华东师范大学计算机科学技术系上机实践报告

课程名称：计算机网络	**年级**：2022	**上机实践成绩**：
指导教师：洪道诚	**姓名**：朱宇笑	**创新实践成绩**：
实验名称：IPv6基础实验	**学号**：10225001410	**上机实践日期**：2023/12/22
座位编号：F	**组号**：6	**上机实践时间**：2 学时

一、 实验目的

1. 了解华为 eNSP 网络仿真平台的使用方法；
2. 了解华为 VRP 操作系统的基本功能；
3. 了解和掌握通过 CLI 界面对华为网络设备进行 IPv6 基本配置（静态 IPv6 地址配置、无状态地址自动配置、DHCPv6 部署于配置）；
4. 了解几个 ICMPv6 的应用（ping、tracert 和 path MTU 发现）的工作原理；
5. 了解 NDP 的基本功能（无状态地址自动配置、地址解析和 DAD）。

二、 实验设备

1. PC 机一台，运行华为 eNSP 仿真软件仿真编辑器
2. 以每人为小组独立完成实验

三、 实验原理

1. 华为 eNSP 仿真平台简介

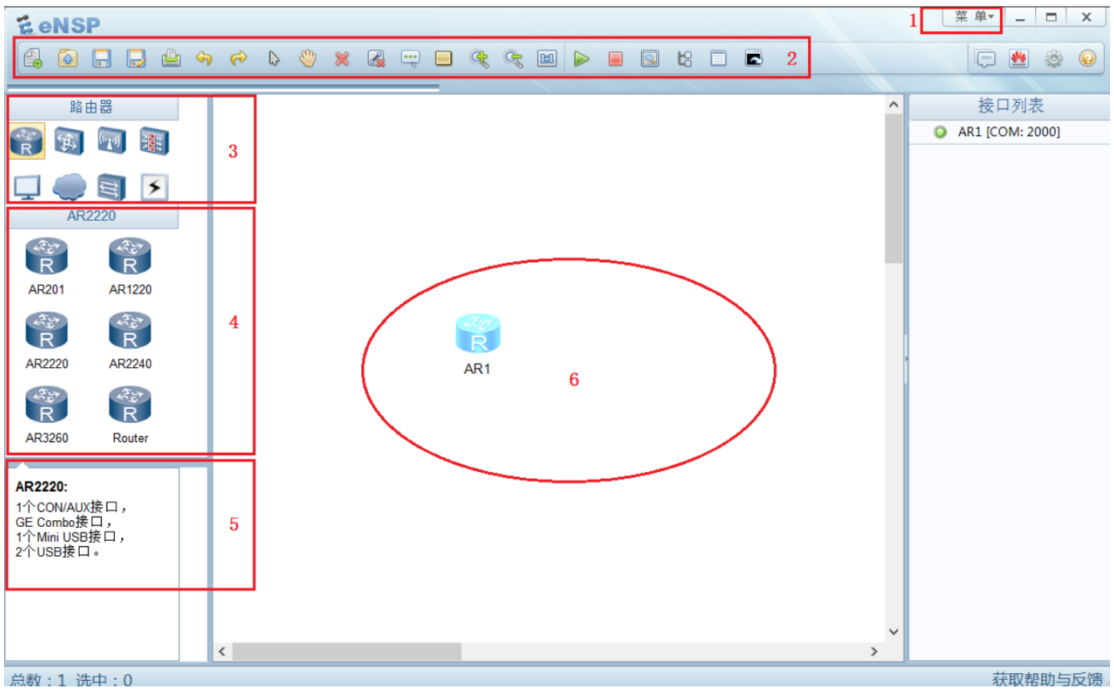
eNSP 全称是 Enterprise Network Simulation Platform。这是一款由华为提供的、免费的、图形化的网络仿真平台，可用于对华为的路由器和交换机等网络设备进行软件仿真，可支持大型的、复杂的网络模拟，以便于用户在没有真实设备的情况下模拟练习，学习和掌握华为网络设备的技术及用法。

该网络模拟器的主要功能如下：

- 模拟实际的设备硬件。
- 能够提供三层交换机的功能。
- 借助于 WireShark 工具，eNSP 能够分析协议报文。
- 支持 IPv6、支持无线网络。
- 具有网络拓扑绘制功能，能够轻松得到设计的网络拓扑图。

运行 eNSP，初始界面如下：

点击“新建拓扑”，即可进入下图所示的界面：



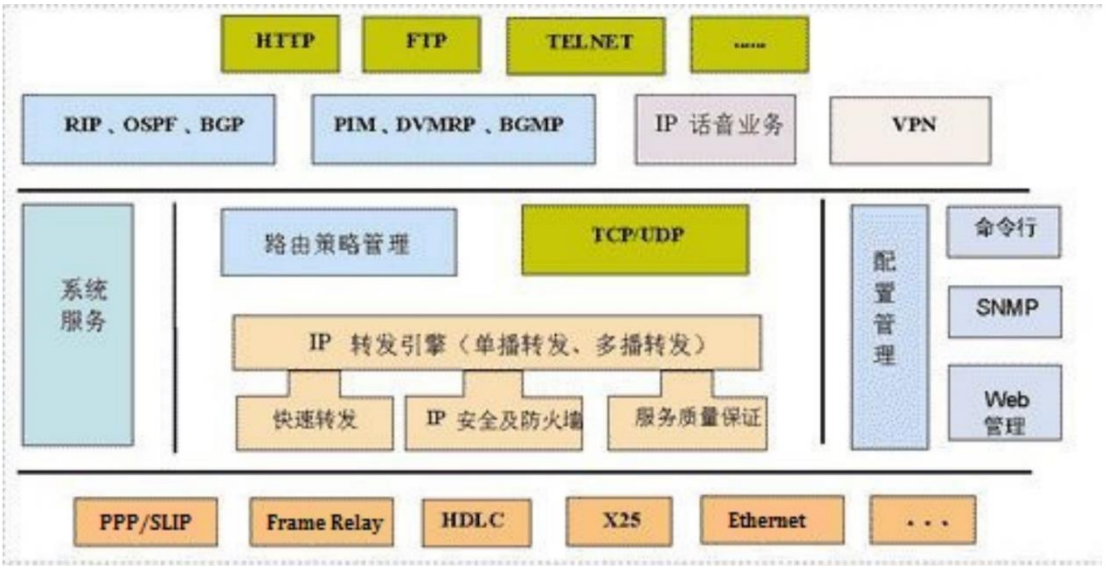
这个界面包含七个功能区域。下面简单地解释其中六个带上数字标记的区域的用途。

- 区域 1 下拉菜单栏：此栏提供“文件”、“编辑”、“视图”、“工具”、和“帮助”菜单。
- 区域 2 工具栏：此工具栏提供指向“文件”和“编辑”菜单命令的快捷图标。此栏还提供“开启设备”、“停止设备”、“数据抓包”、“撤销”、“恢复”、“缩放”、“调色板”等工具按钮。
- 区域 3 设备类型选择区：此框包含可用的设备和连接的类型。具体设备选择区的内容将依据你所选择的设备类型或连接而变化。
- 区域 4 具体设备选择区：可以从中选择哪些具体设备加入到你的网络拓扑中，并建立所需类型的连接。

- 区域 5 设备接口列表区：该区内将列出你选择的具体设备的接口列表。
- 区域 6 工作区：用户将在该区域创建自己所设计的网络拓扑，观察模拟过程和结果，并可查看多种信息和统计结果。

2. 华为 VRP 简介

华为 VRP （Versatile Routing Platform）即通用路由平台，是华为公司从低端到高端的全系列路由器、交换机等数据通信产品的通用网络操作系统。运行 VRP 操作系统的华为产品包括路由器、局域网交换机、ATM 交换机、拨号访问服务器、IP 电话网关、电信级综合业务接入平台、智能业务选择网关，以及专用硬件防火墙等。VRP 可以运行在多种硬件平台之上，并拥有一致的网络界面、用户界面和管理界面，可为用户提供灵活而丰富的应用解决方案。随着网络技术的迅速发展， VRP 在处理机制、业务能力、产品支持等方面也在持续地演进。目前，VRP 的版本已从最初的 VRP l.0 演进到了 VRP8.x，但是对于大部分用于企业网络场景的中低端网络设备都是基于 VRP 5.x 的。
VRP 平台以 TCP/IP 协议栈为核心，实现了数据链路层、网络层和应用层的多种协议，在操作系统中集成了路由技术、QoS 技术、VPN 技术、安全技术和 IP 语音技术等数据通信要件，并以 IP 转发引擎（TurboEngine）技术作为基础，为网络设备提供了出色的数据转发能力。其体系结构图如下：



- IP 转发引擎：包括传统 IP 报文转发、IP 快速转发、QoS 服务质量、策略路由、安全能力及防火墙等。
- 广域网互连：支持 PPP/MP、SLIP、HDLC/SDLC、X.25、Frame Relay、LAPB、ISDN 和 Ethenet 等。
- 路由协议：支持 RIP、OSPF、BGP、IGRP、EIGRP、PIM、DVMRP、BGMP 等。
- IP 业务：支持 ARP/Proxy ARP、NAT、DNS、DHCP 中继、VLAN、SNA、VoIP 和 VPN 等。
- 配置管理能力：支持命令行配置、日志告警、调试信息、SNMP 管理等。

要配置和管理华为的网络设备，首先要学会 VRP 命令行的用法。一条命令行由关键字和参数组成，关键字是一组与命令行功能相关的单词或词组，通过关键字可以唯一确定一条命令行。参数是为了完善命令行的格式或指示命令的作用对象而指定的相关单词或数字等，包括整数、字符串、枚举值等数据类型。例如，测试设备间连通性的命令行 ping *ip-address* 中，
ping 为命令行的关键字，而 *ip-address* 为参数，取值为一个 IP 地址。
VRP 命令的总数有几千之多，为了实现对它们的分级管理， VRP 系统将这些命令按照功能类型的不同分别注册在不同的视图之下。使用某个命令行时，需要先进入到该命令行所在的视图。最常用的命令行视图有用户视图、系统视图和接口视图。用户视图的提示符为“<Huawei>”，其中“< >”表

示用户视图， “Huawei” 是设备默认的主机名。在用户视图下，用户可以了解设备的基础信息、查询设备状态，但不能进行与业务功能相关的配置。系统视图的提示符为“[Huawei]”，在用户视图下使用“system-view”命令进入系统视图，此时才能对设备进行业务功能配置。系统视图下可以使用绝大部分的基础功能配置命令。另外，系统视图还提供了进入其他视图的入口，如果要进入其他视图，则首先必须进入系统视图。注意：除用户视图外，其他任何视图的提示符都是“[ ]”，方括号内的字符串会提示当前所处的视图。

VRP 还对命令和用户进行了分级，每条命令都有相应的级别，每个用户也都有自己的权限级别，并且用户权限级别与命令级别具有一定的对应关系。具有一定权限级别的用户登录以后，只能执行等于或低于自己级别的命令。

VRP 命令级别分为 0~3 级：0 级为参观级、1 级为监控级、2 级为配置级、3 级为管理级。网络诊断类命令属于参观级命令，用于测试网络是否连通等。监控级命令用于查看网络状态和设备基本信息。对设备进行业务配置时，需要使用配置级命令。而对于一些特殊的功能（如上传或下载配置文件），则需要用到管理级命令。

用户权限分为 0~15，共 16 个级别。默认情况下，3 级用户就可以操作 VRP 系统的所有命令，也就是说 4~15 级的用户权限在默认情况下与 3 级用户的权限是一致的。4~15 级的用户权限一般与提升命令级别的功能一起使用，例如当设备管理员较多时，需要在管理员中再进行权限细分，这时可以将某条关键命令所对应的用户级别提高，如提高到 15 级，这样的话，默认的 3 级管理员将不再能使用该关键命令。

命令级别与用户权限级别的对应关系如下表所示：

用户级别	命令级别	说明
0	0	网络诊断类命令(ping，tracert) 、从本设备访问其他设备的命令(telnet) 等
1	0、1	系统维护命令，包括 display 等。但并不是所有的 display 命令都是监控级的，例如 display current-configuration 和 display saved-configuration 都是管理级命令
2	0、1、2	业务配置命令，包括路由、各个网络层次的命令等
3~15	0、1、2、3	涉及系统基本运行的命令，如文件系统、FTP 下载、配置文件切换命令、用户管理命令、命令级别设置命令、系统内部参数设置命令等，还包括故障诊断的 debugging 命令

VRP 系统提供了丰富的命令行输入方法，支持多行输入，每条命令最大长度为 510 个字符，命令关键字不区分大小写，同时支持不完整关键字输入。

在线帮助是 VRP 系统提供的一种实时帮助功能。在命令行输入过程中，用户可以随时键入“?”获得在线帮助信息。命令行在线帮助可分为完全帮助和部分帮助。

为了进一步提高命令行的输入效率，VRP 定义了一些快捷键，称为系统快捷键。系统快捷键功能固定，用户不能重新定义。常见的系统快捷键如下：

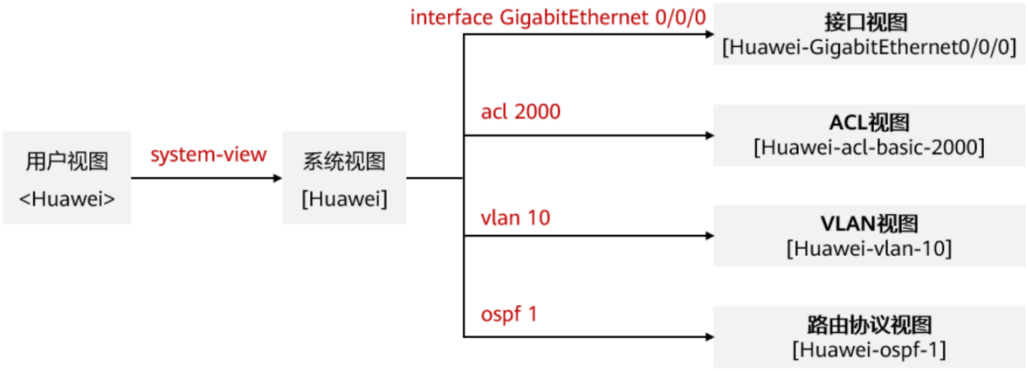
- CTRL+A：将光标移动到当前行的开始
- CTRL+E：将光标移动到当前行的末尾
- ESC+N：将光标向下移动一行
- ESC+P：将光标向上移动一行
- CTRL+C：停止当前正在执行的功能
- CTRL+Z 返回到用户视图，功能相当于 return 命令
- <Tab>键：部分帮助的功能，输入不完整的关键字后按下<Tab>键，系统将自动补全关键字命令行

常见错误信息表：

英文错误信息	错误原因
Error: Unrecognized command found at '^' position.	没有查找到命令
	没有查找到关键字
Error: Wrong parameter found at '^' position.	参数类型错
	参数值越界
Error: Incomplete command found at '^' position.	输入命令不完整
Error: Too many parameters found at '^' position.	输入参数太多
Error: Ambiguous command found at '^' position.	输入命令不明确

3.华为设备的常用配置视图

华为 VRP 的命令行是以所谓的视图来区分使用范围的，视图决定了用户能够使用的命令行。对于华为设备，VRP 提供了用户视图和系统视图两种基本的命令行分级，同时还提供了接口视图、子接口视图、用户界面视图、路由协议视图、VLAN 视图、VLAN 接口视图等多种级别的配置视图，以允许用户对交换机的资源进行配置和管理。下图是华为路由器的几种视图：



以下的视图操作命令均以华为 S3700 交换机为例。

（1）用户视图

当用户通过设备的 Console 连接或 Telnet 方式连接登录到交换机时，此时进入的就是用户视图。对于 eNSP 环境，右击工作区的设备图标，选择“CLI”菜单即可进入用户视图。

在该视图下，用户可了解设备的基础信息、查询设备状态，但不能进行业务功能配置。用户视图的命令行提示符是：<Huawei>，其中的 Huawei 是华为设备的默认主机名。在用户视图下，直接输入？，可获得在该视图下允许执行的命令帮助。

```
<Huawei>?
User view commands:
cd Change current directory
check Check information
clear Clear information
clock Specify the system clock
cluster Run cluster command
cluster-ftp FTP command of cluster
compare Compare function
configuration Configuration interlock
copy Copy from one file to another
```

```
debugging Enable system debugging functions
delete Delete a file
dir List files on a file system
display Display current system information
fixdisk Recover lost chains in storage device
format Format the device
ftp Establish an FTP connection
hwtacacs-user
kill Release a user terminal interface
language-mode Specify the language environment
lldp Link Layer Discovery Protocol
local-user Add/Delete/Set user(s)
lock Lock the current user terminal interface
mkdir Create a new directory
more Display the contents of a file
move Move the file
<Huawei>
```

（2）系统视图

在用户视图下，执行 system-view 命令，将进入系统视图。在该视图下，用户能够执行 VRP 提供的大多数命令。系统视图的命令行提示符为：

[Huawei]

```
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]
```

华为 VRP 没有从用户视图进入系统视图的密码保护，无法设置系统视图密码，因此系统的安全性依赖于 Console 密码。

如果要离开系统视图、返回用户视图，可执行 quit 命令。

在任何视图下，执行 return 命令或按快捷键 Ctrl+Z，将返回用户视图。

（3）接口视图

在系统视图下，执行 interface 命令，即进入接口视图。在该视图下，可对选定的接口（端口）进行配置。接口视图的命令行提示符为：[Huawei-Ethernet0/0/1]

例如，如果要设置华为 S3700 交换机的 Ethernet 0/0/1 以太网端口的通讯速度设置为 100M，全双工方式，则配置命令为：

```
[Huawei]interface e0/0/1
[Huawei-Ethernet0/0/1]speed 100
Error: Please undo negotiation auto first.
[Huawei-Ethernet0/0/1]undo negotiation auto （注：取消自动协商功能）
[Huawei-Ethernet0/0/1]speed 100
[Huawei-Ethernet0/0/1]duplex full
[Huawei-Ethernet0/0/1]
```

【注】这里 X/Y/Z 为接口的编号格式，表示对应的“槽位号/子卡号/接口号”。

（4）用户界面视图

在系统视图下，执行 user-interface console 或 user-interface vty 命令，将进入用户界面视图。该视图主要用于对控制台（Console）端口和虚拟终端（vty，Virtual Type Terminal）进行配置，其主要目的是设置控制台和虚拟终端的用户级登录密码。

用户界面视图的命令行提示符为：[Huawei-ui-console0]或[Huawei-ui-vty0-4]

交换机的控制台端口（console），其编号为 0，利用该端口可进行本地配置和管理。该视图下的主要几个命令帮助为：

```
[Huawei]user-interface console 0
[Huawei-ui-console0]?
user-interface view commands:
authentication-mode Configure the authentication mode for a user
terminal interface
set Set the parameters for a user terminal interface
user Set the parameters of a login user
authentication-mode
```

命令用于设置认证方式，Password 认证、AAA 认证和 None 认证：

- Password 认证:只需输入密码，密码验证通过后，即可登录设备。缺省情况下，设备使用的是 Password 验证方式。使用该方式时，如果没有配置密码，则无法登录设备。
- AAA 认证:需要输入用户名和密码，只有输入正确的用户名和其对应的密码时，才能登录设备。由于需要同时验证用户名和密码，所以 AAA 认证方式的安全性比 Password 认证方式要高，并且该方式可以区分不同的用户，用户之间互不干扰。所以，对于 Telnet 连接方式登录时，一般都采用 AAA 验证方式。
- None 认证:不需要输入用户名和密码，可直接登录设备，即无需进行任何用户身份认证。为安全起见，不推荐使用这种验证方式。用户认证机制保证了用户登录的合法性。为安全起见，实践中应为该端口设置登录密码，

设置密码（Password）认证的方法为：

```
[Huawei-ui-console0]authentication-mode password
[Huawei-ui-console0]set authentication password cipher 12345
[Huawei-ui-console0]user privilege level 3[Huawei-ui-console0]quit
[Huawei]
```

从上面的例子可知，设置控制台登录密码的命令是 set authentication password，本例的密码字为 12345。用户的特权级别为 3。退出用户界面视图，执行 quit 命令。

设置 console 密码后，利用控制台端口登录交换机时，就会首先询问并要求输入该登录密码，密码校验成功后，才能进入到交换机的用户视图。

虚拟终端（即 VTY）用户界面用于以 Telnet 方式登录设备的用户。考虑到 Telnet 是远程登录，容易存在安全隐患，所以在用户验证方式上应该采用 AAA 验证。一般地，设备调试阶段需要登录设备的人员较多，并且需要进行业务方面的配置，所以通常配置最大 VTY 用户界面数为 15，即允许最多 15 个用户同时使用 Telnet 方式登录到设备。同时，应将用户级别设置为 3 级，即配置级（默认为 0），以便可以进行正常的业务配置。交换机支持多个虚拟终端，不同型号的交换机允许开启的虚拟终端数是不同的。对于 S3700 交换机，最大可以开启 15 个虚拟终端，默认为 5 个。虚拟终端只有设置了密码后才允许登录，没有设置密码的虚拟终端是不能登录的。下例是对一台 S3700 交换机设置虚拟

终端 telnet 登录的过程：

- 设置交换机管理 IP 地址：

```
<Huawei>syslem-view
[Huawei]interface vlanif 1
[Huawei-vlanif1]ip address 192.168.10.10 24
[Huawei-vlanif1]quit
```

- 开启 telnet 服务：

```
[Huawei]telnet server enable
```

- 设置最大虚拟终端数：

```
[Huawei]user-interface maximum-vty 15
```

- 设置用户权限级别：

```
[Huawei]user-interface vty 0 14
```

```
[Huawei-ui-vty0-14]user privilege level 15
```

- 设置认证方式为 AAA：

```
[Huawei-ui-vty0-14]authentication-mode aaa
[Huawei-ui-vty0-14]quit
```

- 设置本地认证用户、权限级别、用户访问的服务类型：

```
[Huawei]aaa
[Huawei-aaa]local-user admin password cipher huawei
[Huawei-aaa]local-user admin privilege level 15
[Huawei-aaa]local-user admin service-type telnet
[Huawei-aaa]quit
```

- 设置访问控制列表（ACL），限制从指定主机登录设备：

```
[Huawei]acl 2000
[Huawei-acl-basic-2000]rule permit source 192.168.10.2 0
[Huawei-acl-basic-2000]quit
[Huawei]user-interface vty 0 14
[Huawei-ui-vty0-14]acl 2000 inbound
```

一旦配置了虚拟终端的登录密码，就可以使用 telnet 命令登录设备进行配置和管理。为了防止过长的空闲的连接可能带来的安全隐患，无论通过 console 端口登录或通过 vty 方式的 telnet 登录，可以设置空闲超时的时间，系统默认的空闲超时时间是 10 分钟。

设置空闲超时时间的命令为：idle-timeout 分钟数 秒数

例如，要将 vty 0-14 线路的空闲超时时间设置为 3 分钟 0 秒，则配置命令为：

```
<Huawei>syslem-view
[Huawei]user-interface vty 0 14
[Huawei-ui-vty0-14]idle-timeout 3 0
[Huawei-ui-vty0-14]quit
[Huawei]
```

（5）VLAN 视图

在系统视图下，执行 vlan 命令，将进入 VLAN 配置视图。

在该视图下，可实现对 VLAN（虚拟局域网）的创建、修改或删除等操作。退出 VLAN 视图，返回到系统视图，可执行 quit 命令。

VLAN 视图的命令行提示符为：[Huawei-vlan10] ，其中 10 为 VLAN 号。

下例创建了 VLAN 10：

```
[Huawei]vlan 10
[Huawei-vlan10]
```

为了便于批量创建 VLAN，华为 VRP 提高了一条 vlan batch 命令，具体用法如下：

```
[Huawei]vlan batch 2 to 5
Info: This operation may take a few seconds. Please wait for a
moment...done.
[Huawei]
[Huawei]vlan batch 20 30 40
Info: This operation may take a few seconds. Please wait for a
moment...done.
[Huawei]
```

上述例子中，第一条命令创建了从 2 到 5 的总共 4 个 VLAN，第二条命令创建了 20、30、40 总共 3 个 VLAN。

（6）VLAN 接口视图

VLAN 接口，也可以称为交换虚拟接口（SVI），可以在上面配置 IP 地址参数。对于二层交换机，用于配置交换机的管理 IP 地址；对于三层交换机，这个接口类似于子网的默认网关。

在系统视图下，执行 interface vlanif 命令，将进入 VLAN 接口视图。
在该视图下，可实现对 VLAN 虚拟接口的配置。退出 VLAN 接口视图，返回到系统视图，可执行 quit 命令。
VLAN 接口视图的命令行提示符为：[Huawei-Vlanif10] ，其中 10 为 VLAN 号。
下例配置 VLANIF1 接口的主 IP 地址为 192.168.10.10，从 IP 地址为 192.168.10.20，子网掩码均为 255.255.255.0。

```
<HUAWEI> system-view
[HUAWEI] interface vlanif 1
[HUAWEI-Vlanif2] ip address 192.168.10.10 255.255.255.0
[HUAWEI-Vlanif2] ip address 192.168.10.20 255.255.255.0 sub
```

上例中配置 IP 地址的命令行为：
ip address *ip-address* {*mask*|*mask-length*} [sub]
其中的参数和选项说明如下：

- *ip-address*：指定接口的 IP 地址
- *mask*：二选一参数，指定所设置的 IP 地址对应的子网掩码
- *mask-length*：二选一参数，指定所设置的 IP 地址对应的子网掩码长度
- sub：可选项，指定设置的 IP 地址为从 IP 地址，如果不选择此可选项，则设置的 IP 地址为主 IP 地址。

4. 华为网络设备的配置保存

与思科设备类似，华为的网络设备也包含多种类型的存储器，主要包括只读内存（ROM）、随机存储器（RAM）、闪存（Flash）、还有 CF 卡存储器等，其作用也类似。在此不再赘述。设备的配置文件一般存储在 flash 中。华为与配置文件管理有关的基本概念有 3 个：当前配置、配置文件、下次启动的配置文件。
（1）当前配置
设备内存中的配置信息称为设备的当前配置，它是设备当前正在运行的配置。显然，设备断电后或设备重启时，内存中所有信息（包括配置信息）都会消失。
（2）配置文件
包含设备配置信息的文件称为配置文件，它存在于设备的外部存储器中（注：一般位于闪存中），其文件名的格式一般为“*.cfg” 或“*.zip”。用户可以将当前配置保存到配置文件中。当设备重启时，配置文件的内容可以被重新加载到内存，成为新的当前配置。配置文件除了具有保存配置信息的作用外，还可以方便设备安装和维护人员查看、备份以及移植配置信息用于其他设备。默认情况下，保存当前配置时，设备会将配置信息保存到名为 “vrpcfg.zip”的配置文件中，并存放于设备的外部存储器的根目录下。
（3）下次启动的配置文件
下次启动的配置文件即为设备下次启动时加载至内存的配置文件。设备重启时，会从指定的配置文件中提取配置信息，并加载至内存中。默认情况下，下次启动的配置文件的文件名为“vrpcfg.zip”。
保存当前配置的方式有两种：手动保存和自动保存。
手动保存配置时，用户使用 save [*configuration-file*]命令将当前配置保存到指定配置文件中，参数 *configuration-file* 为指定的配置文件名，格式必须为“*.cfg” 或“*.zip”。如果未指定配置文件名，则配置文件名缺省为“vrpcfg.zip”。用户还可以使用 save backup.zip 命令将当前配置保存到文件名为“backup.zip”的配置文件中，作为对 vrpcfg.zip 的备份。自动保存配置功能可以有效降低用户因忘记保存配置而导致配置丢失的风险。自动保存功能分为周期性自动保存和定时自动保存两种方式。在周期性自动保存方式下，设备会根据用户设定的保存周期，自动完成配置保存；无论设备的当前配置相比配置文件是否有变化，设备都会进行自动保存操作。在定时自动保存方式下，用户设定一

个时间点，设备会每天在此时间点自动进行一次保存。默认情况下，设备的自动保存功能是关闭的，需要用户开启之后才能使用。周期性自动保存的设置方法如下：首先执行命令 autosave interval on，开启设备的周期
性自动保存功能，然后执行命令 autosave interval *time*，设置自动保存周期。*time* 为指定的时间周期，单位为分钟，默认值为 1440 分钟（24 小时）。
定时自动保存的设置方法如下：首先执行命令 autosave time on，开启设备的定时自动保存功能，然后执行命令 autosave time *time-value*，设置自动保存的时间点。*time-value* 为指定的时间点，格式为 hh:mm:ss，默认值为 00:00:00。

5. 华为 VRP 的基本配置命令

由于我们之前已经学习过思科 IOS 的命令行，因此首先把 IOS 与 VRP 的最重要的命令关键字作简单的对比，以便于学习掌握。

VRP 命令	解释
display	显示
undo	删除/取消
local-user	新建用户
return	前者返回特权用户模式；后者返回用户视图
quit	返回上级模式或视图
rip	启动 rip 进程
ospf	启动 ospf 进程
bgp	启动 bgp 进程
sysname	设置设备的主机名
acl	配置控制访问列表
save	保存配置
delete	删除配置
simple	明文形式的密码
cipher	密文形式的密码
ip host	host 名字和 ip 地址映射
link-protocol	配置链路层封装协议
display version	显示版本
display current-configuration	显示当前配置
display saved-configuration	显示已保存的配置
ctrl+z	前者返回特权用户模式；后者返回/用户视图

（1）设置设备的主机名
设置交换机的主机名可在系统视图下，通过 sysname 配置命令来实现，其用法为：
sysname *自定义设备名*
默认情况下，华为交换机、路由器的默认主机名为 Huawei。
（2）设置设备的系统时钟
设置时区：clock timezone time-zone-name { add | minus } offset

相对于 UTC 时间，正向偏移用 add，负向偏移选 minus，offset 为具体的偏移值。例如位于北京的设备可以这样设置时区：clock timezone BJ add 8:00 设置日期时间：clock datetime HH:MM:SS YYYY-MM-DD

（3）设置设备的 IP 地址

配置 IP 地址的命令行为：ip address *ip-address* {*mask*|*mask-length*} [sub]

其中 *ip-address* 为指定接口的 IP 地址；*mask* 为指定所设置的 IP 地址对应的子网掩码；*mask-length* 为指定所设置的 IP 地址对应的子网掩码长度；sub 指定设置的 IP 地址为从 IP 地址，如果不选择此可选项，则设置的 IP 地址为主 IP 地址。

（4）设置端口工作模式

配置端口工作模式的命令行为：port link-type {trunk|access|hybrid}

（5）设置 trunk 允许的 VLAN

设置 trunk 链路允许的 VLAN 的命令行为：port trunk allow-pass vlan {*vid*|all}

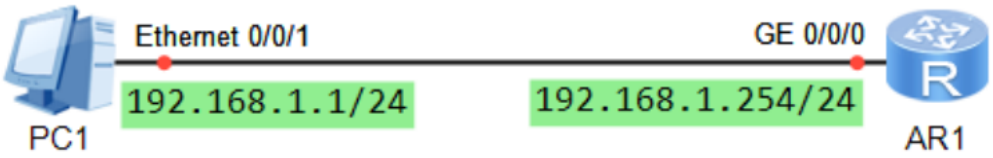
（6）设置当前端口的默认 VLAN（即 pvid）

设置当前端口的默认 VLAN 的命令行为：port default vlan *vid*

四、 实验步骤

6. 使用 eNSP 完成第一个实验

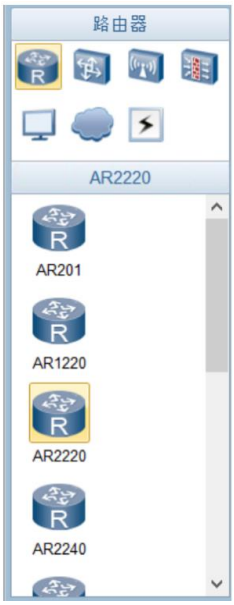
（1）实验拓扑



网络拓扑如上图所示：一台 PC 与一台路由器通过网线直连。现在我们要在 eNSP 中搭建这个实验环境，并且完成 PC 和路由器的配置，使得 PC 能够访问路由，即能够 ping 通路由器。PC1 配置的 IP 地址是 192.168.1.1/24，其默认网关的 IP 地址是 192.168.1.254；路由器 AR1 的 GE0/0/0 接口配置的 IP 地址是 192.168.1.254/24。

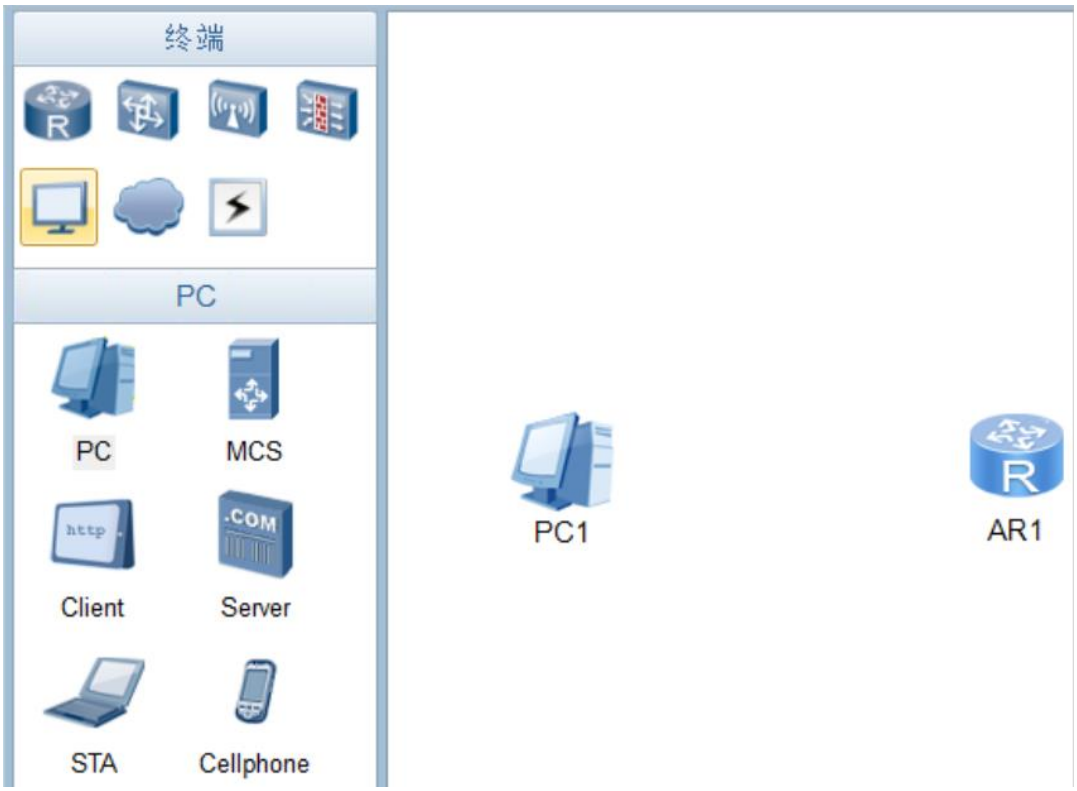
（2）环境搭建

使用 eNSP 搭建上述环境是非常简单的。新建一个拓扑后，从左侧的设备列表中选择“路由器”：

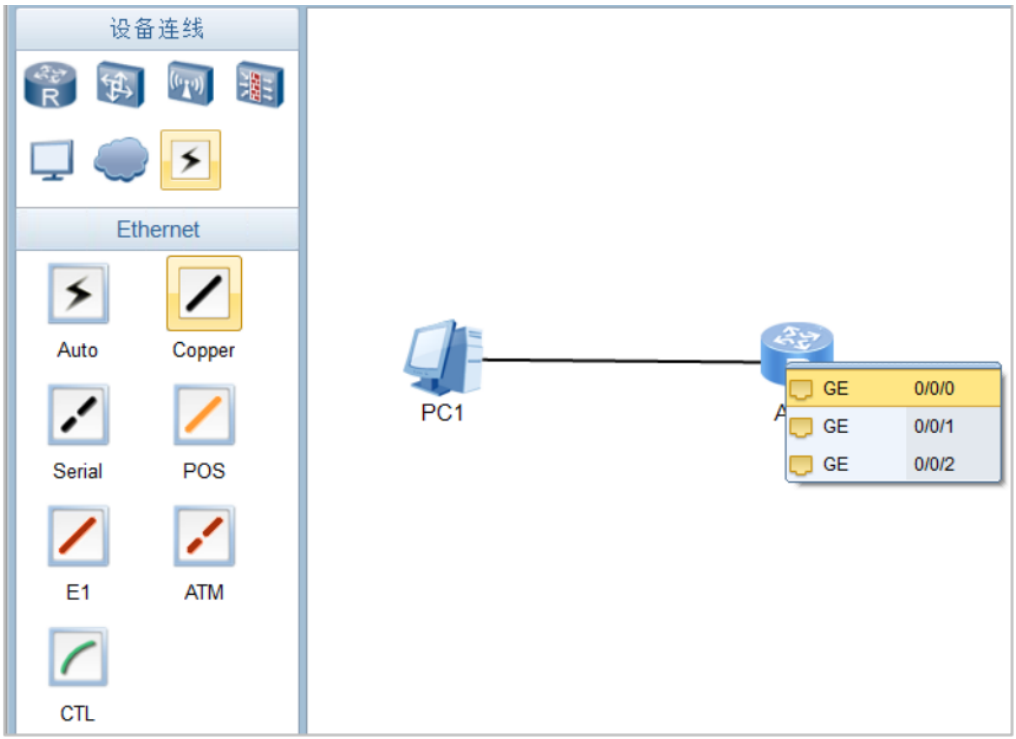


此处有多种型号的路由器可以选择，在本实验中，我们选取的是 AR2220，因此用鼠标左键按住 AR2220 的图标不放，然后将其拖到工作区中再松开即可：

接下来继续添加 PC，点击设备列表里的“终端”类型，选择 PC，然后拖放到工作区上：



接下来完成设备的连线，在左侧设备列表中选择“设备连线”图标，在线缆列表中选择铜缆“Copper”，点选成功后，鼠标指针会发生变化，随后在路由器和 PC 上分别点击并选择相应的互联接口，即可实现设备相应端口之间的连线：

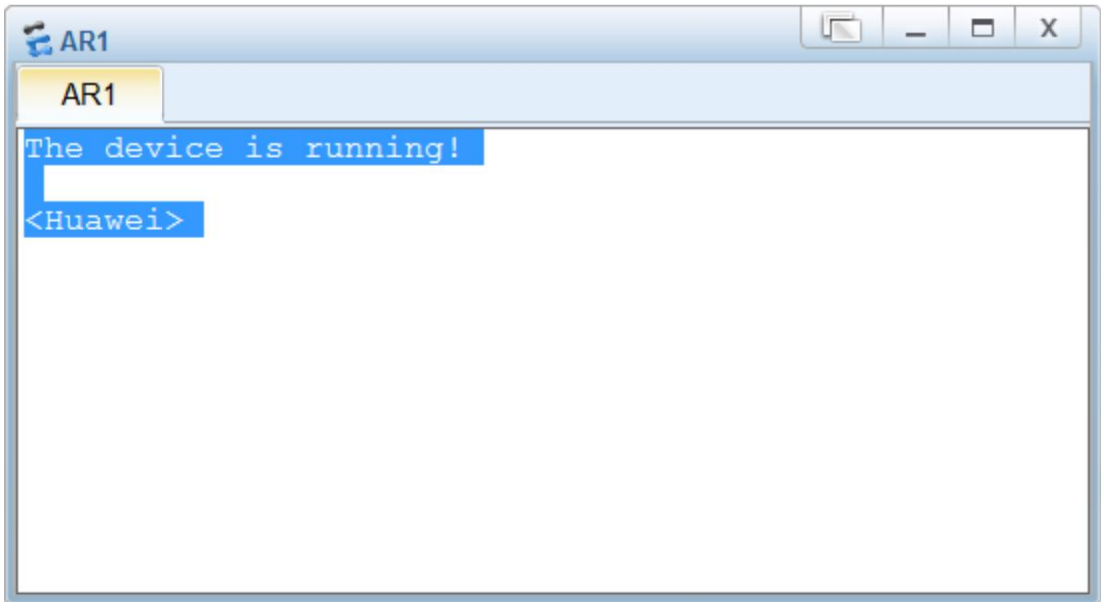


这样网络拓扑就已经搭建完成了，现在点击工具栏上的启动按钮：

点击上面的按钮后将启动所有设备。如果实验拓扑比较大，建议不要使用上面的启动按钮同时启动设备，可以对设备进行逐台启动，也就是分别对设备点击右键，然后选择“启动”菜单。待所有设备都启动完毕后即可开始实验。

（3）配置实现

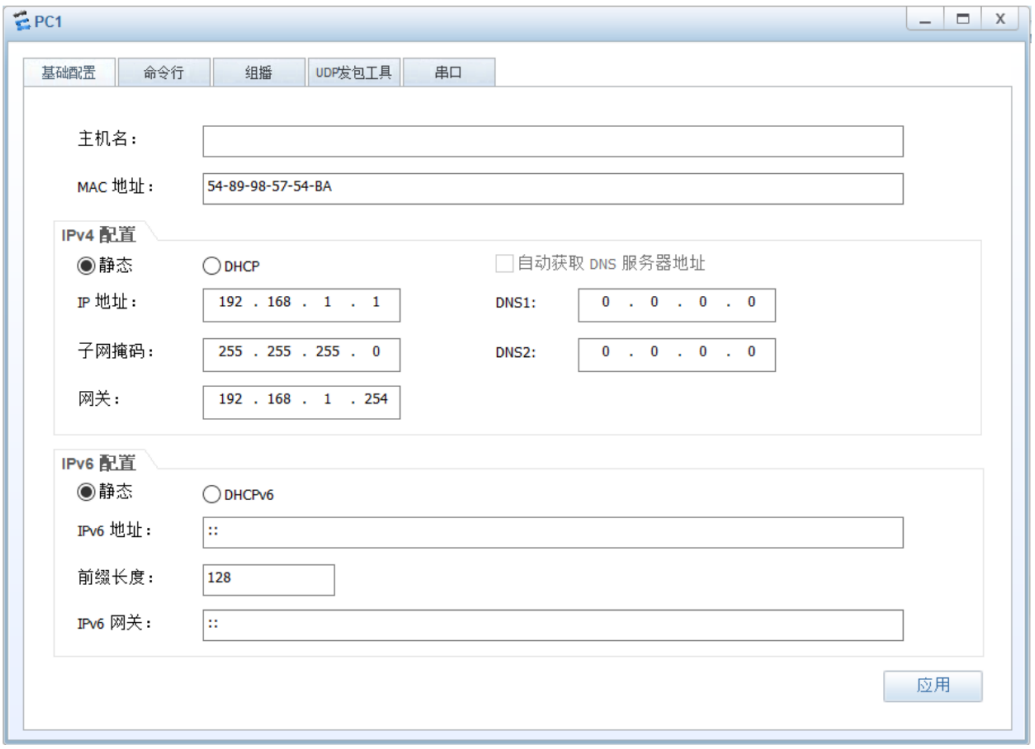
首先完成路由器的配置，双击路由器的图标即可打开命令行界面：



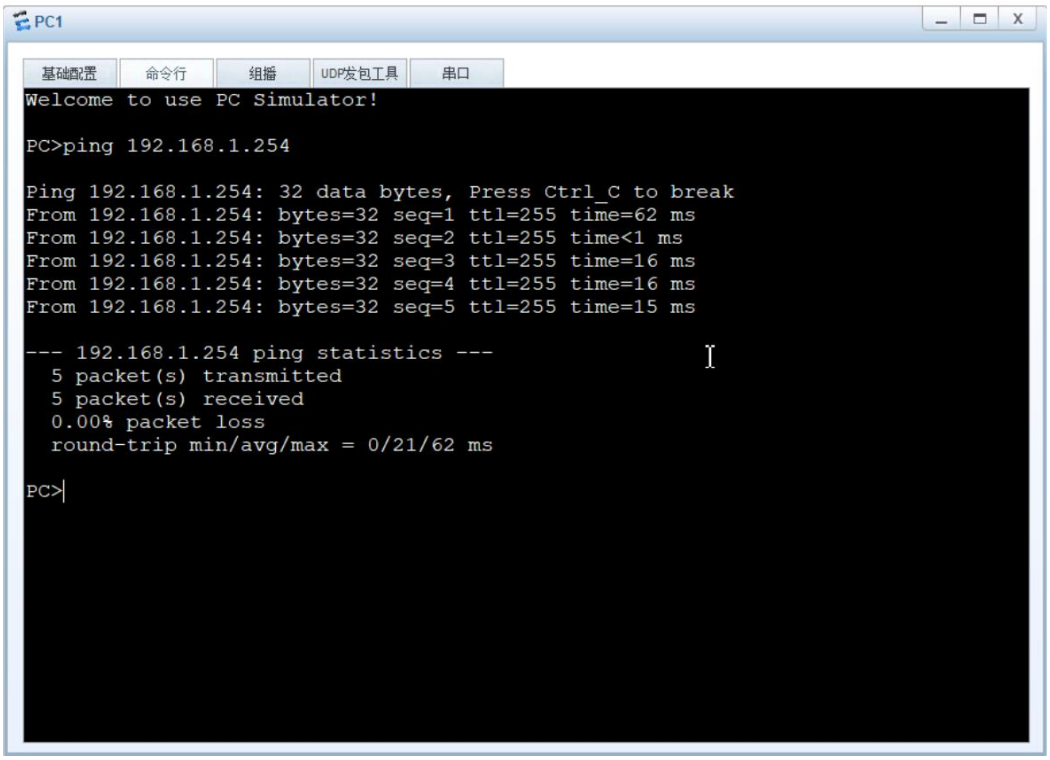
在该命令行界面中完成对路由器的基本配置，如下：

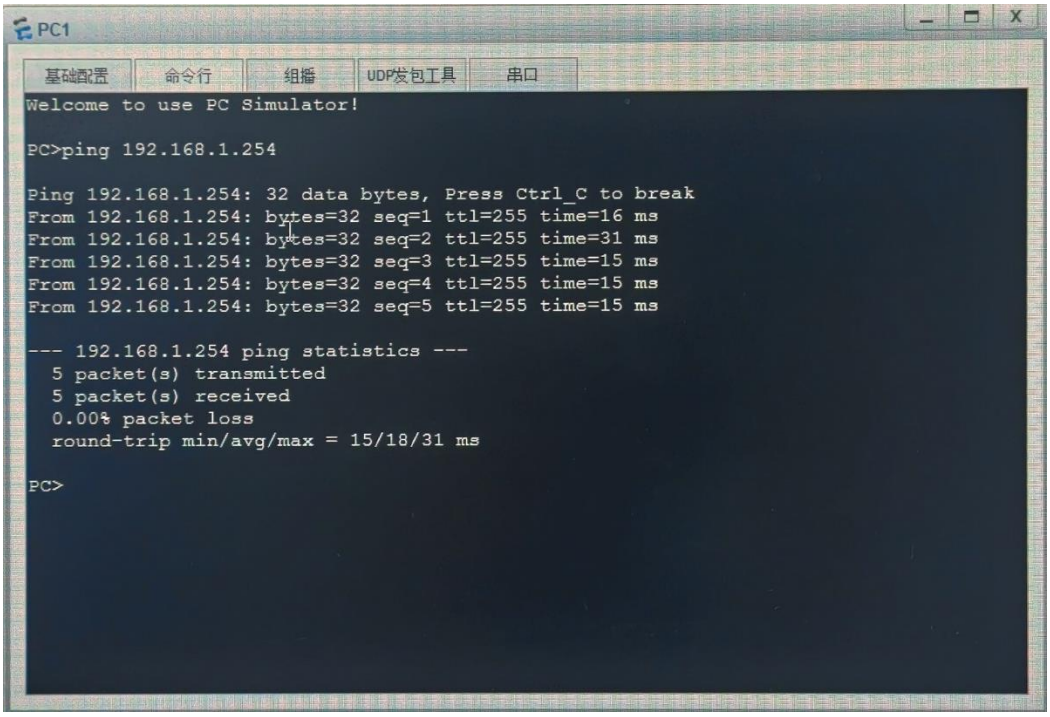
```
<Huawei> system-view
Enter system view, return user view with Ctrl+Z.
[Huawei] interface GigabitEthernet 0/0/0
[Huawei-GigabitEthernet0/0/0] ip address 192.168.1.254 24
```

现在开始配置 PC，双击 PC1，在出现的配置界面中如下填写参数：



填写完成之后，点击“应用”即可。接下来就可以进行连通性测试了。双击 PC 的图标，选择“命令行”选项卡，然后就能看到 CMD 界面，在 CMD 界面中可以进行基本的 ping、tracert 等操作，例如测试 PC 到路由器的联通性，可以 ping 192.168.1.254：





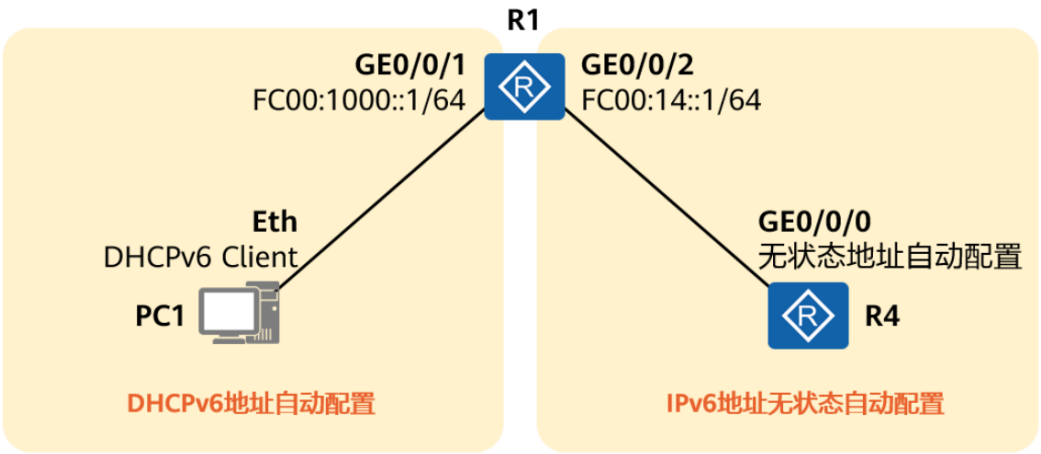
从 ping 的回显能看到，PC1 现在能够 ping 通路由器了，至此实验已经成功了。eNSP 能够保存实验拓扑及配置以便下次继续进行操作，非常方便。如果需要保存实验环境以及拓扑中各设备的配置，则在完成实验操作后，先为实验拓扑中每台设备（PC、Client 无需做这个操作）使用 save 命令保存配置（注意，务必要先在设备的 CLI 界面中使用 save 命令保存配置）：

<huawei> save

然后再点击 eNSP 工具栏的“保存”按钮将拓扑及配置文件保存在指定目录即可。

7. IPv6 地址配置实验

（1）实验拓扑



在本实验中，R1 是一台网关路由器，它通过两个物理接口分别连接物联网终端 R4（通过一台路由器模拟）及 PC1。【注】R1 及 R4 推荐使用 AR2220 及以上设备。

（2）实验要求

- 完成 R1 的 IPv6 基础配置。
- 在 R1 的 GE0/0/2 接口上启动 RA 报文通告，使得物联网终端 R4 的 GE0/0/0 接口能够通过无状态自动配置获取 IPv6 地址。

- 在 R1 的 GE0/0/1 接口上部署 DHCPv6，使得 PC1 能够通过 DHCPv6 协议自动获取 IPv6 地址。

（3）配置思路

- 完成 R1 的 IPv6 基础配置。
- 完成 IPv6 地址无状态自动配置。
- 完成 DHCPv6 部署与配置。
- 测试 IPv6 网络联通性。

（4）操作步骤

- 完成 R1 的 IPv6 基础配置

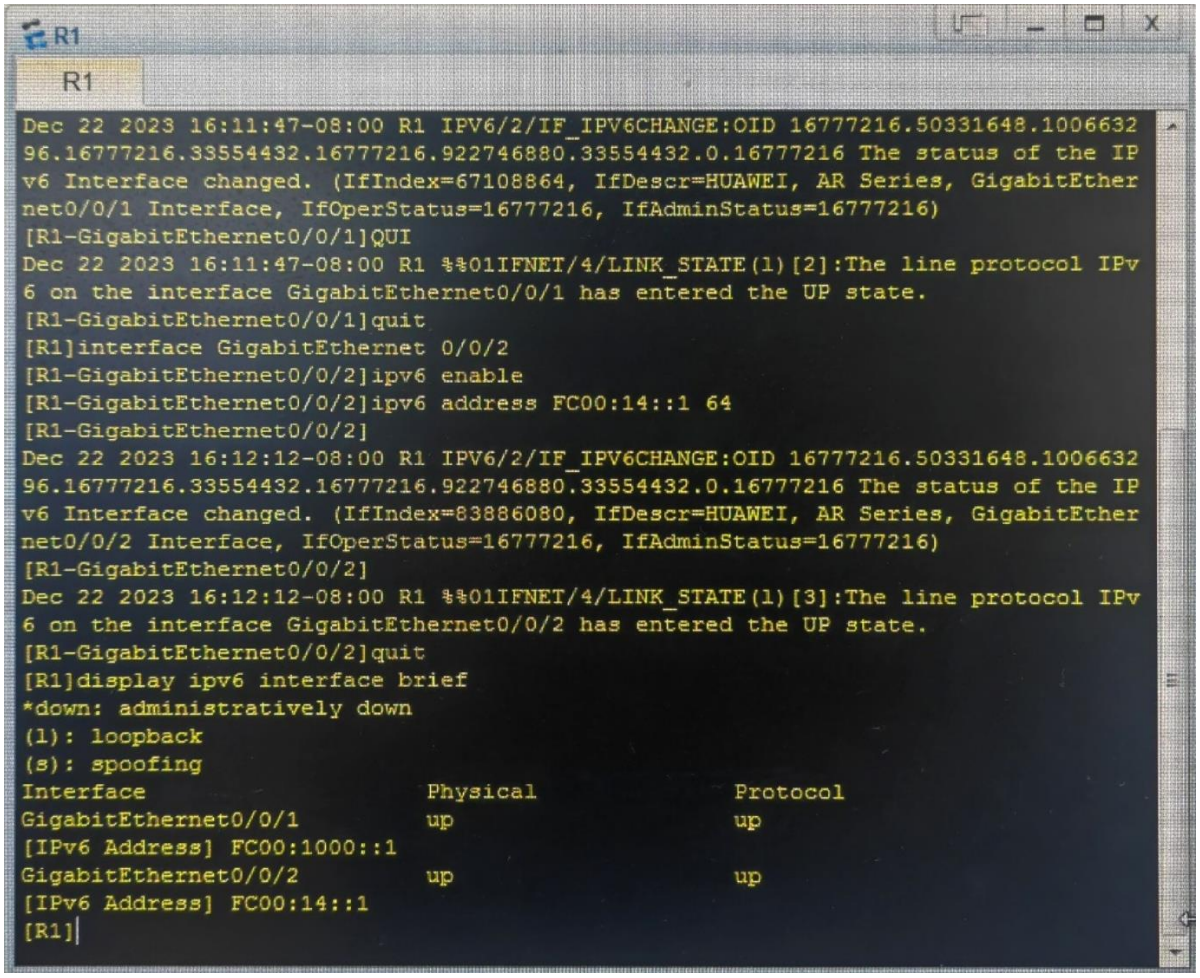
在 R1 上完成如下配置：

```
<Huawei> system-view
[Huawei] sysname R1
[R1] ipv6 #全局使能 IPv6
[R1] interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1] ipv6 enable #在接口上使能 IPv6
[R1-GigabitEthernet0/0/1] ipv6 address FC00:1000::1 64 # 手工配置 IPv6 地址
[R1-GigabitEthernet0/0/1] quit
[R1] interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2] ipv6 enable
[R1-GigabitEthernet0/0/2] ipv6 address FC00:14::1 64
[R1-GigabitEthernet0/0/2] quit
```

以上配置展示的是在华为路由器上通过手工方式配置静态 IPv6 地址的过程。在多数情况下，网络设备的 IPv6 地址需要固定，因此大多采用手工配置的方式为设备配置静态 IPv6 地址。

完成上述配置后，R1 的 GE0/0/1 与 GE0/0/2 接口便获得了静态 IPv6 地址。此时可在 R1 上查看 IPv6 接口地址信息，在设备上执行 display ipv6 interface brief 命令可查看设备的 IPv6 接口信息，其中包括接口 IPv6 地址、接口物理状态及协议状态：

```
[R1] display ipv6 interface brief
*down: administratively down
(l): loopback
(s): spoofing
Interface Physical Protocol
GigabitEthernet0/0/1 up up
[IPv6 Address] FC00:1000::1
GigabitEthernet0/0/2 up up
[IPv6 Address] FC00:14::1
```



从以上输出可以看到，R1 的 GE0/0/1 及 GE0/0/2 接口已经分别获得了对应的 IPv6 地址。

- 完成 IPv6 地址无状态自动配置

在 R1 上完成如下配置：

```
[R1] interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2] undo ipv6 nd ra halt
```

在华为路由器上 ipv6 nd ra halt 命令用来取消使能设备发布 RA 报文功能，缺省情况下，设备发布 RA 报文功能处于未使能状态，可以认为该命令缺省时已经被配置在接口上了。在本实验中，我们需要在 R1 的 GE0/0/2 接口上发布 RA 报文，因为 RA 报文中携带 IPv6 地址前缀信息，通过 RA 报文的通告，R4 可解析出报文中携带的 IPv6 地址前缀，并且使用该前缀结合 R4 本地生成的接口 ID 构造一个 IPv6 地址，这个过程被称为无状态地址自动配置。为实现这个功能，需要在 R1 的 GE0/0/2 接口上使能发布 RA 报文功能，即执行 undo ipv6 ndra halt 命令。

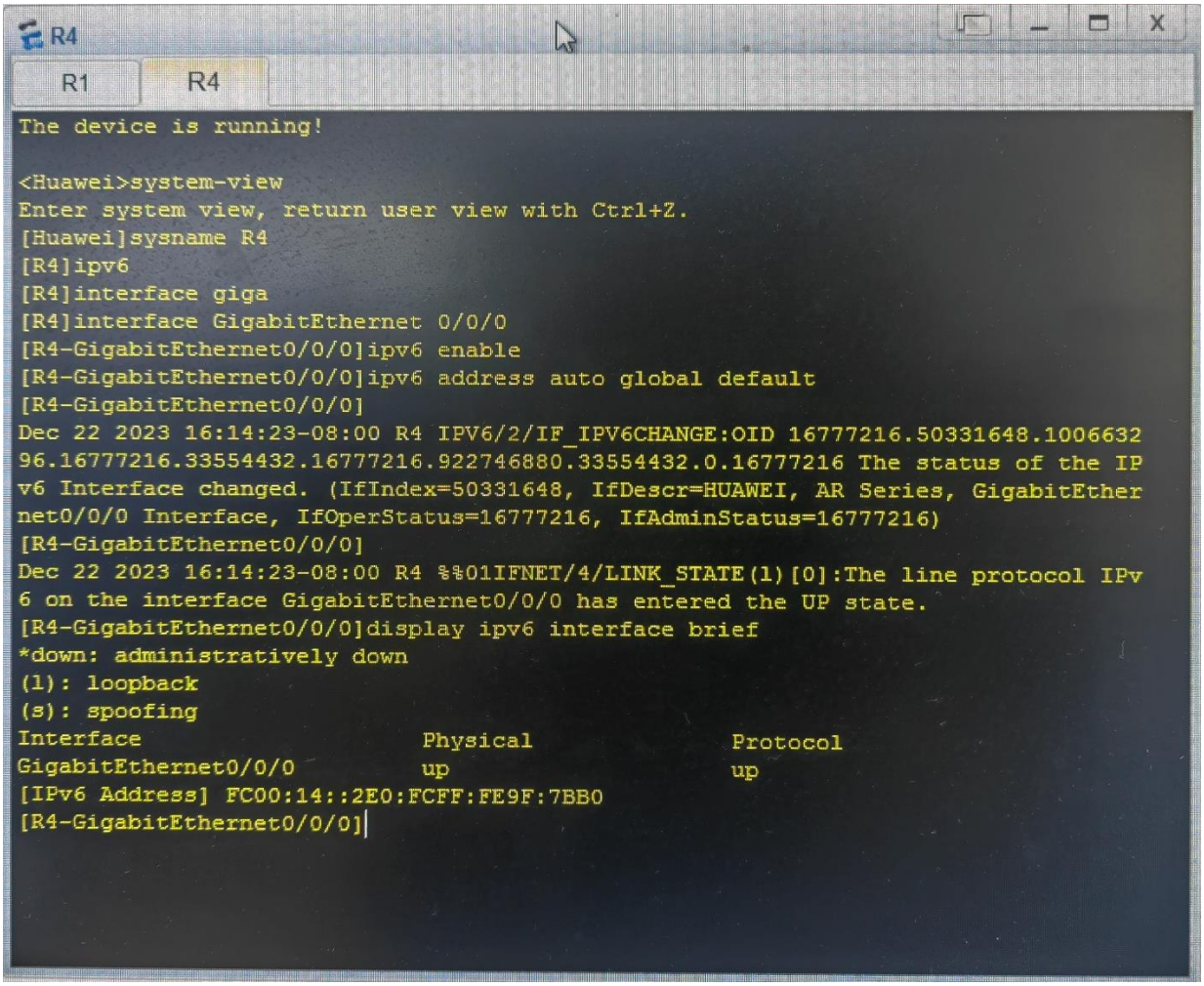
在物联网终端 R4 上完成如下配置：

```
<Huawei> system-view
[Huawei] sysname R4
[R4] ipv6
[R4] interface GigabitEthernet 0/0/0
[R4-GigabitEthernet0/0/0] ipv6 enable
[R4-GigabitEthernet0/0/0] ipv6 address auto global default
```

在以上配置中，ipv6 address auto global 命令用来使能无状态自动生成 IPv6 全局地址功能，命令末尾的 default 关键字用于指定学习缺省路由，这样一来 R4 在收到 RA 报文生成 IPv6 地址同时，还可以学习 RA 报文中的源 IPv6 地址，并且把它作为 IPv6 缺省路由的下一跳地址。

完成上述配置后，R4 便会通过 IPv6 无状态地址自动配置功能在 GE0/0/0 接口上自动配置一个 IPv6 地址。在 R4 上查看 IPv6 接口地址信息：

```
[R4] display ipv6 interface brief
*down: administratively down
(l): loopback
(s): spoofing
Interface Physical Protocol
GigabitEthernet0/0/0 up up
[IPv6 Address] FC00:14::2E0:FCFF:FECB:6980
```

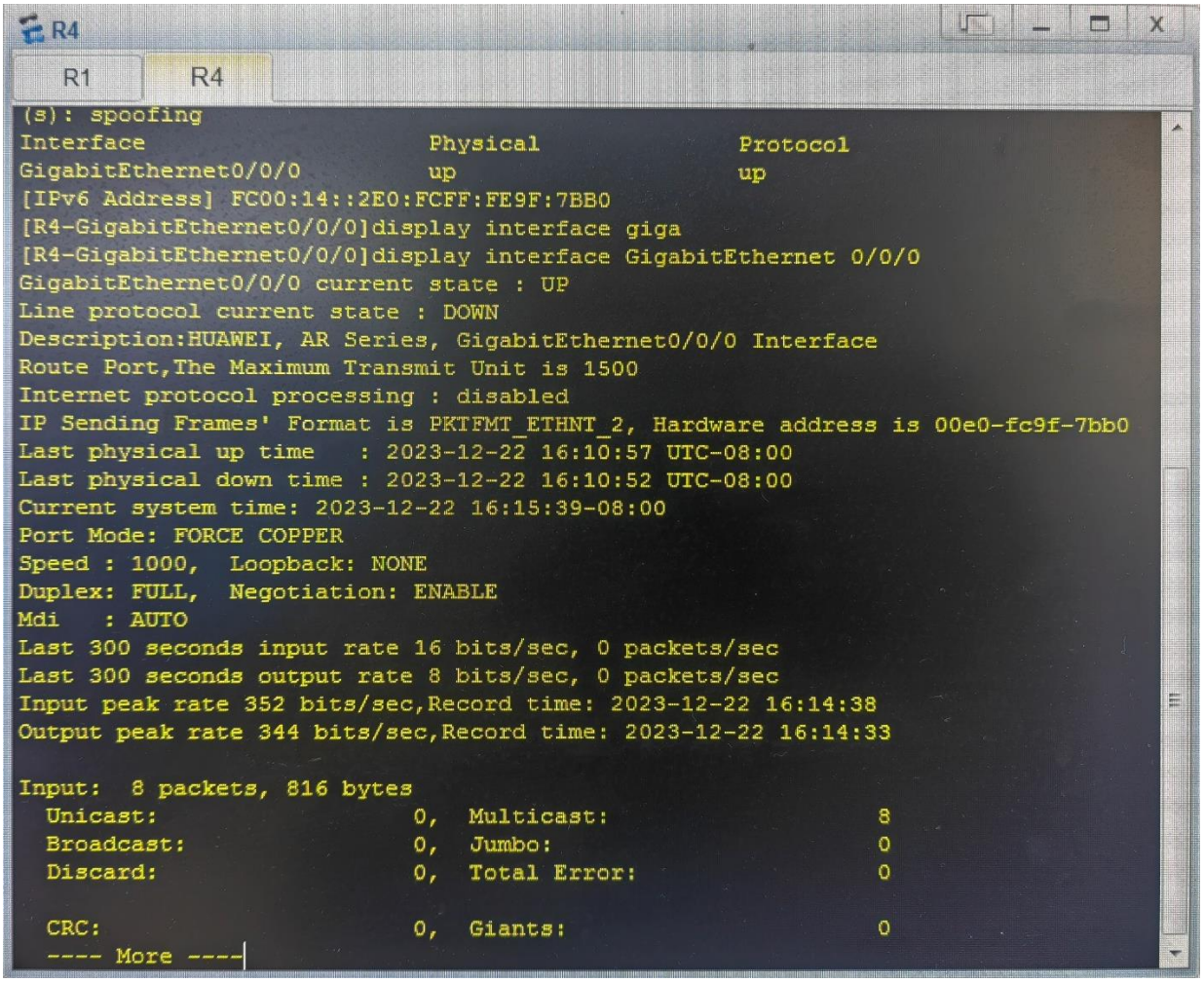


值得一提的是，R1 通告给 R4 的 RA 报文中缺省时携带的 IPv6 地址前缀是前者 GE0/0/2 接口的 IPv6 地址前缀 FC00:14::/64，R4 将该 64bit 前缀与自己本地 GE0/0/0 接口 IPv6 接口 ID（2E0:FCFF:FECB:6980）构成了一个 IPv6 地址：FC00:14::2E0:FCFF:FECB:6980。其中接口 ID 2E0:FCFF:FECB:6980 是 R4 在 GE0/0/0 接口上根据接口 MAC 地址自动生成的，采用的是 EUI-64 规范，该规范可确保生成的接口 ID 唯一。

查看 R4 的 GE0/0/0 接口的相关信息：

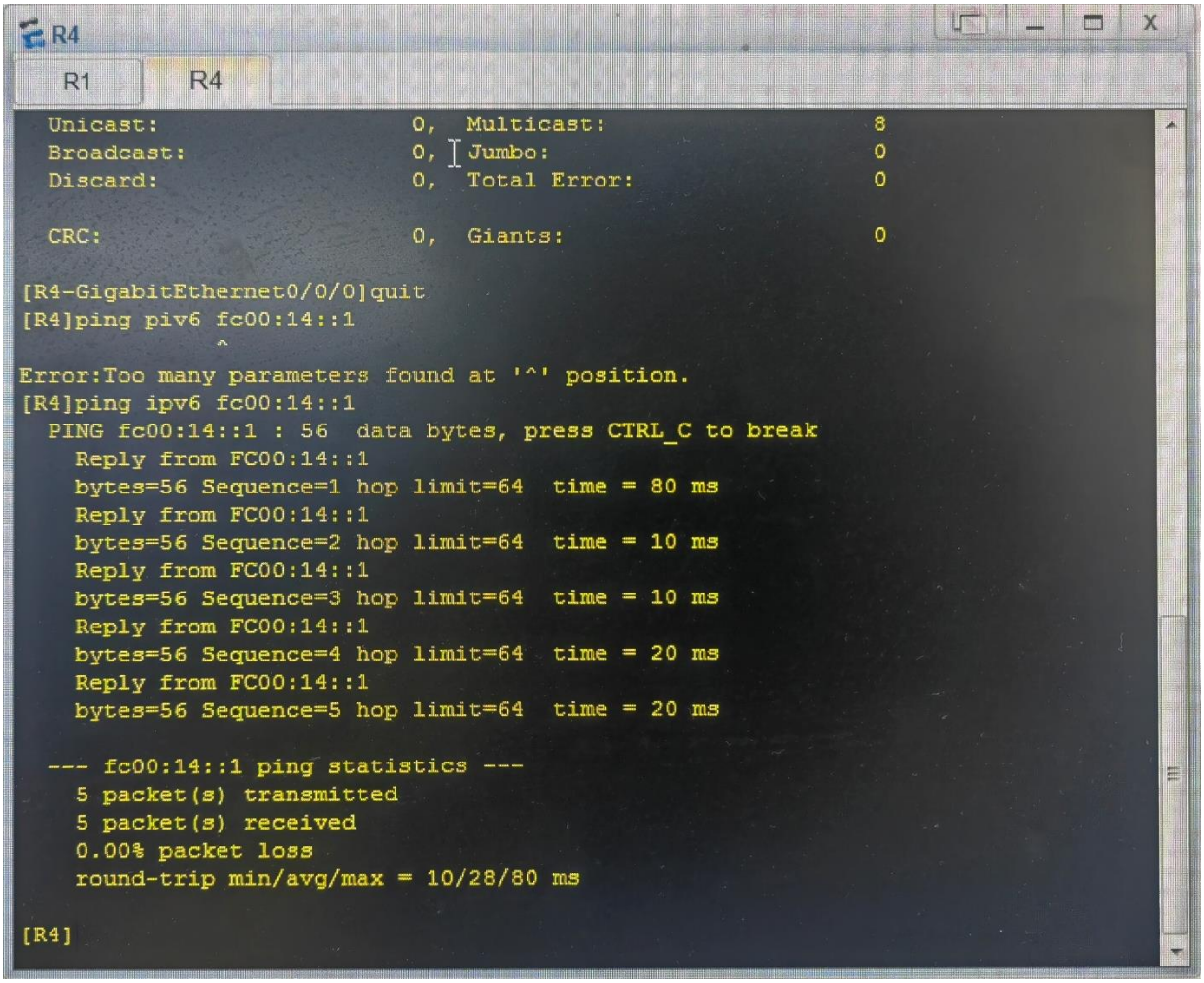
```
[R4] display interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 current state : UP
Line protocol current state : DOWN
Description:HUAWEI, AR Series, GigabitEthernet0/0/0 Interface
Route Port,The Maximum Transmit Unit is 1500
Internet protocol processing : disabled
```

```
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 00e0-
fccb-6980
Last physical up time : 2021-11-04 23:18:47 UTC-08:00
Last physical down time : 2021-11-04 23:18:37 UTC-08:00
……
```



在 R4 上 ping R1，可以看到 R4 已经能够与 R1 成功通信：

```
[R4] ping ipv6 fc00:14::1
PING fc00:14::1 : 56 data bytes, press CTRL_C to break
Reply from FC00:14::1
bytes=56 Sequence=1 hop limit=64 time = 80 ms
Reply from FC00:14::1
bytes=56 Sequence=2 hop limit=64 time = 30 ms
Reply from FC00:14::1
bytes=56 Sequence=3 hop limit=64 time = 30 ms
Reply from FC00:14::1
bytes=56 Sequence=4 hop limit=64 time = 20 ms
Reply from FC00:14::1
bytes=56 Sequence=5 hop limit=64 time = 20 ms
--- fc00:14::1 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 20/36/80 ms
```



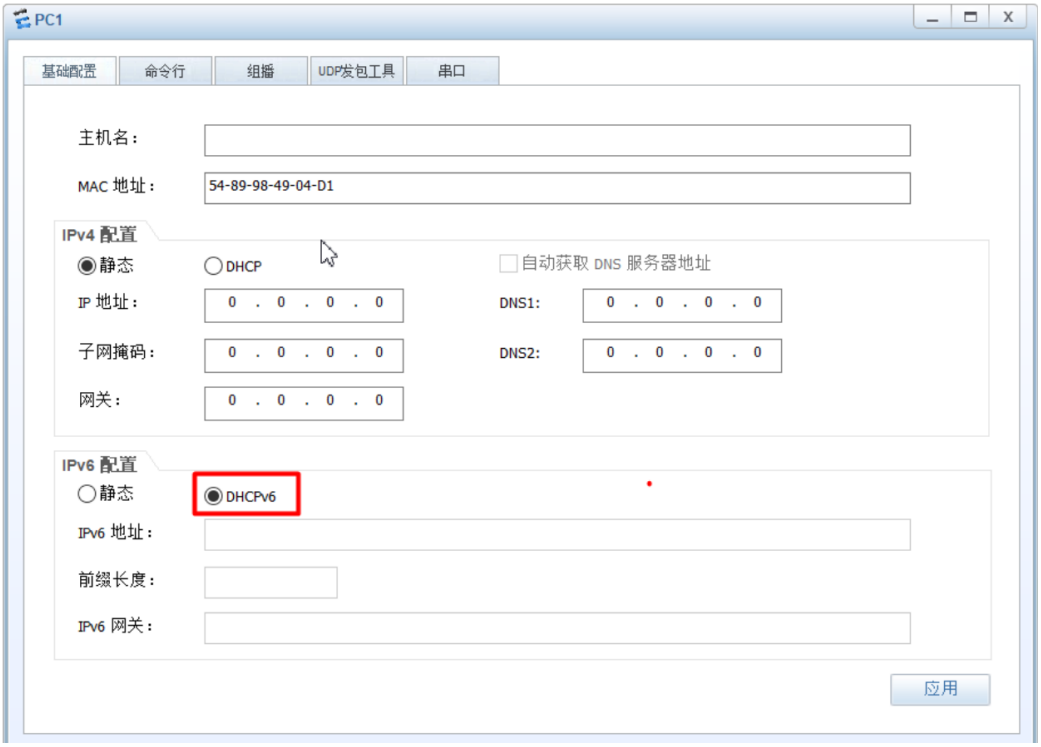
- 完成 DHCPv6 部署与配置

在 R1 上完成如下配置：

```
[R1] dhcp enable #使能 DHCP 服务
[R1] dhcpv6 pool pool1 #创建 IPv6 地址池
[R1-dhcpv6-pool-pool1] address prefix fc00:1000::/64
[R1-dhcpv6-pool-pool1] excluded-address fc00:1000::1
[R1-dhcpv6-pool-pool1] quit
[R1] interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1] dhcpv6 server pool1 #在接口上绑定地址池
[R1-GigabitEthernet0/0/1] quit
```

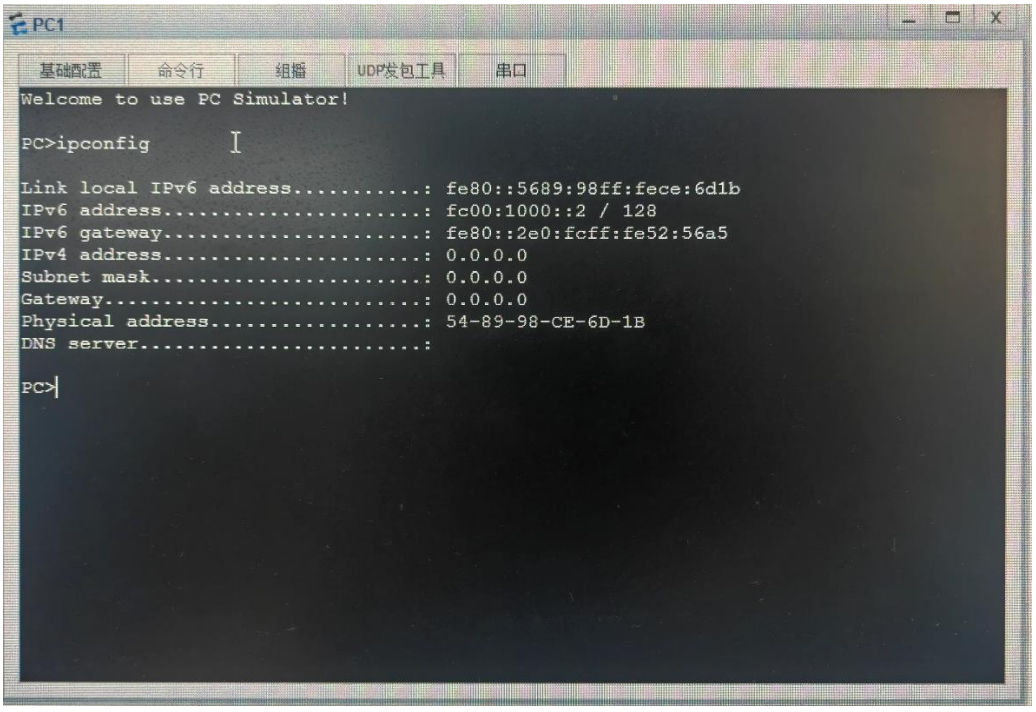
在以上配置中，我们在 R1 上创建了一个 DHCPv6 地址池 pool1，并为该地址池配置了地址前缀 FC00:1000::/64，该前缀与 R1 的 GE0/0/1 接口使用的前缀相同，此外，地址池 pool1 还将 R1 的 GE0/0/1 接口已使用的 IPv6 地址排除在池外，以防止这个地址被分配给其他设备。最后，该地址池被绑定在 GE0/0/1 接口上，此后，R1 的 GE0/0/1 接口将响应 DHCPv6 请求。

接下来配置 PC1 的以太网卡，开启 DHCPv6 客户端功能：



点击“应用”按钮完成上述配置后，在 PC1 的配置界面上选择“命令行”选项卡，然后执行 ipconfig 查看网卡信息：

```
PC> ipconfig
Link local IPv6 address...........: fe80::5689:98ff:fe49:4d1
IPv6 address......................: fc00:1000::2 / 128
IPv6 gateway......................: fe80::2e0:fcff:fe31:2798
IPv4 address......................: 0.0.0.0
Subnet mask.......................: 0.0.0.0
Gateway...........................: 0.0.0.0
Physical address..................: 54-89-98-49-04-D1
DNS server........................:
```

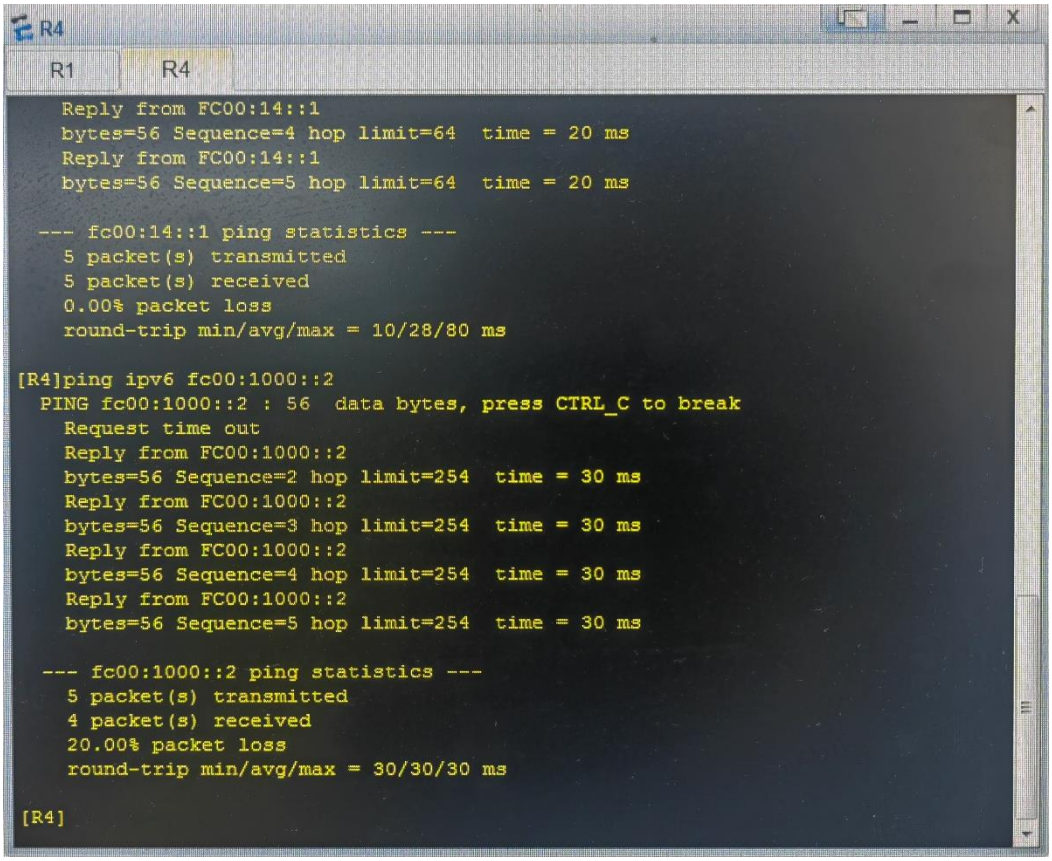


从上述输出可以看到，PC1 已经通过 DHCPv6 获取到了地址 FC00:1000::2。

- 测试 IPv6 网络联通性。

在 R4 上执行如下命令，测试到达 PC1 的连通性，可以发现二者已经可以正常通信：

```
<R4> ping ipv6 fc00:1000::2
PING fc00:1000::2 : 56 data bytes, press CTRL_C to break
Request time out
Reply from FC00:1000::2
bytes=56 Sequence=2 hop limit=254 time = 30 ms
Reply from FC00:1000::2
bytes=56 Sequence=3 hop limit=254 time = 30 ms
Reply from FC00:1000::2 bytes=56 Sequence=4 hop limit=254 time = 20ms
Reply from FC00:1000::2
bytes=56 Sequence=5 hop limit=254 time = 30 ms
--- fc00:1000::2 ping statistics ---
5 packet(s) transmitted
4 packet(s) received
20.00% packet loss
round-trip min/avg/max = 20/27/30 ms
```



（5）参考配置

- R1 的参考配置如下：

```
sysname R1
#
snmp-agent local-engineid 800007DB03000000000000
snmp-agent
#
clock timezone China-Standard-Time minus 08:00:00
```

```
#
portal local-server load portalpage.zip
#
drop illegal-mac alarm
#
ipv6
#
set cpu-usage threshold 80 restore 75
#
dhcp enable
#
dhcpv6 pool pool1
address prefix FC00:1000::/64
excluded-address FC00:1000::1
#
aaa
authentication-scheme default
authorization-scheme default
accounting-scheme default
domain default
domain default_admin
local-user admin password cipher %$%$K8m.Nt84DZ}e#<0`8bmE3Uw}%$%$
local-user admin service-type http
#
firewall zone Local
priority 15#
interface GigabitEthernet0/0/0
#
interface GigabitEthernet0/0/1
ipv6 enable
ipv6 address FC00:1000::1/64
dhcpv6 server pool1
#
interface GigabitEthernet0/0/2
ipv6 enable
ipv6 address FC00:14::1/64
undo ipv6 nd ra halt
#
interface NULL0
#
user-interface con 0
authentication-mode password
user-interface vty 0 4
user-interface vty 16 20
#
wlan ac
#
return
```

- R4 的参考配置如下：

```
sysname R4
#
```

```
snmp-agent local-engineid 800007DB03000000000000
snmp-agent
#
clock timezone China-Standard-Time minus 08:00:00
#
portal local-server load portalpage.zip
#
drop illegal-mac alarm
#
ipv6
#
set cpu-usage threshold 80 restore 75
#
aaa
authentication-scheme default
authorization-scheme default
accounting-scheme defaultdomain default
domain default_admin
local-user admin password cipher %$%$K8m.Nt84DZ}e#<0`8bmE3Uw}%$%$
local-user admin service-type http
#
firewall zone Local
priority 15
#
interface GigabitEthernet0/0/0
ipv6 enable
ipv6 address auto global default
#
interface GigabitEthernet0/0/1
#
interface GigabitEthernet0/0/2
#
interface NULL0
#
user-interface con 0
authentication-mode password
user-interface vty 0 4
user-interface vty 16 20
#
wlan ac
#
return
```

（6）思考题

- IPv6 无状态地址自动配置与 DHCPv6 地址自动配置的区别是？

回答：

1. 配置方式：IPv6 无状态地址自动配置通过设备的接口 ID 和 IPv6 地址前缀来自动构建唯一地址，而不需要服务器的参与。而 DHCPv6 则需要借助服务器来进行地址分配。

2. 地址分配：在 IPv6 无状态地址自动配置中，如果一个设备移到新的子网，它将会自动构建一个新的地址，而不需要从服务器获取。而在 DHCPv6 中，设备需要从服务器获取新的地址。

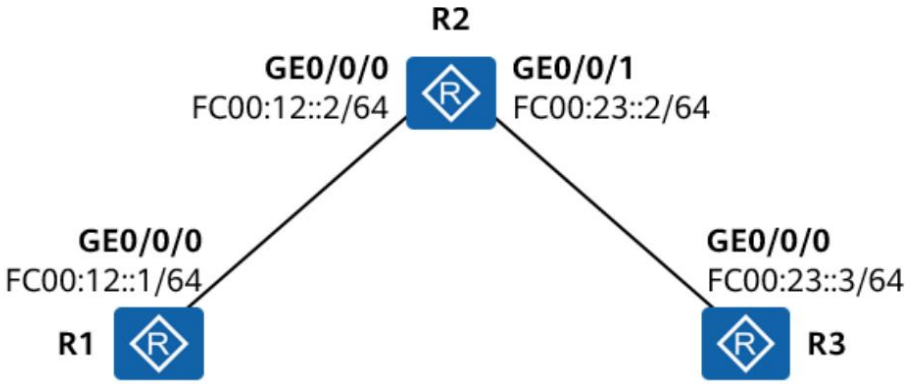
3. 可扩展性：对于大型网络，DHCPv6 可能会成为瓶颈，因为每个设备都需要与服务器进行交互。相比之下，IPv6 无状态地址自动配置的扩展性更好。
4. 安全性：DHCPv6 提供了一些安全性功能，例如地址租约期限和地址冲突检测。而 IPv6 无状态地址自动配置没有这些功能。
5. 应用场景：IPv6 无状态地址自动配置适用于固定设备和少量移动设备的场景，例如家庭网络或小型企业网络。而对于大量移动设备的场景，例如大型企业或运营商网络，DHCPv6 可能更为适用。

- 在本实验中，我们使用路由器作为 IPv6 无状态地址自动配置的客户端，它依据什么规范生成的 IPv6 接口 ID 并在获取 IPv6 地址前缀后最终形成单播地址？这个规范具体的操作过程是什么？

回答：EUI-64（扩展唯一标识符 64 位）规则。
具体操作过程：
1. 路由器通过识别网络中的路由器 RA（路由器宣告）信息获取网络前缀。
2. 路由器使用自己的 MAC 地址生成一个全球唯一的 IPv6 地址。
3. 路由器将自己的 IPv6 地址和网络前缀进行组合，生成一个完整的 IPv6 地址。
4. 路由器检查生成的 IPv6 地址是否与已有的地址重复。
5. 如果 IPv6 地址不重复，则路由器将其设置为自己的网络接口地址。
6. 路由器使用其他配置参数，例如 DNS 服务器地址等，完成自己的网络接口配置。

8. ICMPv6 与 NDP 实验

（1）实验拓扑



（2）实验要求

- 在本实验拓扑中完成基础 IPv6 配置，观察各类常见的 ICMPv6 报文在网络中的功能与应用。

（3）配置思路

- 完成 R2 的基础配置。
- 观察 RA 报文与无状态地址自动配置过程。
- 观察 DAD 过程。
- 观察地址解析过程。
- 捕获 Ping 报文。
- 捕获 Tracert 报文。
- 观察 IPv6 PMTUD 机制。

（4）操作步骤

- 完成 R2 的基础配置

在 R2 上完成如下配置：

```
<Huawei> system-view
[Huawei] sysname R2
```

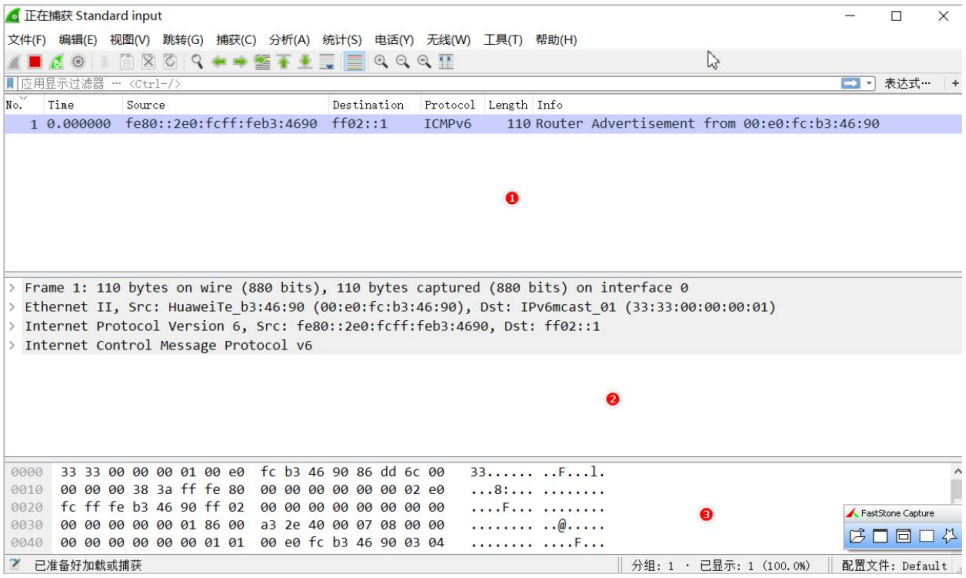
```
[R2] ipv6
[R2] interface GigabitEthernet 0/0/0
[R2-GigabitEthernet0/0/0] ipv6 enable
[R2-GigabitEthernet0/0/0] ipv6 address fc00:12::2 64
[R2-GigabitEthernet0/0/0] undo ipv6 nd ra halt
[R2-GigabitEthernet0/0/0] quit
```

在上述配置中，我们为 R2 的 GE0/0/0 接口配置了静态 IPv6 地址 FC00:12::2/64，并使能该接口发布 RA 报文的功能，此后，该接口将周期性地向外发送 RA 报文。

- 观察 RA 报文与无状态地址自动配置过程

如上图所示，在 R2 的 GE0/0/0 接口上右击鼠标键，选择 GE0/0/0 接口开始进行报文抓取，该操作将调用 WireShark 捕获 R2 的 GE0/0/0 接口上的入向与出向报文。

此时，我们将观察到如下窗口：

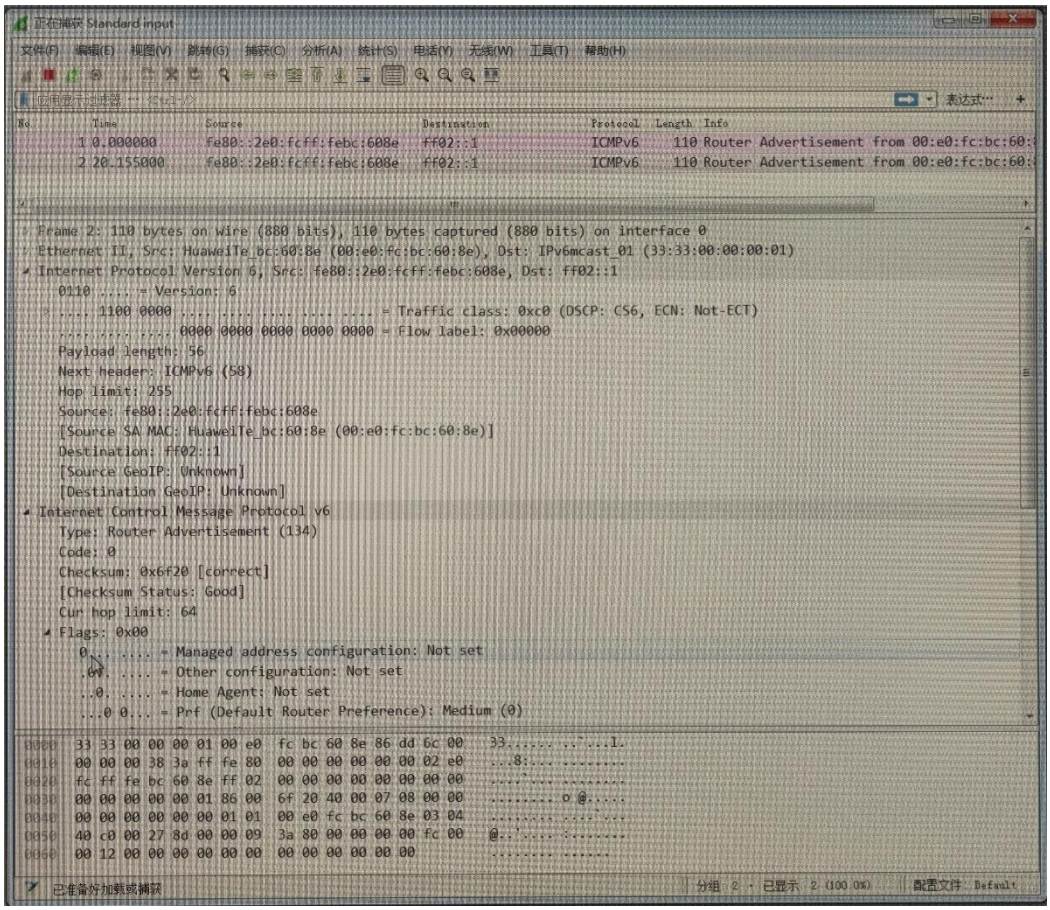


在上图所示的界面中，分栏①显示的是 Wireshark 捕获的报文列表，被选中的报文详情将出现在分栏②中，此时可在其中查看该报文的详细信息，包括二层数据帧头、三层 IPv6 头以及报文载荷，分栏③则以 16 进制形式显示报文的内容。

图中的“ICMPv6 Router Advertisement”报文便是 R2 周期性发送的 RA 报文。双击该条目，或者在分栏②中展开相应内容查看 RA 报文的详细信息如下：

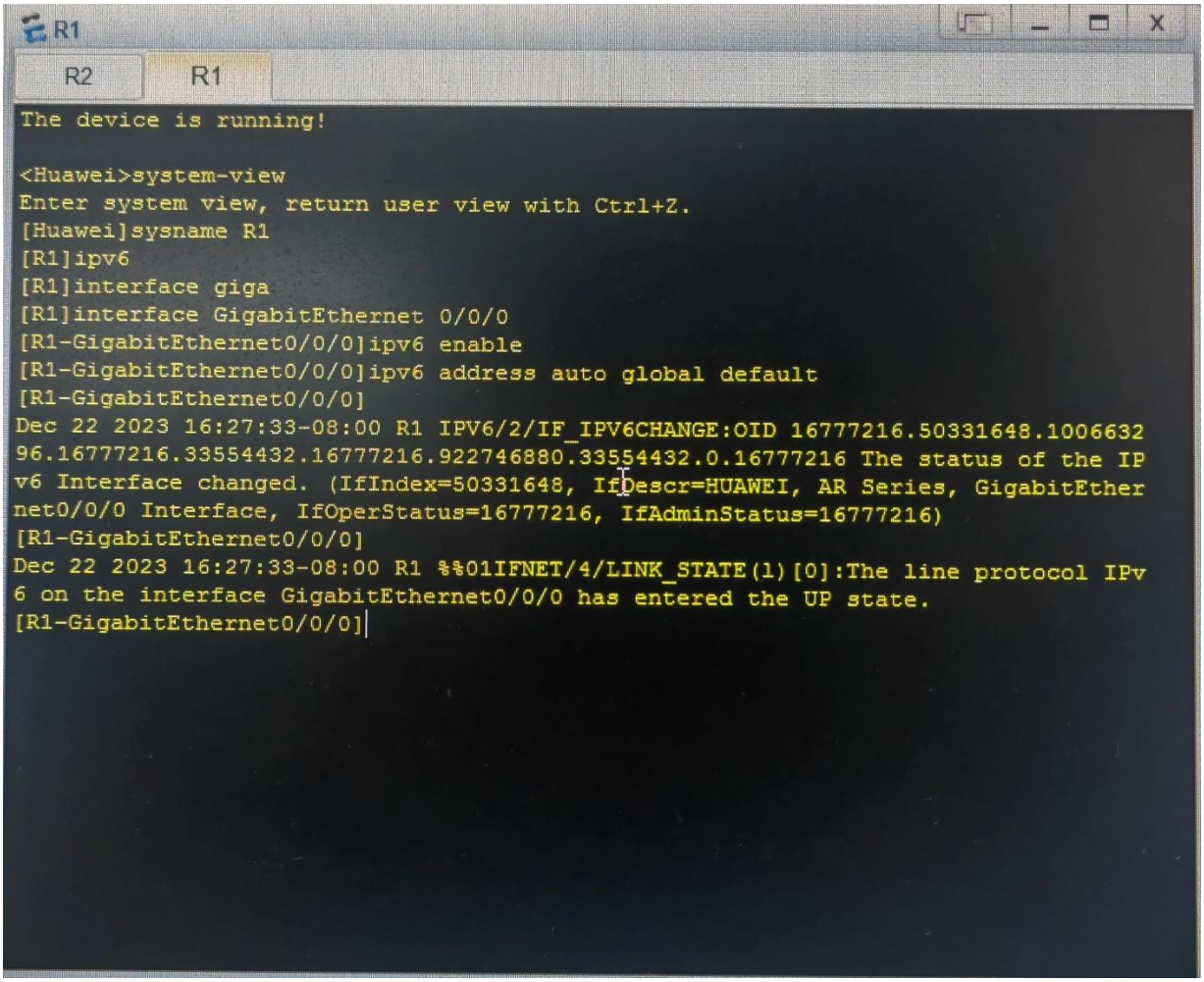
```
> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: HuaweiTe_b3:46:90 (00:e0:fc:b3:46:90), Dst: IPv6mcast_01 (33:33:00:00:00:01) ❶
v Internet Protocol Version 6, Src: fe80::2e0:fcff:feb3:4690, Dst: ff02::1 ❷
    0110 .... = Version: 6
  > .... 1100 0000 .... .... .... .... .... = Traffic class: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
    Payload length: 56
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::2e0:fcff:feb3:4690
    [Source SA MAC: HuaweiTe_b3:46:90 (00:e0:fc:b3:46:90)]
    Destination: ff02::1
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
v Internet Control Message Protocol v6 ❸
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0xa32e [correct]
    [Checksum Status: Good]
    Cur hop limit: 64
  > Flags: 0x00
    Router lifetime (s): 1800
    Reachable time (ms): 0
    Retrans timer (ms): 0
  > ICMPv6 Option (Source link-layer address : 00:e0:fc:b3:46:90)
  > ICMPv6 Option (Prefix information : fc00:12::/64)
```

如上图所示，从数据帧头①可以看出，报文的目的 MAC 地址为 33:33:00:00:00:01，这是一个组播 MAC 地址，对应组播 IPv6 目的地址 FF02::1，这个组播地址对应本链路上的所有 IPv6 节点，这表明该 RA 报文发往链路上的所有节点。从 IPv6 包头②可以看出该报文发往 FF02::1，并且 NextHeader 为 58，对应 ICMPv6，标明该头部后面跟随的是 ICMPv6 报文。从 ICMPv6 报文③可以看出该报文的类型为 134（Router Advertisement，RA）报文，且报文携带两个可选字段（ICMPv6 Option），其中一个描述 R2 的接口 MAC 地址，另一个则描述 R2 通告的 IPv6 地址前缀 FC00:12::/64，该前缀可用于实现无状态地址自动配置。

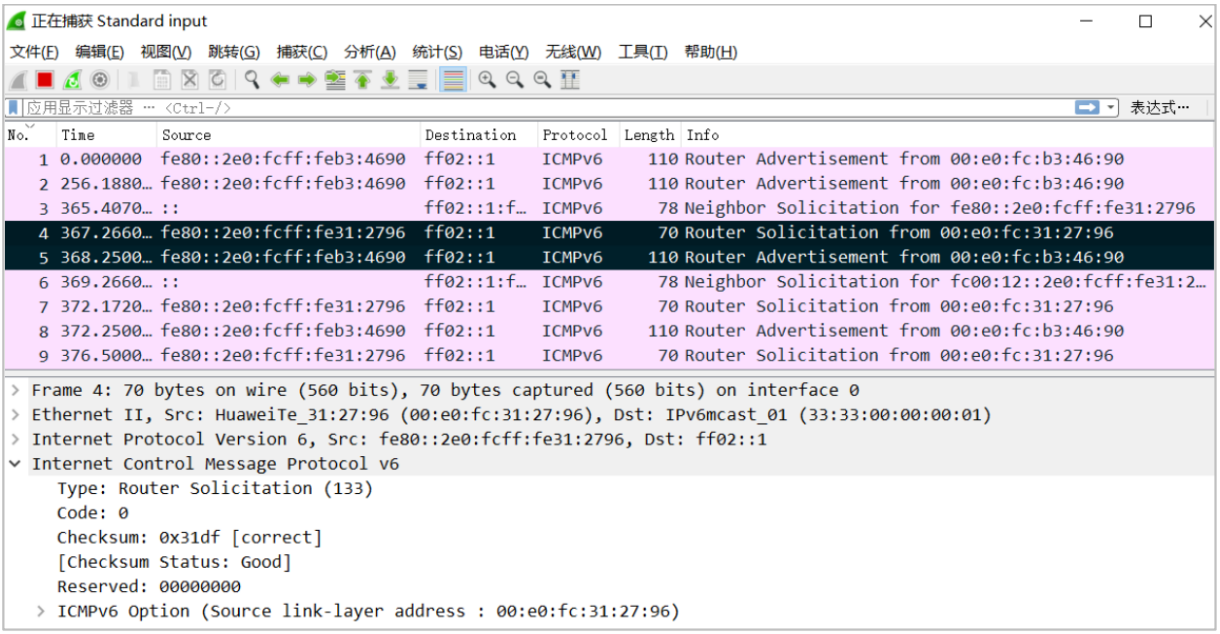


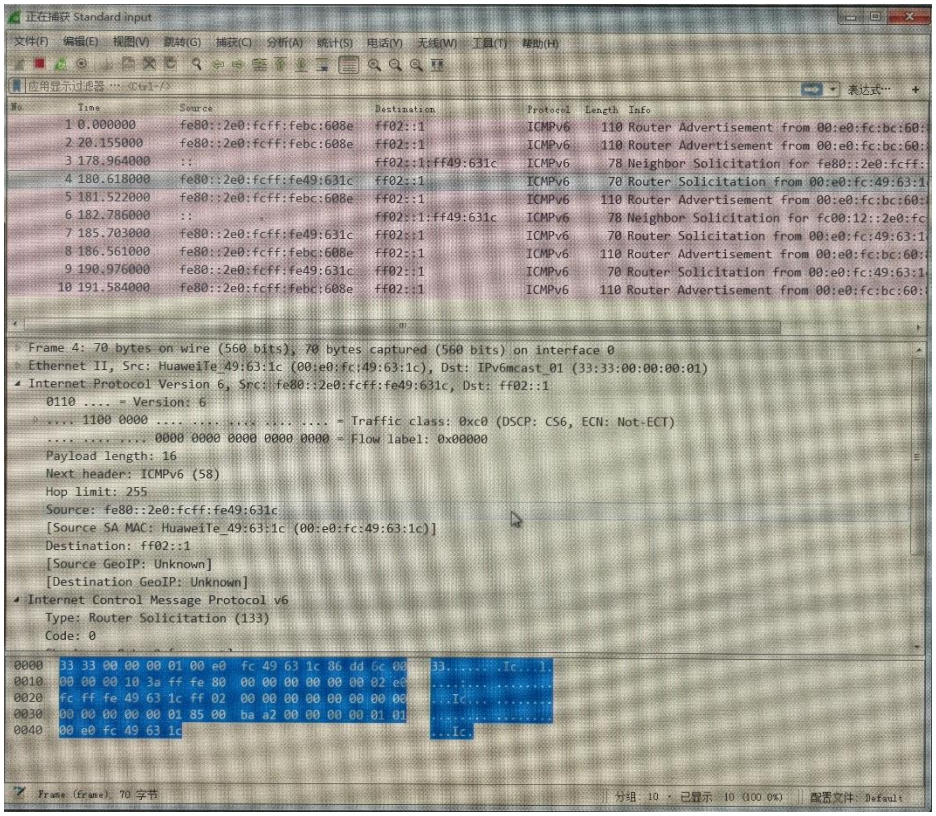
接下来我们在 R1 上配置其 GE0/0/0 接口：

```
[R1] ipv6
[R1] interface GigabitEthernet 0/0/0
[R1-GigabitEthernet0/0/0] ipv6 enable
[R1-GigabitEthernet0/0/0] ipv6 address auto global default
```



完成配置后，R1 将主动发送 RS 报文，请求 R2 发送 RA 路由器通告报文：





当然，在这个过程中，R2 依然会周期性发送 RA 报文，当 R2 收到 R1 发送的 RS 报文时，也将立即使用 RA 报文进行回应。

此时 R1 已经通过无状态地址自动配置方式获得 IPv6 地址：

```
<R1> display ipv6 interface brief
*down: administratively down
(l): loopback
(s): spoofing
Interface Physical Protocol
GigabitEthernet0/0/0 up up
[IPv6 Address] FC00:12::2E0:FCFF:FE31:2796
```

- 观察 DAD 过程

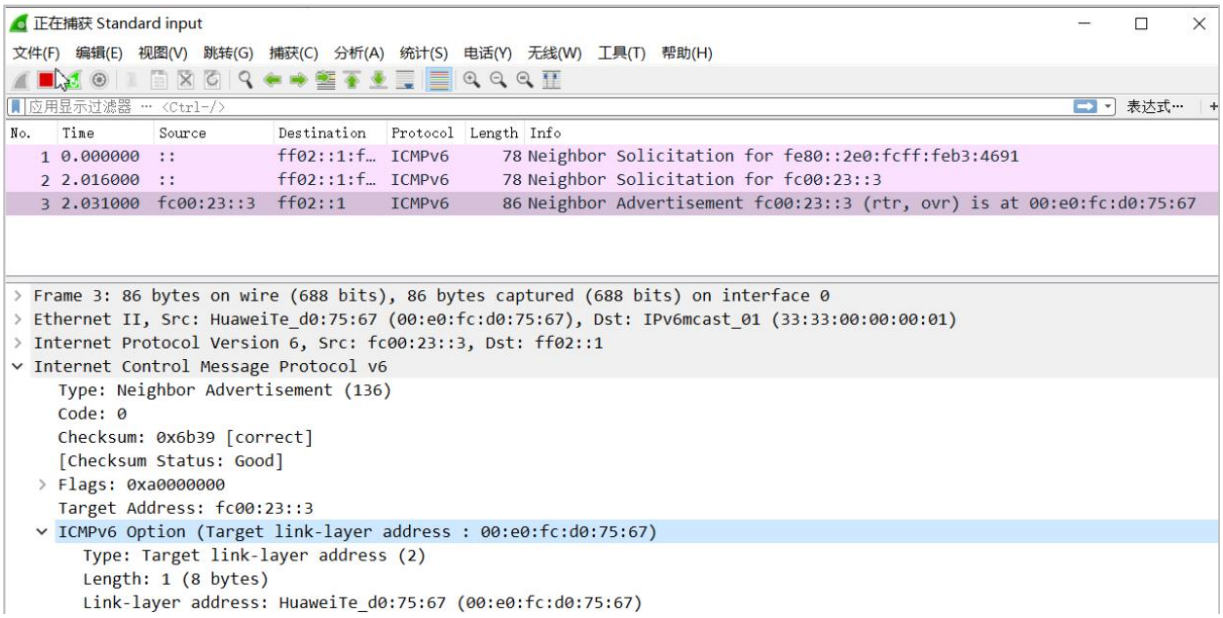
在 R3 上配置静态 IPv6 地址：

```
<Huawei> system-view
[Huawei] sysname R3
[R3] ipv6
[R3] interface GigabitEthernet 0/0/0
[R3-GigabitEthernet0/0/0] ipv6 enable
[R3-GigabitEthernet0/0/0] ipv6 address fc00:23::3 64
```

接下来在 R2 的 GE0/0/1 接口上开始捕获数据报文，然后在 R2 上完成如下配置：

```
[R2] interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1] ipv6 enable[R2-GigabitEthernet0/0/1] ipv6
address fc00:23::3 64
```

值得大家注意的是，我们在对 R2 的上述配置命令中，将其 GE0/0/1 接口的 IPv6 地址设置为 FC00:23::3，与 R3 的 GE0/0/0 的 IPv6 地址相同，从而故意制造出 IPv6 地址冲突的现象。此时可以捕获到如下报文：



从上图可以分析出，R2 在配置好该 IPv6 地址后，首先在接口上以组播方式发送一个 NS（Neighbor Solicitation, NS）邻居请求报文，如上图所示的序号（No.）为 2 的报文，这是一个 ICMPv6 报文，在 ICMPv6 载荷中写入了 DAD 探测的目标地址 FC00:23::3，此时 R3 将会收到这个 NS 报文，由于它已经使用了该地址，因此它立即回应一个 NA（Neighbor Advertisement, NA）邻居通告报文，以便告知 R2 它已经使用了该地址，在这个 ICMPv6 报文的载荷中写入了 R3 的 MAC 地址。R2 收到 NA 报文后，得知网络中已经有其他节点使用了这个 IPv6 地址，因此将该地址置为“DUPLICATE”（重复）状态，不会使用该地址进行数据通信。

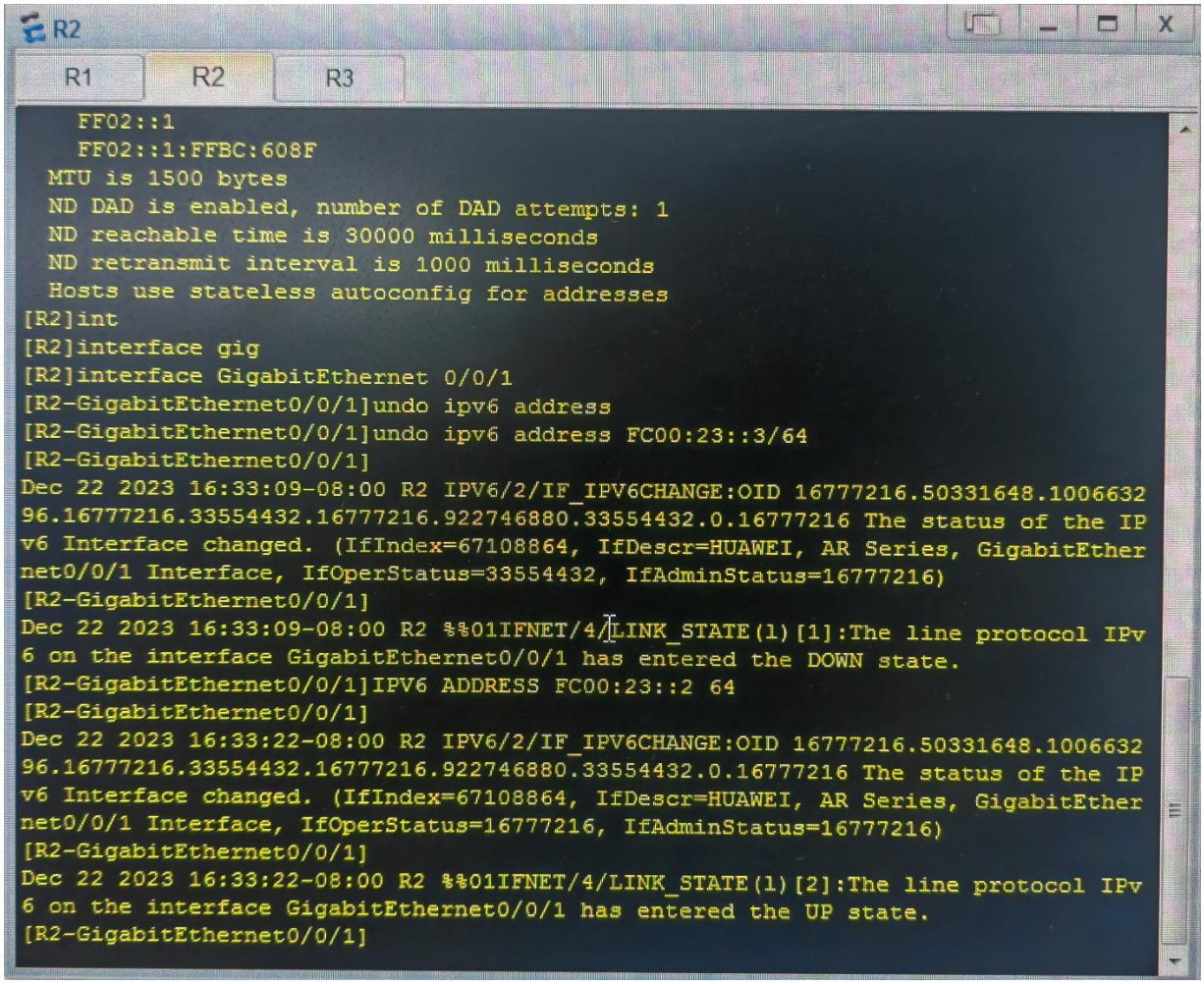
在 R2 上执行如下命令可观察到接口的地址状态：

```
<R2> display ipv6 interface GigabitEthernet 0/0/1
GigabitEthernet0/0/1 current state : UP
IPv6 protocol current state : UP
IPv6 is enabled, link-local address is FE80::2E0:FCFF:FEB3:4691
Global unicast address(es):
FC00:23::3, subnet is FC00:23::/64 [DUPLICATE]
Joined group address(es):
FF02::1:FF00:3
FF02::2
FF02::1
FF02::1:FFB3:4691
MTU is 1500 bytes
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND retransmit interval is 1000 milliseconds
Hosts use stateless autoconfig for addresses
```

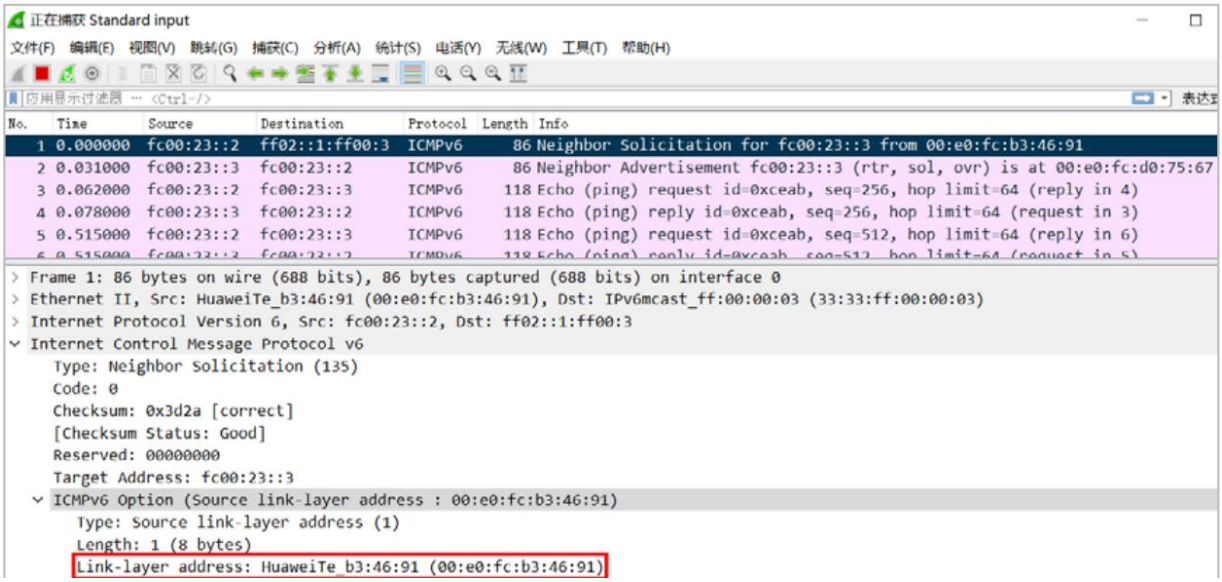
- 观察地址解析过程

现在，将 R2 的接口地址修改为正确的地址：

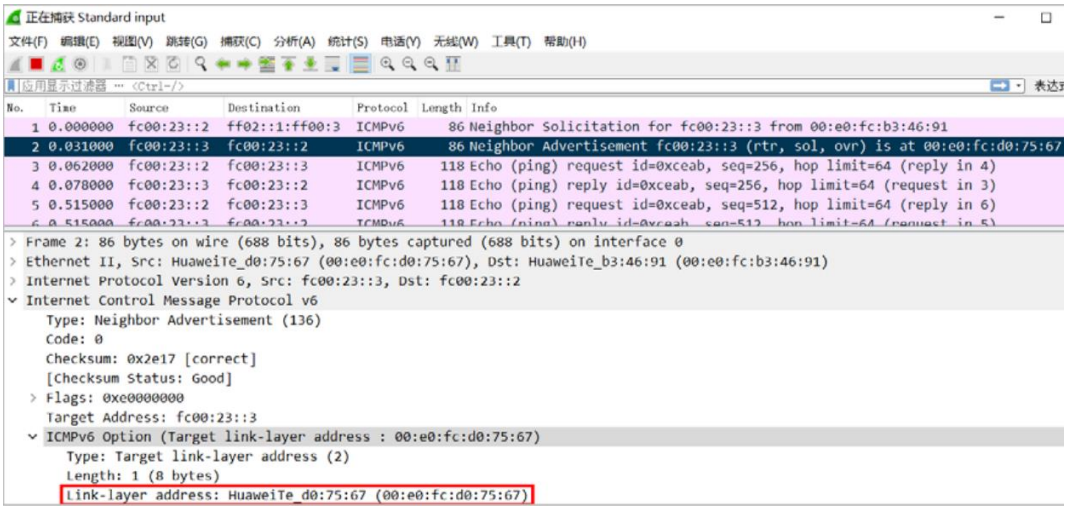
```
[R2] interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1] undo ipv6 address FC00:23::3/64
[R2-GigabitEthernet0/0/1] ipv6 address fc00:23::2 64
```



在 DAD 检测通过后，R2 正式启用 FC00:23::2 地址，此时我们依然在 R2 的 GE0/0/1 接口上进行报文抓取，然后在 R2 上 ping FC00:23::3。



从上图可以看到，R2（FC00:23::2）首先发送了一个 NS 报文，该报文的 ICMPv6 载荷中带有 R2 接口的 MAC 地址信息，这个报文发往目标地址 FC00:23::3 对应的被请求节点组播地址 FF02::1:FF00:3，R3 恰恰在侦听这个地址，于是使用 NA 报文进行回应：

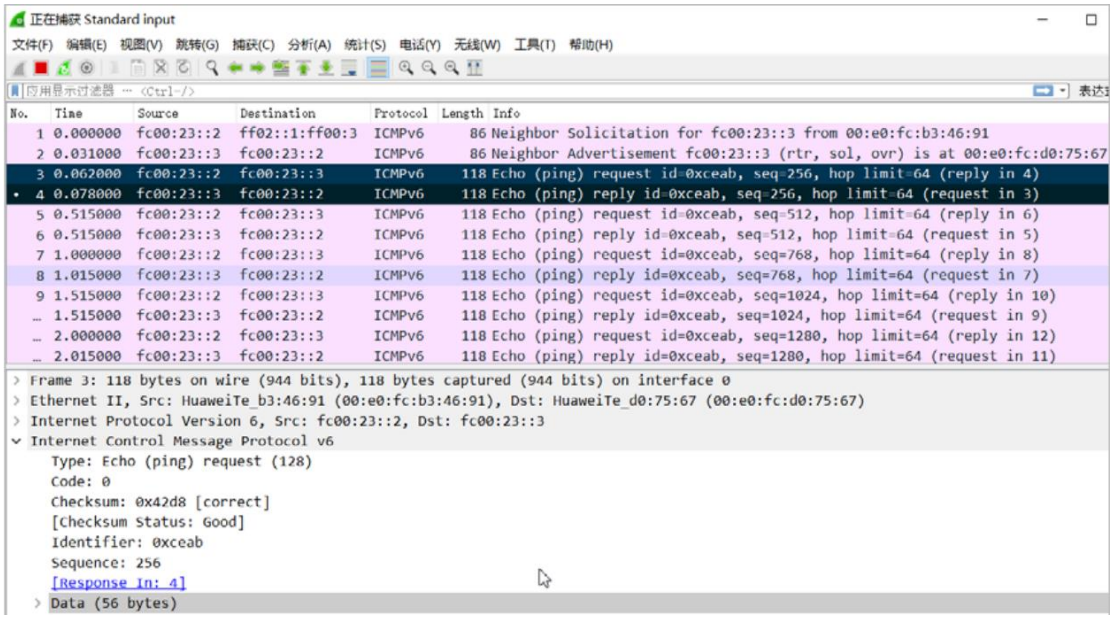


如上图所示，这个 NA 报文直接单播发给了 R2，其中填充着 R3 的接口 MAC 地址。

如此一来，R2 与 R3 便相互知晓了对方的 MAC 地址，此后就能正常交互 IPv6 报文。

- 捕获 Ping 报文

在 ICMPv6 报文中，Echo Request 和 Echo Reply 报文是非常基础且重要的报文，被用于 Ping 应用程序等，当我们在一个 IPv6 节点上执行 Ping 操作探测到某个目的地址的可达性时，实际上该应用将触发一个 ICMPv6 Echo Request 报文发往目的地址，如果收到了对方回应的 Echo Reply，则认为网络是可达的。下图展示的是当 R2 ping R3 的 FC00:23::3 地址时，捕获到的 Echo Request 和 Echo Reply 报文：



- 捕获 Tracert 报文

在网络日常运维和管理过程中，Tracert 是被广泛使用的应用程序，该应用也使用 ICMPv6 的相关报文来实现其功能。Tracert 可以帮助网络管理员检测从源节点到目的节点之间所经过的逐跳设备。

在 R3 上添加默认路由，下一跳为 R2：

```
[R3] ipv6 route-static :: 0 fc00:23::2
```

完成上述配置后，R1 与 R3 即可互通。

然后在 R1 的 GE0/0/0 接口上开始捕获报文。

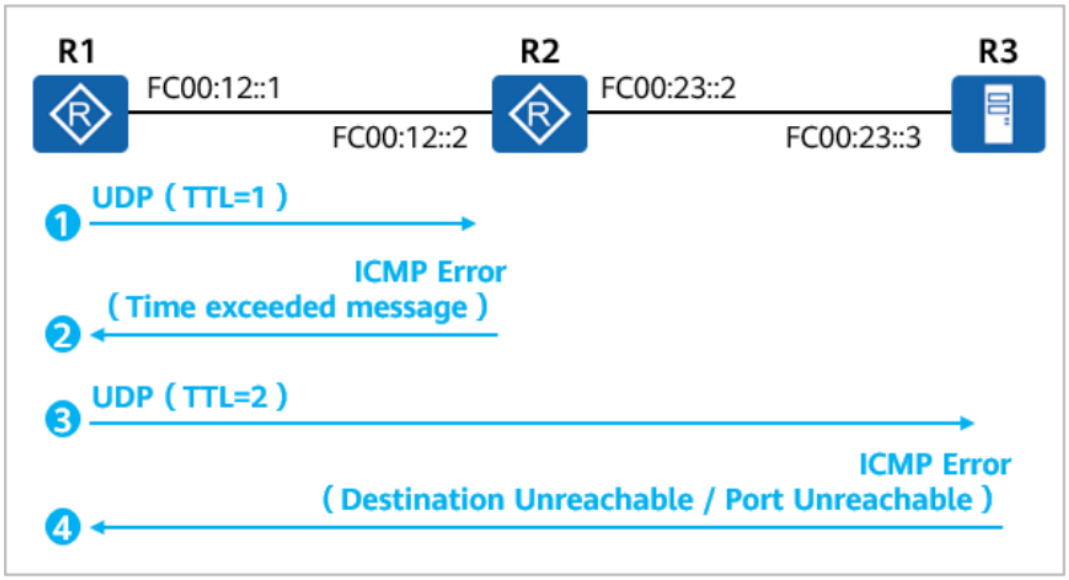
此时我们在 R1 上执行如下命令：

```
<R1> tracert ipv6 fc00:23::3
traceroute to fc00:23::3 30 hops max,60 bytes packet
```

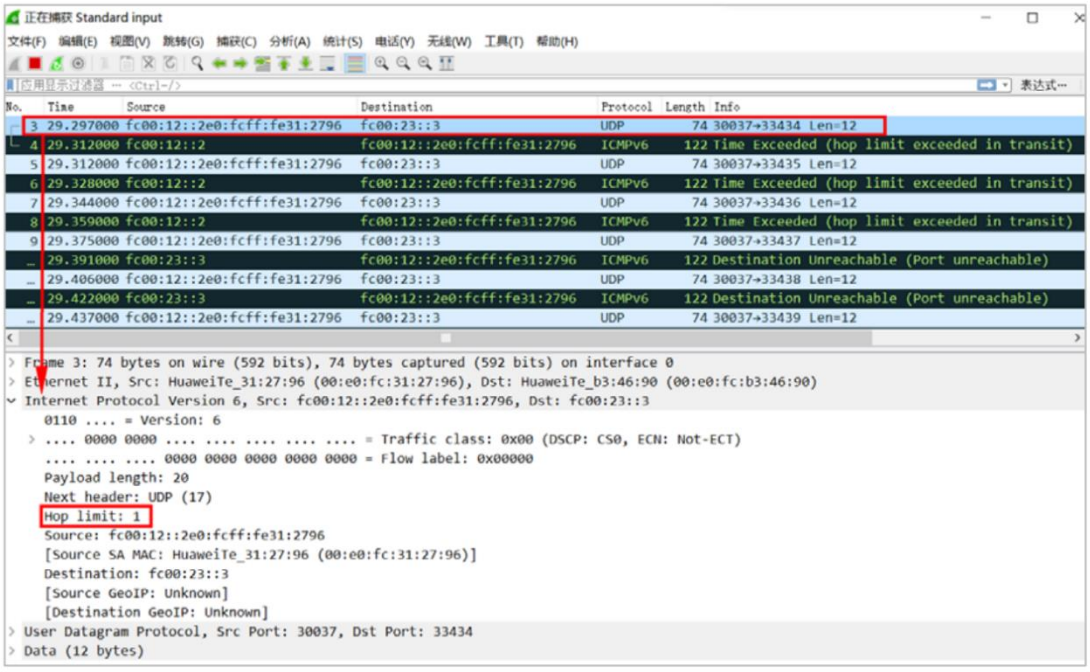
```
1 FC00:12::2 20 ms 20 ms 30 ms
2 FC00:23::3 30 ms 40 ms 20 ms
```

从上述结果可得知，从 R1 到 R3 经过了 FC00:12::2，最终到达 FC00:23::3。当源与目的节点之间存在多跳设备时，Tracert 执行的结果更加直观。因此面对一个复杂的网络时，这个工具可以方便地帮助网络管理员识别流量的转发路径。

Tracert 的实现原理如下：



R1 首先构造第一个发往目标地址 FC00:23::3 的 UDP（UDP 目的端口为特殊的 33434，该端口不会被具体的应用所使用）报文，这个报文的内容是随机填充的，没有实际意义，但是在该报文的 IPv6 头部中，R1 将 Hop Limit 字段设置为 1，这意味着报文在发出去之后，只能传递一跳。R1 可能一次会发出多个相同的 UDP 报文。如下图所示：

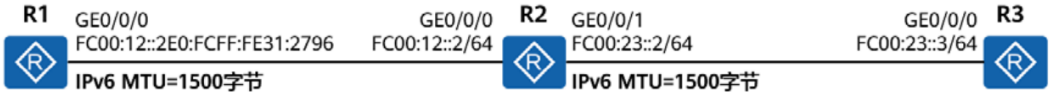


R2 收到该报文后将 Hop Limit 字段值减 1 后发现值已为 0，因此立即向 R1 发送 ICMPv6 错误消息，告知报文的生存时间截止，这个错误消息的源地址为 R2 的接口地址。R1 收到这个报错消息后，获得了第一跳设备 R2 的接口地址，然后将该地址打印在回显中。接着 R2 以 Hop Limit=2 继

续发送 UDP 报文，如此反复，直到报文到达目的地 R3。由于 R1 在 Tracert 中所使用的 UDP 端口在 R3 处并未侦听，因此 R3 回应 ICMPv6 差错报文，告知 R1 目的端口不可达。R1 收到该差错报文后即知晓最后一跳已到达。

- 观察 IPv6 PMTUD 机制

完成上述配置后，R1 与 R3 已经能够相互通信。



如上图所示，默认时路由器接口的 IPv6 MTU 值为 1500 字节。如果此时在 R1 的 GE0/0/0 接口上抓取报文，并在 R1 上执行 ping ipv6 -s 1452 fc00:23::3 命令，这将触发 R1 发出载荷为 1452 字节的 ICMPv6 Echo Request 报文。由于 IPv6 标准头部的大小为 40 字节，而 ICMPv6Echo Request 报文头的大小为 8 字节，因此 R1 产生的该 IPv6 报文总长度为 1500 字节，等于 R1 的 GE0/0/0 接口的 IPv6 MTU 值，因此 R1 并不会对报文进行分片。

然而如果此时在 R1 上执行 ping ipv6 -s 1453 fc00:23::3，则会在 R1 的 GE0/0/0 接口上捕获到如下报文：

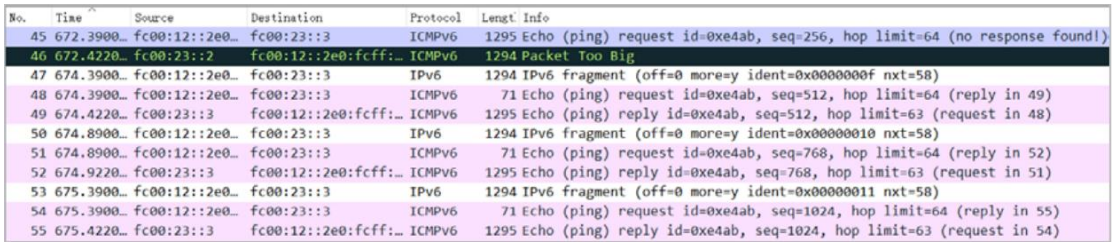
No.	Time	Source	Destination	Protocol	Length	Info
13	8.954000	fc00:12::2e0…	fc00:23::3	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000005 nxt=58)
14	8.954000	fc00:12::2e0…	fc00:23::3	ICMPv6	75	Echo (ping) request id=0xdcab, seq=256, hop limit=64 (reply in 16)
15	8.985000	fc00:23::3	fc00:12::2e0:fcff:…	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000005 nxt=58)
16	8.985000	fc00:23::3	fc00:12::2e0:fcff:…	ICMPv6	75	Echo (ping) reply id=0xdcab, seq=256, hop limit=63 (request in 14)
17	9.438000	fc00:12::2e0…	fc00:23::3	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000006 nxt=58)
18	9.438000	fc00:12::2e0…	fc00:23::3	ICMPv6	75	Echo (ping) request id=0xdcab, seq=512, hop limit=64 (reply in 20)
19	9.469000	fc00:23::3	fc00:12::2e0:fcff:…	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000006 nxt=58)
20	9.469000	fc00:23::3	fc00:12::2e0:fcff:…	ICMPv6	75	Echo (ping) reply id=0xdcab, seq=512, hop limit=63 (request in 18)
21	9.938000	fc00:12::2e0…	fc00:23::3	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000007 nxt=58)
22	9.938000	fc00:12::2e0…	fc00:23::3	ICMPv6	75	Echo (ping) request id=0xdcab, seq=768, hop limit=64 (reply in 24)
23	9.969000	fc00:23::3	fc00:12::2e0:fcff:…	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000007 nxt=58)
24	9.969000	fc00:23::3	fc00:12::2e0:fcff:…	ICMPv6	75	Echo (ping) reply id=0xdcab, seq=768, hop limit=63 (request in 22)
25	10.422000	fc00:12::2e0…	fc00:23::3	IPv6	1510	IPv6 fragment (off=0 more=y ident=0x00000008 nxt=58)

从上图可以看出，R1 将一个载荷长度为 1453 字节的 ICMPv6 报文进行了分片，每个单独的报文被分为 2 片发往目的地 FC00:23::3。由于 R1 是以上报文的始发节点，因此它可以对报名进行分片，报文分片到达 R3 后，R3 再将分片进行重新组装。值得注意的是，在 IPv6 中，中间转发设备不对 IPv6 报文进行分片，报文的分片将在源（始发）节点进行。因此，如果在本例中，若 R1 发出了长度超过 R2 的 GE0/0/1 接口 IPv6 MTU 的报文，则 R2 是无法对其进行分片处理的，也无法转发该报文。Path MTU 发现（PMTUD）机制用于解决该问题。PMTUD 的主要目的是发现路径上的 MTU，当数据包被从源转发到目的地的过程中便可避免分片。我们继续在 R1 的 GE0/0/0 接口上获取报文头，然后将 R2 的 GE0/0/1 接口的 IPv6 MTU 值修改为一个较小的值：1280 字节。

```
[R2] interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1] ipv6 mtu 1280
[R2-GigabitEthernet0/0/1] quit
```

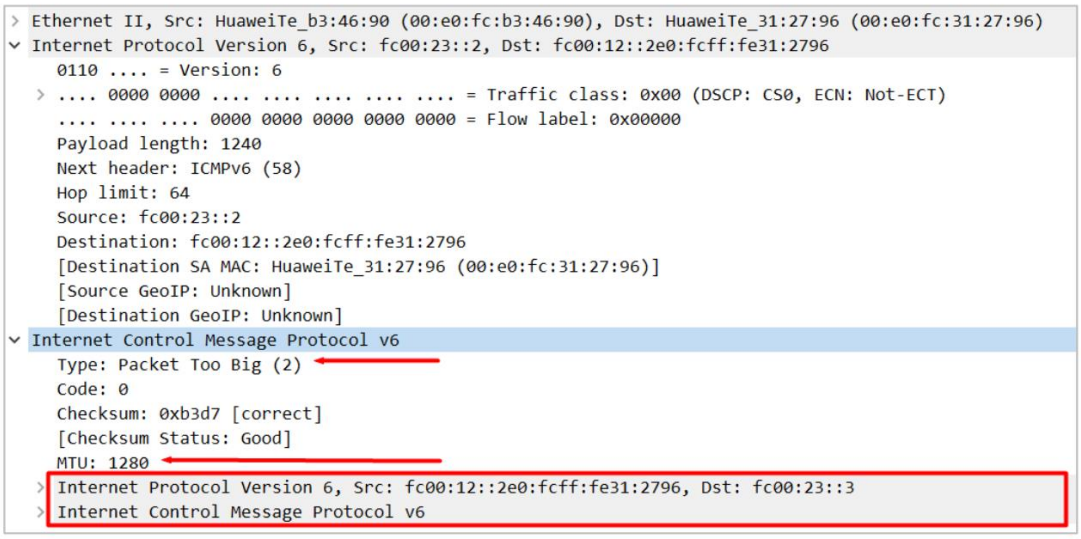
此时在 R1 上执行 ping ipv6 -s 1232 fc00:23::3 命令，可以触发 R1 产生一个人载荷长度 1232 字节的 ICMPv6 Echo Request 报文，这个长度加上 40 字节 IPv6 基本头部及 8 字节 ICMPv6 Echo Request 头部，总和是 1280 字节，正好等于报文到达 R3 的途中需经过的 R2 的 GE0/0/1 接口的 IPv6 MTU 值。

该命令执行后，从 R2 的 GE0/0/1 接口所捕获的报文中不会发现异常。接下来，将在 R1 上执行的命令变更为 ping ipv6 -s 1233 fc00:23::3，会发现能够 Ping 通 R3，但是报文交互有了变化：



No.	Time	Source	Destination	Protocol	Lengt	Info
45	672.3900…	fc00:12::2e0…	fc00:23::3	ICMPv6	1295	Echo (ping) request id=0xe4ab, seq=256, hop limit=64 (no response found!)
46	672.4220…	fc00:23::2	fc00:12::2e0:fcff:…	ICMPv6	1294	Packet Too Big
47	674.3900…	fc00:12::2e0…	fc00:23::3	IPv6	1294	IPv6 fragment (off=0 more=y ident=0x0000000f nxt=58)
48	674.3900…	fc00:12::2e0…	fc00:23::3	ICMPv6	71	Echo (ping) request id=0xe4ab, seq=512, hop limit=64 (reply in 49)
49	674.4220…	fc00:23::3	fc00:12::2e0:fcff:…	ICMPv6	1295	Echo (ping) reply id=0xe4ab, seq=512, hop limit=63 (request in 48)
50	674.8900…	fc00:12::2e0…	fc00:23::3	IPv6	1294	IPv6 fragment (off=0 more=y ident=0x00000010 nxt=58)
51	674.8900…	fc00:12::2e0…	fc00:23::3	ICMPv6	71	Echo (ping) request id=0xe4ab, seq=768, hop limit=64 (reply in 52)
52	674.9220…	fc00:23::3	fc00:12::2e0:fcff:…	ICMPv6	1295	Echo (ping) reply id=0xe4ab, seq=768, hop limit=63 (request in 51)
53	675.3900…	fc00:12::2e0…	fc00:23::3	IPv6	1294	IPv6 fragment (off=0 more=y ident=0x00000011 nxt=58)
54	675.3900…	fc00:12::2e0…	fc00:23::3	ICMPv6	71	Echo (ping) request id=0xe4ab, seq=1024, hop limit=64 (reply in 55)
55	675.4220…	fc00:23::3	fc00:12::2e0:fcff:…	ICMPv6	1295	Echo (ping) reply id=0xe4ab, seq=1024, hop limit=63 (request in 54)

在上图中，第 45 个报文为 R1 发出的首个 ICMPv6 Echo Request 报文，这个报文到达 R2 后，因为长度超出了其出站接口 GE0/0/1 的 IPv6 MTU，故被丢弃，R1 将无法收到对于这个 Echo Request 报文的应答。此时 R2 立即通过 ICMPv6 差错报文通知 R1，这个通知在第 46 个报文中体现，这个报文的详细内容如下：



```
> Ethernet II, Src: HuaweiTe_b3:46:90 (00:e0:fc:b3:46:90), Dst: HuaweiTe_31:27:96 (00:e0:fc:31:27:96)
v Internet Protocol Version 6, Src: fc00:23::2, Dst: fc00:12::2e0:fcff:fe31:2796
    0110 .... = Version: 6
  > .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
    Payload length: 1240
    Next header: ICMPv6 (58)
    Hop limit: 64
    Source: fc00:23::2
    Destination: fc00:12::2e0:fcff:fe31:2796
    [Destination SA MAC: HuaweiTe_31:27:96 (00:e0:fc:31:27:96)]
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
v Internet Control Message Protocol v6
    Type: Packet Too Big (2)
    Code: 0
    Checksum: 0xb3d7 [correct]
    [Checksum Status: Good]
    MTU: 1280
  > Internet Protocol Version 6, Src: fc00:12::2e0:fcff:fe31:2796, Dst: fc00:23::3
  > Internet Control Message Protocol v6
```

R2 在这个 ICMPv6 差错报文（报文类型为 Packet Too Big）中，将本地出站接口的 IPv6 MTU 值 1280 带给了 R1，同时也在该报文中将此前被其丢弃的、R1 所发出的 Echo Request 报文附上了。R1 收到上述报文后，得知自己发出的报文因为尺寸过大被丢弃，而且报文转发路径上目前探知的最小 MTU 为 1280，于是形成如下缓存表项：

```
<R1> display ipv6 pathmtu all
IPv6 Destination Address ZoneID PathMTU LifeTime(M) Type FF
FC00:23::3
0
1280 2 Dynamic No
--------------------------------------------------------------------------------
Total: 1 Dynamic: 1 Static: 0
```

如此一来，后续再发往 FC0023::3 的报文，将会以 1280 字节作为 MTU，如果报文的长度超出该值，则始发路由器 R1 将直接对齐进行分片，因此当 R1 Ping FC00:23::3 时，首个 ICMPv6 报文被丢弃，后续的报文则可以被顺利转发。

（5）思考题

- 当我们在路由器的 IPv6 接口上执行 undo ipv6 nd ra halt 命令后，该接口将周期性地发送 RA 报文，这些报文的目的 IPv6 地址是？该报文的载荷有什么内容？

回答：这些报文的目的 IPv6 地址是 IPv6 所有路由器组播地址（FF02::2）。

RA 报文的载荷包含以下内容：

1. 路由器 ID（Router ID）：一个 32 位的值，标识路由器的 IPv6 地址。
2. 当前时间：路由器宣告报文发送的时间。
3. 路由器优先级（Router Priority）：一个 8 位的值，用于在选举过程中确定路由器的优先级。

4. 保留：用于将来使用的保留字段。
5. 跳限制（Hop Limit）：一个 8 位的值，表示数据包在路由器之间传输的最大跳数。
6. 寿命（Lifetime）：一个 32 位的值，表示地址配置的持续时间。
7. 本地数据段前缀：一个 64 位的值，表示本地数据段的前缀和前缀长度。
8. 默认路由（Default Route）：一个 32 位的值，表示是否将数据包发送到默认路由。
9. 其他选项：其他可选字段，如 MTU、主机名字等。

- 当一台设备的接口获得 IPv6 地址后，设备立即启动 DAD 过程并在接口上发送一个 NS 报文用于检测该地址是否已被使用，这个 NS 报文的目的 IPv6 地址是什么？这个地址是如何形成的？

回答：NS 报文的目的 IPv6 地址是通过将设备接口的 MAC 地址和本地链路前缀组合生成的。具体来说，设备会使用 EUI-64 格式的 MAC 地址，并将其嵌入到 IPv6 地址中的 64 位中，然后添加本地链路前缀，从而形成一个完整的 IPv6 地址。

- IPv6 报文头部中的“Hop Limit”字段有什么用途？

回答：IPv6 报文头部中的“Hop Limit”字段是用来防止数据报在网络中无限期存在的。这个字段的长度为 8 位，类似于 IPv4 中的 Time to Live 字段。
源点在每个数据报发出时设定某个跳数限制（最大为 255）。每个路由器在转发数据报时，要先把跳数限制字段中的值减 1。当跳数限制的值为 0 时，则把这个数据报丢弃。这样可以防止数据报在网络中无限期存在，造成网络拥堵。