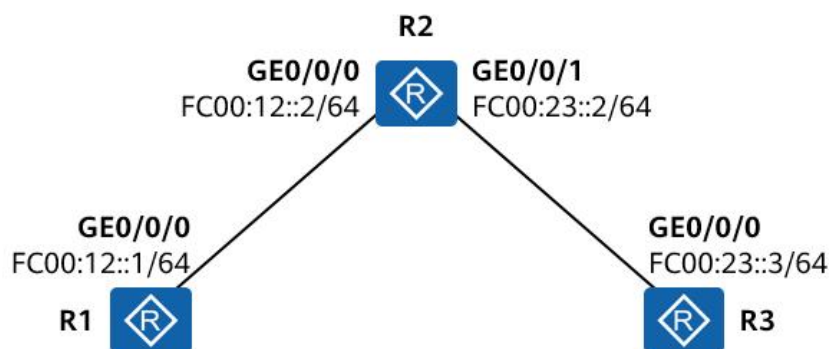


实验 2：ICMPv6 与 NDP 基础实验

2.1 实验说明

实验拓扑



实验目的

1. 掌握数据报文捕获及分析方法。
2. 理解 RA 报文及无状态地址自动配置过程。
3. 理解 DAD 地址冲突检测机制工作过程。
4. 理解 IPv6 网络中的地址解析过程。
5. 分析 Ping 与 Tracert 应用所使用的 ICMPv6 报文及工作原理。
6. 理解 IPv6 PMTUD 机制及其工作原理。

实验需求

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。

2.2 配置思路

1. 完成 R2 的基础配置。
2. 观察 RA 报文与无状态地址自动配置过程。
3. 观察 DAD 过程。
4. 观察地址解析过程。
5. 捕获 Ping 报文。
6. 捕获 Tracert 报文。
7. 观察 IPv6 PMTUD 机制。

2.3 操作步骤

实验注意事项：

- 1) 路由器最好选用 AR2220
- 2) 提醒：打开路由器终端默认进入的是用户视图，[R1]是系统视图，需要使用 system-view 命令进入。
- 3) 如果中途退出实验，需要关闭软件时，记得保存已经完成的路由器配置：在用户视图下使用 save 命令（例如：<R1> save）

1. 完成R2的基础配置

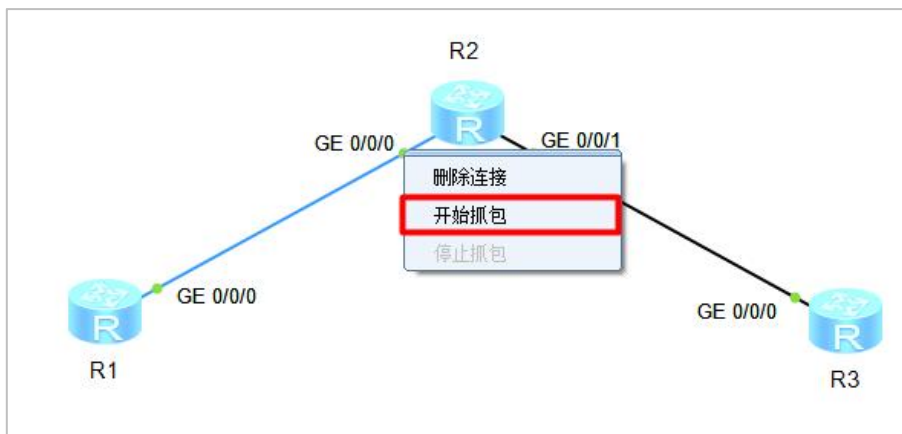
在 R2 上完成如下配置：

```
<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
```

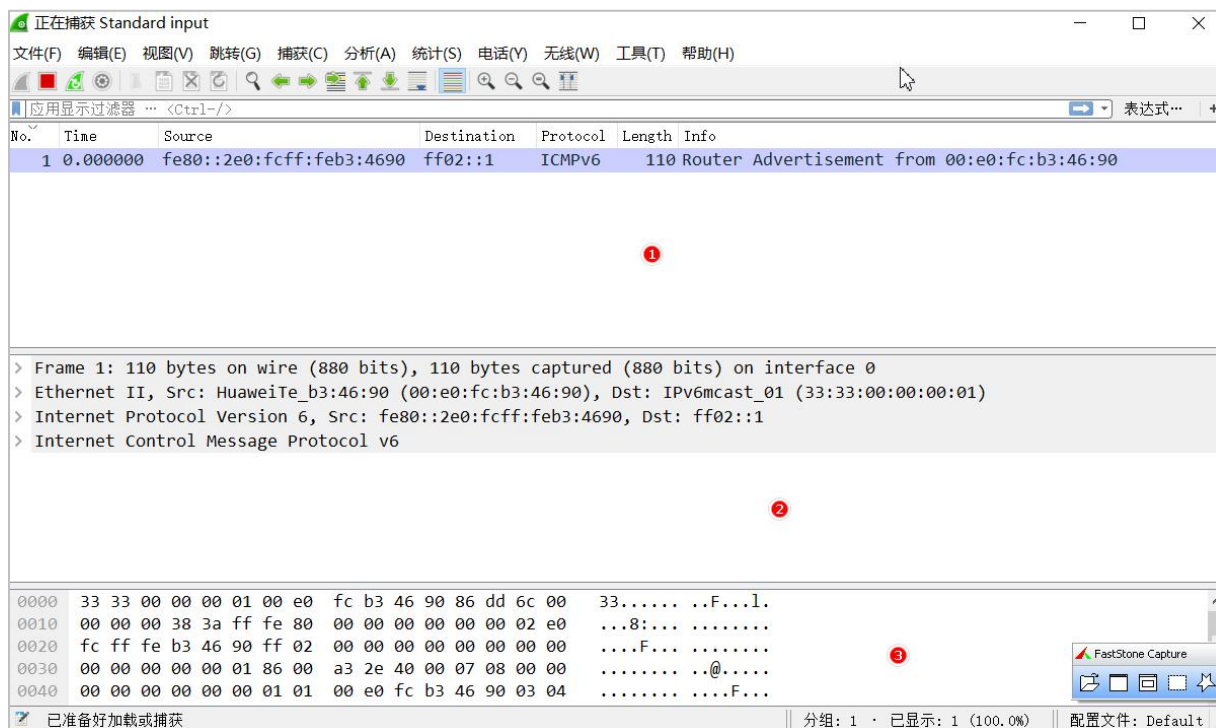
在上述配置中，我们为 R2 的 GE0/0/0 接口配置了静态 IPv6 地址 FC00:12::2/64，并使其能接口发布 RA 报文的功能，如此一来，该接口将周期性地向外发送 RA 报文。

2. 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 接口上单机鼠标右键，点击“**开始抓包**”启动抓包程序 Wireshark，该动作将捕获 R2 的 GE0/0/0 接口上的入向与出向报文，并通过 Wireshark 实现报文解析。（选择“数据抓包”-- GE0/0/0）

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 R1 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) ①
> Internet Protocol Version 6, Src: fe80::2e0:fcff:feb3:4690, Dst: ff02::1 ②
    0110 .... = Version: 6
    > .... 1100 0000 .... = Traffic class: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    .... 0000 0000 0000 0000 = Flow label: 0x000000
    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]
> Internet Control Message Protocol v6 ③
    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，这实际上是一个组播 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（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：

正在捕获 Standard input						
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)						
应用显示过滤器 ... <Ctrl-/> 表达式...						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::2e0:fcff:feb3:4690	ff02::1	ICMPv6	110	Router Advertisement from 00:e0:fc:b3:46:90
2	256.1880...	fe80::2e0:fcff:feb3:4690	ff02::1	ICMPv6	110	Router Advertisement from 00:e0:fc:b3:46:90
3	365.4070...	::	ff02::1:f...	ICMPv6	78	Neighbor Solicitation for fe80::2e0:fcff:fe31:2796
4	367.2660...	fe80::2e0:fcff:fe31:2796	ff02::1	ICMPv6	70	Router Solicitation from 00:e0:fc:31:27:96
5	368.2500...	fe80::2e0:fcff:feb3:4690	ff02::1	ICMPv6	110	Router Advertisement from 00:e0:fc:b3:46:90
6	369.2660...	::	ff02::1:f...	ICMPv6	78	Neighbor Solicitation for fc00:12::2e0:fcff:fe31:2...
7	372.1720...	fe80::2e0:fcff:fe31:2796	ff02::1	ICMPv6	70	Router Solicitation from 00:e0:fc:31:27:96
8	372.2500...	fe80::2e0:fcff:feb3:4690	ff02::1	ICMPv6	110	Router Advertisement from 00:e0:fc:b3:46:90
9	376.5000...	fe80::2e0:fcff:fe31:2796	ff02::1	ICMPv6	70	Router Solicitation from 00:e0:fc:31:27:96
> Frame 4: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0 > Ethernet II, Src: HuaweiTe_31:27:96 (00:e0:fc:31:27:96), Dst: IPv6mcast_01 (33:33:00:00:00:01) > Internet Protocol Version 6, Src: fe80::2e0:fcff:fe31:2796, Dst: ff02::1 > Internet Control Message Protocol v6 Type: Router Solicitation (133) Code: 0 Checksum: 0x31df [correct] [Checksum Status: Good] Reserved: 00000000 > ICMPv6 Option (Source link-layer address : 00:e0:fc:31:27:96)						

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
```

3. 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
```

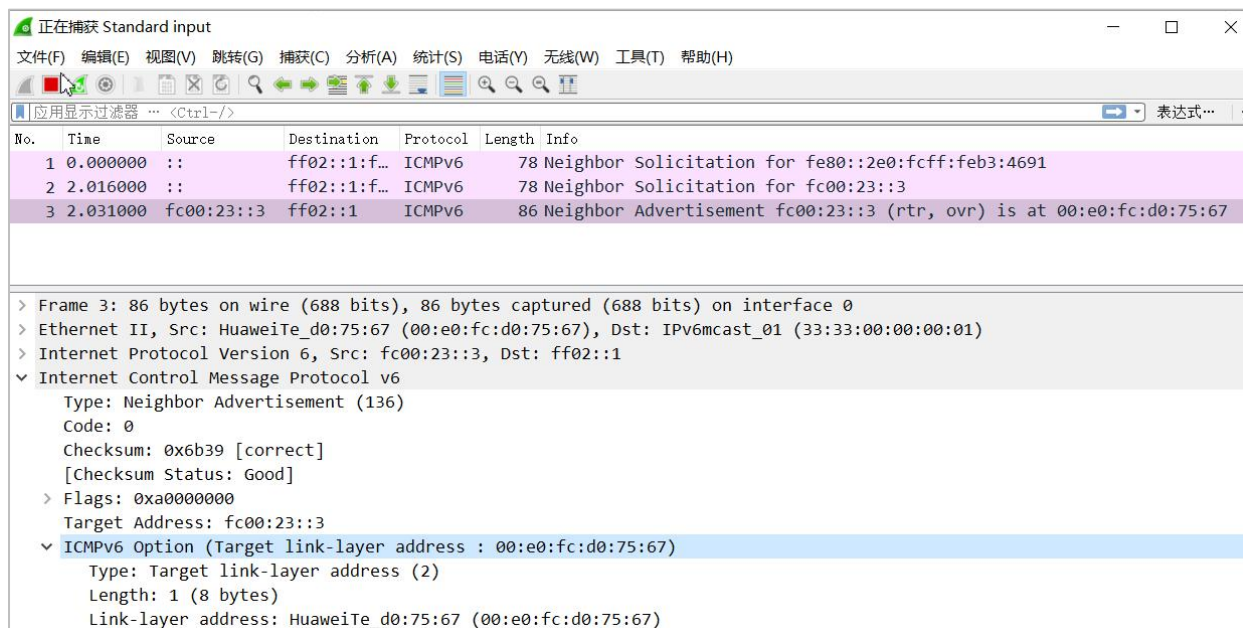
接下来在 R2 的 GE0/0/1 接口上开始抓包。

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
```


值得注意的是，在 R2 的上述配置中，我们故意将其 GE0/0/1 接口的 IPv6 地址设置为 FC00:23::3，与 R3 的 GE0/0/0 地址相同，从而制造 IPv6 地址冲突的现象。

此时可以捕获到如下报文：



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::	ff02::1:f...	ICMPv6	78	Neighbor Solicitation for fe80::2e0:fcff:feb3:4691
2	2.016000	::	ff02::1:f...	ICMPv6	78	Neighbor Solicitation for fc00:23::3
3	2.031000	fc00:23::3	ff02::1	ICMPv6	86	Neighbor Advertisement fc00:23::3 (rtr, ovr) is at 00:e0:fc:d0:75:67

> Frame 3: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: HuaweiTe_d0:75:67 (00:e0:fc:d0:75:67), Dst: IPv6mcast_01 (33:33:00:00:00:01)
> Internet Protocol Version 6, Src: fc00:23::3, Dst: ff02::1
> Internet Control Message Protocol v6
Type: Neighbor Advertisement (136)
Code: 0
Checksum: 0x6b39 [correct]
[Checksum Status: Good]
> Flags: 0xa0000000
Target Address: fc00:23::3
> ICMPv6 Option (Target link-layer address : 00:e0:fc:d0:75:67)
Type: Target link-layer address (2)
Length: 1 (8 bytes)
Link-layer address: HuaweiTe_d0:75:67 (00:e0:fc:d0:75:67)

从上图可以分析出，R2 获得该 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，不会使用这个地址进行业务通信。

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
```

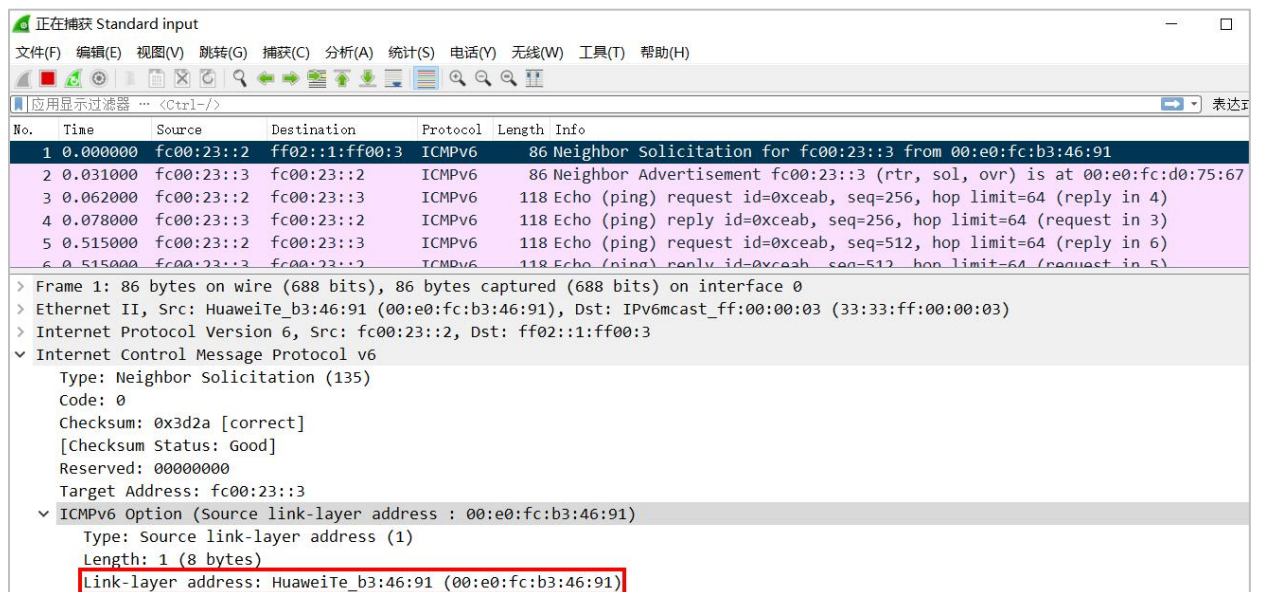
```
ND reachable time is 30000 milliseconds
ND retransmit interval is 1000 milliseconds
Hosts use stateless autoconfig for addresses
```

4. 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 接口上进行抓包，然后在 R2 上 ping FC00:23::3（命令：ping ipv6 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：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fc00:23::2	ff02::1:ff00:3	ICMPv6	86	Neighbor Solicitation for fc00:23::3 from 00:e0:fc:b3:46:91
2	0.031000	fc00:23::3	fc00:23::2	ICMPv6	86	Neighbor Advertisement fc00:23::3 (rtr, sol, ovr) is at 00:e0:fc:d0:75:67
3	0.062000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=256, hop limit=64 (reply in 4)
4	0.078000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=256, hop limit=64 (request in 3)
5	0.515000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=512, hop limit=64 (reply in 6)
6	0.515000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=512, hop limit=64 (request in 5)

> Frame 2: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0

> Ethernet II, Src: HuaweiTe_d0:75:67 (00:e0:fc:d0:75:67), Dst: HuaweiTe_b3:46:91 (00:e0:fc:b3:46:91)

> Internet Protocol Version 6, Src: fc00:23::3, Dst: fc00:23::2

> Internet Control Message Protocol v6

Type: Neighbor Advertisement (136)

Code: 0

Checksum: 0x2e17 [correct]

[Checksum Status: Good]

> Flags: 0xe0000000

Target Address: fc00:23::3

> ICMPv6 Option (Target link-layer address : 00:e0:fc:d0:75:67)

Type: Target link-layer address (2)

Length: 1 (8 bytes)

Link-layer address: HuaweiTe_d0:75:67 (00:e0:fc:d0:75:67)

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，可以正常交互 IPv6 报文。

5. 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：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fc00:23::2	ff02::1:ff00:3	ICMPv6	86	Neighbor Solicitation for fc00:23::3 from 00:e0:fc:b3:46:91
2	0.031000	fc00:23::3	fc00:23::2	ICMPv6	86	Neighbor Advertisement fc00:23::3 (rtr, sol, ovr) is at 00:e0:fc:d0:75:67
3	0.062000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=256, hop limit=64 (reply in 4)
4	0.078000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=256, hop limit=64 (request in 3)
5	0.515000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=512, hop limit=64 (reply in 6)
6	0.515000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=512, hop limit=64 (request in 5)
7	1.000000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=768, hop limit=64 (reply in 8)
8	1.015000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=768, hop limit=64 (request in 7)
9	1.515000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=1024, hop limit=64 (reply in 10)
...	1.515000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=1024, hop limit=64 (request in 9)
...	2.000000	fc00:23::2	fc00:23::3	ICMPv6	118	Echo (ping) request id=0xceab, seq=1280, hop limit=64 (reply in 12)
...	2.015000	fc00:23::3	fc00:23::2	ICMPv6	118	Echo (ping) reply id=0xceab, seq=1280, hop limit=64 (request in 11)

> Frame 3: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0

> Ethernet II, Src: HuaweiTe_b3:46:91 (00:e0:fc:b3:46:91), Dst: HuaweiTe_d0:75:67 (00:e0:fc:d0:75:67)

> Internet Protocol Version 6, Src: fc00:23::2, Dst: fc00:23::3

> Internet Control Message Protocol v6

Type: Echo (ping) request (128)

Code: 0

Checksum: 0x42d8 [correct]

[Checksum Status: Good]

Identifier: 0xceab

Sequence: 256

[\[Response In: 4\]](#)

> Data (56 bytes)

6. 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 接口上开始抓包。

此时我们在 R1 上执行如下命令：

```
<R1> tracert ipv6 fc00:23::3

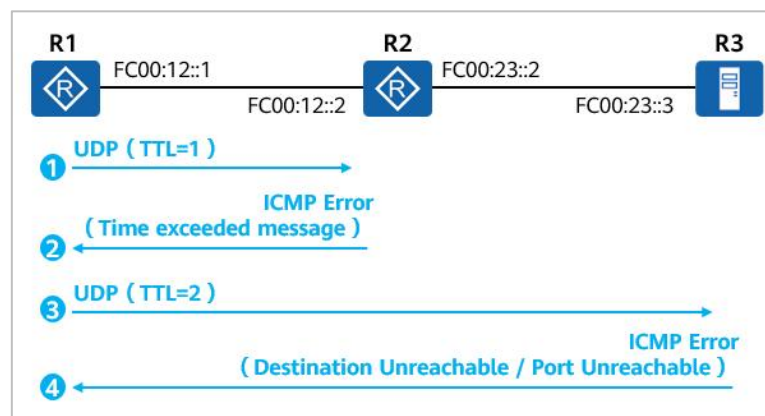
traceroute to fc00:23::3 30 hops max, 60 bytes packet

 1 FC00:12::2 20 ms 20 ms 30 ms

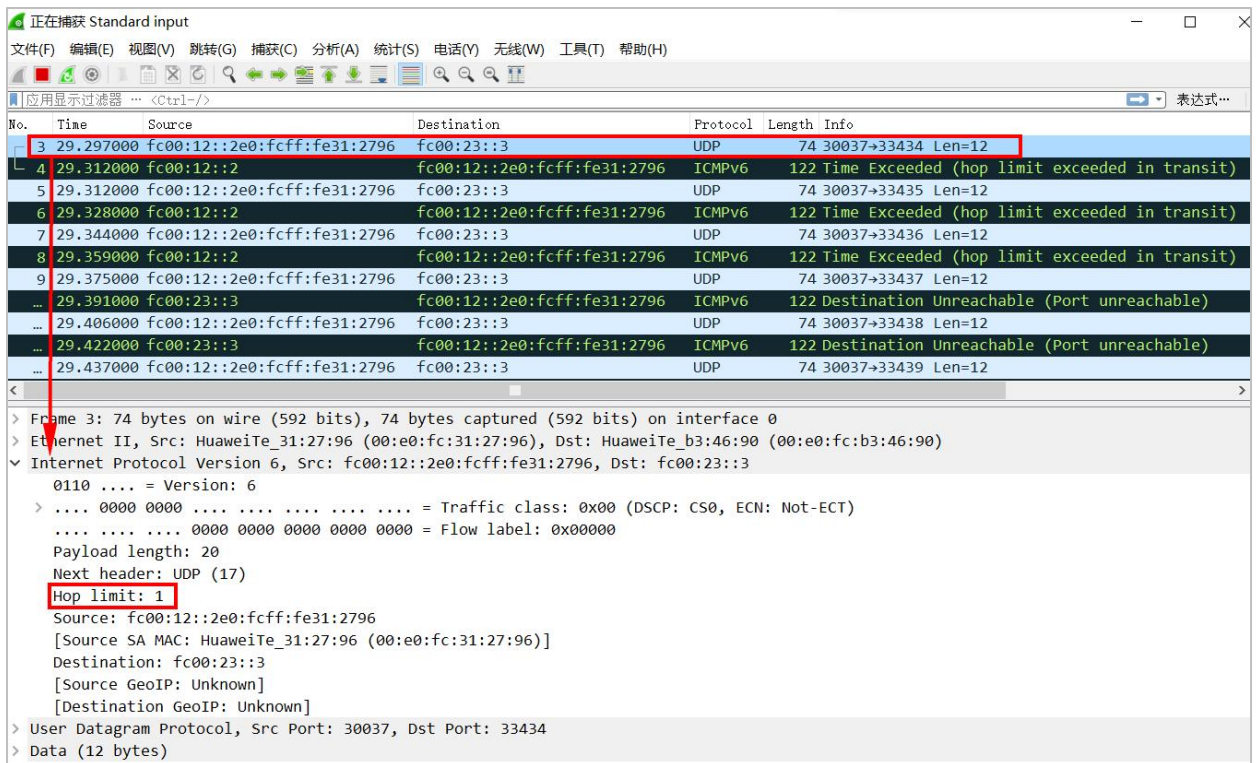
 2 FC00:23::3 30 ms 40 ms 20 ms
```

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。

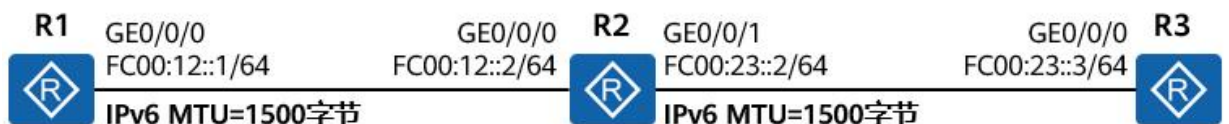
接着 R2 以 Hop Limit=2 继续发送 UDP 报文, 如此反复。

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。

7. 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, 缺省时, 路由器接口的 IPv6 MTU 值为 1500 字节。如果此时在 R1 的 GE0/0/0 接口上抓包, 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 ICMPv6 Echo 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, 因此它可以对报文进行分片, 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 接口上抓包, 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, 正好是 1280 字节——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, 但是抓包的结果有了变化:

No.	Time	Source	Destination	Protocol	Length	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 = 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。

2.4 思考题

1. 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？
2. 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？
3. IPv6 报文头部中的 “Hop Limit” 字段有什么用途？