

第五讲



登录路由器与路由协议配置

- 路由器介绍
- 登录路由器
- 路由协议回顾
- 路由协议配置





路由器介绍

- 路由器硬件结构
- 路由器软件结构
- 路由器上的各种接口
- 华3全系列路由器



路由器硬件结构

路由器就是一种具有专门用途的计算机

- CPU (Central Processing Unit)
- RAM (Random Access Memory)
 - 存储正在运行的操作系统和配置文件
- FLASH
 - 相当于硬盘，存储操作系统和配置文件等
- NVRAM (Non-Volatile RAM)
 - 备份当前RAM中的配置文件，以免断电丢失
- ROM (Read-Only Memory)
 - 存储路由器初始化和操作系统引导程序
- 各种接口
 - 将路由器与其它网络设备或PC机连接起来



路由器软件结构

- 两部分组成：

路由器初始化、操作系统引导
程序 BootRom (BootWare)

Comware网络操作系统平台



路由器上的各种接口

指示灯



电源开关

电源插座

配置口

备份口

以太网口

同异步串口

同异步串口

接V.35
cable

配置口: CONSOLE, 备份口: AUX, 以太网口: 100M ETH或LAN 口, 同异步串口: SERIAL或 WAN口



路由器上的各种接口

CONSOLE和AUX接口

MSR30-20

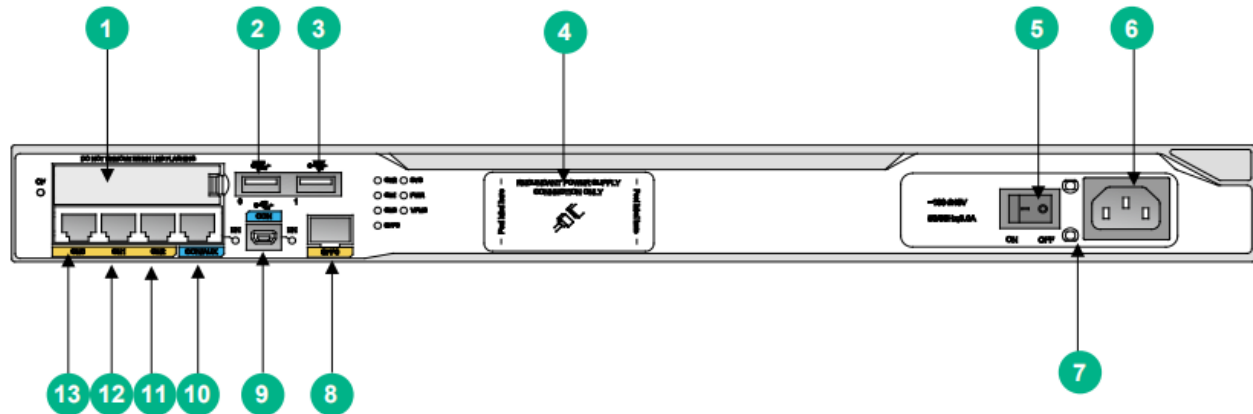


配置口: CONSOLE, 备份口: AUX, 以太网口: GE口或 LAN 口, 同异步串口: SERIAL或 WAN口



路由器上的各种接口（续一）

图A-5 MSR 36-20 AC 前视图

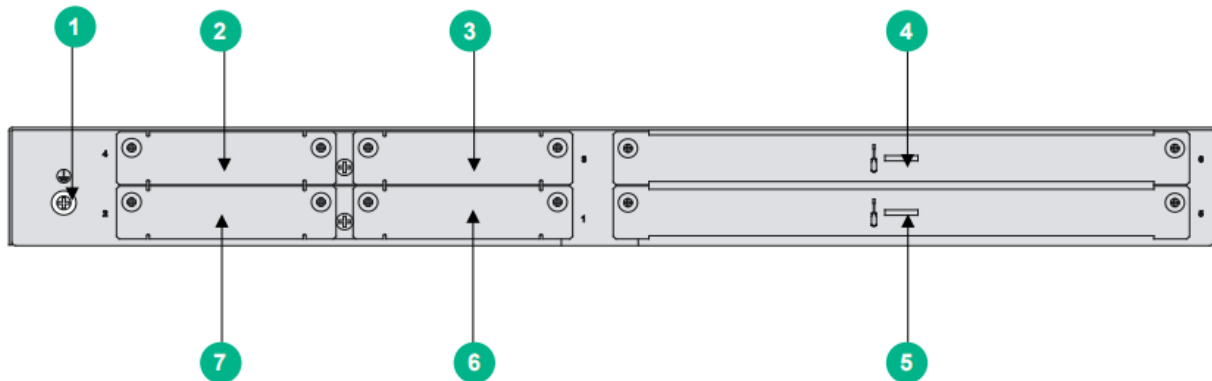


1: CF卡保护盖	2: USB接口0	3: USB接口1
4: RPS电源挡板	5: 电源开关	6: 交流电源插座
7: 电源线卡钩槽	8: 光模块接口 SFP0	9: USB配置口 CON
10: 串行配置口 CON/AUX	11: 千兆以太网接口 GE2	12: 千兆以太网接口 GE1
13: 千兆以太网接口 GE0		



路由器上的各种接口（续二）

图A-6 MSR 36-20 AC 后视图



1: 接地端子

2: SIC接口模块插槽4

3: SIC接口模块插槽3

4: HMIM接口模块插槽6

5: HMIM接口模块插槽5

6: SIC接口模块插槽1

7: SIC接口模块插槽2



H3C系列路由器（部分）

高端核心路由器



SR8802



SR8805



SR8808



SR8812

开放多核路由器



SR6602



SR6604

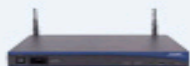


SR6608



SR6616

多业务开放路由器



H3C MSR20-1X



H3C MSR20



H3C MSR30



H3C MSR50

ER系列路由器



H3C ER3100



H3C ER3260



H3C ER5100



H3C ER5200



目前H3C全系列路由器



新华三集团

快速导航

产品技术

解决方案

技术咨询与服务

产品支持与服务

如何购买

新华三大学

合作伙伴

关于我们

H3C ICT融合网关

- > H3C MSR3600 系列ICT融合网关

5G IP-RAN 系列路由器

- > 5G IP-RAN接入路由器
- > 5G IP-RAN汇聚路由器
- > 5G IP-RAN核心路由器

H3C CR系列核心路由器

- > H3C CR19000 T级集群路由器
- > H3C CR16000-X核心路由器
- > H3C CR16000-F 核心路由器
- > H3C CR16000核心路由器

H3C SR系列高端路由器

- > H3C SR8800-F 核心路由器
- > H3C SR8800 系列路由器
- > H3C SR6600-F系列业务承载路由器
- > H3C SR6600-X云业务汇聚路由器
- > H3C SR6602-X双万光业务路由器
- > H3C SR6600开放多核路由器

H3C MSR系列开放多业务路由器

- > H3C MSR2600-XS 系列路由器
- > H3C MSR3600-XS 路由器
- > H3C MSR3600-XS 系列ICT 融合业务网关
- > H3C MSR 5600路由器

H3C LA系列融合网关

- > H3C LA46 系列ICT融合网关
- > H3C LA66 系列车载融合网关
- > H3C LA3608E系列无线网关产品
- > H3C LA3616系列无线网关产品

H3C MSR Winet智慧路由器网关

- > H3C MSR830-WiNet系列多业务网关
- > H3C MSR5600-WiNet路由器
- > H3C MSR3600-WiNet系列路由器产品
- > H3C MSR2600-WiNet系列路由器产品
- > H3C MSR810-WiNet无线部署路由器
- > H3C MSR830-WiNet系列路由器产品

H3C ICG 系列路由器

- > H3C ICG3000F信息通信网关
- > H3C ICG 5000系列信息通信网关
- > H3C ICG2000D信息通信网关
- > H3C ICG 6000 信息通信网关
- > H3C ICG 3000D&3000E 信息通信网关

H3C Aolynki以太网接入宽带路由器

- > Aolynki BR104H 家庭宽带路由器

H3C MER系列路由器

- > H3C MER3220路由器
- > H3C MER5200路由器

H3C ER G3系列路由器

- > H3C ER3200G3全新一代高性能企业级路由器

H3C ER G2系列路由器

- > H3C ERG2-450W第二代企业级商用无线网关
- > H3C ERG2-1350W第二代企业级无线网关
- > H3C ER2200G2 新一代企业级路由器
- > H3C ERG2-1200W第二代企业级无线网关
- > H3C ER8300G2-X 新一代企业级千兆路由器
- > H3C ER8300G2 新一代企业级千兆路由器
- > H3C ER6300G2 新一代企业级千兆路由器
- > H3C ER5200G2 新一代企业级千兆路由器
- > H3C ER5100G2 新一代企业级千兆路由器
- > H3C ER3260G2 新一代企业级千兆路由器
- > H3C ER3200G2 新一代企业级千兆路由器
- > H3C ER3100G2 新一代企业级千兆路由器

H3C GR系列路由器

- > H3C GR1100-P企业级千兆路由器
- > H3C GR1108-P企业级千兆路由器
- > H3C GR-1200W企业级千兆无线网关
- > H3C GR2200 企业级千兆路由器
- > H3C GR3200 企业级千兆路由器
- > H3C GR5200 企业级千兆路由器
- > H3C GR8300 高性能企业级千兆路由器

H3C ER 系列路由器



登录路由器

- **通过Console 口登录**
- 通过AUX口登录
- **通过Telnet方式登录**
- 通过SSH方式登录(部分设备支持)
- 通过哑终端方式登录
- 通过FTP方式传送配置文件



通过Console口登录

- 将RJ45的一端插入到路由器的Console口中，另外一端为9针的串口接口或一个25针的串口接口，接在计算机合适的串口上



通过Console口登录(续)

- 在PC机上运行终端仿真程序（如Windows XP/Windows 2000的超级终端等，以下配置以Windows XP为例。PC上选择“开始>程序>附件>通讯>超级终端”），选择与设备相连的串口（PC上选择“我的电脑>管理>设备管理器>端口”来查看当前使用的串口），设置终端通信参数。这些参数的值必须和设备上的值一致，缺省情况下：传输速率为9600bit/s、8位数据位、1位停止位、无校验和无流控。
- 如果PC使用的是Windows Server 2008、Windows Vista、Windows 7或其它操作系统，请准备第三方的终端控制软件，或者另行安装超级终端。



通过Telnet方式登录

- 将PC机的以太网接口与路由器的以太网接口相连
- 配置上述两接口的IP地址，使它们在同一网段上（不在同一网段亦可，但必须保障路由可达）
- 在路由器配置相关接口的IP地址命令
[RT-Ethernet *x/x*]ip address ip-address {mask|mask-length}
某些路由器上没有Ethernet接口，是GE接口



通过Telnet方式登录配置举例 (Comware V5)

在路由器上配置Telnet用户和密码举例 (Comware V5)

//打开Telnet服务器，缺省关闭，必须打开

```
[h3c] telnet server enable
```

//设置scheme认证

```
[H3C] user-interface vty 0 4
```

```
[H3C-ui-vty0-4]authentication-mode scheme
```

//创建本地帐号与密码(进入本地用户视图)

```
[H3C]local-user h3c
```

```
[H3C-luser-manage-h3c]password simple 123456
```

//设置服务类型为telnet(本地用户视图)

```
[H3C-luser-manage-h3c]service-type telnet
```



通过Telnet方式登录配置举例 (Comware V5) 续

//设置用户优先级为15 (本地用户视图)

```
[H3C-luser-manage-h3c] level 15 ( 如不支持level命令，输入  
authorization-attribute level 15 )
```

注意：某些路由器上用户优先级最高为3

//连接到telnet主机客户端 (如果接口已配IP地址，此步省略)

```
[H3C]interface G0/0
```

```
[H3C-GigabitEthernet0/0]ip address 192.168.0.1 24| 255.255.255.0  
(24或255.255.255.0 都可以)
```



通过Telnet方式登录配置举例 (Comware V7)

在路由器上配置Telnet用户和密码举例 (Comware V7)

//打开Telnet服务器，缺省关闭，必须打开

```
[h3c] telnet server enable
```

//进入一个或多个VTY用户线视图

```
[h3c]line vty 0 4
```

//设置登录用户的认证方式为通过scheme认证

```
[H3C-line-vty0-4]authentication-mode scheme
```

//创建本地帐号与密码(进入本地用户视图)

```
[H3C]local-user h3c
```

```
[H3C-luser-manage-h3c]password simple h3c
```



通过Telnet方式登录配置举例 (Comware V7) 续

//设置服务类型为telnet(本地用户视图)

```
[H3C-luser-manage-h3c]service-type telnet
```

//设置用户角色 (即优先级) (本地用户视图)

```
[H3C-luser-manage-h3c] authorization-attribute user-role {  
network-admin| level-15 }
```

//连接到telnet主机客户端 (如果接口已配IP地址 , 此步省略)

```
[H3C]interface G0/0
```

```
[H3C-GigabitEthernet0/0]ip address 192.168.0.1 24| 255.255.255.0  
(24或255.255.255.0 都可以)
```



路由器视图一览

系统视图

[H3C]

Quit

其它配置视图

视图

接口视图

路由协议视图

安全策略视图

提示

[H3C-Serial0]
[H3C-Ethernet0]

[H3C-rip]
[H3C-ospf]

[H3C-acl-1]



清空路由器配置信息

MSR路由器

<H3C> reset saved-configuration

<H3C> The saved configuration file will be erased. Are you sure? [Y/N]: 键入Y

<H3C> reboot

<H3C> Start to check configuration with next startup configuration file, please wait.....DONE!

Current configuration may be lost after the reboot, save current configuration? [Y/N]: 键入N

This command will reboot the device. Continue? [Y/N]: 键入Y



路由协议介绍

- 距离向量 (Distance Vector)路由算法
- 链路状态 (Link State)路由算法
- RIP (Routing Information Protocol)
- OSPF (Open Shortest Path First)
 - 推迟到综合实验时介绍



距离向量路由算法 — 概述

- 网络上的每个节点都周期性的与相邻节点交换信息。
- 信息的内容主要包括由本节点到网络上所有其它节点的最短距离。这些距离构成一个距离数组，被称为距离向量 (Distance Vector)。



距离向量路由算法 — 数据结构

每个节点 X 都维护如下数据结构：

- Cost List: 由节点 X 到每个邻居节点的距离。
 - 其中由节点 X 到节点 Y 的距离记为： $c(X, Y)$
- Distance Table：由节点 X 经由每个邻居节点到网络上其它节点的距离。
 - 其中节点 X 经由节点 Z 到节点 Y 的距离记为：
 $D^X(Y, Z)$, $D^X(Y, Z) = c(X, Z) + Z \text{到} Y \text{的最短距离}$

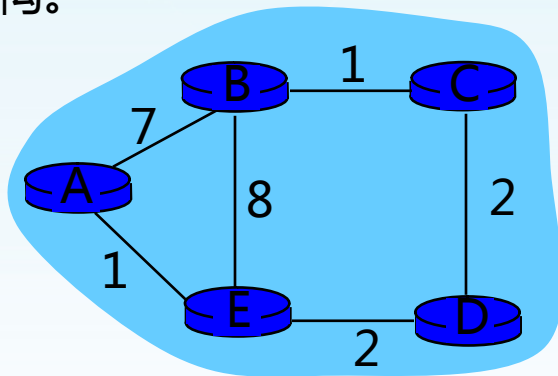


距离向量路由算法 — 数据结构（续）

举例：如右图，考虑节点E的数据结构。

Distance Table :

destination	邻居节点			
	$D^E()$	A	B	D
	A	1	14	5
	B	7	8	5
	C	6	9	4
	D	4	11	2



Cost List :

$$c(E, A) = 1$$

$$c(E, B) = 8$$

$$c(E, D) = 2$$



距离向量路由算法 — 数据结构（续）

Distance Table实际上给出了Routing Table:

		邻居节点					下一步节点, 距离	
destination	$D^E()$	A	B	D	destination			
	A	1	14	5		A	A,	1
	B	7	8	5		B	D,	5
	C	6	9	4		C	D,	4
	D	4	11	2		D	D,	2

Distance table



Routing table



距离向量路由算法 — 算法描述

考虑任意节点X, 初始化:

对节点X的每一个邻居节点 V:

$c(X, V) = X \text{ 到 } V \text{ 的距离} ;$

对网络上每一个非 X 的节点 *: $D^X(*, V) = \infty ;$

$D^X(V, V) = c(X, V) ;$



距离向量路由算法 — 算法描述（续）

任意节点X执行如下循环:

向每个邻居节点发送自己到所有其它节点的最短距离所构成的距离向量；

接收每个邻居节点所发送的距离向量；

对来自每个邻居节点 V 的距离向量中的每个元素

$\min \{D^V(Y, -)\}$ ，更新到Y的距离：

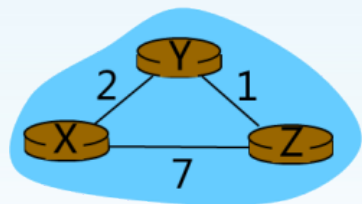
$D^X(Y, V) = c(X, V) + \min \{D^V(Y, -)\}$ ；

如果 $c(X, V)$ 发生变化，则对网络上每一个目的节点 Y ，更新经由V的距离：

$D^X(Y, V) = \text{新的 } c(X, V) + \min \{D^V(Y, -)\}$;



距离向量路由算法 — 举例



		cost via	
		Y	Z
destination	X		
	D ^X		
Y		2	∞
Z		∞	7

		cost via	
		X	Z
destination	Y		
	D ^Y		
X		2	∞
Z		∞	1

		cost via	
		X	Y
destination	Z		
	D ^Z		
X		7	∞
Y		∞	1

		cost via	
		Y	Z
destination	X		
	D ^X		
Y		2	8
Z		3	7

$$D^X(Z, Y) = c(X, Y) + \min \{D^Y(Z, -)\}$$

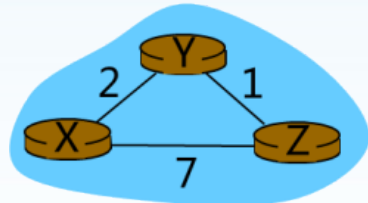
$$= 2 + 1 = 3$$

$$D^X(Y, Z) = c(X, Z) + \min \{D^Z(Y, -)\}$$

$$= 7 + 1 = 8$$



距离向量路由算法 — 举例（续）



		cost via	
		Y	Z
destination	X		
	Y	2	∞
	Z	∞	7

		cost via	
		X	Z
destination	Y		
	X	2	∞
	Z	∞	1

		cost via	
		X	Y
destination	Z		
	X	7	∞
	Y	∞	1

		cost via	
		Y	Z
destination	X		
	Y	2	8
	Z	3	7

		cost via	
		X	Z
destination	Y		
	X	2	8
	Z	9	1

		cost via	
		X	Y
destination	Z		
	X	7	3
	Y	9	1

		cost via	
		Y	Z
destination	X		
	Y		
	Z		

		cost via	
		X	Z
destination	Y		
	X		
	Z		

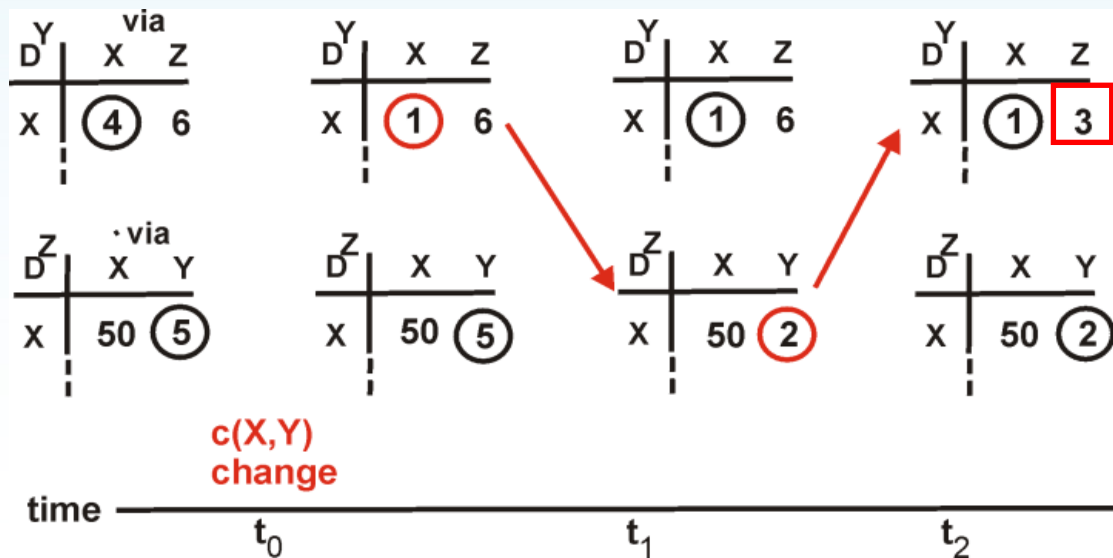
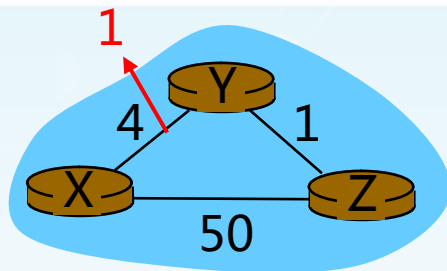
		cost via	
		X	Y
destination	Z		
	X		
	Y		



距离向量路由算法 — 距离改变

对于好消息，算法收敛迅速。例1

:



算法收敛



距离向量路由算法 — 距离改变（续）

对于好消息，算法收敛迅速。

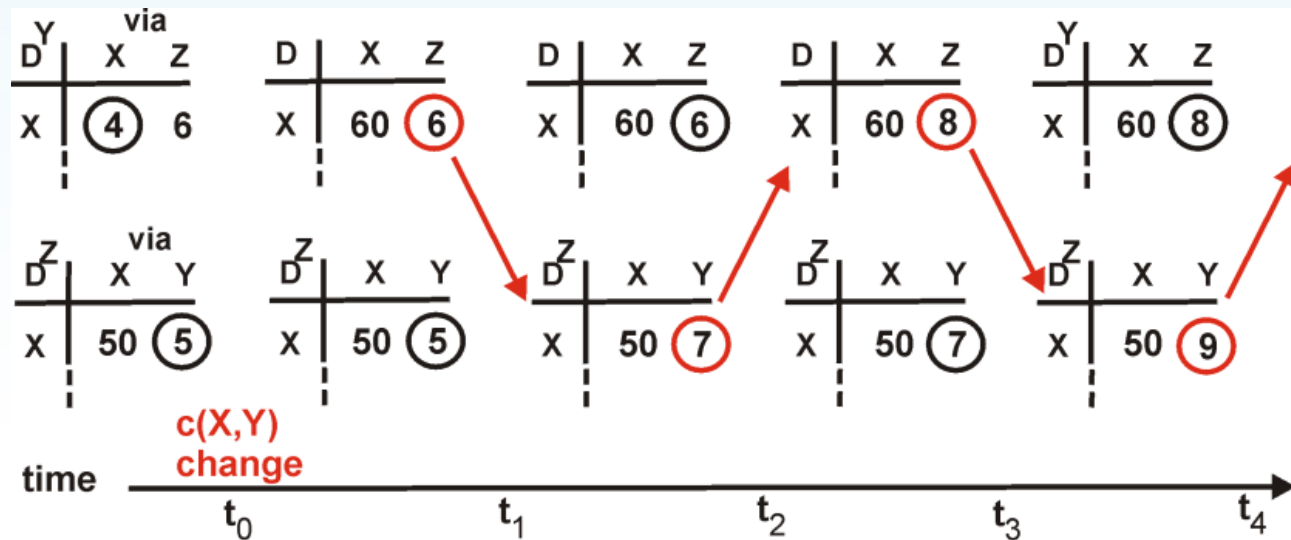
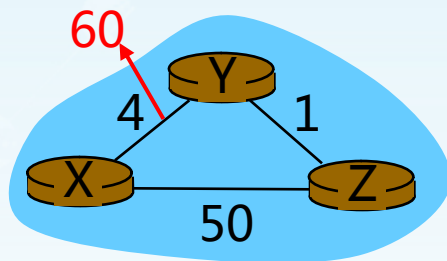
例2，下图中AB 链路由断开到重新连通：

A	B	C	D	E	
●	●	●	●	●	Initially
	●	●	●	●	After 1 exchange
	1	●	●	●	After 2 exchanges
	1	2	●	●	After 3 exchanges
	1	2	3	●	After 4 exchanges
	1	2	3	4	After 4 exchanges



距离向量路由算法 — 距离改变（续）

对于坏消息，算法收敛缓慢，即存在 “Count to Infinity Problem”。例1：



算法仍不收敛



距离向量路由算法 — 距离改变（续）

对于坏消息，算法收敛缓慢，即存在 “Count to Infinity Problem”。例2：下图中AB 间链路突然断开。

A	B	C	D	E	
●	●	●	●	●	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
	⋮				
●	●	●	●		



距离向量路由算法 — Split Horizon

Split Horizon: 如果 Z 到 X 的最短路径经过 Y，那么 Z 不告诉 Y “Z 到 X 的路径信息”，这样 Y 就不可能选择经由 Z 到达 X 的路线。



Tells Z that I can reach X

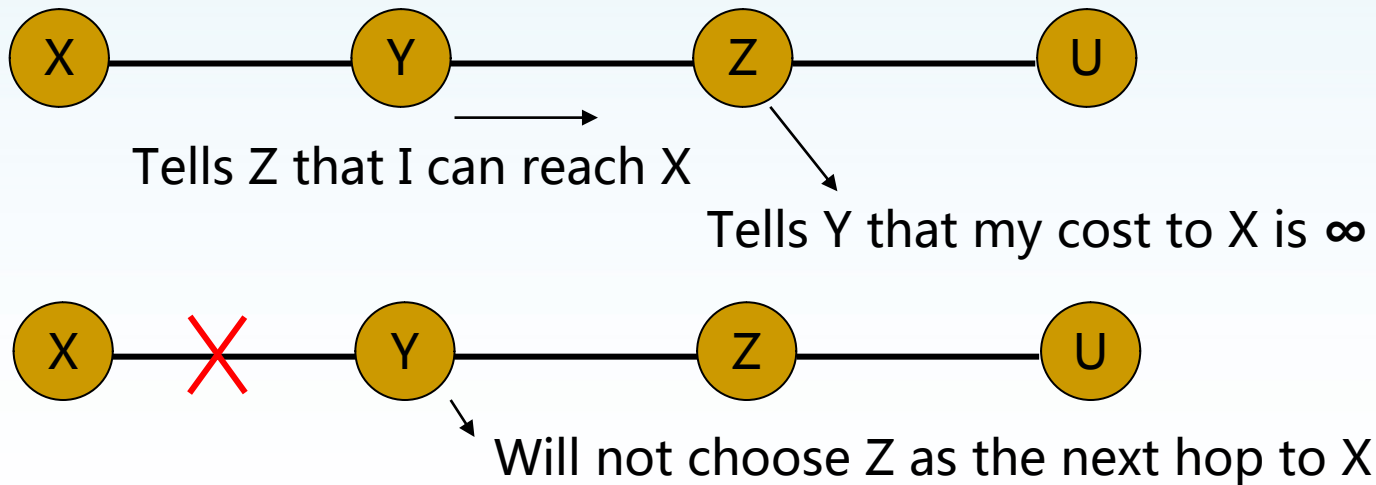


Will not choose Z as the next hop to X,
i.e., decides that it cannot reach X



距离向量路由算法 — Poisoned Reverse

Poisoned Reverse: 如果 Z 到 X 的最短路径经过 Y，那么 Z 告诉 Y “Z 到 X 的最短距离是 ∞ ”，这样 Y 就不会选择经由 Z 到达 X 的路线。

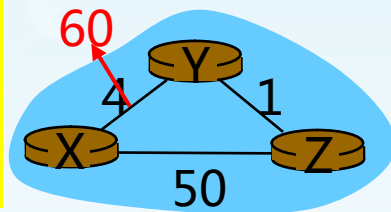


下一页图可以显示出Poisoned Reverse的优点

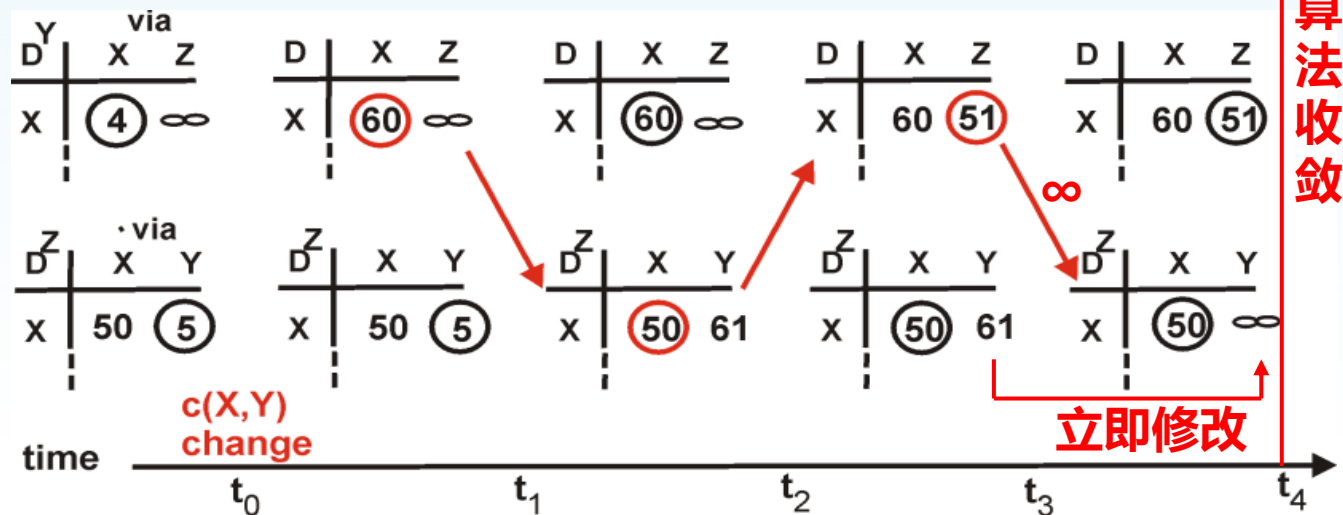


距离向量路由算法 — Poisoned Reverse

Poisoned Reverse: 如果 Z 到 X 的最短路径经过 Y，那么 Z 告诉 Y “Z 到 X 的最短距离是 ∞ ”

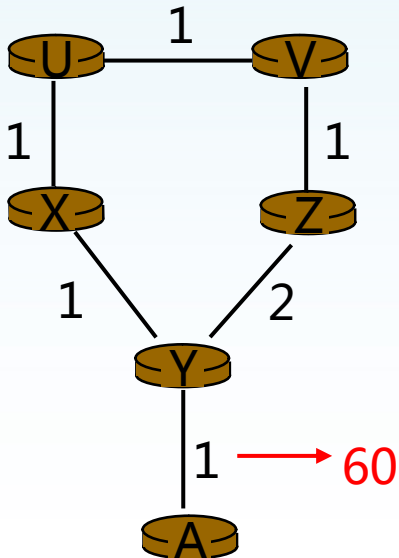
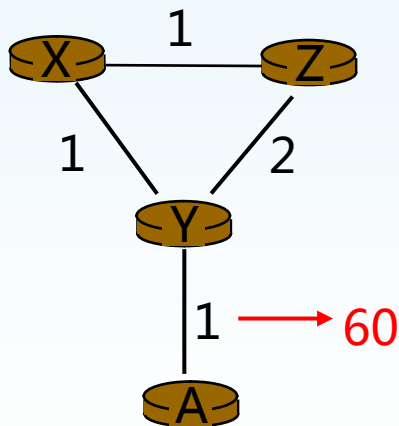


这样 Y 就不会选择经由 Z 到达 X 的路线。



距离向量路由算法 — 注意

Split Horizon 或 Poisoned Reverse 并没有完全解决 Count to Infinity Problem。考虑下图：



链路状态路由算法

- 每个节点都周期性的向网络上所有其它节点广播自己到邻居节点的距离。
- 结果：每个节点都具有完整的网络信息，即知道网络上每条边的距离。
- 于是，每个节点都可以彼此独立的使用Dijkstra最短路径算法来计算网络上其它节点的最短路径。



比较：DV versus LS

Distance Vector

- 仅与邻居节点交换消息
- 消息包括到所有节点的最短距离
- 收敛速度比较慢
- 有Count to Infinity Problem

Link State

- 向网络上所有其它节点广播消息
- 消息仅包括到邻居节点的距离
- 收敛速度比较快
- 没有Count to Infinity Problem



RIP — 历史

- Late 1960s : ARPANET 最初采用距离向量路由算法
- 1982: RIP首次在BSD Unix中被实现
- 1988: RIP-1 (RFC 1058)
 - Classful Routing
- 1993: RIP-2 (RFC 1388) , 支持 CIDR
 - 每个路由表项都包含子网掩码
 - 合并路由 (Route Summary/Aggregation)
- 1998: 最新版 -- RIP-2 (RFC 2453)



RIP — 概览

- 采用距离向量路由算法
- 对距离的度量是跳数 (number of hops)
- 允许的最大跳数为15，认为16代表 ∞ 。
- 一般的配置是每30秒钟交换一次距离向量
- 如果持续3分钟没有收到某个路由线路的更新，则认为该路由线路失效。
- 使用Split Horizon或Poisoned Reverse技术来解决Count-to-Infinity Problem.

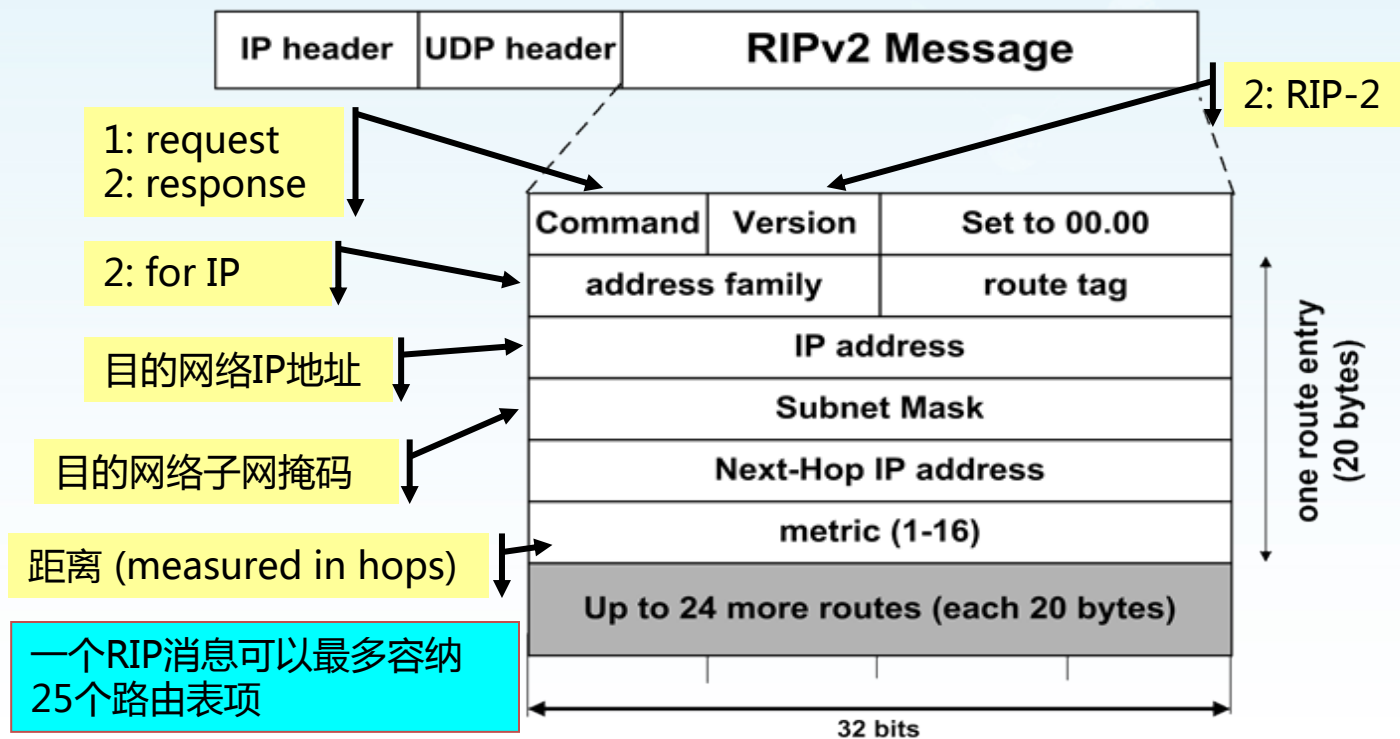


RIP — 消息 (Messages)

- RIP-1和RIP-2消息都使用UDP协议来传送，UDP端口号为520。
- RIP-1在IP层使用广播来发布距离向量
- RIP-2在IP层使用广播或组播来发布距离向量
 - 组播IP地址为224.0.0.9
- RIP共使用两种消息类型
 - RIP Request: 用于向邻居节点请求距离向量更新。
 - RIP Response: 用于向邻居节点发送距离向量更新。

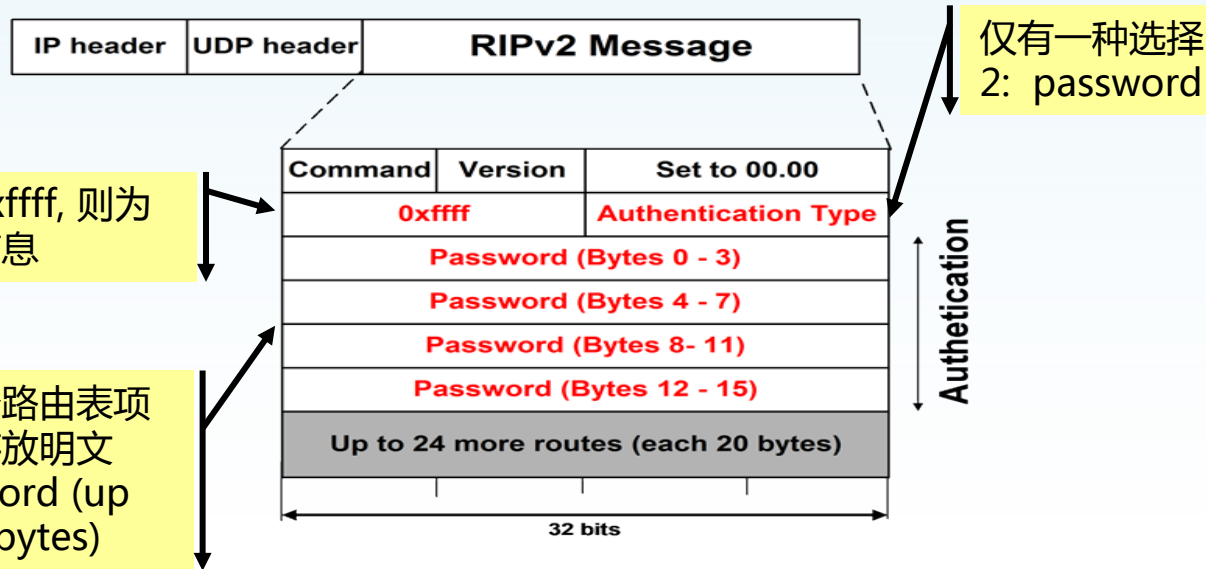


RIP — 消息格式



RIP — 安全 (Security)

- 问题: 发送假距离向量
- RIP-1: 没有安全措施
- RIP-2: 提供简单的安全措施





路由协议配置

- 静态路由配置
- RIP配置



静态路由配置 — 命令

静态路由的配置命令:

```
[H3C] [undo] ip route-static dest-address { mask-length |  
  mask } { interface-type interface-number [ next-hop-  
  address ] [ preference value ]
```

例如：

```
[H3C] ip route-static 129.1.0.0 16 10.0.0.2
```

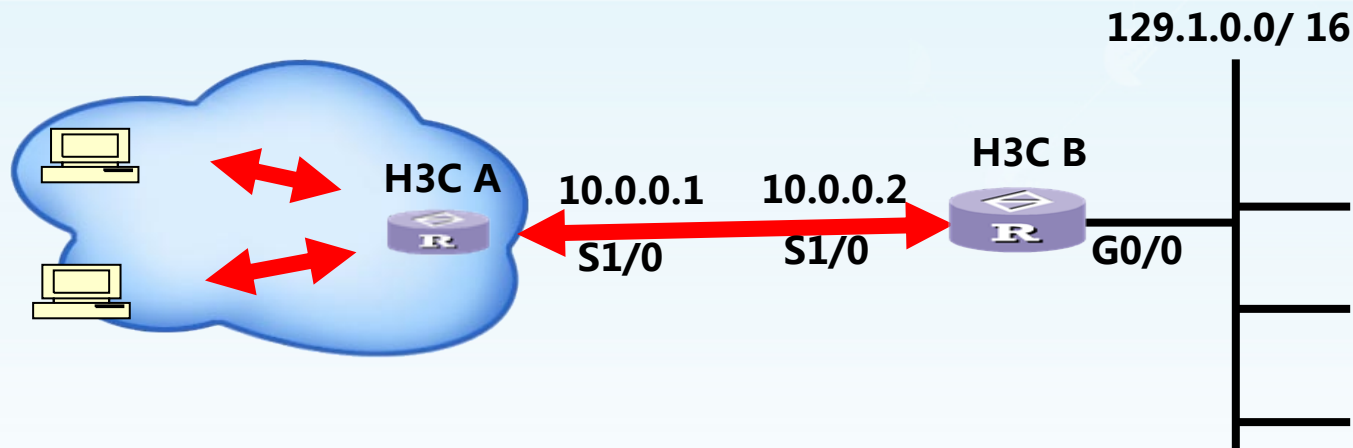
```
[H3C] ip route-static 129.1.0.0 255.255.0.0 10.0.0.2
```

```
[H3C] ip route-static 129.1.0.0 16 Serial 2
```

注意：只有下一跳所属的接口是点对点（PPP、HDLC）的接口时，才可以填写<interface-name>，否则必须填写<nexthop-address>。



静态路由配置 — 举例



在路由器 H3C A上配置：

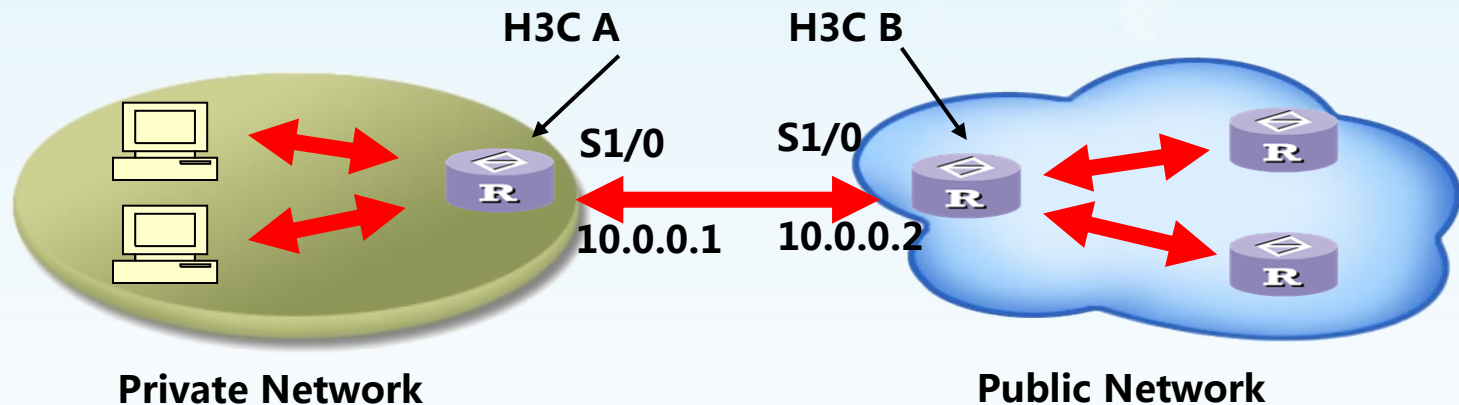
```
ip route-static 129.1.0.0 255.255.0.0 10.0.0.2
```

```
ip route-static 129.1.0.0 16 10.0.0.2
```

```
ip route-static 129.1.0.0 16 S1/0
```



静态路由配置 — 缺省路由



- 在路由器 H3C A上配置：

```
ip route-static 0.0.0.0 0 10.0.0.2
```

- Internet上大约99%的路由器上都存在一条缺省路由！



RIP配置 — 启动/关闭

- 启动RIP并进入RIP配置视图
[H3C] rip
[H3C-rip-1]
- 关闭RIP
[H3C]undo rip
Undo RIP process? [Y/N]:y

(1) 缺省情况下，路由器不启动RIP。

(2) RIP 的大部分特性都需要在RIP 视图下配置，接口视图下也有部分RIP 相关属性的配置。目前，系统支持RIP多进程。当在一台路由器上启动多个RIP进程时，需要指定不同的进程号。RIP进程号是本地概念，不影响与其它路由器之间的报文交换。因此，不同的路由器之间，即使进程号不同也可以进行报文交换。



RIP配置 — 指定工作网段

- 在路由器所连接的一个网段启动/关闭RIP
`[H3C-rip-1] [undo] network network-address [wildcard-mask]`
network-address: 路由器相应接口的IP地址
wildcard-mask: 通配符掩码（以后章节介绍）
- 在路由器所连接的所有网段启动/关闭RIP
`[H3C-rip-1] [undo] network 0.0.0.0`

注意：RIP只在指定网段上的接口运行；因此，RIP 启动后必须指定其工作网段。



RIP配置 — RIP版本

- 指定接口的版本为RIP-1
[H3C-Serial1/0] rip version 1
- 指定接口的版本为RIP-2
[H3C-Serial1/0] rip version 2 [broadcast | multicast]
- 配置全局RIP版本
[H3C-RIP-1] version { 1 | 2 }

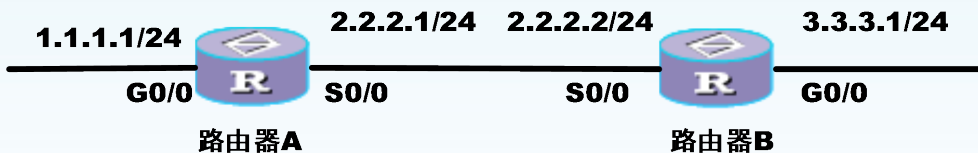
用户可以在RIP视图下配置RIP版本，也可在接口上配置RIP版本：

- 当全局和接口都没有进行RIP版本配置时，接口发送RIP-1广播报文，可以接收RIP-1广播/单播报文、RIP-2广播/组播/单播报文。
- 如果接口上配置了RIP版本，以接口配置的为准；如果接口没有进行RIP版本配置，接口运行的RIP版本将以全局配置的版本为准。



RIP配置 — 水平分割

- 启动/关闭水平分割
[H3C-Serial1/0] [undo] rip split-horizon
- 在缺省情况下，RIP启动水平分割



启动水平分割时
：

send from 2.2.2.1:
dest 1.1.1.0, mask 255.255.255.0, metric 1

关闭水平分割时
：

send from 2.2.2.1:
dest 1.1.1.0, mask 255.255.255.0, metric 1
dest 3.3.3.1, mask 255.255.255.0, metric 2



RIP配置 — 路由聚合与路由引入

- 启动/关闭RIP-2的路由聚合功能
[H3C-rip-1] [undo] summary
- 引入/取消其它协议的路由
[H3C-rip-1] [undo] import-route *protocol*
protocol: Direct, Static, OSPF, BGP, IS-IS

(1) 路由聚合仅在RIP-2下工作；在缺省情况下，RIP-2启动路由聚合。

(2) 在缺省情况下，RIP不引入其它协议的路由。

(3) 只能引入路由表中状态为active的路由，是否为active状态可以通过display ip routing-table protocol命令来查看

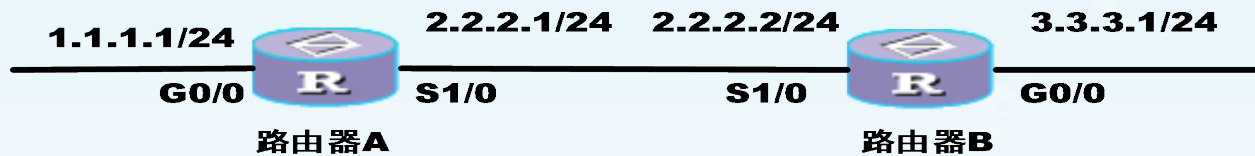


RIP配置 — 显示与调试

- 显示当前RIP运行状态与配置信息
[任意视图] display rip *Process ID*
- 对RIP报文进行调试
 - <H3C> terminal debugging
 - <H3C> terminal monitor
 - <H3C> debugging rip 1 packet
- 关闭RIP报文调试
 - <H3C> undo debugging rip 1 packet



RIP配置 — 举例



```
interface GigabitEthernet 0/0  
ip addr 1.1.1.1 255.255.255.0
```

```
interface serial 1/0  
ip addr 2.2.2.1 255.255.255.0
```

```
rip  
network 1.1.1.1  
network 2.2.2.1
```

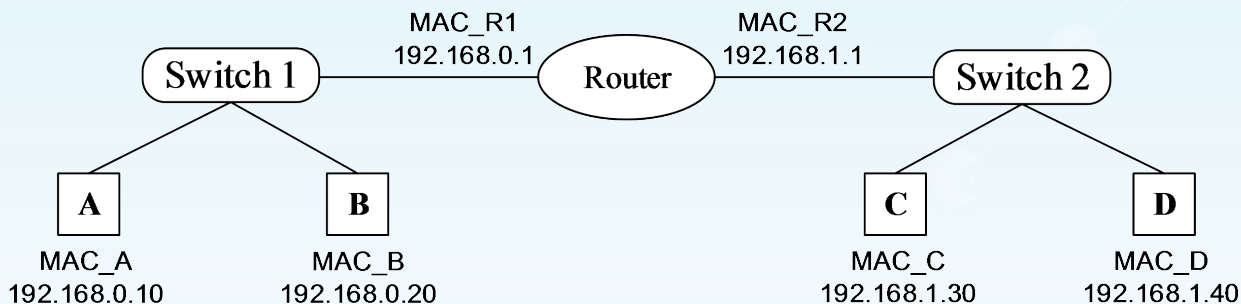
```
interface GigabitEthernet 0/0  
ip addr 3.3.3.1 255.255.255.0
```

```
interface serial 1/0  
ip addr 2.2.2.2 255.255.255.0
```

```
rip  
network 2.2.2.2  
network 3.3.3.1
```



附录 — IP包在网络上的传输过程



主机A发送一个IP分组给主机D：

ARP Request (MAC_A, ff:ff:ff:ff:ff:ff, who has IP 192.168.0.1)

ARP Reply (MAC_R1, MAC_A)

运载IP分组的帧 (MAC_A, MAC_R1, 192.168.0.10, 192.168.1.40)

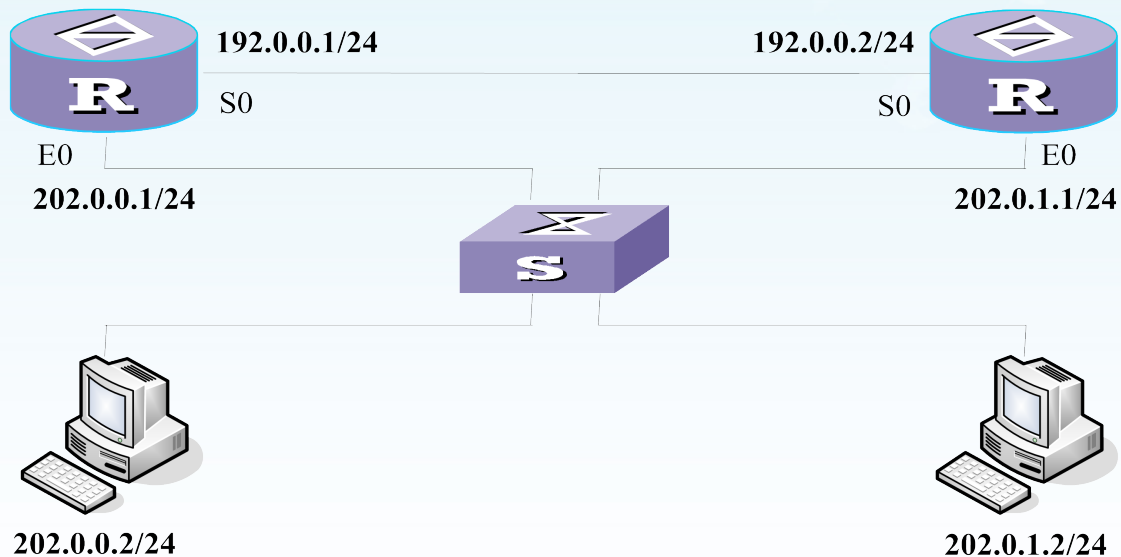
ARP Request (MAC_R2, ff:ff:ff:ff:ff:ff, who has IP 192.168.1.40)

ARP Reply (MAC_D, MAC_R2)

运载IP分组的帧 (MAC_R2, MAC_D, 192.168.0.10, 192.168.1.40)



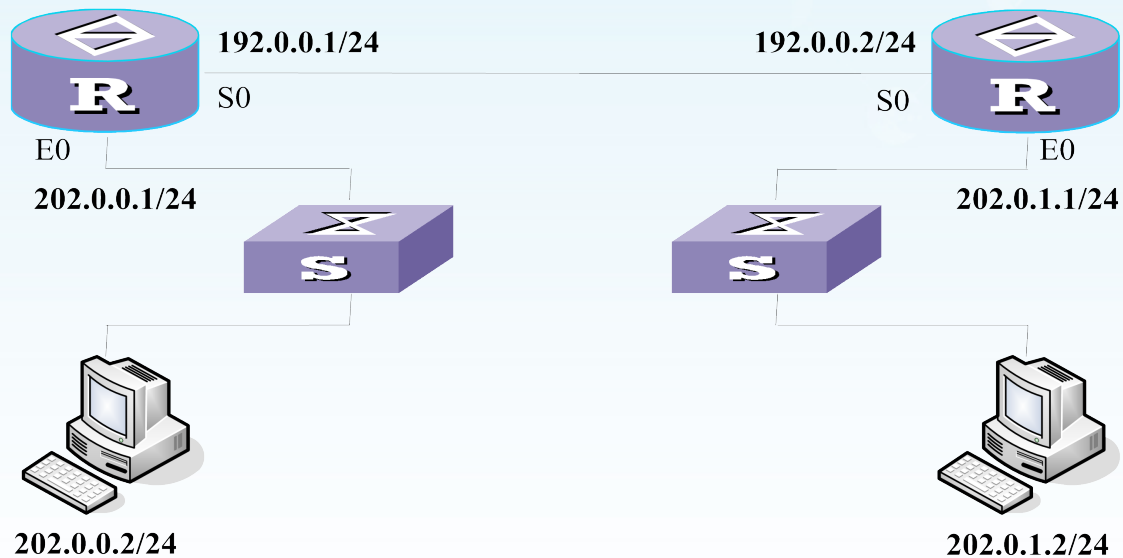
附录 — 实验环境讲解



完全等价于下一页的实验环境



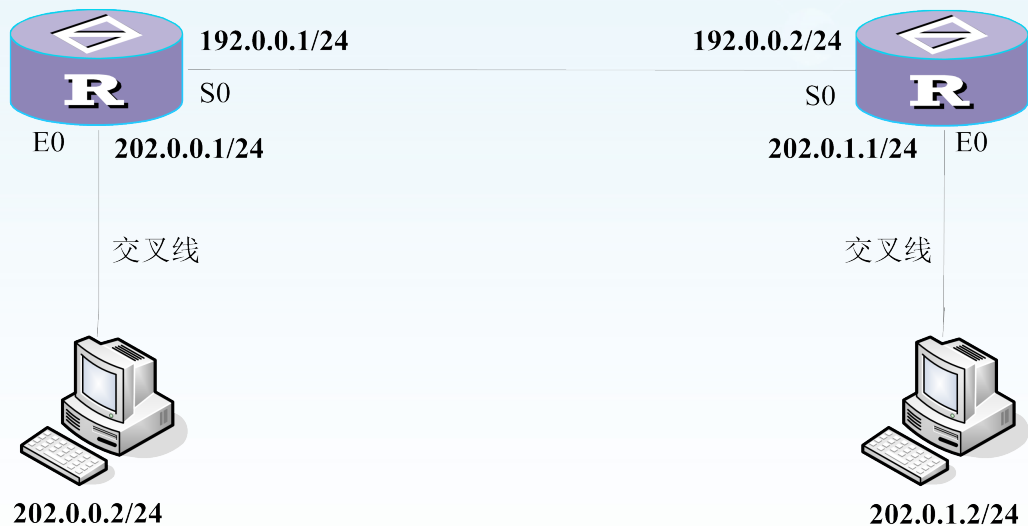
附录 — 实验环境讲解（续）



完全等价于下一页的实验环境



附录 — 实验环境讲解（续）



附录一 显示路由表

[H3C] display ip routing-table

Routing Tables:

Destination/Mask	Proto	Pref	Metric	Nexthop	Interface
127.0.0.0/8	Direct	0	0	127.0.0.1	LoopBack0
127.0.0.1/32	Direct	0	0	127.0.0.1	LoopBack0
192.0.0.0/24	Direct	0	1	192.0.0.2	Serial0
192.0.0.1/32	Direct	0	0	127.0.0.1	LoopBack0
192.0.0.2/32	Direct	0	1	192.0.0.2	Serial0
202.0.0.0/24	Direct	0	1	202.0.0.1	Ethernet0
202.0.0.1/32	Direct	0	0	127.0.0.1	LoopBack0
202.0.1.0/24	Static	60	2	192.0.0.2	Serial0



附录 — dis cu

[任意视图] **display current-configuration**

#

sysname H3c

#

local-user *sws* service-type administrator password simple *123456*

#

interface GigabitEthernet0/0

description Don't change the configuration please

ip address 10.110.98.137 255.255.255.0

#

interface Serial1/0

link-protocol ppp

ip address 100.110.1.1 255.255.255.0

ppp authentication-mode pap

(1) 列出所有视图，以及在该视图下生效的命令；

(2) 所列命令的格式严格符合语法；

