 15)，这些字母是不区别大小写的。

但是由于 IPv6 拥有 128 位的长度，所以不能直接表示，必须像 IPv4 那样进行分段表示。IPv6 将整个地址分为 8 段来表示，每段之间用冒号隔开，每段的长度为 16 位，表示如下：

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

从上面可以看出，IPv6 中每一个段是 16 位，每段共四个 X，其中 X 使用 4 bit 表示，一个 X 就表示一个数字或字母，一个完整的地址共 128 bit。

一个 X 使用 4 bit 表示，那么 XXXX 的取值范围就应该从 0000 到 FFFF。

IPv6 地址表示方法

对于一个完整的 IPv6 地址，需要写 128 位，已经被分成了 8 段，每段 4 个字符，也就是说完整地表示一个 IPv6 地址，需要写 32 个字母，这是相当长的，并且容易混淆和出错，所以 IPv6 在地址的表示方法上，是有讲究的，到目前为止，IPv6 地址的表示方法分为三种，分别是：

1. 首选项格式

2. 压缩表示

3. IPv4 内嵌在 IPv6 中

下面分别详细介绍这三种 IPv6 地址表示方法：

1. 首选格式

首选格式的表达方法其实没有任何讲究，就是将 IPv6 中的 128 位，也就是共 32 个字符完完整整，一个不漏地全写出来，比如下面就是一些 IPv6 地址的首选格式表示形式：

0000:0000:0000:0000:0000:0000:0000:0000

0000:0000:0000:0000:0000:0000:0000:0001

2001:0410:0000:1234:FB00:1400:5000:45FF

3ffe:0000:0000:0000:1010:2a2a:0000:0001

FE80:0000:0000:0000:0000:0000:0000:0009

FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

从上面 IPv6 地址的首选格式表示中可以看出，每一个地址，都将 32 个字符全部写了出来，即使地址中有许多个 0，或者有许多个 F，也都有一个不漏地写了出来，由此可见，首选格式只需要将地址完整写出即可，没有任何复杂的变化，但是容易出错。

2. 压缩格式

从前面一个 IPv6 地址表示方法首选格式表示方法中可以看出，一个完整的 IPv6 地址中，会经常性的出现许多个 0，而我们知道，许多时候，0 是毫无意义的，0 表示没有，写出来，也表示没有，不写，也同样表示没有，那么我们就考虑能否将不影响地址结果的 0 给省略不写，这样就可以大大节省时间，也方便人们阅读和书写，这样的将地址省略 0 的表示方法，称为压缩格式。

而压缩格式的表达中，分三种情况，下面来分别介绍三种压缩格式：

第一种情况：

在 IPv6 中，地址分为 8 个段来表示，每个段共 4 个字符，但是一个完整的 IPv6 地址会经常碰到整个段 4 个字符全部都为 0，所以我们将整个段 4 个字符全部都为 0 的使用双冒号 :: 来表示，如果连续多个段全都为 0，那么也可以同样将多个段都使用 双冒号 :: 来表示，如果是多个段，并不需要将双冒号写多次，只需要写一次即可，比如一个地址 8 个段，其中有三个段全都为 0，那么我们就将这全为 0 的三个段共 48 位用 :: 来表示，再将其它 5 个段照常写出即可，当计算机读到这样一个不足 128 位的地址时，比 128 位少了多少位，就在 :: 的地方补上多少个 0，比如上面的 :: 代替为 48 位，那么计算机就会在这个地址的 :: 位置补上 48 位的 0，这样就正确地将地址还原回去了。

下面来看一些整个段 4 个字符都为 0 的 IPv6 地址使用压缩格式来表示：

例 1：

压缩前：

0000:0000:0000:0000:0000:0000:0000:0000

压缩后：

::

说明：可以看出，由于这个地址的 8 个段全部都为 0，所以只用 :: 就将整个地址表示出来，当计算机拿到这个压缩后的地址时，发现比正常的 128 位少了 128 位，那么就会在 :: 的地方补上 128 个 0，结果为：

0000:0000:0000:0000:0000:0000:0000:0000

可以看出，计算机还原的地址就是压缩之前的真实地址。

例 2:

压缩前:

0000:0000:0000:0000:0000:0000:0000:0001

压缩后:

::0001

说明: 可以看出，压缩后的地址比正常的 128 位少了 112 位，计算机就会在 :: 的地方补上 112 个 0，结果为:

0000:0000:0000:0000:0000:0000:0000:0001

可以看出，计算机还原的地址就是压缩之前的真实地址。

例 3:

压缩前:

2001:0410:0000: 0000:FB00:1400:5000:45FF

压缩后:

2001:0410 :: FB00:1400:5000:45FF

说明: 可以看出，压缩后的地址比正常的 128 位少了 32 位，计算机就会在 :: 的地方补上 32 个 0，结果为:

2001:0410:0000: 0000:FB00:1400:5000:45FF

可以看出，计算机还原的地址就是压缩之前的真实地址。

例 4:

压缩前:

3ffe:0000:0000:0000:1010:2a2a:0000:0001

压缩后:

3ffe::1010:2a2a::0001

说明: 当计算机拿到这个压缩后的地址，发现比正常的 128 位少了 64 位，计算机就会试图在::的地方补上少了的 64 个 0，但是我们可以看到，压缩后的地址有两个::，而计算机要补上 64 个 0，所以这时补出来的结果很可能是以下几种:

3ffe:0000:1010:2a2a: 0000:0000:0000:0001

或

3ffe:0000:00001010:2a2a::0000:0000:0001

或

3ffe:0000:0000:0000:1010:2a2a:0000:0001

从结果中可以发现，当一个 IPv6 地址被压缩后，如果计算机出现两个或多个::的时候，计算机在将地址还原时，就可能出现多种情况，这将导致计算机还原后的地址不是压缩之前的地址，将导致地址错误，最终通信失败。

所以，在压缩 IPv6 地址时，一个地址中只能出现一个::。

第二种情况:

在压缩格式的第一种情况的表示中,是在地址中整个段 4 个字符都为 0 时,才将其压缩为::来表示,但是在使用第一种情况压缩之后,我们仍然可以看见地址中还存在许多毫无意义的 0,比如 0001, 0410。我们知道,0001 中,虽然前面有三个 0,但是如果我们将前面的 0 全部省略掉,写为 1,结果是等于 0001 的,而 0410 也是一样,我们将前面的 0 省略掉,写成 410,也同样等于 0410 的,所以我们在省略数字前面的 0 时,是不影响结果的,那么这个时候,表示 IPv6 地址时,允许将一个段中前导部分的 0 省略不写,因为不影响结果。但是需要注意的是,如果 0 不是前导 0,比如 2001,我们就不能省略 0 写成 21,因为 21 不等于 2001,所以在中间的 0 不能省略,只能省略最前面的 0。下面来看一些省略前导 0 的地址表示形式:

例 1:

压缩前:

0000:0000:0000:0000:0000:0000:0000:0000

压缩后:

0:0:0:0:0:0:0:0

从结果中可以看出,计算机根本就不需要对这样的地址还原,压缩后的结果和压缩前的结果是相等的。

例 2:

压缩前:

0000:0000:0000:0000:0000:0000:0000:0001

压缩后:

0:0:0:0:0:0:0:1

从结果中可以看出，计算机根本就不需要对这样的地址还原，压缩后的结果和压缩前的结果是相等的。

例 3:

压缩前:

2001:0410:0000:1234:FB00:1400:5000:45FF

压缩后:

2001:410:0:1234:FB00:1400:5000:45FF

从结果中可以看出，计算机根本就不需要对这样的地址还原，压缩后的结果和压缩前的结果是相等的。

第三种情况:

在前面两种 IPv6 地址的压缩表示方法中，第一种是在整段 4 个字符全为 0 时，才将其压缩后写为 **::**，而第二种是将无意义的 0 省略不写，可以发现两种方法都能节省时间，方便阅读。第三种压缩方法就是结合前两种方法，既将整段 4 个字符全为 0 的部分写成 **::**，也将无意义的 0 省略不写，结果就可以出现以下一些最方便的表示方法:

例 1:

压缩前:

0000:0000:0000:0000:0000:0000:0000:0001

压缩后:

:: 1

可以看到，结合了两种压缩格式的方法，但为简便。

例 2:

压缩前:

2001:0410:0000:0000:FB00:1400:5000:45FF

压缩后:

2001:410::FB00:1400:5000:45FF

可以看到，结合了两种压缩格式的方法，但为简便。

3.IPv4 内嵌在 IPv6 中

在网络还没有全部从 IPv4 过渡到 IPv6 时，就可能出现某些设备即连接了 IPv4 网络，又连接了 IPv6 网络，对于这样的情况，就需要一个地址即可以表示 IPv4 地址，又可以表示 IPv6 地址。

因为一个 IPv4 地址为 32 位，一个 IPv6 地址为 128 位，要让一个 IPv4 地址表示为 IPv6 地址，明显已经少了 96 位，那么就将一个正常的 IPv4 地址通过增加 96 位，结果变成 128 位，来与 IPv6 通信。在表示时，是在 IPv4 原有地址的基础上，增加 96 个 0，结果变成 128 位，增加的 96 个 0 再结合原有的 IPv4 地址，表示方法为

0:0:0:0:0:A.B.C.D 或者 **::A.B.C.D.**，如下:

0000: 0000: 0000: 0000: 0000:0000:A.B.C.D

96 个 0 32 位

例:

IPv4 地址为 138.1.1.1

表示 IPv6 地址为 0:0:0:0:0:0:138.1.1.1

注：IPv6 中没有广播地址，IPv6 不建议划子网，如果需要划子网，网络位请不要低于 48 位。

IPv6 地址类型

在 IPv4 地址中，地址分许多类型，比如代表节点自己的 127.0.0.0/8，私有地址段，组播地址段，广播地址，以及一些不可用的地址。在 IPv6 中，同样地址也像 IPv4 那样分了许多类型，我们需要了解的有 3 种类型，为 Unicast（单播），Anycast（任意播）和 Multicast（组播），下面分别来详细介绍这几种地址类型。

Unicast（单播）

即使是在 IPv4 中，单播地址的类型也分好多种，就是我们常用的也分私有，公有，还有回环地址，在 IPv6 中，单播地址也分好几种，我们需要知道的有：Link-Local Address（链路本地地址），Unique Local Address（本地站点地址），Aggregatable Global Address（可聚合全球），回环地址。

下面详细介绍几种单播地址：

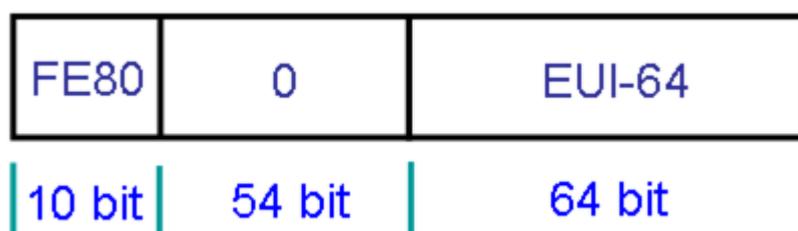
Link-Local Address（链路本地地址）

即使网络再大，每两点之间，都有链路相连，在一个节点将数据包发给下一个节点时，必须在数据包中封装三层 IP 地址，再封装下一节点的二层链路地址（如以太网中的 MAC 地址），才能将数据包发给下一节点，并且只有当封装的二层链路地址确实为下一节点的真实链路地址时，对方才能接收，这就是普通二层链路地址的功能，这样的地址在一条链路的范围内明确了每个节点，并且这样的地址是不能被路由的。

而在 IPv6 网络中，两个 IPv6 的节点通过链路相连，必须在这条链路之间为各自确立一个 Link-Local Address（即链路本地地址），在一条链路上，IPv6 节点能够确定对方节点的身份，能够将数据包发向对方节点，必须知道对方节点的链路本地地址，如果不知道，将是不能通信的，所以一条链路中的 IPv6 节点要通信，必须拥有链路本地地址，并且这个链路本地地址只在一条链路中有效，也不能被路由，而不同链路的链路本地地址是可以重复的。

因为在链路上没有链路本地地址的情况下，IPv6 是不能通信的，所以每个节点必须拥有一个链路本地地址，当一个节点上正常启动了 IPv6 之后，链路本地地址是不需要人工干预，会自己生成的，但也可以自己手工配置链路本地地址。

自动生成的链路本地地址，有默认的特殊格式，是以 FE80::/10 (1111 1110 10)打头，再加 54 个 0，还差 64 位，这后面的 64 位，再使用 EUI-64 来填充，表示如下：



EUI-64 结构

一个链路本地地址的后 64 位使用 EUI-64 来填充，EUI-64 其实就是接口的 MAC 地址，而 MAC 地址共长度为 48 位，要填充 64 位的 EUI-64，还少 16 位。一个完整的 EUI-64 是将 MAC 地址的 48 位平均分成两部分，前面 24 位，后面 24 位，然后在中间补上 FFFE(16 位)，如一个 MAC 地址为 00:12:33:5C:82:E1，将其变为 EUI-64 的结果如下表示：



只有以太网链路才会有 MAC 地址，而串行链路是没有 MAC 地址的，当一个接口上启用 IPv6 之后，此接口会自动产生一个链路本地地址，而链路本地地址需要借用接口上的 MAC 地址才能产生，这在有 MAC 地址的以太网接口上可以轻松实现，但是当一个没有 MAC 地址的串行接口上开启 IPv6 之后，由于自己没有 MAC 地址，所以不能产生 EUI-64，也就无法完成链路本地地址。在这种情况下，所有没有 MAC 地址的接口，如串行接口，在开启 IPv6 后，需要产生 EUI-64 时，统统借用设备上第一个以太网插槽的第一个接口，也可以理解为没有 MAC 地址的接口，统统使用设备上 MAC 地址池中的第一个地址，比如设备上为接口 s2/3 开启 IPv6 之后，就很有可能借用 F0/0 或 G0/0 接口的 MAC 地址。

EUI-64 不仅在产生链路本地地址时可以使用，在正常配置 IPv6 地址时，同样可以使用 EUI-64 来填充后 64 位。

本地站点地址

本地站点地址是单播中一种受限制的地址，只在一个站点内使用，不会默认启用，这个地址不能在公网上路由，只能在一个指定的范围内路由，需要手工配置。IPv6 中的本地站点地址类似 IPv4 中私有地址，如 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16。

得不到合法 IPv6 地址的机构可配置本地站点地址，表示方法为：

FC00::/7 + 41bit 子网标识 +16bit 链路标识+ EUI-64

可聚合全球单播地址

可聚合全球单播地址相当于 IPv4 的公网地址，可以被路由的，可以正常使用的地址，但网络位最少为 48 位。可聚合全球单播地址的范围是

2000:0000:0000:0000:0000:0000:0000:0000

到

3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

以上面可以看出，可聚合全球单播地址也就是 2 和 3 打头的地址，因为 IPv6 使用 16 进制来表示，一个字符的取值范围从 0 到 F 共 16 个，而可聚合全球单播地址地址占了 2 和 3 两个，由此说明，可聚合全球单播地址占 IPv6 总地址空间的 8 分之 1，也就是说，所有 IPv6 地址中，只有 8 分之 1 是可以给网络正常使用的。

回环地址

回环地址表示节点自身，类似 IPv4 的 127.0.0.0/8

回环地址表示为

0000:0000:0000:0000:0000:0000:0000:0001

0:0:0:0:0:0:0:1

::1

任意播地址

任意播地址表示一组接口，当一个发向某个任意播地址的数据包，只被最近的接口收到，这个地址是由路由协议定义的，不能手工配置，但是我们无法看到一个地址就能区别出到底是单播地址还是任意播地址，因为任意播地址的表示格式和单播地址是一样的，也就是说任意播地址就是用普通的单播地址来表示的。任意播地址只能出现在路由器上，并且不能作为数据包的源地址来使用。

组播

组播地址就是一个目标为组播地址的数据包将被多个节点收到，地址以 FF00::/8 (1111 1111)打头，.表示为

```
FF00:0000:0000:0000:0000:0000:0000/8
```

```
FF00:0:0:0:0:0:0:0/8
```

```
FF00::/8
```

详细的 IPv6 多播知识将在后面的 IPv6 Multicast 部分介绍。

配置 IPv6 地址

1. 激活 IPv6 功能

默认情况下，Cisco 设备的 IPv6 流量转发功能是关闭的，需要使用 IPv6，必须先开启 IPv6 流量转发功能。

(1) 开启 IPv6 流量转发功能

```
r1(config)#ipv6 unicast-routing
```

2. 配置正常的 IPv6 地址

(1) 在接口下配置正常 IPv6 地址

```
r1(config)#int f0/0
```

```
r1(config-if)#ipv6 address 2011:1:2:3:1:1:1:1/64
```

说明：配置的地址前 64 位为网络地址，即 2011:1:2:3；后 64 位为主机位，即 1:1:1:1。

(2) 查看接口的 IPv6 地址

```
r1#show ipv6 interface brief f0/0
```

```
FastEthernet0/0    [up/up]
```

```
FE80::C200:EFF:FEB0:0
```

```
2011:1:2:3:1:1:1:1
```

```
r1#
```

说明：可以看到接口 F0/0 已经接受我们配置的地址 2011:1:2:3:1:1:1:1。

3.使用 EUI-64 格式配置静态地址：

(1)配置包含 EUI-64 的 IPv6 地址

```
r1(config)#int f0/1
```

```
r1(config-if)#ipv6 address 2022:2:2:22::/64 eui-64
```

(2) 查看接口的 IPv6 地址

```
r1#show interfaces f0/0
```

```
FastEthernet0/0 is up, line protocol is up
```

```
Hardware is Gt96k FE, address is c000.0eb0.0000 (bia c000.0eb0.0000)
```

```
r1#show ipv6 interface brief f0/1
```

```
FastEthernet0/1    [up/up]
```

```
FE80::C200:EFF:FEB0:1
```

```
2022:2:2:22:C200:EFF:FEB0:1
```

```
r1#
```

说明：可以看到，F0/1 成功使用接口上的 MAC 地址为 EUI-64 来填充后 64 位。

4.仅启用接口 IPv6 功能

说明：一个接口上可以仅启用 IPv6 功能，而不配置 IPv6 地址

(1) 启用接口 IPv6 功能

```
r1(config)#int s1/0
```

```
r1(config-if)#ipv6 enable
```

(2) 查看接口 IPv6 状态

```
r1#show ipv6 interface brief serial 1/0
```

```
Serial1/0          [up/up]
```

```
FE80::C200:EFF:FE80:0
```

```
r1#
```

可以看到，接口 S1/0 可以只开启 IPv6 功能而不配地址，但开了 IPv6 功能的接口也会自动产生一个链路本地地址。

5.配置无编号地址

当地址紧缺时，可以配置一个接口使用另外一个接口的地址，这样的地址称为无编号地址，即 **unnumbered** 地址，当从无编号接口产生数据包时，该接口使用借用的那个接口的地址作为源地址，配置这样的地址，需要允许双方不同网段协议的配合。

(1) 为接口配置无编号地址

```
r1(config)#int s1/1
```

```
r1(config-if)#ipv6 unnumbered f0/0
```

(2) 查看接口 IPv6 地址情况。

```
r1#show ipv6 interface brief serial 1/1
```

```
Serial1/1          [up/up]
```

FE80::C200:EFF:FEB0:0

unnumbered (FastEthernet0/0)

r1#

说明：可以看到结果显示为接口 S1/1 借用 F0/0 的地址。