

实验：EIGRP

分解实验 - EIGRP

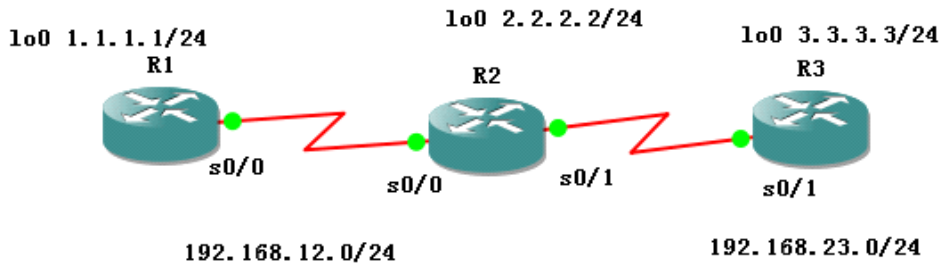
臧家林制作



- EIGRP 实验 1：基本实验
- EIGRP 实验 2：邻居的建立
- EIGRP 实验 3：度量值的计算
- EIGRP 实验 4：默认路由
- EIGRP 实验 5：手工汇总
- EIGRP 实验 6：负载均衡
- EIGRP 实验 7：身份验证
- EIGRP 实验 8：被动接口
- EIGRP 实验 9：eigrp stub

EIGRP 实验 1：基本实验

全网运行 EIGRP 90，能够实现全网的互通，掌握几种常用的 show 命令



1. GNS 3 拓扑的搭建

2. 按照图例，配置 IP 地址，测试直连的连通性

3. 配置 EIGRP 90，查看路由学习是否完整

4. show 命令

show ip eigrp neighbor

show ip route eigrp

show ip route

show ip protocols

show ip eigrp interface

show ip eigrp topology

show ip eigrp traffic

5. 使用偏移列表修改 metric 值, 值只能增不能减少

R1 :

access-list 1 permit 2.2.2.0

router ei 90

offset-list 1 in 14

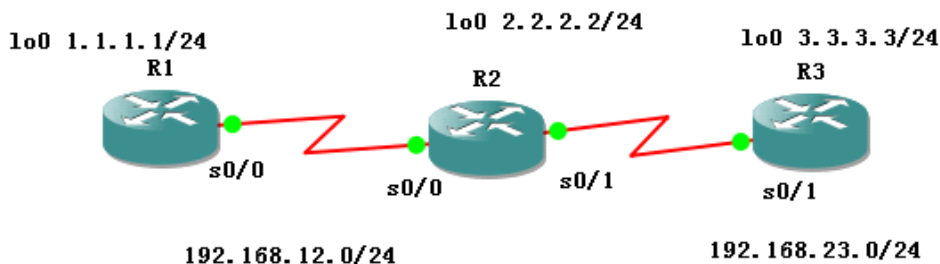
14 是增加的值，原来是 2297856，偏移之后，变为 2297856 + 14 = 2297870

R1 : show ip route 主要关注 2.2.2.0

```

R1#show ip route eigrp
 2.0.0.0/24 is subnetted, 1 subnets
D    2.2.2.0 [90/2297870] via 192.168.12.2, 00:03:41, Serial0/0
 3.0.0.0/24 is subnetted, 1 subnets
D    3.3.3.0 [90/2809856] via 192.168.12.2, 00:03:34, Serial0/0
D    192.168.23.0/24 [90/2681856] via 192.168.12.2, 00:07:50, Serial0/0
R1#
  
```

EIGRP 实验 2：邻居的建立



EIGRP 邻居起不来的原因有三种：

1) AS 号不匹配

AS 号不同的两个 EIGRP 是建不起邻居的，因为它们的进程号不相同。

```
no router eigrp 90
```

```
router eigrp 91
```

2) K 值不相同

EIGRP 计算开销有 5 个因素：带宽，延迟，负载，可靠性，MTU。用 K1，K2，K3，K4，K5 分别代替这几个值。当 K 值不同时，计算开销的方式就不同了，就不能传递路由了，即不能建邻居

默认为：metric weights 0 1 0 1 0 0

R1#show ip protocols 可以查看到 K 值

修改：

```
router eigrp 90
```

```
metric weights 0 1 1 1 1 1
```

```
metric weights tos K1 K2 K3 K4 K5
```

3) HELLO 时间不同

EIGRP 的 hello 包更新时间为 5 秒，死亡时间为 15 秒，如果有一个进程的 hello 时间改成大于了 5 秒，在 5 秒内对方

没有收到 hello 包就会认为没有邻居，此时也起不来邻居，但是过了改了的时间后又会给对方一个 hello 包，对方过 5 秒就回了 hello 包，此时邻居又会起来，所以 hello 时间不同时会出现邻居时而有时而又断了的情况。

```
int s0/0
```

```
ip hello-interval eigrp 90 10
```

查看 hello 时间 show ip eigrp interface detail

EIGRP 实验 3：度量值的计算

计算 EIGRP 度量值时使用 5 个变量,但默认情况下,只使用其中的两个(带宽.延迟)

带宽:源和目的地之间的最小带宽

延迟:路径上接口的累计延迟

可靠性:源和目的地之间的最低可靠性,基于存活消息

负载:源和目的地之间链路上的最重负载,基于分组速率和接口的配置带宽

MTU:最大传输单元,路径上最小的 MTU

EIGRP 度量值的计算公式= $[K1 * \text{Bandwidth} + (K2 * \text{Bandwidth}) / (256 - \text{Load}) + K3 * \text{Delay}] * [K5 / (\text{Reliability} + K4)] * 256$

默认情况下， $K1 = K3 = 1$ ， $K2 = K4 = K5 = 0$

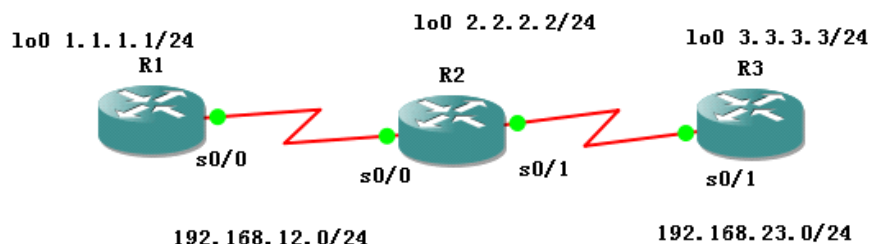
$\text{Bandwidth} = 10^7 / \text{所经由链路中入口带宽 (单位为 Kbps)}$ 的最小值

$\text{Delay} = \text{所经由链路中入口的延迟之和 (单位为 } \mu\text{s)} / 10$

也就是说默认情况下，计算公式= $[10^7 / \text{Bandwidth} + \text{Delay}] * 256$

6

R2 上看 R1 的 1.1.1.1 metric 值为：2297856



1.带宽应该是从 R1 的 lo0 到 R2 的最小带宽，应该是 R2 的 s0/0 接口为 1544

show int lo0

show int s0/0

2.延迟是路由器 R1 的 lo0 和路由器 R2 的 s0/0 接口的延迟之和

lo0 : 5000 s0/0:20000

$(5000+20000)/10=2500$

$[10^7/1544+2500]*256$

取整 $6476+2500=8976$

$8976*256=2297856$

R2： 修改带宽值， 修改 R1 的 s0/0 是不会发生变化的

int s0/0

bandwidth 128

查看 1.1.1.1 的 metric 发生变化了吗？

EIGRP 实验 4：默认路由

要配置 EIGRP 默认路由，可使用全局配置命令 ip default-network。

配置中包含该命令的路由器将 network number 视为最后求助（last-resort）网关，并将其通告给其它路由器。

仅当使用该命令的路由器能够到达指定的网络时，才会将其作为候选默认路由器通告给其他 EIGRP 路由器

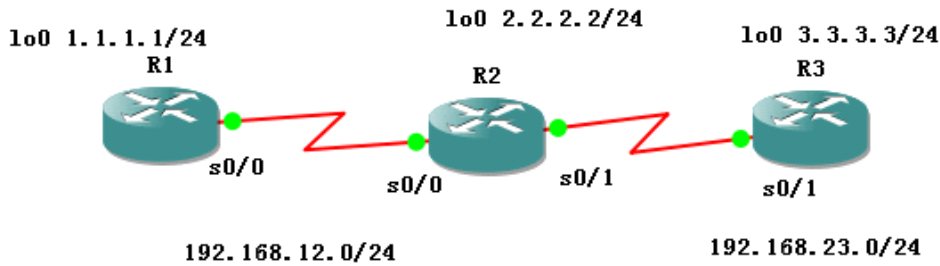
默认路由是静态路由中的一种，用途：用通俗的话来说就是数据包没有别的路径选择了，最后才选择默认路由，若没有默认路由目的地址在路由表中没有匹配表项的包将被丢弃

默认路由是一种特殊的静态路由,指的是当路由表中与包的目的地址之间没有匹配的表项时路由器能够做出的选择.如果没有默认路由,那么目的地址在路由表中没有匹配表项的包将被丢弃. 默认路由在某些时候非常有效,当存在末梢网络时,默认路由会大大简化路由器的配置,减轻管理员的工作负担,提高网络性能.

默认路由 (Default route) ，是对 IP 数据包中的目的地址找不到存在的其他路由时，路由器所选择的路由。目的地不在路由器的路由表里的所有数据包都会使用默认路由。这条路由一般会连去另一个路由器，而这个路由器也同样处理数据包: 如果知道应该怎么路由这个数据包，则数据包会被转发到已知的路由；否则，数据包会被转发到默认路由，从而到达另一个路由器。每次转发，路由都增加了一跳的距离。

`ip route 0.0.0.0 0.0.0.0 null 0`

保证所有不匹配的路由都丢弃到 NULL0 口 (也就是丢弃所有无法匹配的报文)
如果一个报文找不到一条匹配的路由，会被送到路由器 CPU，当有大量的这样的路由出现的时候，会导致 CPU 负载很高，所以这样做可以减轻路由器 CPU 负荷



```

router eigrp 90
no net 1.1.1.0 0.0.0.255
net 1.0.0.0
auto-summary
ip default-network 1.0.0.0
  打开自动汇总
ip default-network 命令后面必须跟主类网络
  
```

1.重分布静态 (1、 R1: 0.0.0.0 S0/0 2、 Redistribute)
 2.network 0.0.0.0 (1、 R1: 0.0.0.0 null0 2、 network 1.0.0.0)
 3.ip default-network (ip default-net 1.0.0.0 2、 net 1.0.0.0 3、 本地有此路由)
 4..在接口上汇总路由 (ip summary-address eigrp 90 0.0.0.0 0.0.0.0)

1. 重 分 布 静 态 (1 、 R1: 0.0.0.0 S0/0 2、 Redistribute)

```

R1:
ip route 0.0.0.0 0.0.0.0 1.1.1.254
router eigrp 90    redis static
  
```

```

show ip route
D*EX 0.0.0.0/0 [170/2297856] via 192.168.12.1
  
```

**2.network 0.0.0.0 (1 、 R1: 0.0.0.0 null0
2、 network 1.0.0.0)**

R1:

```
no ip route 0.0.0.0 0.0.0.0 1.1.1.254
router eigrp 90
no redis static
```

```
ip route 0.0.0.0 0.0.0.0 null 0
router eigrp 90
net 0.0.0.0
```

```
show ip route
```

```
D* 0.0.0.0/0 [90/2169856] via 192.168.12.1
```

3.ip default-network

(ip default-network 1.0.0.0 2、 network 1.0.0.0 3、 本地有此路由)

R1:

```
no ip route 0.0.0.0 0.0.0.0 null 0
router eigrp 90
no net 0.0.0.0
```

```
ip default-network 1.0.0.0
router eigrp 90
auto-summary
net 1.0.0.0
```

```
show ip route
```

```
D* 1.0.0.0/8 [90/2297856] via 192.168.12.1
```

4..在接口上汇总路由 (ip summary-address eigrp 90 0.0.0.0 0.0.0.0)

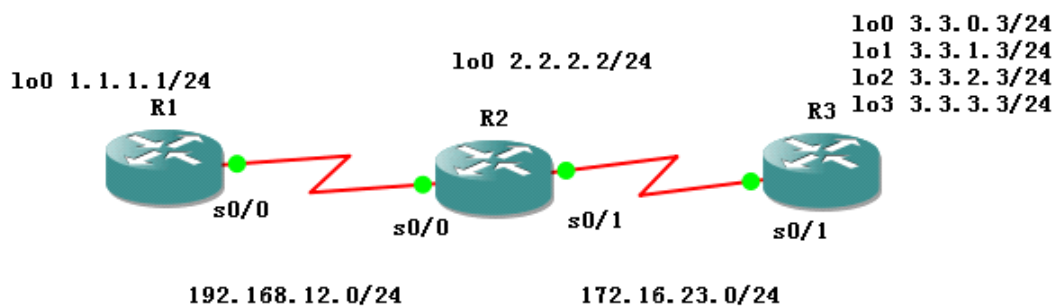
R1:


```
no ip default-network 1.0.0.0
router eigrp 90
no auto-summary
no net 1.0.0.0
```

```
ip route 0.0.0.0 0.0.0.0 null 0
int s0/0
ip summary-address eigrp 90 0.0.0.0 0.0.0.0
```

```
show ip route
D*EX 0.0.0.0/0 [170/2169856] via 192.168.12.1
```

EIGRP 实验 5 : 手工汇总



默认时 EIGRP 的自动汇总总是开启的，自动汇总只对本地产生的 EIGRP 路由汇总，通过 no auto-summary 命令关闭自动汇总，然后进行手工汇总

手工汇总是基于接口配置的

R3 :

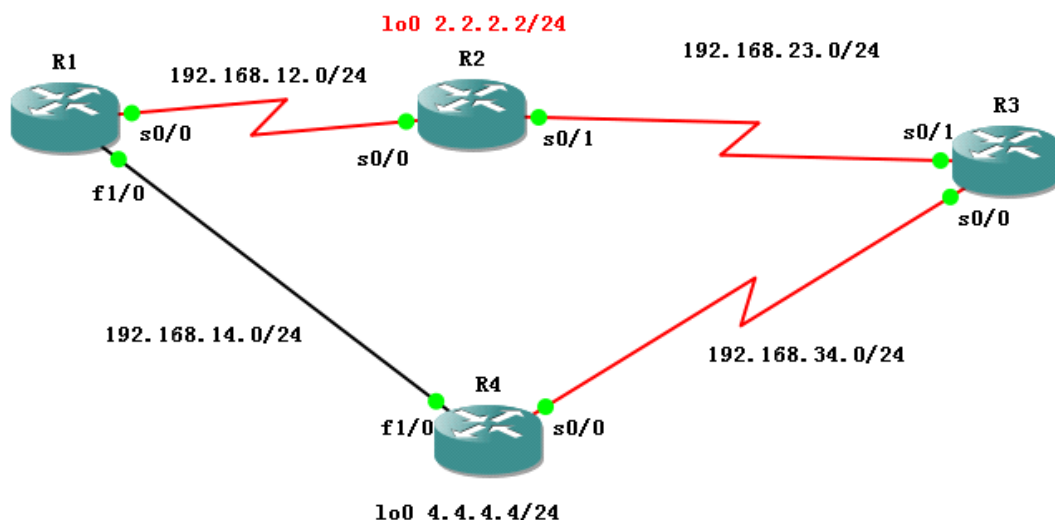
```
int s0/1
```

```
ip summary-address eigrp 90 3.3.0.0 255.255.252.0
```

在路由器 R3 上的 s0/1 执行手工汇总后，会在自己的路由表中产生一条指向 null0 的 EIGRP 路由主要是为了防止路由环路

当被汇总的明细路由全部 down 掉以后，汇总路由才自动从路由表里被删除，从而可以有效避免路由抖动

EIGRP 实验 6：负载均衡



关注路由器 R2 的环回口 lo0

R4: show ip route show ip eigrp topology

R4 把 F1/0 延迟改为 2000，默认为 100

int f1/0

delay 2000

clear ip route *

show ip route 可以看到 2.2.2.0 有两条等价路由了
表明 EIGRP 是支持等价负载均衡的

不等价负载均衡

将延迟改默认值,通过 variance 命令来研究 EIGRP 非等价负载均衡

Variance 是 EIGRP 做非等价负载均衡用的

$FD \leq FD_{min} * Variance$ 的路由可以进入路由表

比如说到 2.2.2.2 有 3 条路径 FD 分别为 10 20 40，最小的 FD 值为 10，即 $FD_{min}=10$

默认情况下 Variance 为 1，所以只有 FD 为 10 的路由可以进

入路由供路由选路使用

但如果你希望 FD 为 20 的路由也进入路由表，那么你可以把 Variance 调到 2 以上，
那么 FD 为 20 的路由也能进入路由表，两条路由实现负载均衡

```
P 2.2.2.0/24, 1 successors, FD is 2300416  
    via 192.168.14.1 (2300416/2297856),  
FastEthernet1/0  
    via 192.168.34.3 (2809856/2297856), Serial0/0
```

EIGRP 可以支持非等价负载均衡，最多支持 6 条，默认为 4 条，但非等价负载均衡功能默认为关闭状态。EIGRP 只能将拓扑数据库中的备用链路放入路由表执行负载均衡，拓扑数据库中可能有多条备用链路，而且多条链路的 Metric 值也可能各不相同，当启用非等价负载均衡时，需要定义什么样的 Metric 范围可以用来负载均衡，这需要通过控制 Metric 的变量 (Variance) 值来控制，具体方法如下：

路由表中正在使用的最优路由的 Metric 值为 FD，而拓扑数据库中备用路由的 Metric 值肯定是大于 FD 的，Variance 值通过控制备用链路的 Metric 值与 FD 的倍数关系来控制，就是 Variance 值取多少，备用链路的 Metric 在 FD 的 Variance 值倍数范围内就有资格执行负载均衡，例如当前 FD 为 20，3 条备用链路 Metric 分别为 30，50，55，如果 Variance 值取 2，那么 Metric 值范围在 $20 \times 2 = 40$ 的链路都可以执行负载均衡，所以 Metric 值为 30 的链路可以执行负载均衡，而 Metric 值为 50 和 55 的却不可以，因为大于 40，只有当 Variance 值取 3 时，Metric 值范围在 $20 \times 3 = 60$ 的链路才可以执行负载均衡，所以 Metric 值为 50 和 55 只有在 Variance 值取 3 时才可以执

行负载均衡。

注：Variance 值默认取值为 1，也就是不执行非等价负载均衡，但会执行等价负载均衡。

R4：

```
int f1/0
```

```
no delay 2000
```

```
router eigrp 90
```

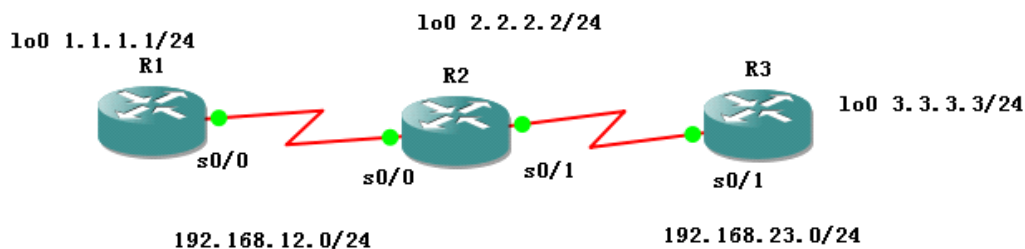
```
variance 2
```

show ip route 路由条目 2.2.2.0 有两条路径，但是它们的度量值不同，从而证明 EIGRP 支持非等价负载均衡

```
R4#show ip protocols
```

```
R4#show ip protocols
Routing Protocol is "eigrp 90"
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 90
  EIGRP NSF-aware route hold timer is 240s
```

EIGRP 实验 7：身份验证



通过对邻居路由器进行身份验证，可以避免路由器收到伪造的路由更新。

通过配置 EIGRP 邻居身份验证，可以让路由器根据预定义的

密码参与路由选择。

路由器使用两种身份验证方式：

明文身份验证

MD5 身份验证

EIGRP 只支持 MD5 验证

EIGRP 认证是基于接口来设置的

全局命令，进入密钥链配置模式

key chain ccnp

key 1 --> 密钥 ID

key-string cisco --> 指定密钥字符串（密码）

接口配置

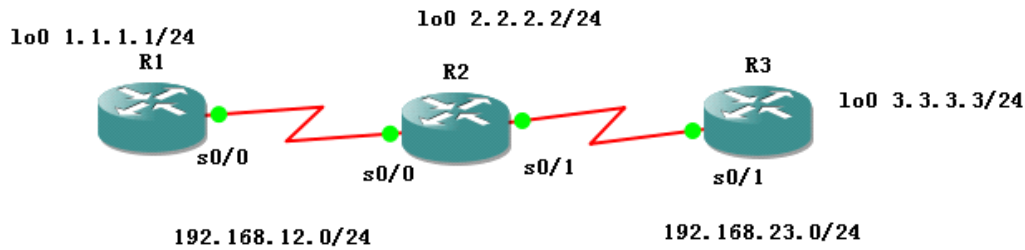
int s0/0

ip authentication mode eigrp 90 md5 --> 指定对 EIGRP 分组使用 MD5 身份验证

ip authentication key-chain eigrp 90 ccnp --> 指定使用密钥链中的密钥对 EIGRP 分组进行身份验证

EIGRP 实验 8：被动接口

R1 身后的网络 1.1.1.0 下面没有任 EIGRP 邻居，这样让 EIGRP 发送各种更新到这些接口很明显是没有必要的，我们可以使用 passive 命令让这些接口不发送更新



```

debug eigrp packets
un all
show run | sec eigrp

```

配置被动接口

```

router eigrp 90
passive-interface lo0

```

如果被动接口很多，可以全部改成被动接口，再将部分接口关闭被动

```

router eigrp 90
passive-interface default
no passive-interface s0/0

```

debug eigrp packets 更新包不再发给 1.1.1.1

EIGRP 被动接口，不发送也不接收

```

router eigrp 90
passive-interface s0/0

```

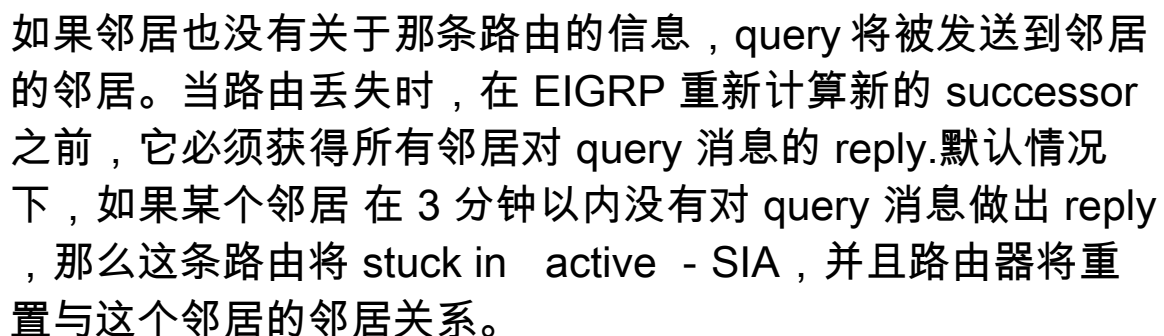
会影响邻居的建立

EIGRP 实验 9 : eigrp stub

EIGRP stub 特性提高网络的稳定性,减少资源使用简单化了分支路由器(spoke) 配置.Stub 路由一般用于 hub-and-

当邻居收到末节路由器的通告后，不会向此路由器发出任何路由的查询请求。

当某条路由丢失，并且没有可行后继进 FS，EIGRP 会发送 query，丢失的那条路由处于 active 状态 query 将被发向所有的邻居，除了原生的后继者 successor.



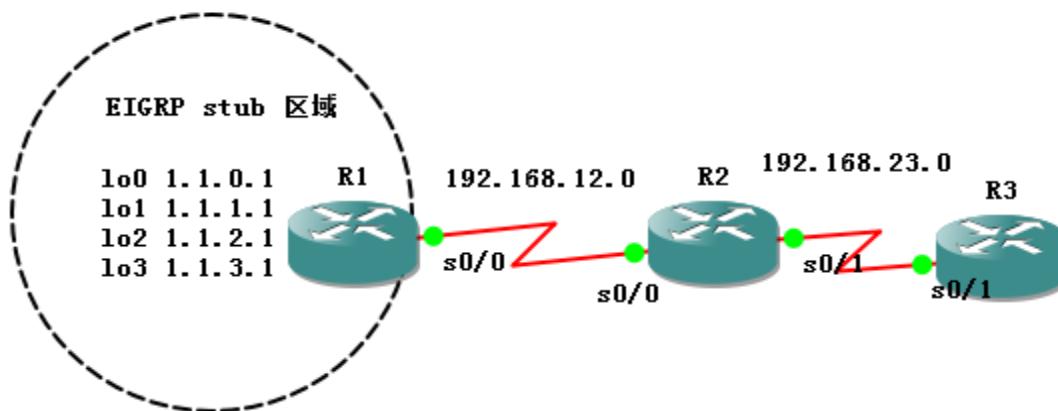
如果 EIGRP 的网络多达数十台及数百台路由，那么如果某个网络出错，则有可能引起整个网络产生大量的查询回复包。

所以为了避免这样的问题，可以使用 ip summary 命令或 stub 参数来进行配置 EIGRP,限制 EIGRP 的查询范围。

```
Router(config-router)#
```

```
eigrp stub [receive-only|connected|static|summary]
```

- receive-only: 阻止末节路由器发送任何路由器更新。
- connected: 允许末节发送直连的路由信息(可能仍然需要重发布)。
- static: 允许末节发送静态路由信息(必须重发布)。
- summary: 允许发送汇总路由信息。
- 缺省参数为 connected 和 summary。



eigrp stub 命令可以阻止向末节路由器发送查询信息，加入不同参数可控制发送更新的内容

R1

起一条静态路由

```
ip route 5.5.5.0 255.255.255.0 null 0
```

把静态路由重分布进 EIGRP

```
router eigrp 90
```

```
redistribute static metric 10000 100 255 1 1500
```


手工汇总

int s0/0

ip summary-address eigrp 90 1.1.0.0 255.255.252.0

R2: 查看路由表

C 192.168.12.0/24 is directly connected, Serial0/0

1.0.0.0/22 is subnetted, 1 subnets

D 1.1.0.0 [90/2297856] via 192.168.12.1, 00:00:03, Serial0/0

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

5.0.0.0/24 is subnetted, 1 subnets

D EX 5.5.5.0 [170/2195456] via 192.168.12.1, 00:00:03, Serial0/0

C 192.168.23.0/24 is directly connected, Serial0/1

connect 参数

R1: connect 允许 stub 路由器发送直连路由更新

router eigrp 90

eigrp stub connect

connect 允许发布直连的网络

R2#show ip route eigrp

1.0.0.0/24 is subnetted, 4 subnets

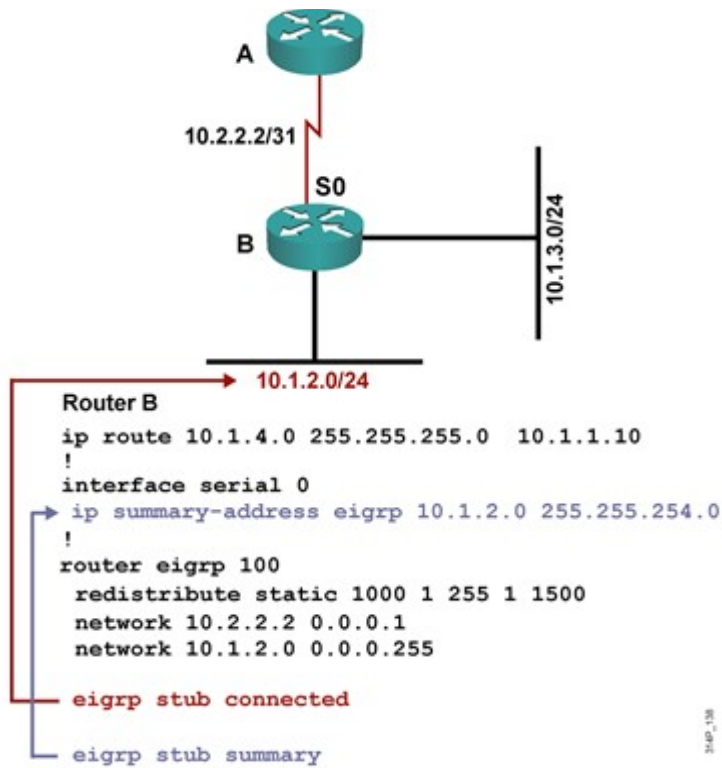
D 1.1.0.0 [90/2297856] via 192.168.12.1,

D 1.1.1.0 [90/2297856] via 192.168.12.1,

D 1.1.2.0 [90/2297856] via 192.168.12.1,

D 1.1.3.0 [90/2297856] via 192.168.12.1,

R2#



summary 参数

R1: summary 允许 stub 路由器发送汇总路由信息

router eigrp 90

eigrp stub summary

R2#show ip route eigrp

1.0.0.0/22 is subnetted, 1 subnets

D 1.1.0.0 [90/2297856] via 192.168.12.1

R2#

static 参数

R1: static 允许 stub 路由器发送重发布静态的路由

router eigrp 90

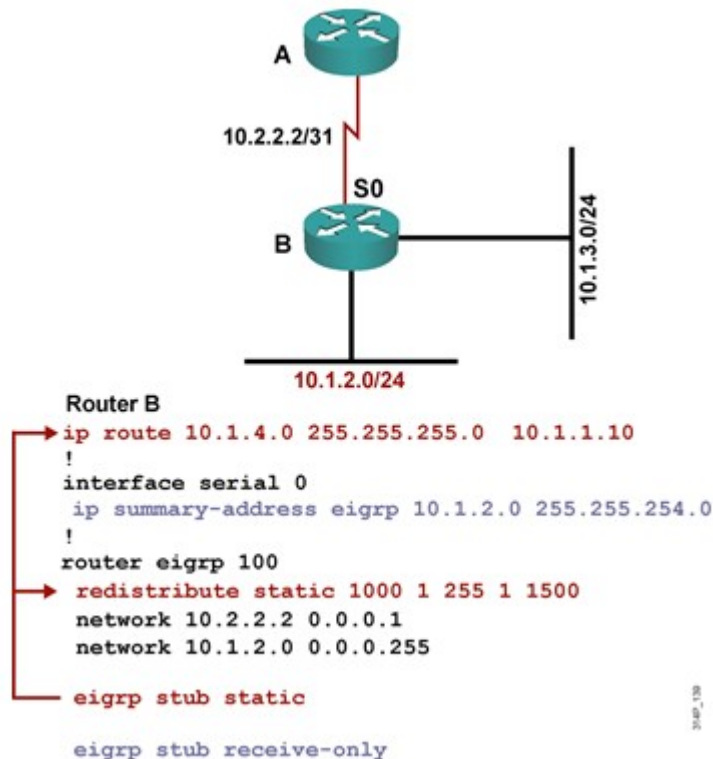
eigrp stub static

R1:show ip route 只有 R2 重发布进来的路由，没有直连路由

```

R2#show ip route eigrp
      5.0.0.0/24 is subnetted, 1 subnets
D EX    5.5.5.0 [170/2195456] via 192.168.12.1
R2#

```



receive-only 参数

R1: receive-only 防止 stub 路由器发送路由更新

```
router eigrp 90
```

```
eigrp stub receive-only
```

show ip route 没有 EIGRP 路由，说明路由器没有发送

```
R2#show ip route eigrp
```

```
R2#
```

默认参数：connected summary

```
router eigrp 90
```

```
eigrp stub
```

默认 stub 选项后面是 connected summary

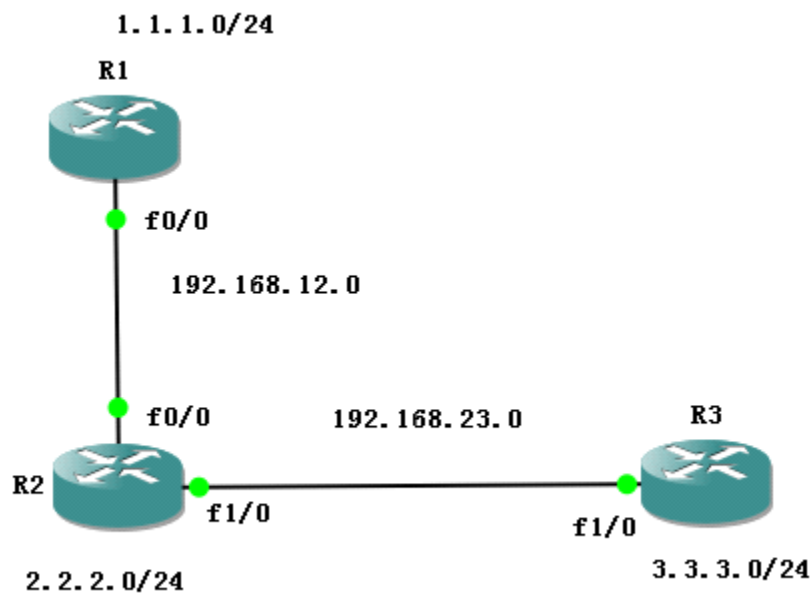
```
R2#show ip route eigrp
      1.0.0.0/22 is subnetted, 1 subnets
D      1.1.0.0 [90/2297856] via 192.168.12.1
R2#
```

总结

router eigrp 90

eigrp stub

connect	允许 stub 路由器发送直连路由更新
summary	允许 stub 路由器发送汇总路由信息
static	允许 stub 路由器发送重发布静态的路由
receive-only	防止 stub 路由器发送路由更新
默认 stub	选项后面是 connected summary



R1 R2 运行 EIGRP

R2 做为 STUB 路由器，R2 做静态路由指向 R3 的网络。同时

R2 对 R1 方向做汇总路由，STUB 后面的参数只指定 eigrp stub connect,查看实验现象

```
R2:
int lo0
ip add 2.2.2.2 255.255.255.0
int f0/0
ip add 192.168.12.2 255.255.255.0
ip summary-add eigrp 90 4.4.0.0 255.255.248.0 5
no shut
int f1/0
ip add 192.168.23.2 255.255.255.0
no shut

router eigrp 90
no aut
net 2.2.2.0 0.0.0.255
net 192.168.12.0
net 192.168.23.0
redis static metric 10000 100 255 1 1500
exit
ip route 3.3.3.0 255.255.255.0 192.168.23.3

router eigrp 90
eigrp stub connect
```

R2 没做 stub 之前，R1 上有 3.3.3.0

R1:show ip route eigrp

```
R1#show ip route eigrp
  2.0.0.0/24 is subnetted, 1 subnets
D       2.2.2.0 [90/156160] via 192.168.12.2, 00:01:00, FastEthernet0/0
  3.0.0.0/24 is subnetted, 1 subnets
D EX    3.3.3.0 [170/284160] via 192.168.12.2, 00:01:00, FastEthernet0/0
D       192.168.23.0/24 [90/30720] via 192.168.12.2, 00:01:00, FastEthernet0/0
R1#
```

R2 使用 `eigrp stub connect`，则没有了
`connect` 允许发布直连的网络，没有重发布的路由
`connect` 允许 STUB 路由器发送直连路由更新

R2 `static` 允许 STUB 路由器发送重发布静态的路由
`eigrp stub static`

R1: `show ip route` 只有 R2 重发布进来的路由，没有直连路由

```
R1#show ip route eigrp
      3.0.0.0/24 is subnetted, 1 subnets
D EX    3.3.3.0 [170/284160] via 192.168.12.2, 00:00:00
R1#
```

R2 : `receive-only` 防止 STUB 路由器发送路由更新
`eigrp stub receive-only`

R1: `show ip route` 没有 EIGRP 路由，说明 R2 没有发送

```
R1#show ip route eigrp
R1#
```

参数可以加在一起

R2 :

`router eigrp 90`

`eigrp stub connected static summary`

默认 STUB 选项后面是 `connected summary`