

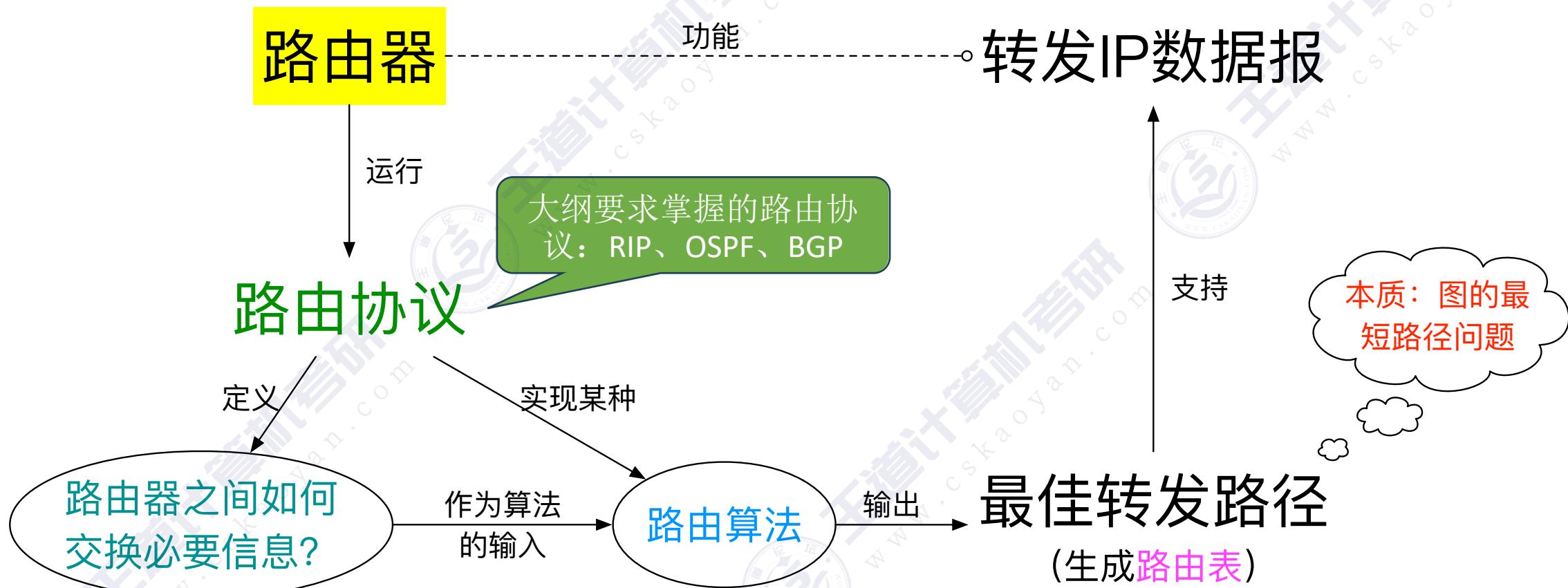
本节内容

RIP

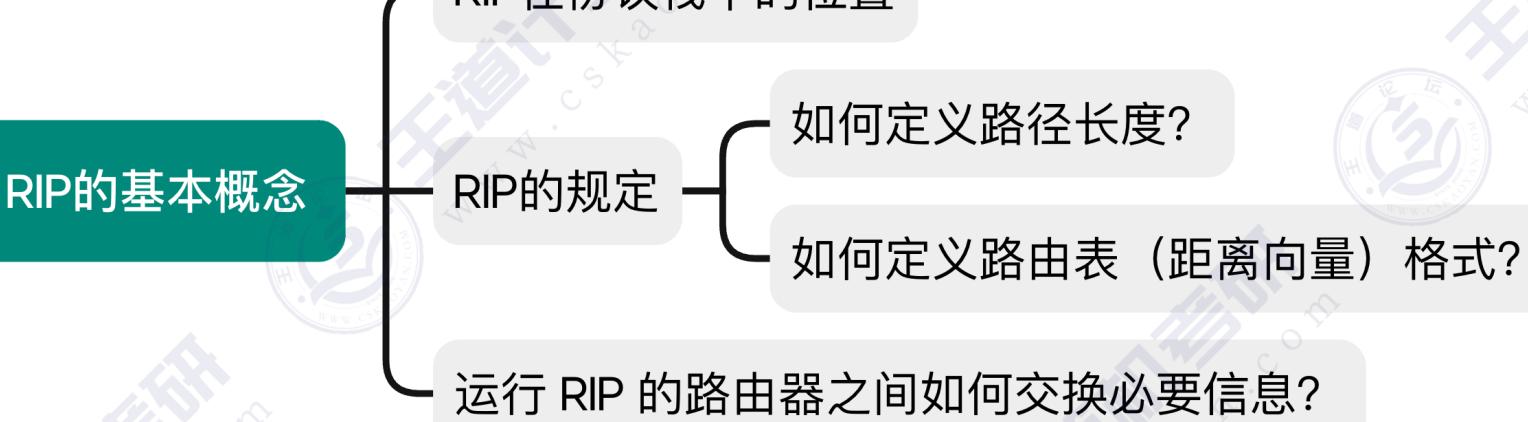
路由信息协议

(Routing Information Protocol)

路由算法、路由协议之间的关系

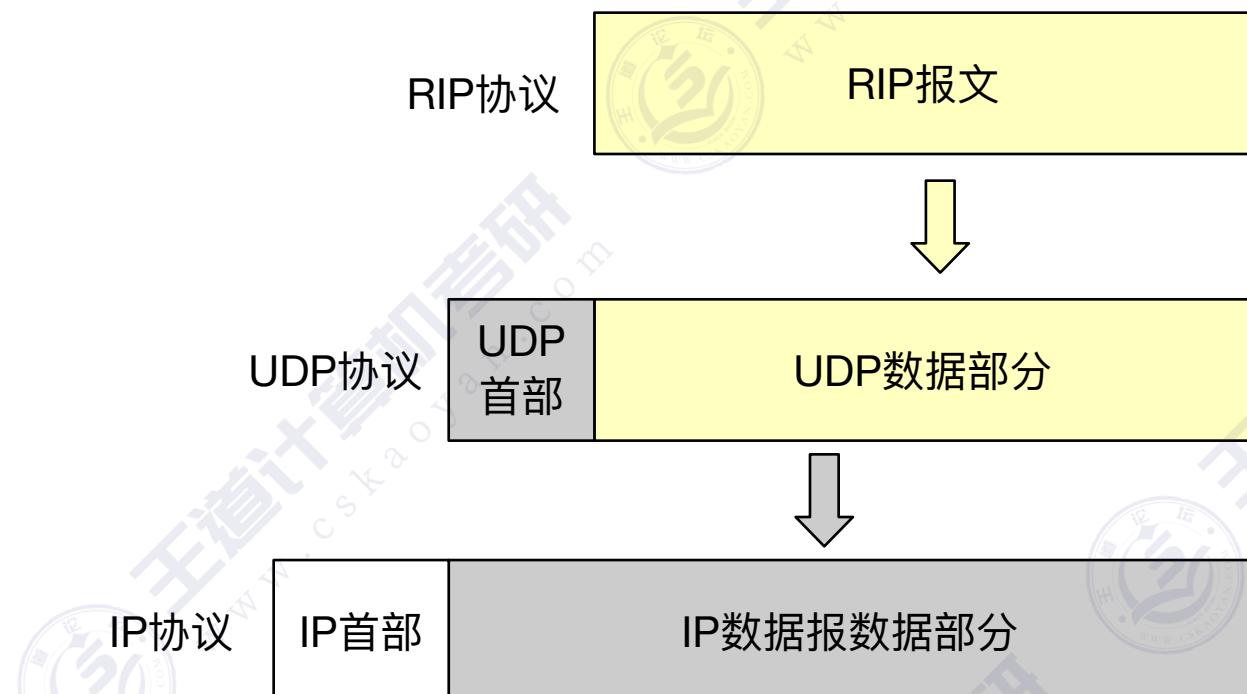


知识总览



RIP 在协议栈中的位置

RIP 属于应用层，它使用 UDP 传送数据（端口=520）



RIP协议定义了两种报文：

Request 报文：请求邻居路由器发送其路由表，用于启动或查询。

Response 报文：响应请求或定期发送给邻居路由器，**携带完整或部分**路由表。

解释：由于每个**Response**报文最多携带 25 个路由表项信息，因此如果路由表太大，就需要拆分为多个**Response**报文，**每个报文携带部分**路由表。

RIP 的规定

RIP如何
定义路
径长度?

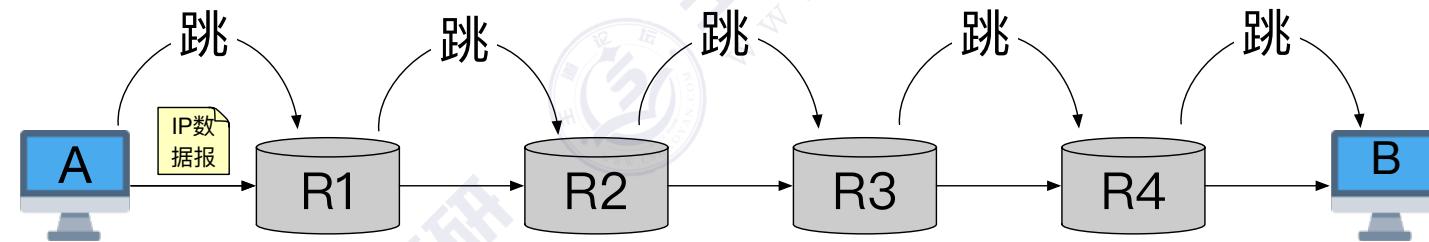
- 1) RIP 使用跳数 (Hop Count) (或称距离) 来衡量到达目的网络的距离。
规定: 路由器到直连网络的距离为 1; 而每经过一个路由器, 距离就加 1。
- 2) RIP 认为好的路由就是跳数最少的。
- 3) RIP 允许一条合法路径距离不能超过 15。距离等于 16 时表示网络不可达。可见 RIP 只适用于小型自治系统。

RIP如何
定义路由
表格式?

- 4) 每个路由器都要维护自己的路由表, 路由表项有三个关键字段: <目的网络 N, 距离 d, 下一跳路由器地址 X >。
- 5) 每个路由器都要维护从它自身到其他每个目的网络的距离记录, 即距离向量 (事实上, 距离向量信息包含在路由表中, RIP 协议使用“距离向量路由算法”计算最短路径)

某路由器的路由表示例

目的网络	距离	下一跳路由器
Net1	1	直接交付
Net2	16	NULL
Net3	2	R6
Net4	2	R9



运行 RIP 的路由器之间如何交换必要信息？

1) Who, 和谁交换信息：仅和直接相邻的路由器交换信息。

其实就是
距离向量

2) What, 交换什么信息：交换的信息是本路由器所知道的全部信息，即自己的路由表。

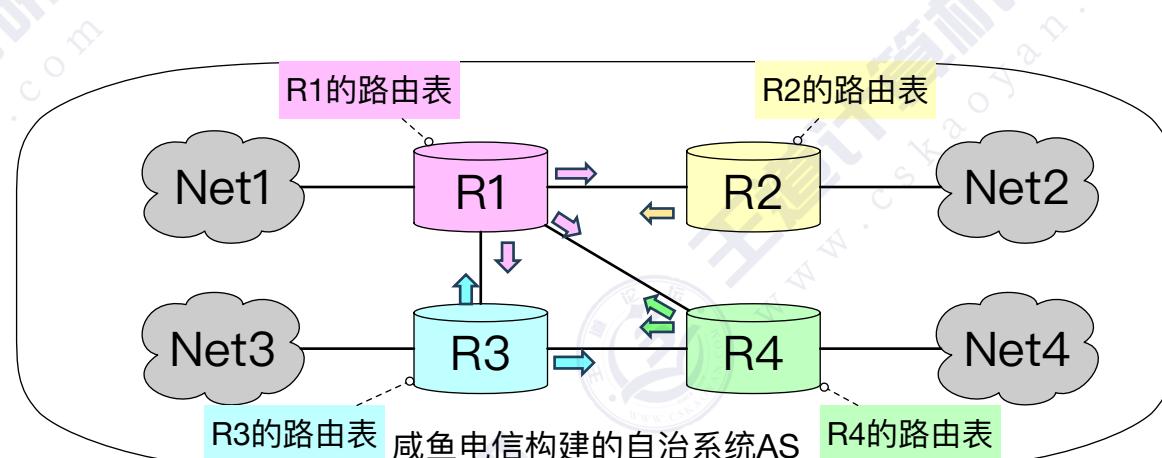
3) When, 何时交换信息：通常按固定的时间间隔（通常为 30 秒）交换路由信息。

注：为了加快RIP收敛速度，也可以引入触发更新（Triggered Update）机制——当路由器发现网络拓扑发生变化时，立即向相邻路由器通告拓扑变化后的路由信息，而不必等待30秒。

R1 路由表

目的网络	距离	下一跳路由器
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	2	R4

路由表是通过
RIP逐步生成的



根据题目
信息确定

本节内容

RIP

的工作过程示例
(从路由器启动到收敛)

RIP 的工作过程（基于 距离向量算法）

周期=30秒

路由器刚开始工作时，知道自己到直接相连的几个网络的距离为 1。每个路由器仅和相邻路由器周期性地交换并更新路由信息。经过若干次交换和更新后，所有的路由器最终都会知道到达本自治系统内任何网络的最短距离和下一跳路由器的地址，称为收敛。

对每个相邻路由器发送来的 RIP 报文，执行如下步骤：

- 1) 对地址为 X 的相邻路由器发来的 RIP 报文，先修改该报文中的所有项目：
把“下一跳”字段中的地址都改为 X，并把所有“距离”字段的值加 1。

本质是距离向量路由算法

- 2) 对修改后的 RIP 报文中的每个项目，执行如下步骤：

IF (若原来的路由表中没有目的网络 N)

 则把该项目添加到路由表中（表明这是新的目的网络）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址是 X)

 用收到的项目替换原路由表中的项目（因为要以更新的消息为准）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址不是 X)

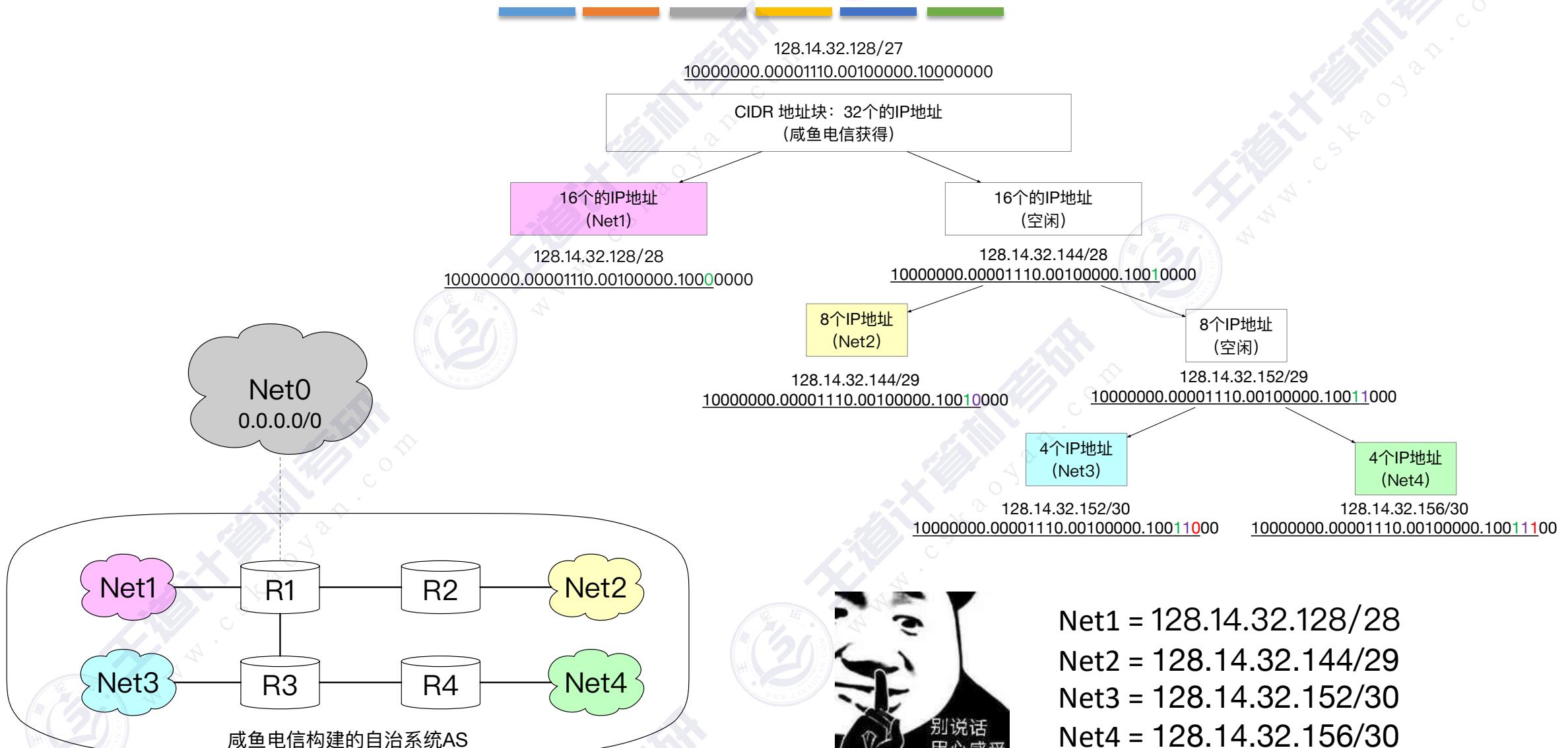
 若收到的项目中的距离 d 小于路由表中的距离，则进行更新

ELSE 什么也不做。

- ①有没有发现新网络？
- ②旧网络的距离是否要更新？
 - 找到一条新路径，距离更近
 - 继续走老路，但老路距离有变化

- 3) 若 180 秒 (RIP 默认超时时间) 还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达的路由器，即把距离设置为 16 (表示不可达)
- 4) 返回

例：咸鱼电信构建的自治系统



初始：各路由器同时开机，构建初始路由表

R1 的初始路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付

R2 的初始路由表

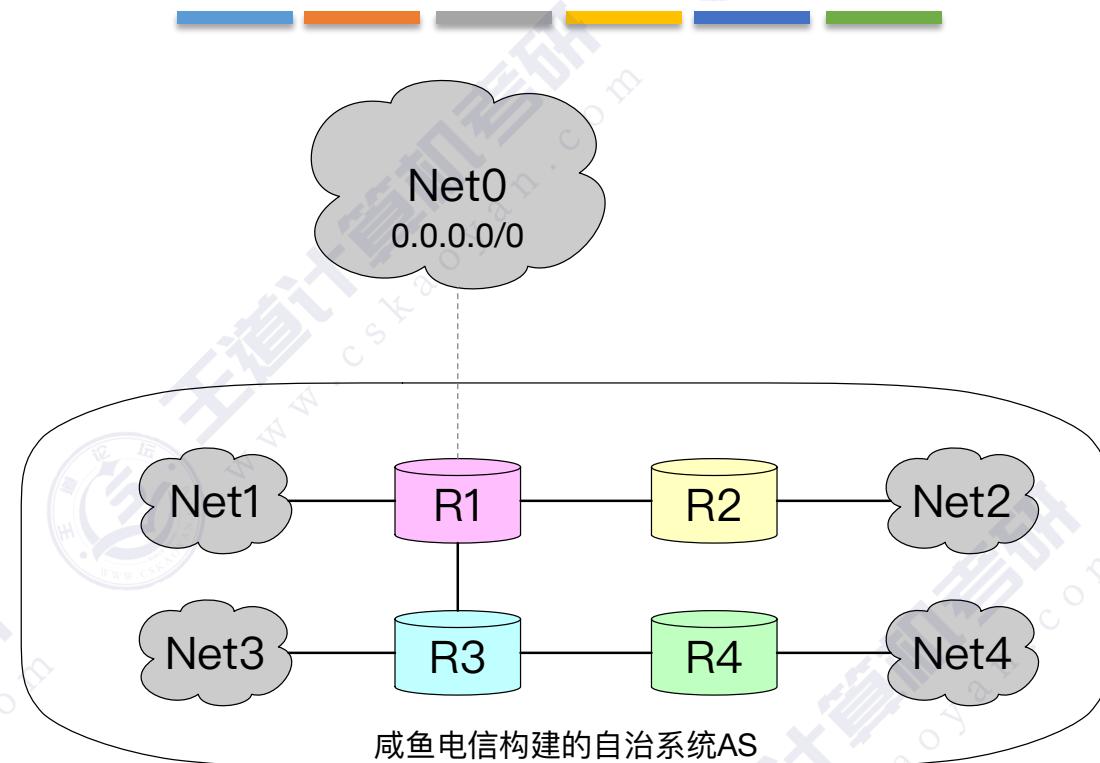
目的网络	距离	下一跳路由器
Net2	1	直接交付

R3 的初始路由表

目的网络	距离	下一跳路由器
Net3	1	直接交付

R4 的初始路由表

目的网络	距离	下一跳路由器
Net4	1	直接交付



路由器刚开始工作时，知道自己到直接相连的几个网络的距离为1

0时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 的初始路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付

R2 的初始路由表

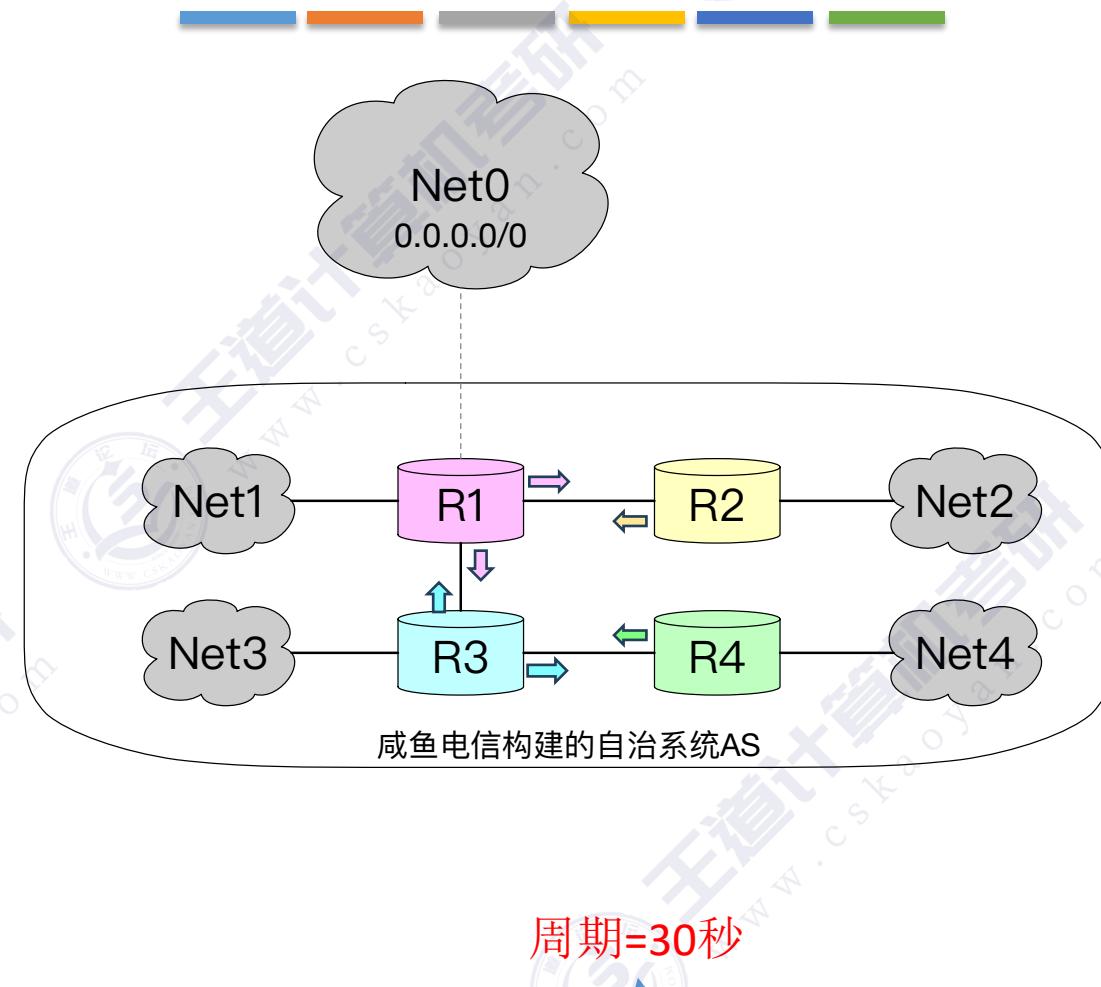
目的网络	距离	下一跳路由器
Net2	1	直接交付

R3 的初始路由表

目的网络	距离	下一跳路由器
Net3	1	直接交付

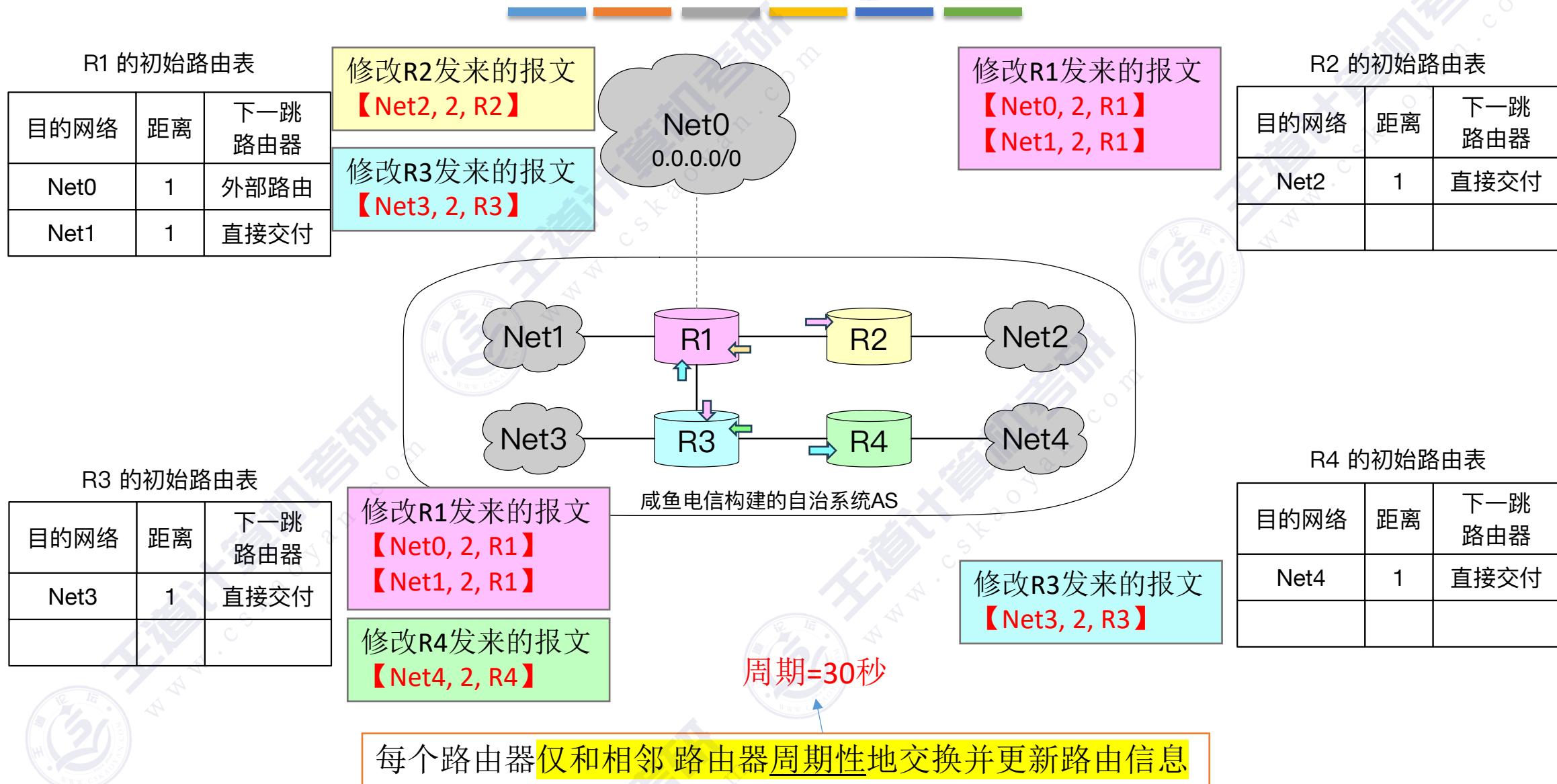
R4 的初始路由表

目的网络	距离	下一跳路由器
Net4	1	直接交付



每个路由器仅和相邻 路由器周期性地交换并更新路由信息

0时刻：收到邻居的RIP报文后，更新自己的路由信息



0时刻：收到邻居的RIP报文后，更新自己的路由信息

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3

修改R2发来的报文
【Net2, 2, R2】

修改R3发来的报文
【Net3, 2, R3】

修改R1发来的报文
【Net0, 2, R1】
【Net1, 2, R1】

R2 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net3	1	直接交付
Net4	2	R4

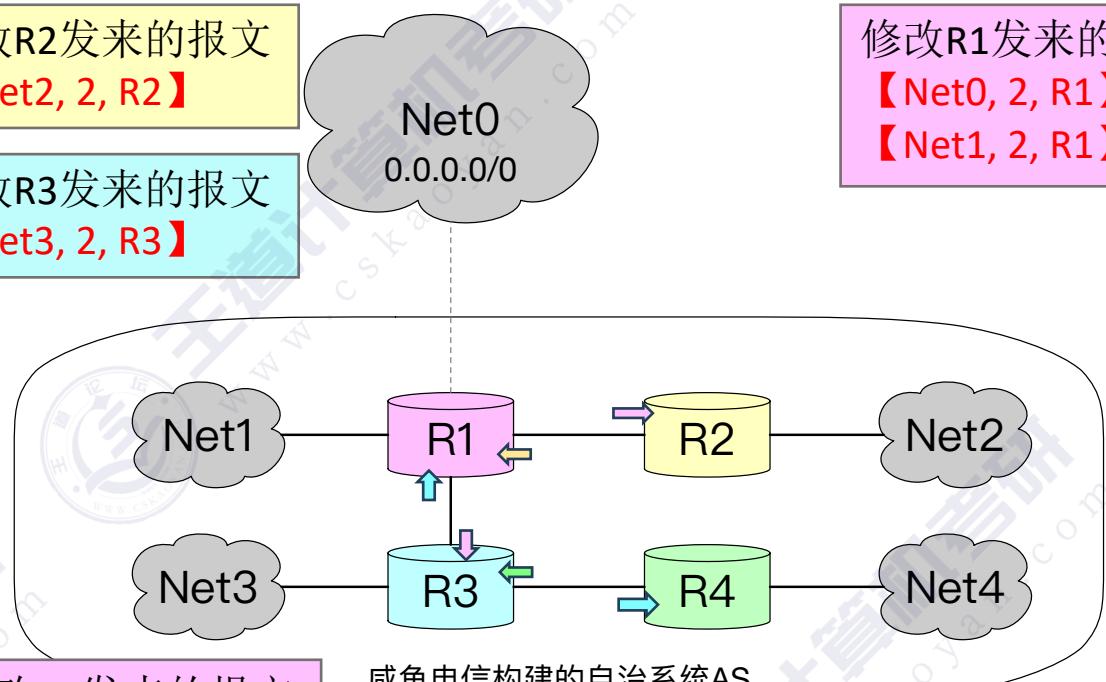
修改R1发来的报文
【Net0, 2, R1】
【Net1, 2, R1】

修改R4发来的报文
【Net4, 2, R4】

修改R3发来的报文
【Net3, 2, R3】

R4 路由表

目的网络	距离	下一跳路由器
Net3	2	R3
Net4	1	直接交付



检查：
①有没有发现新网络？
②旧网络的距离是否更新？

30时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3

R2 路由表

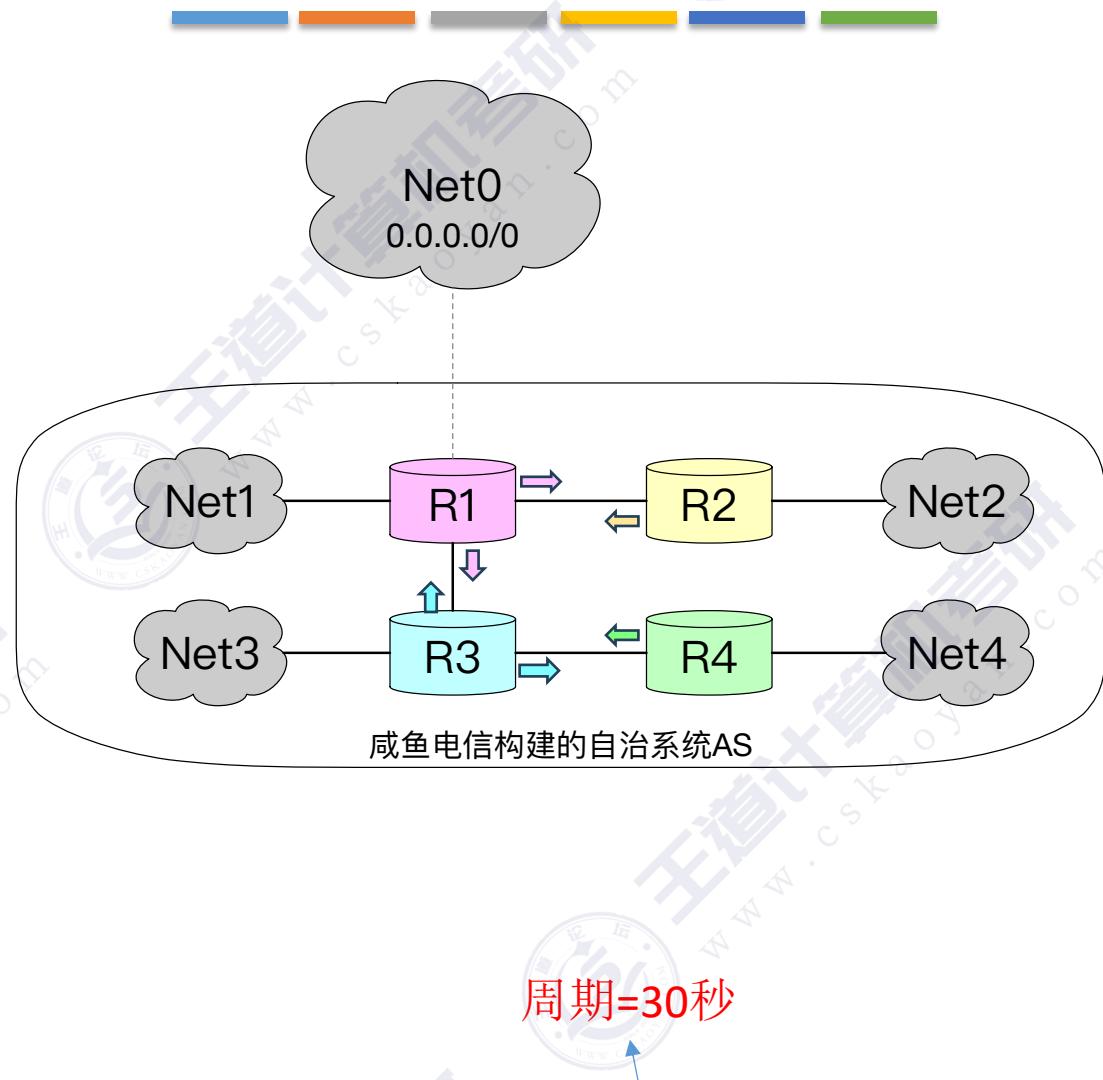
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

目的网络	距离	下一跳路由器
Net3	2	R3
Net4	1	直接交付



每个路由器仅和相邻路由器周期性地交换并更新路由信息

30时刻：收到邻居的RIP报文后，更新自己的路由信息

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3

修改R2发来的报文

【Net0, 3, R2】
【Net1, 3, R2】
【Net2, 2, R2】

修改R3发来的报文

【Net0, 3, R3】
【Net1, 3, R3】
【Net3, 2, R3】
【Net4, 3, R3】

修改R1发来的报文

【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 3, R1】
【Net3, 3, R1】

R2 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net3	1	直接交付
Net4	2	R4

修改R1发来的报文

【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 3, R1】
【Net3, 3, R1】

修改R4发来的报文

【Net3, 3, R4】
【Net4, 2, R4】

R4 路由表

目的网络	距离	下一跳路由器
Net3	2	R3
Net4	1	直接交付

检查： ①有没有发现新网络？
②旧网络的距离是否更新？

30时刻：收到邻居的RIP报文后，更新自己的路由信息

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	3	R3

修改R2发来的报文

【Net0, 3, R2】
【Net1, 3, R2】
【Net2, 2, R2】

修改R3发来的报文

【Net0, 3, R3】
【Net1, 3, R3】
【Net3, 2, R3】
【Net4, 3, R3】

修改R1发来的报文

【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 3, R1】
【Net3, 3, R1】

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

修改R1发来的报文

【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 3, R1】
【Net3, 3, R1】

修改R4发来的报文

【Net3, 3, R4】
【Net4, 2, R4】

R2 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1

检查：①有没有发现新网络？

②旧网络的距离是否更新？

R4 路由表

目的网络	距离	下一跳路由器
Net0	3	R3
Net1	3	R3
Net3	2	R3
Net4	1	直接交付



60时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	3	R3

R2 路由表

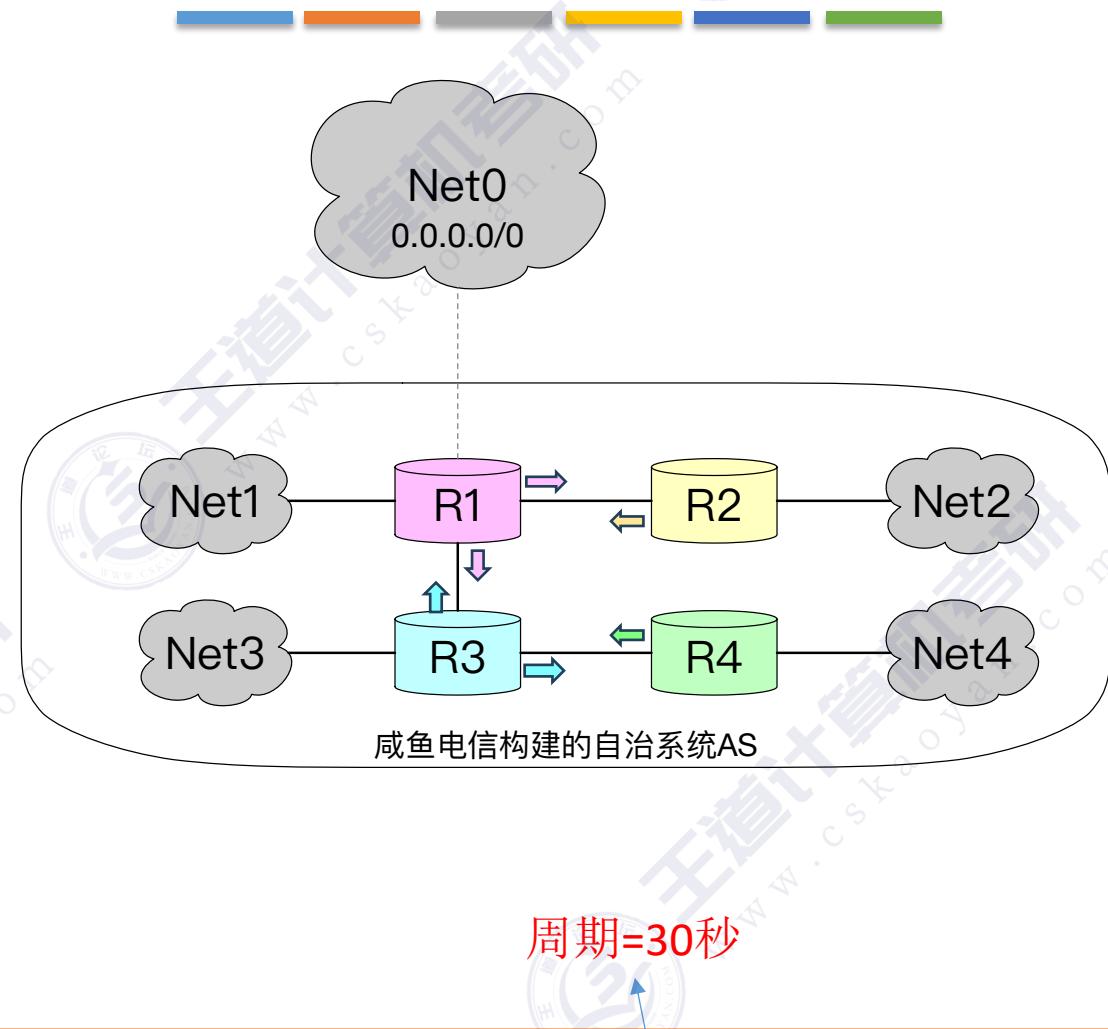
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1

R3 路由表

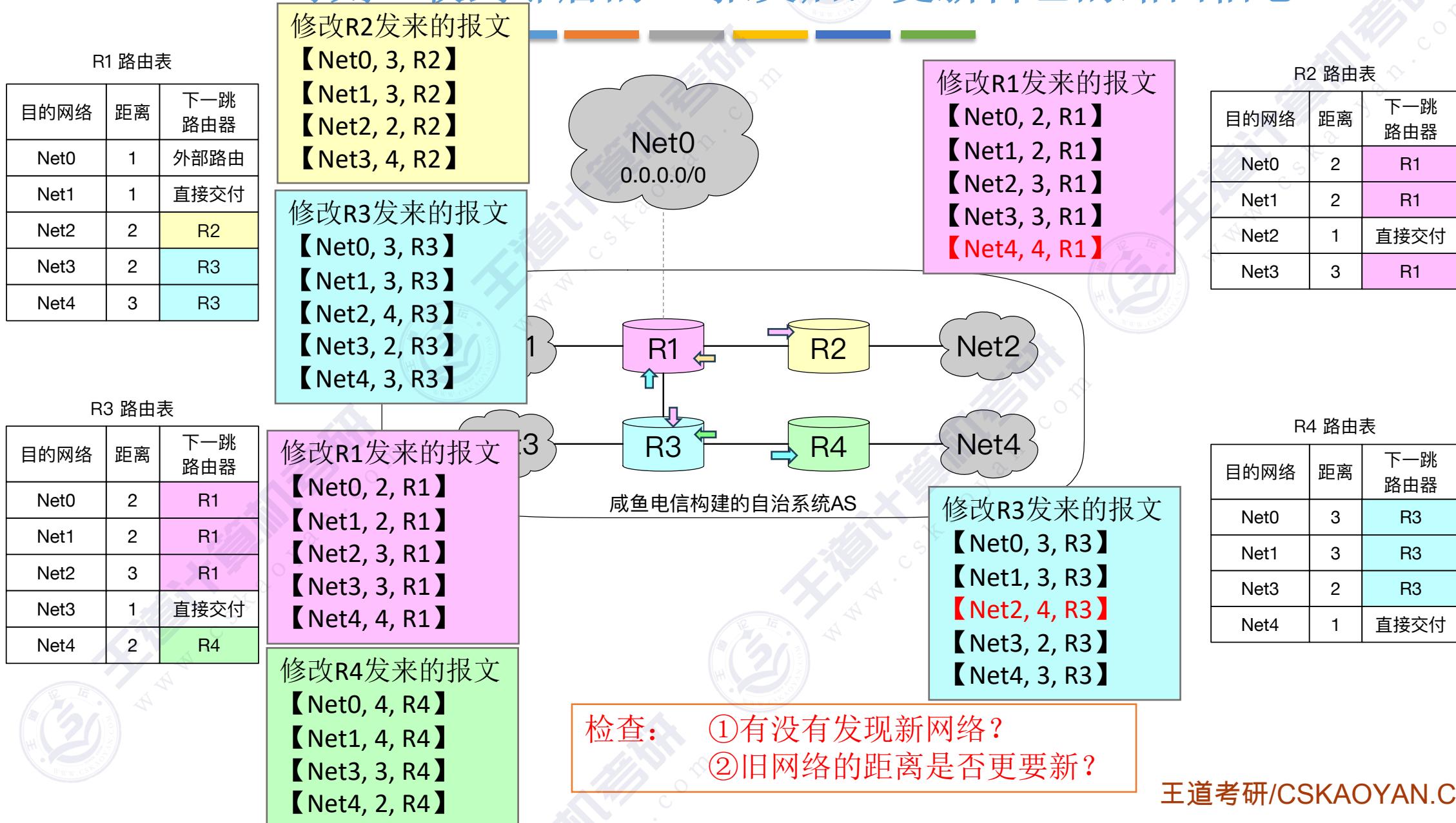
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

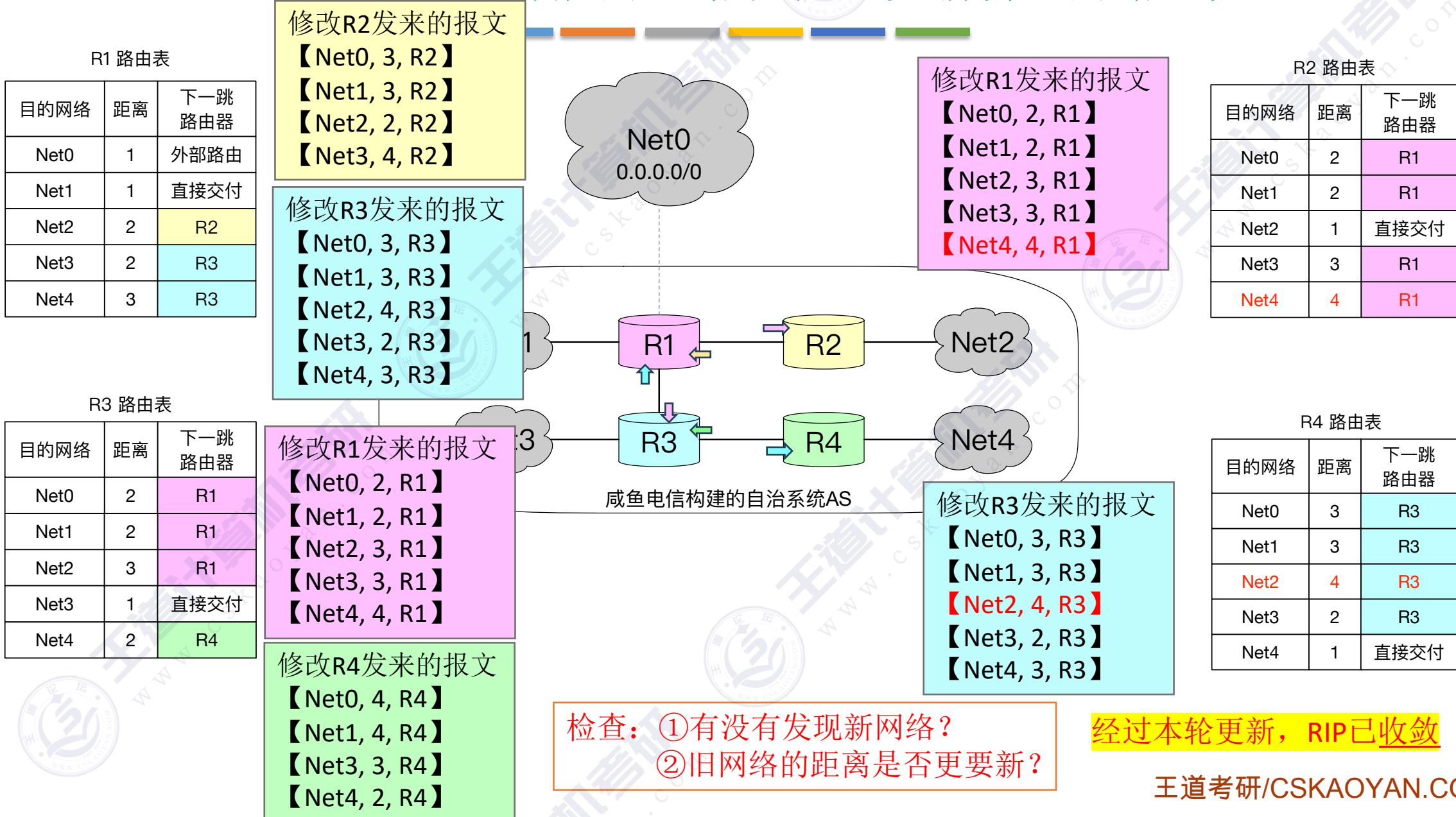
目的网络	距离	下一跳路由器
Net0	3	R3
Net1	3	R3
Net3	2	R3
Net4	1	直接交付



60时刻：收到邻居的RIP报文后，更新自己的路由信息



60时刻：收到邻居的RIP报文后，更新自己的路由信息



90时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	3	R3

R2 路由表

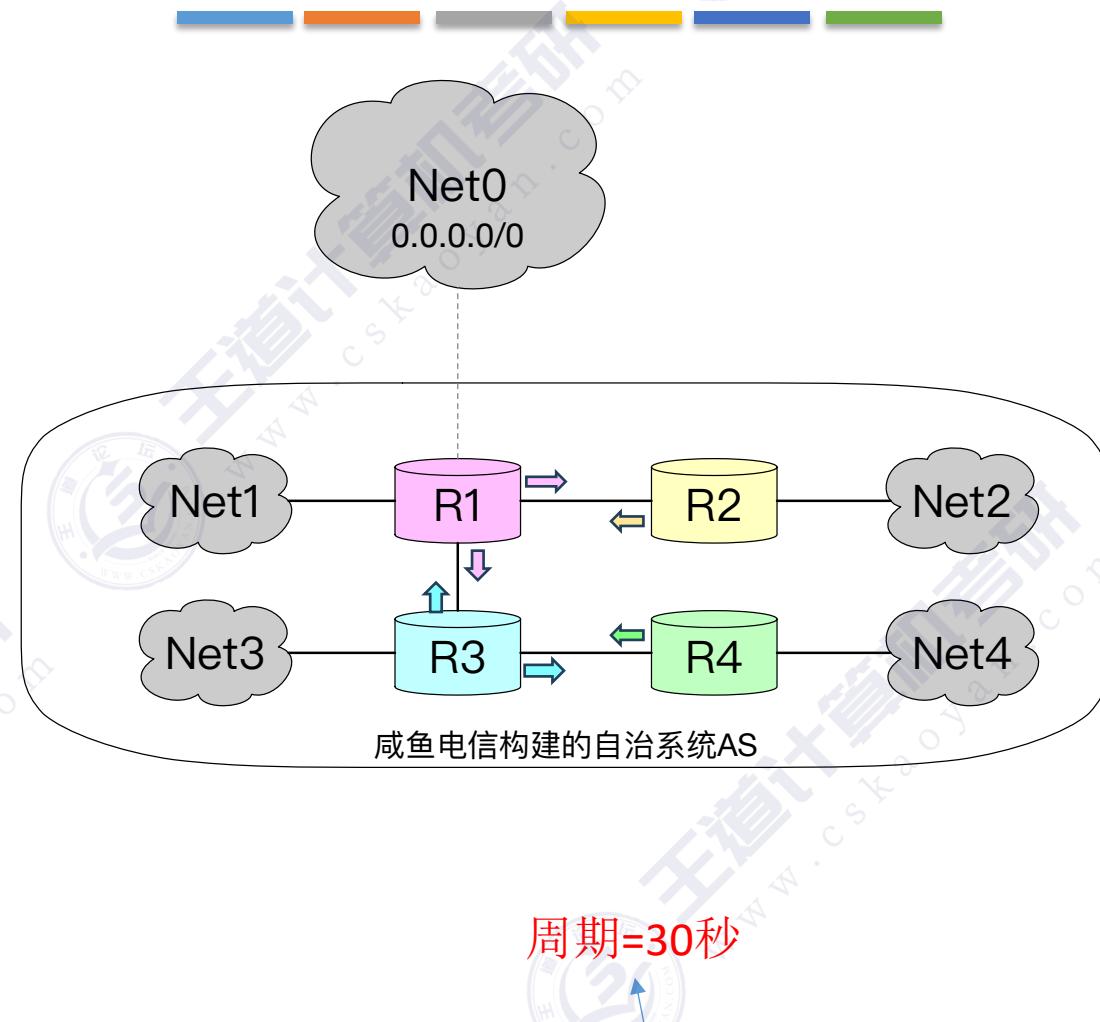
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1
Net4	4	R1

R3 路由表

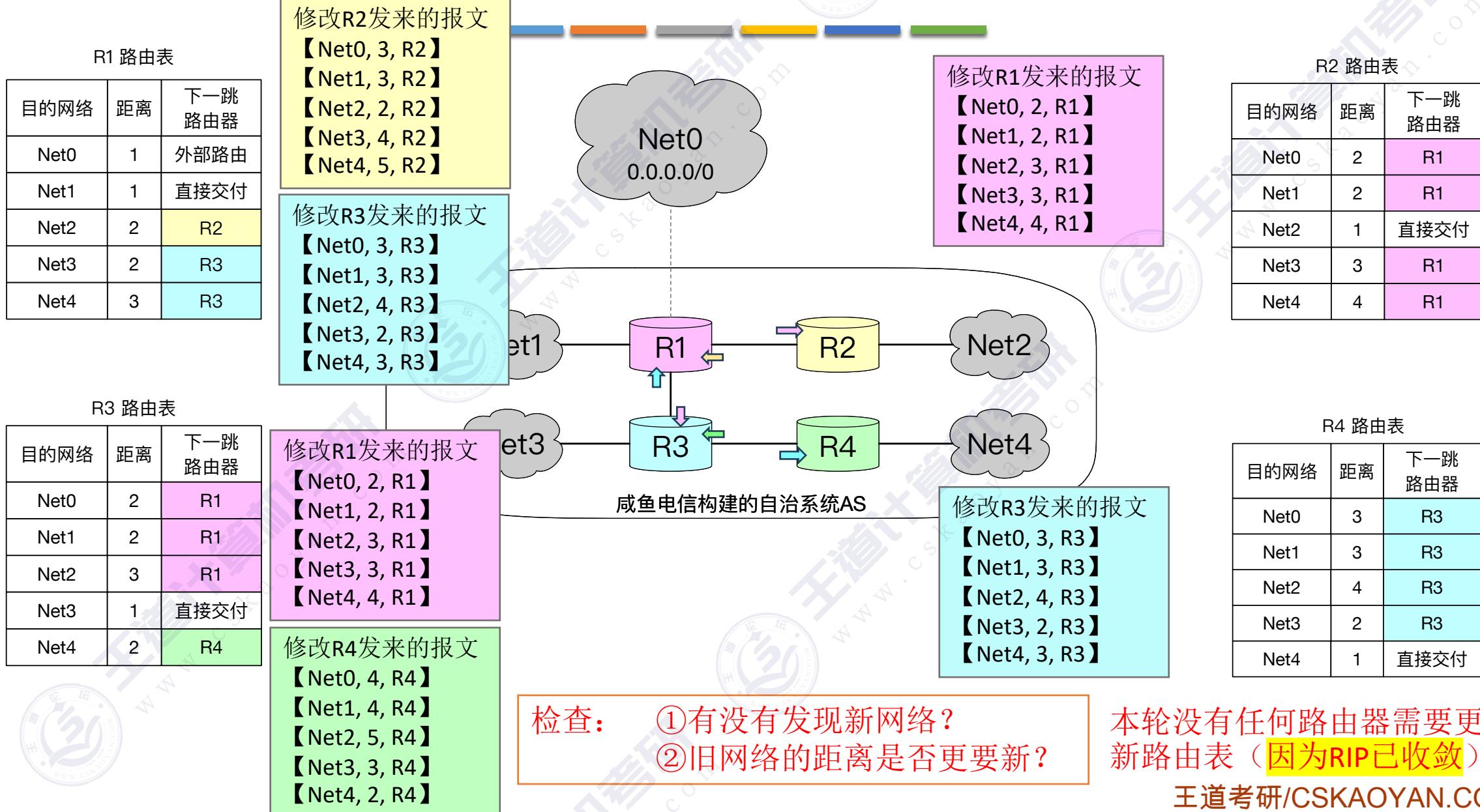
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

目的网络	距离	下一跳路由器
Net0	3	R3
Net1	3	R3
Net2	4	R3
Net3	2	R3
Net4	1	直接交付



90时刻：收到邻居的RIP报文后，更新自己的路由信息



本轮没有任何路由器需要更新路由表（因为RIP已收敛）

本节内容

RIP

的工作过程示例
(动态适应网络拓扑变化)

RIP 的工作过程（基于 距离向量算法）

周期=30秒

路由器刚开始工作时，知道自己到直接相连的几个网络的距离为 1。每个路由器仅和相邻路由器周期性地交换并更新路由信息。经过若干次交换和更新后，所有的路由器最终都会知道到达本自治系统内任何网络的最短距离和下一跳路由器的地址，称为收敛。

对每个相邻路由器发送来的 RIP 报文，执行如下步骤：

- 1) 对地址为 X 的相邻路由器发来的 RIP 报文，先修改该报文中的所有项目：
把“下一跳”字段中的地址都改为 X，并把所有“距离”字段的值加 1。

本质是距离向量路由算法

- 2) 对修改后的 RIP 报文中的每个项目，执行如下步骤：

IF (若原来的路由表中没有目的网络 N)

 则把该项目添加到路由表中（表明这是新的目的网络）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址是 X)

 用收到的项目替换原路由表中的项目（因为要以更新的消息为准）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址不是 X)

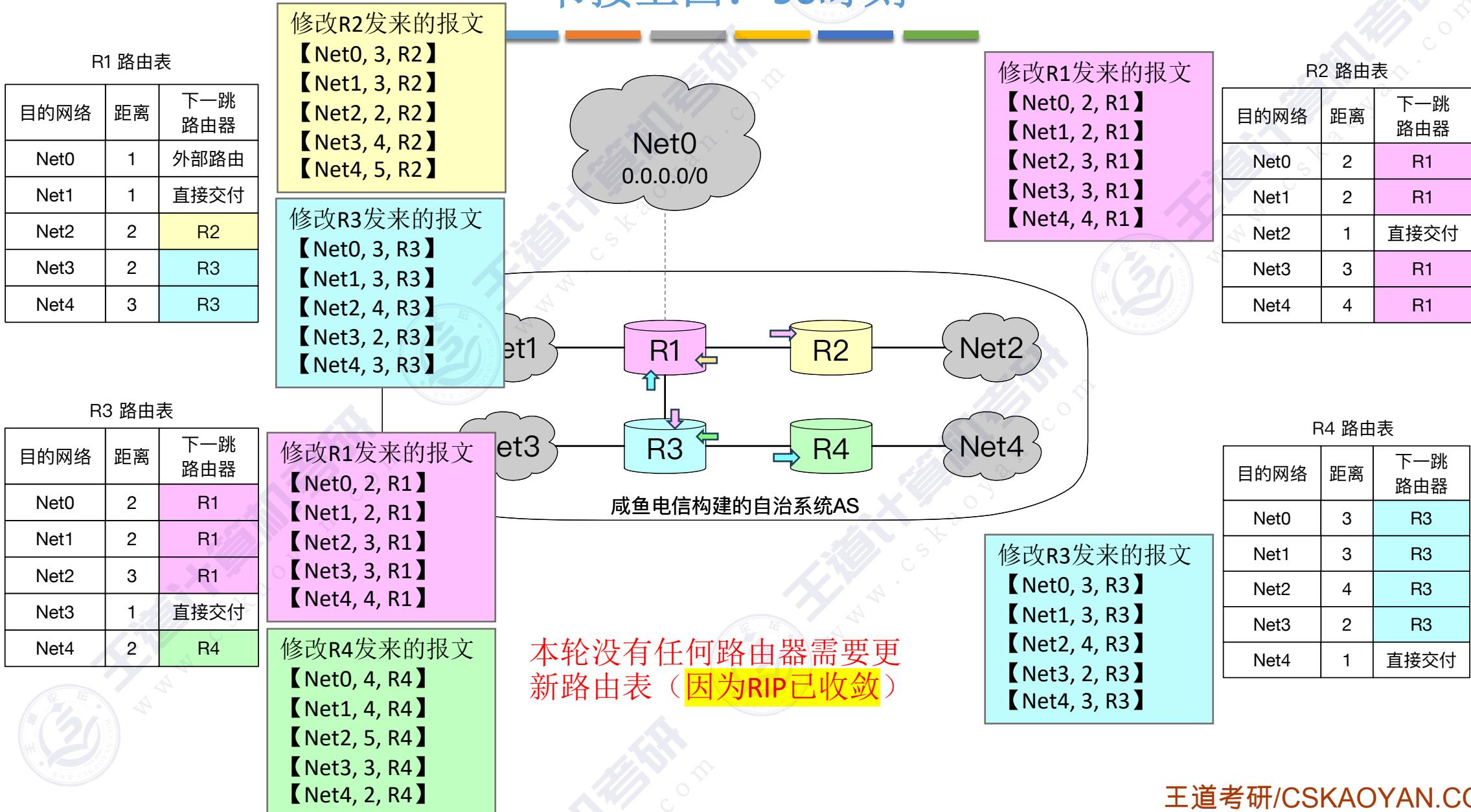
 若收到的项目中的距离 d 小于路由表中的距离，则进行更新

ELSE 什么也不做。

- ①有没有发现新网络？
- ②旧网络的距离是否要更新？
 - 找到一条新路径，距离更近
 - 继续走老路，但老路距离有变化

- 3) 若 180 秒 (RIP 默认超时时间) 还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达的路由器，即把距离设置为 16 (表示不可达)
- 4) 返回

书接上回：90时刻



100时刻：假设R1、R4之间新增一条链路

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	3	R3

R2 路由表

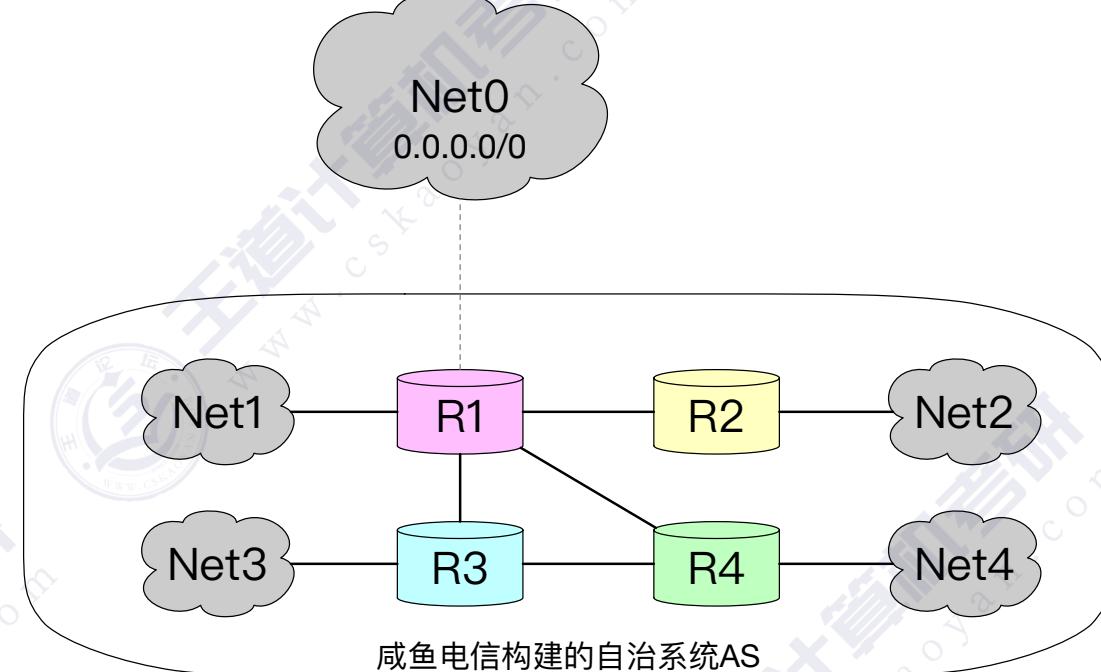
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1
Net4	4	R1

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

目的网络	距离	下一跳路由器
Net0	3	R3
Net1	3	R3
Net2	4	R3
Net3	2	R3
Net4	1	直接交付



每个路由器仅和相邻路由器周期性地交换并更新路由信息

注：为了加快RIP收敛速度，也可以引入**触发更新（Triggered Update）机制**——当路由器发现网络拓扑发生变化时，立即向相邻路由器通告拓扑变化后的路由信息，而不必等待30秒。
本例中，我们假设不引入触发更新机制。

120时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	3	R3

R2 路由表

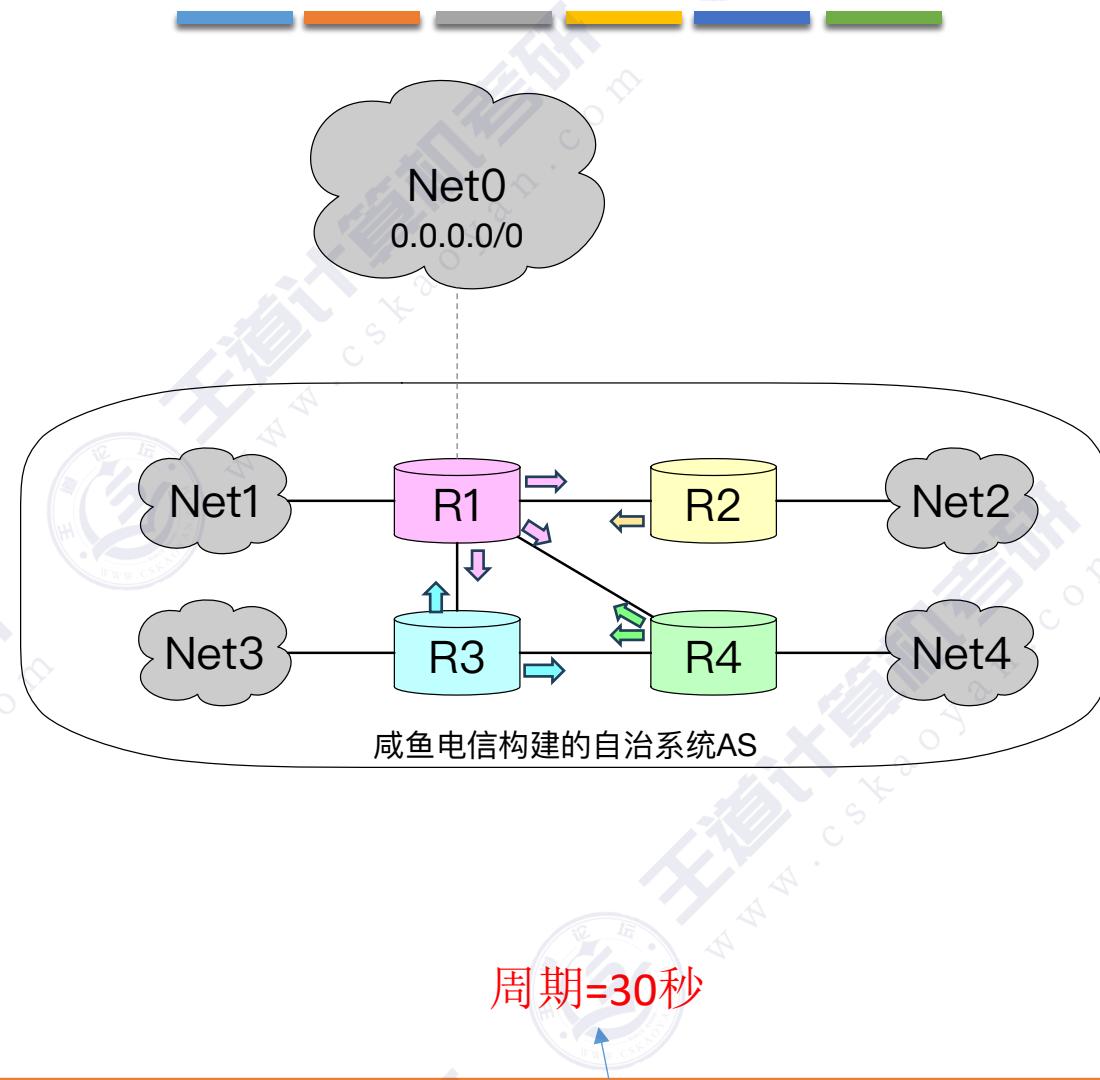
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1
Net4	4	R1

R3 路由表

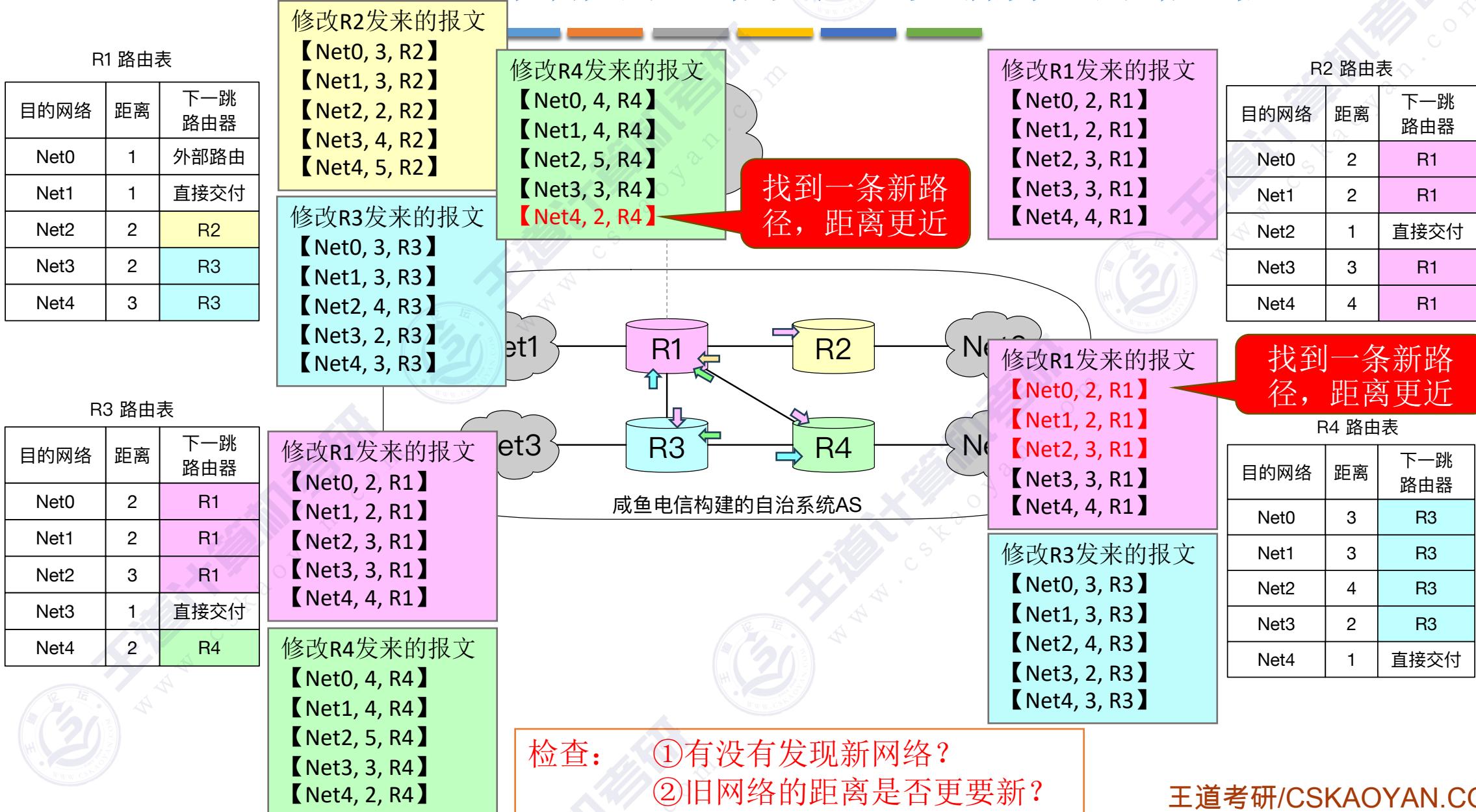
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

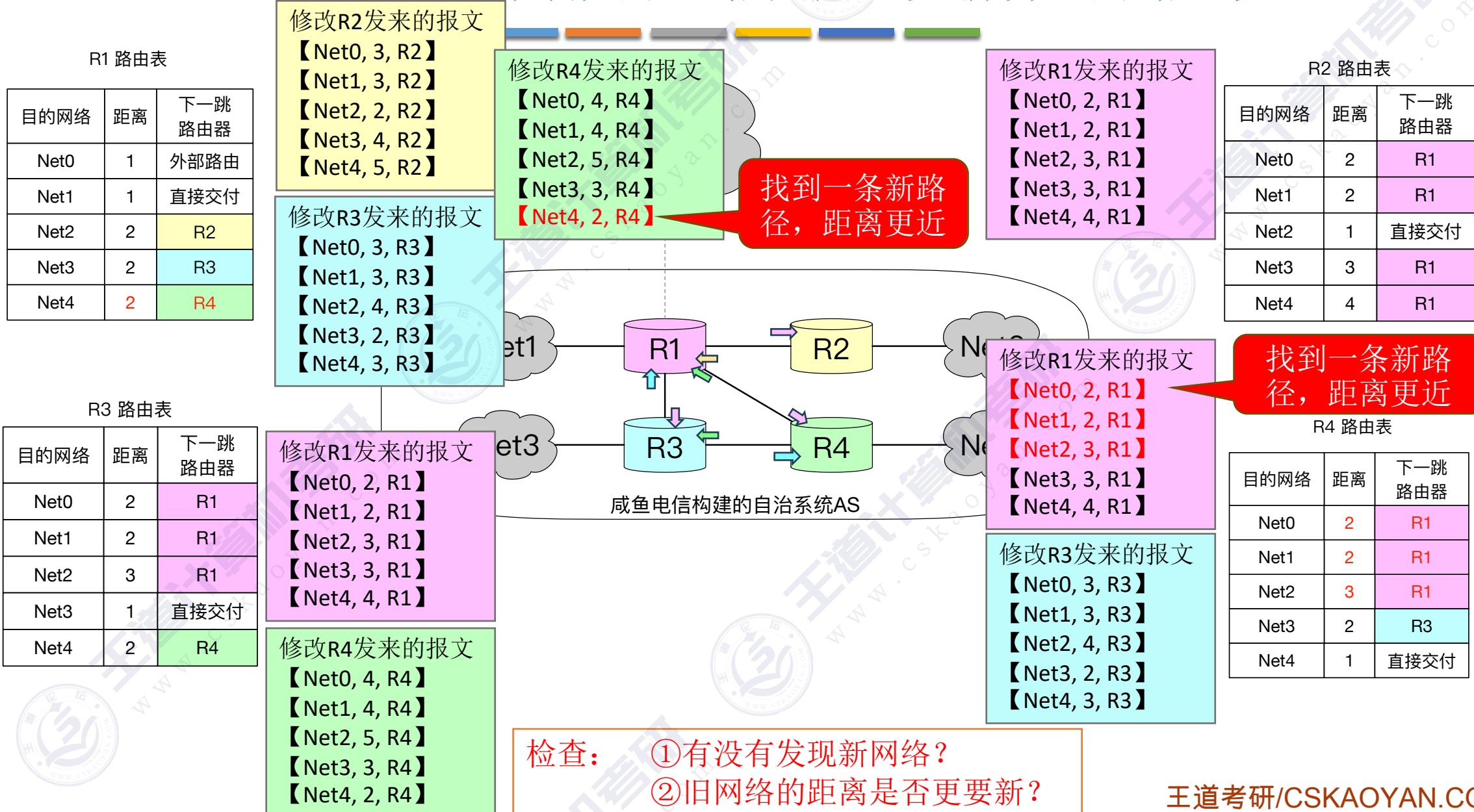
目的网络	距离	下一跳路由器
Net0	3	R3
Net1	3	R3
Net2	4	R3
Net3	2	R3
Net4	1	直接交付



120时刻：收到邻居的RIP报文后，更新自己的路由信息



120时刻：收到邻居的RIP报文后，更新自己的路由信息



150时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	2	R4

R2 路由表

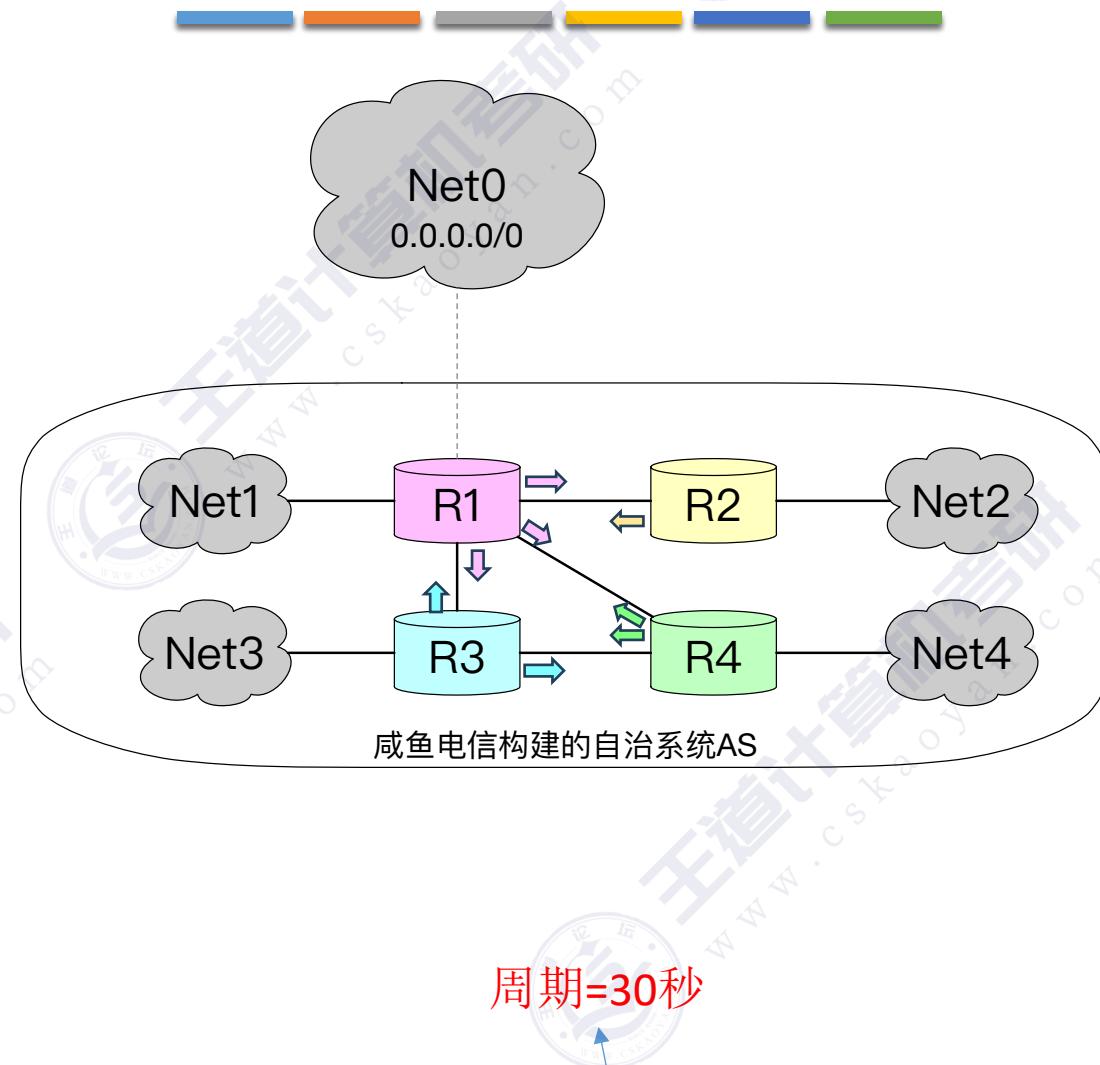
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1
Net4	4	R1

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

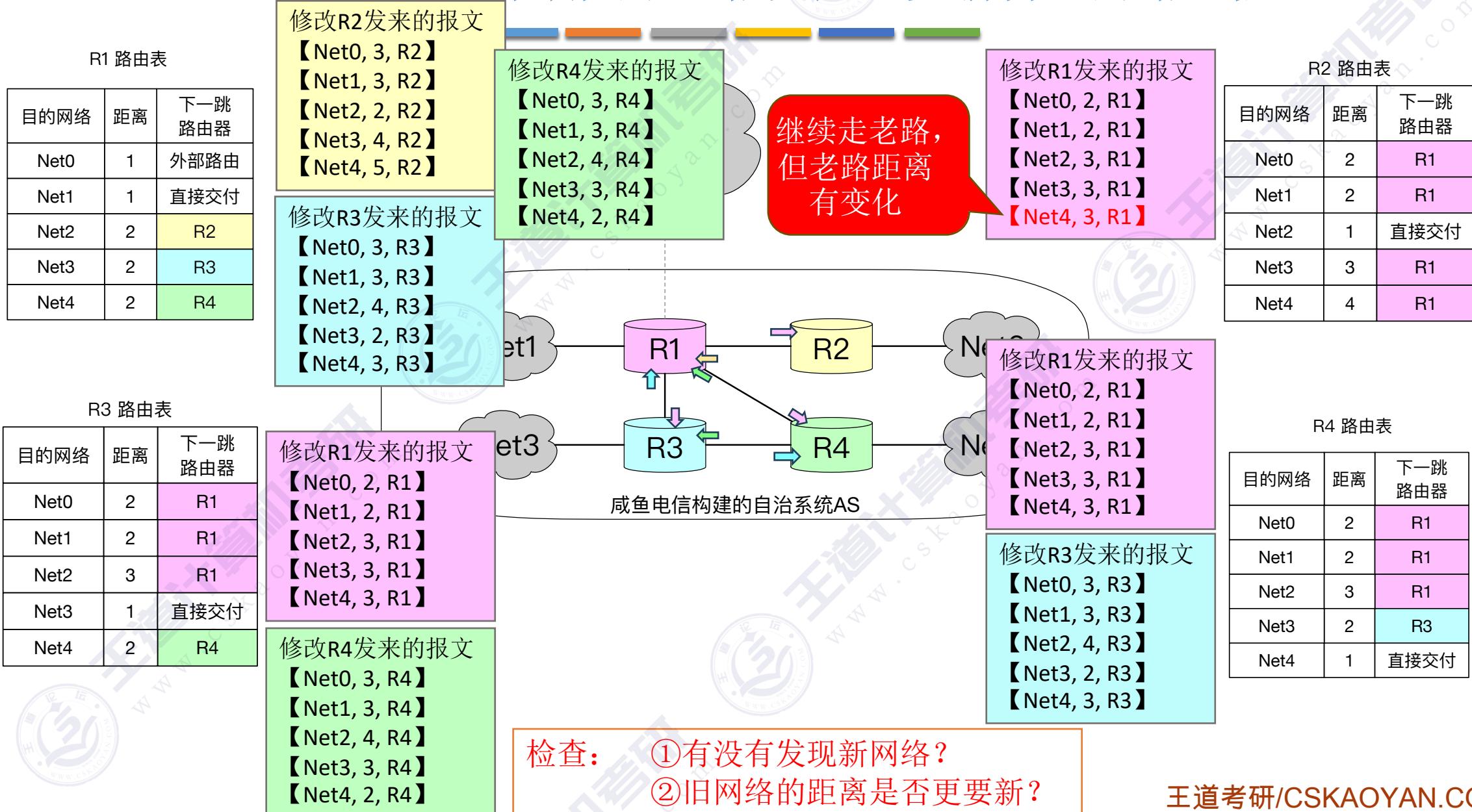
R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	2	R3
Net4	1	直接交付

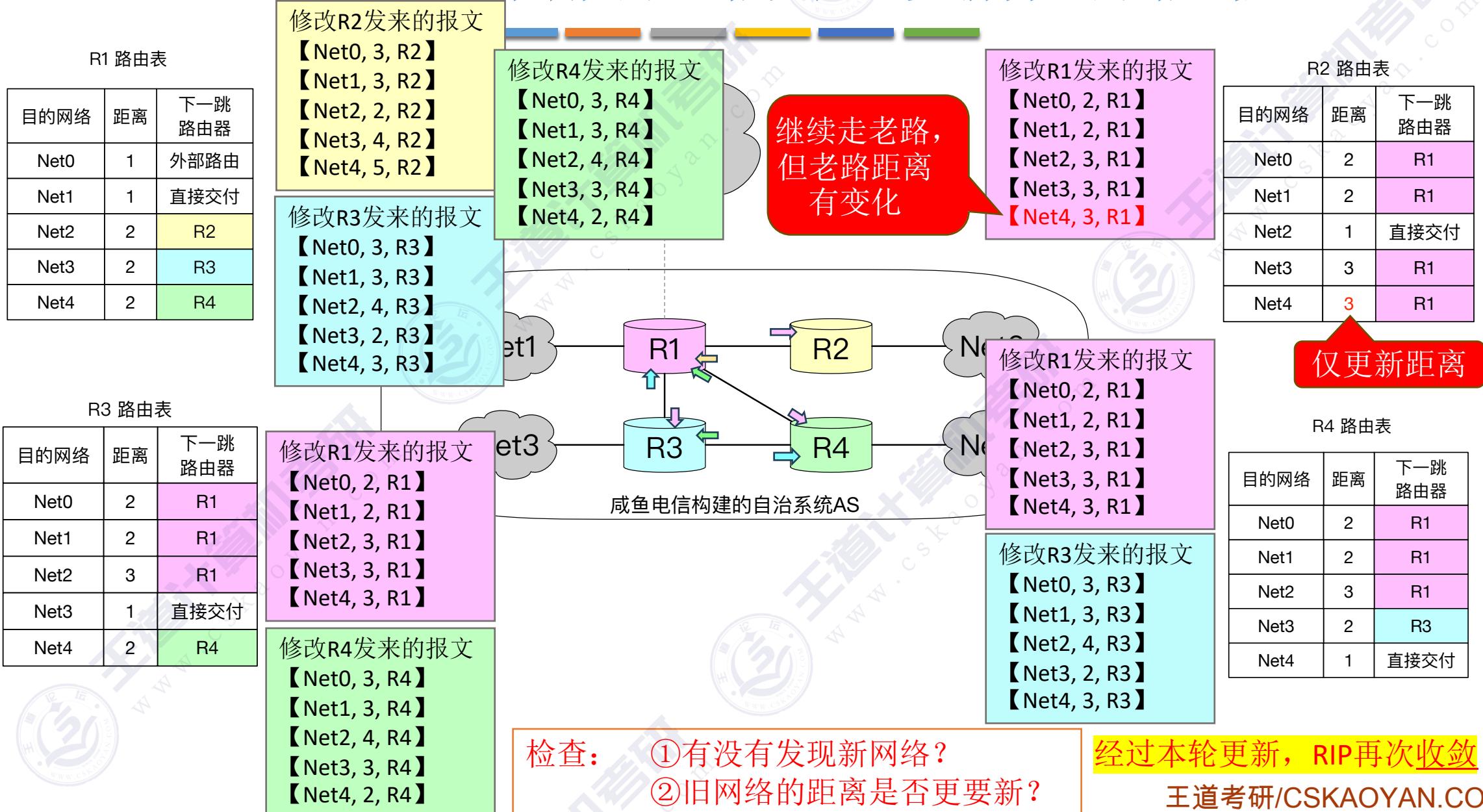


每个路由器仅和相邻路由器周期性地交换并更新路由信息

150时刻：收到邻居的RIP报文后，更新自己的路由信息



150时刻：收到邻居的RIP报文后，更新自己的路由信息



本节内容

RIP

的工作过程示例
(反映RIP的缺点)

RIP 的优缺点



RIP 的优点：

- 1) 实现简单、开销小、收敛过程较快。
- 2) 若一个路由器发现了更短的路由，则这种更新信息就传播得很快，在较短时间内便可被传至所有路由器，俗称“好消息传播得快”。

RIP 的缺点：

- 1) RIP 限制了网络的规模，它能使用的最大距离为 15（16 表示不可达）。
- 2) 路由器之间交换的是路由器中的完整路由表，因此网络规模越大，开销也越大。
- 3) 当网络出现故障时，路由器之间需反复多次交换信息才能完成收敛，要经过较长时间才能将故障消息传送到所有路由器（慢收敛现象），俗称坏消息传播得慢。

注：距离向量路由可能会出现环路的情况，RIP 规定路径上的最高跳数=16 是一种防止环路的简单粗暴的方式。

RIP 的工作过程（基于 距离向量算法）

周期=30秒

路由器刚开始工作时，知道自己到直接相连的几个网络的距离为 1。每个路由器仅和相邻路由器周期性地交换并更新路由信息。经过若干次交换和更新后，所有的路由器最终都会知道到达本自治系统内任何网络的最短距离和下一跳路由器的地址，称为收敛。

对每个相邻路由器发送来的 RIP 报文，执行如下步骤：

- 1) 对地址为 X 的相邻路由器发来的 RIP 报文，先修改该报文中的所有项目：
把“下一跳”字段中的地址都改为 X，并把所有“距离”字段的值加 1。

本质是距离向量路由算法

- 2) 对修改后的 RIP 报文中的每个项目，执行如下步骤：

IF (若原来的路由表中没有目的网络 N)

 则把该项目添加到路由表中（表明这是新的目的网络）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址是 X)

 用收到的项目替换原路由表中的项目（因为要以更新的消息为准）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址不是 X)

 若收到的项目中的距离 d 小于路由表中的距离，则进行更新

ELSE 什么也不做。

- ①有没有发现新网络？
- ②旧网络的距离是否要更新？
 - 找到一条新路径，距离更近
 - 继续走老路，但老路距离有变化

- 3) 若 180 秒 (RIP 默认超时时间) 还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达的路由器，即把距离设置为 16 (表示不可达)
- 4) 返回

书接上回：150时刻

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	2	R4

R2 路由表

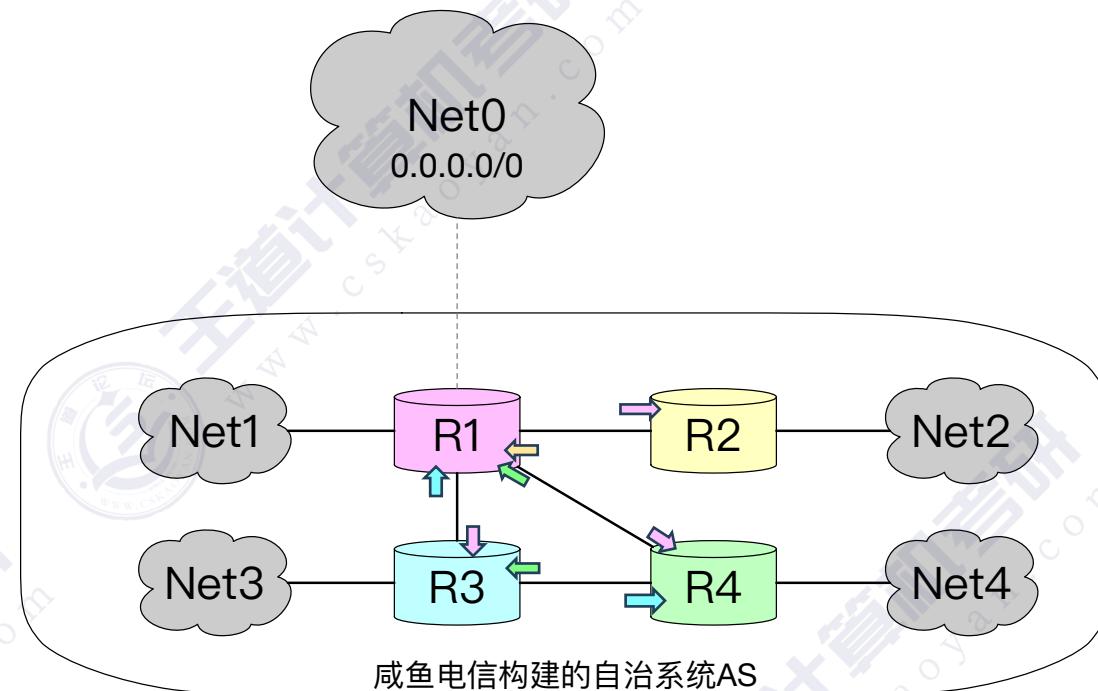
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1
Net4	3	R1

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	2	R3
Net4	1	直接交付



检查：①有没有发现新网络？
②旧网络的距离是否更新了？

经过本轮更新，RIP再次收敛
王道考研/CSKAOYAN.COM

160时刻：路由器R2被老鼠咬坏

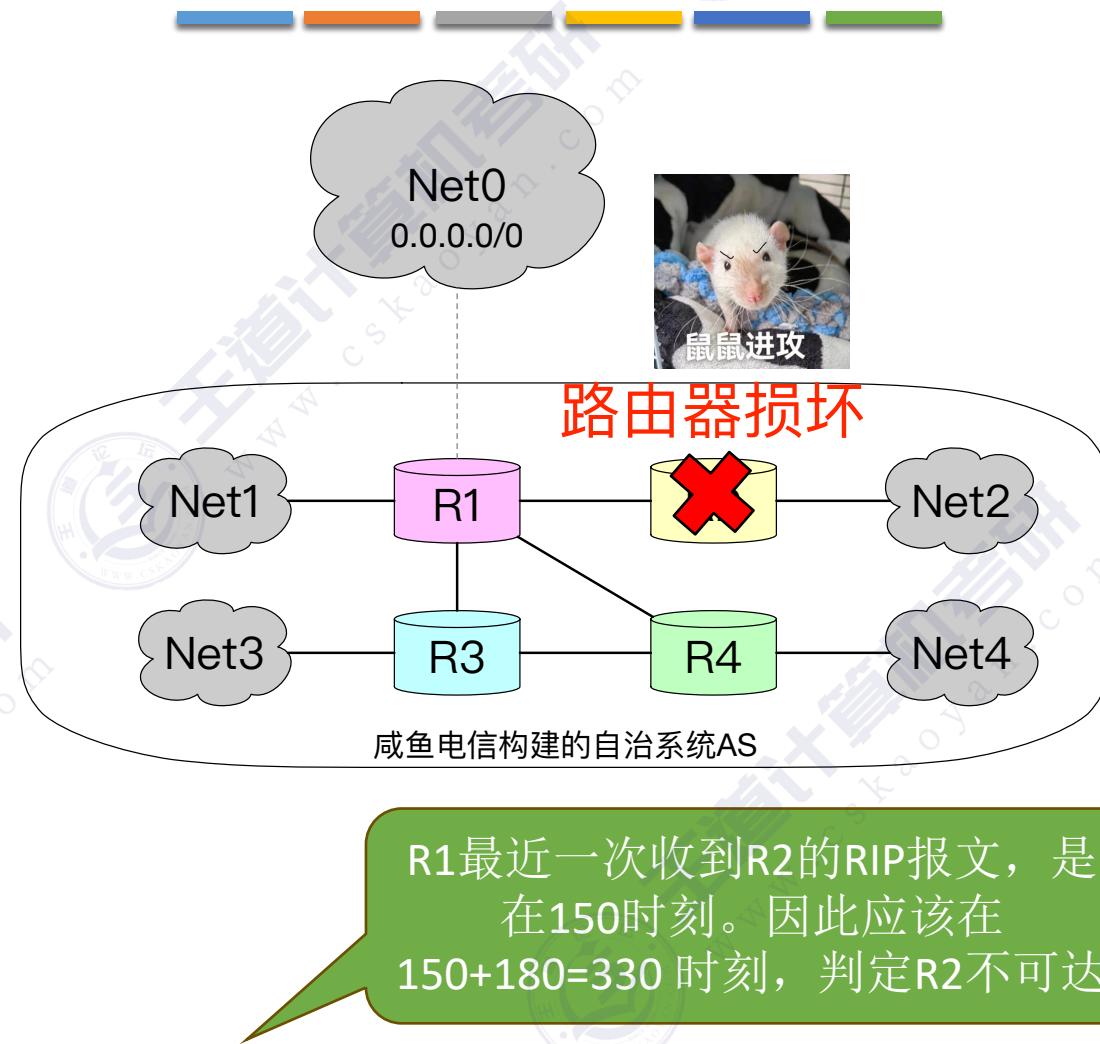
路由器损坏

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	2	R4

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4



R2 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	1	直接交付
Net3	3	R1
Net4	3	R1

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	2	R3
Net4	1	直接交付

若 180 秒 (RIP 默认超时时间) 还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达的路由器，即把距离设置为 16 (表示不可达)

150+180=330 时刻：路由器R1判定R2不可达

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	2	R2
Net3	2	R3
Net4	2	R4

此刻，R1认为
R1→R2 的距离=16

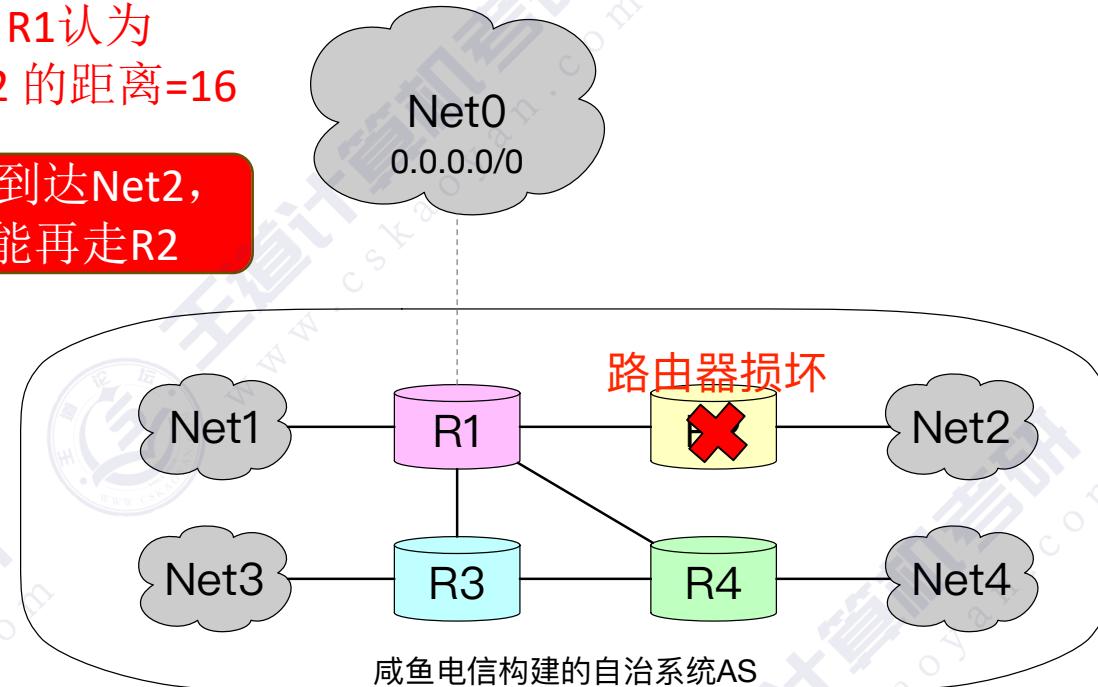
想要到达Net2，
不能再走R2

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	2	R3
Net4	1	直接交付



若 180 秒（RIP 默认超时时间）还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达的路由器，即把距离设置为 16（表示不可达）

330 时刻：路由器R1判定R2不可达，并更新自己的路由表

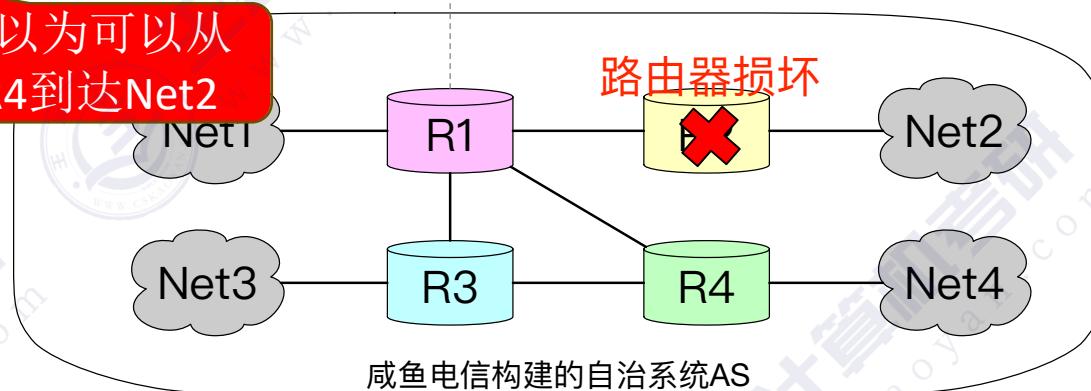
R1 路由表		
目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	4	R3/R4
Net3	2	R3
Net4	2	R4

修改R3发来的报文
 【Net0, 3, R3】
 【Net1, 3, R3】
【Net2, 4, R3】
 【Net3, 2, R3】
 【Net4, 3, R3】

修改R4发来的报文
 【Net0, 3, R4】
 【Net1, 3, R4】
【Net2, 4, R4】
 【Net3, 2, R4】
 【Net3, 3, R4】
 【Net4, 2, R4】

原因：在过去一段时间，R1一直收到R3、R4发来的RIP报文。R1误以为从R3、R4可以到达Net2

R1误以为可以从R3/R4到达Net2



目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	2	R3
Net4	1	直接交付

注1：RIP 协议支持**多条等价路径共存**，将它们同时加入路由表，实现**负载均衡转发**（如：轮询转发）

注2：此时，转发至 Net2 的路径**形成环路**，IP数据报会“兜圈子”，直到TTL=0才会被丢弃

330 时刻：各路由器向邻居发送RIP报文（包含完整路由表）

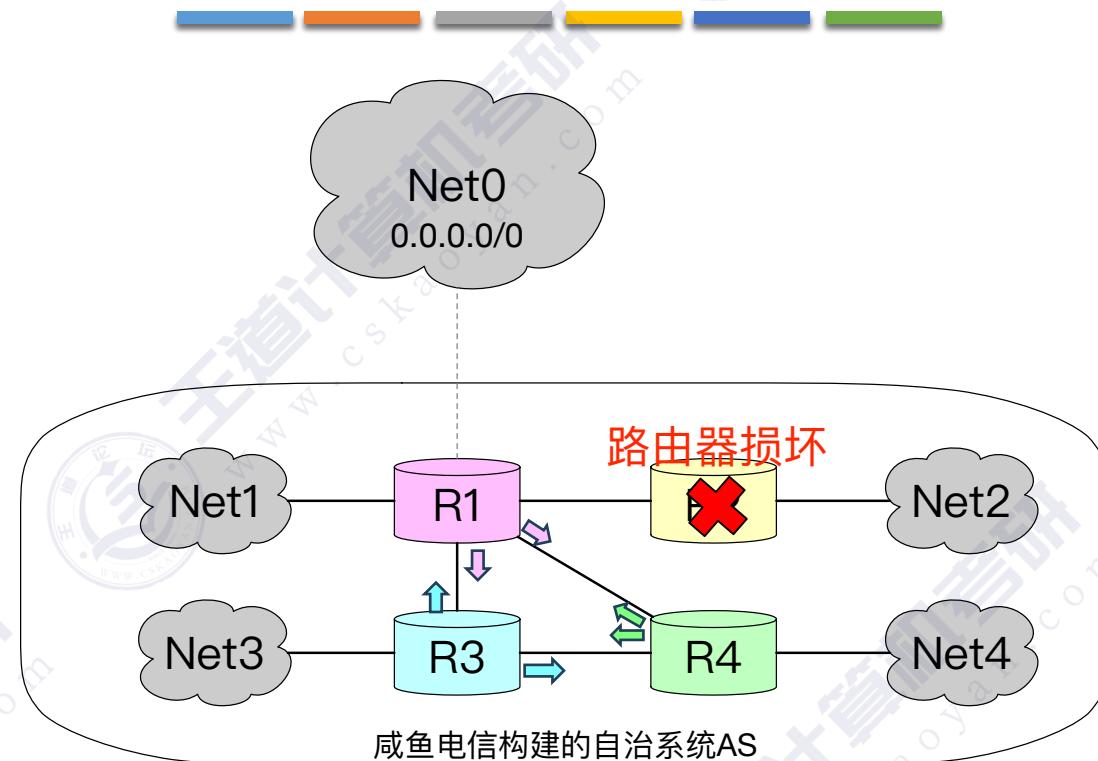
R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	4	R3/R4
Net3	2	R3
Net4	2	R4

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	1	直接交付
Net4	2	R4

王道计算机考研
www.cskaoyan.com



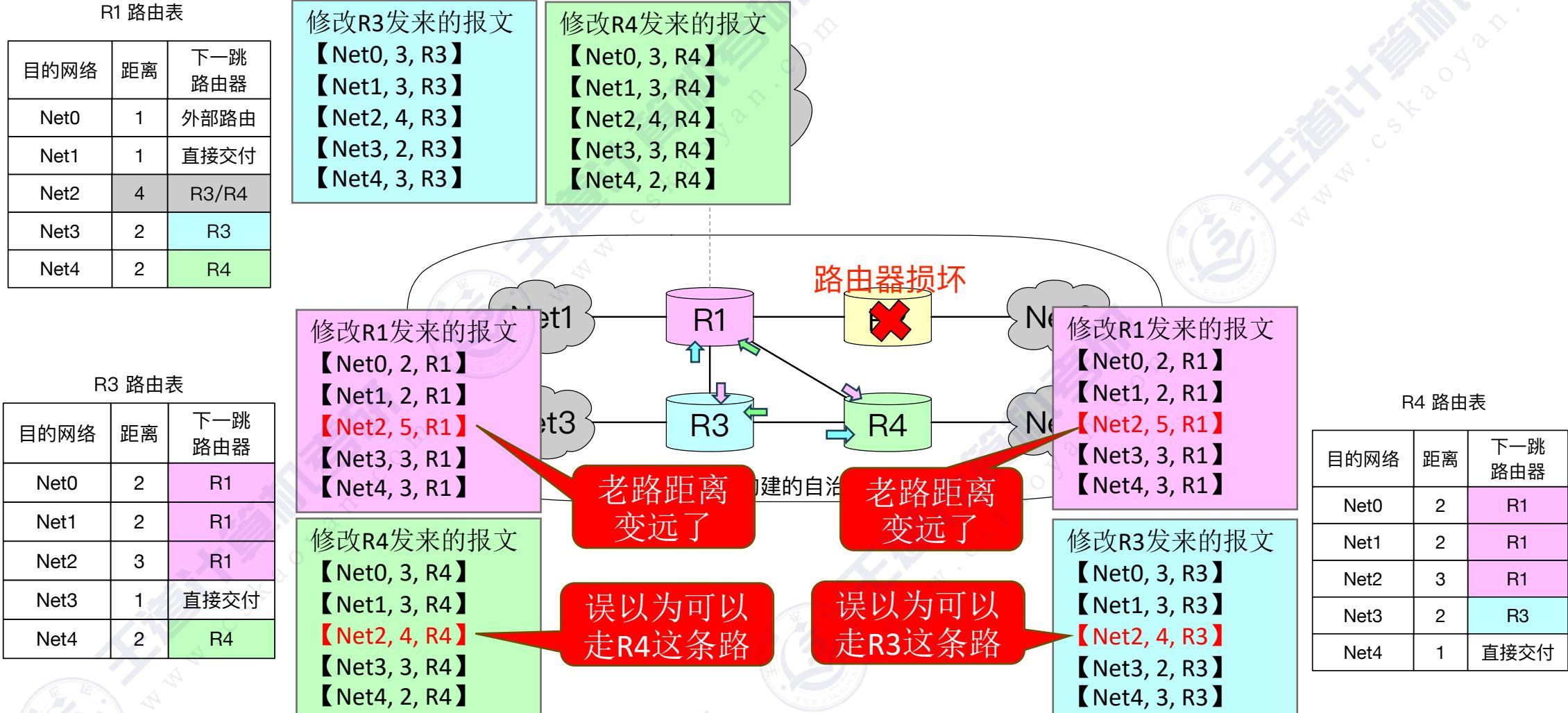
周期=30秒

每个路由器仅和相邻 路由器周期性地交换并更新路由信息

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	3	R1
Net3	2	R3
Net4	1	直接交付

330 时刻：各路由器向邻居发送RIP报文（包含完整路由表）



330 时刻：收到邻居的RIP报文后，更新自己的路由信息

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	4	R3/R4
Net3	2	R3
Net4	2	R4

修改R3发来的报文

【Net0, 3, R3】
【Net1, 3, R3】
【Net2, 4, R3】
【Net3, 2, R3】
【Net4, 3, R3】

修改R4发来的报文

【Net0, 3, R4】
【Net1, 3, R4】
【Net2, 4, R4】
【Net3, 3, R4】
【Net4, 2, R4】

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	4	R4
Net3	1	直接交付
Net4	2	R4

修改R1发来的报文

【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 5, R1】
【Net3, 3, R1】
【Net4, 3, R1】

修改R4发来的报文

【Net0, 3, R4】
【Net1, 3, R4】
【Net2, 4, R4】
【Net3, 3, R4】
【Net4, 2, R4】

王道计算机考研
www.cskaoyan.com

检查：①有没有发现新网络？

②旧网络的距离是否更新？

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	4	R3
Net3	2	R3
Net4	1	直接交付

路由器损坏

老路距离变远了

老路距离变远了

误以为可以走R4这条路

误以为可以走R3这条路

360 时刻：各路由器向邻居发送RIP报文（包含完整路由表）

R1 路由表

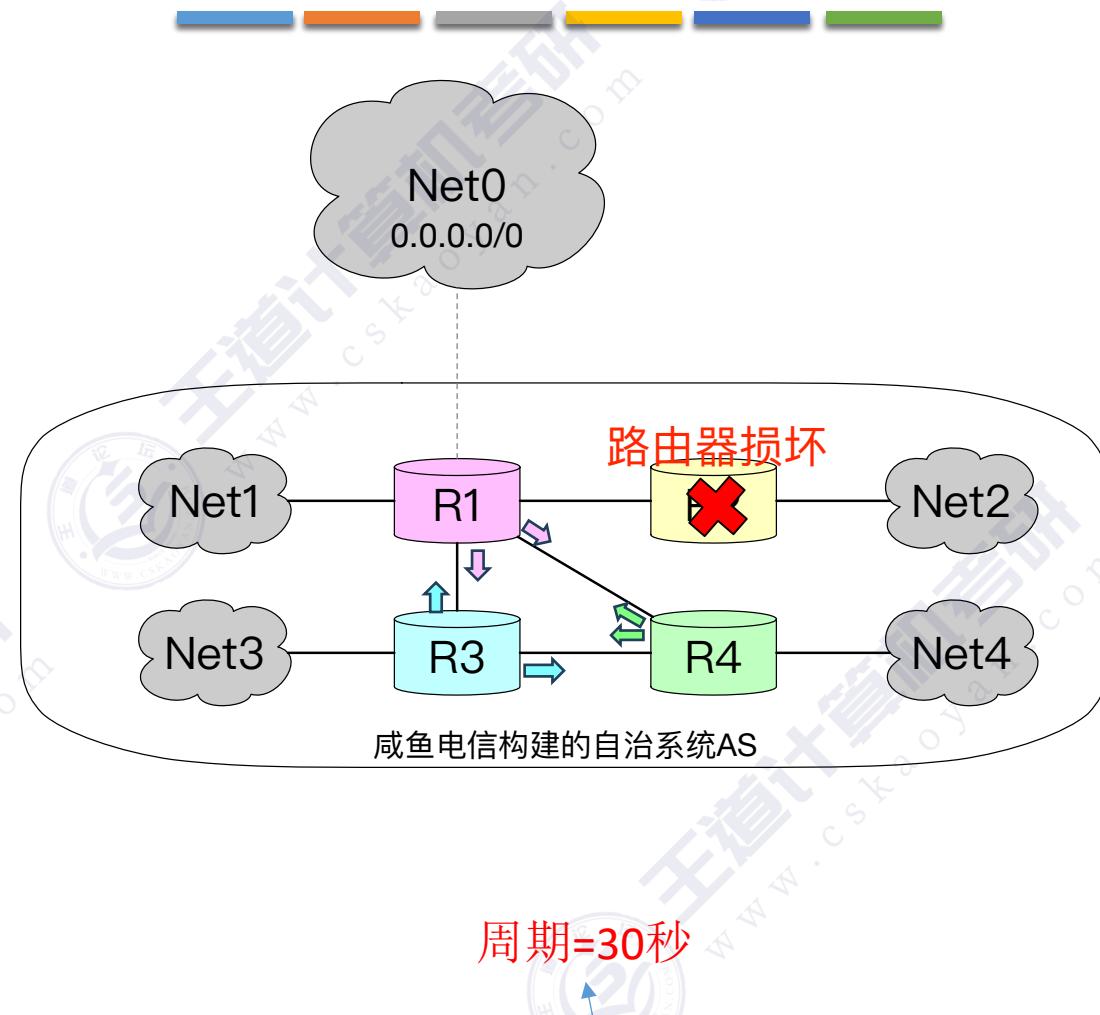
目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	4	R3/R4
Net3	2	R3
Net4	2	R4

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	4	R4
Net3	1	直接交付
Net4	2	R4

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	4	R3
Net3	2	R3
Net4	1	直接交付



每个路由器仅和相邻路由器周期性地交换并更新路由信息

360 时刻：收到邻居的RIP报文后，更新自己的路由信息

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	4	R3/R4
Net3	2	R3
Net4	2	R4

修改R3发来的报文

【Net0, 3, R3】
【Net1, 3, R3】
【Net2, 5, R3】
【Net3, 2, R3】
【Net4, 3, R3】

修改R4发来的报文

【Net0, 3, R4】
【Net1, 3, R4】
【Net2, 5, R4】
【Net3, 2, R4】
【Net4, 2, R4】

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	4	R4
Net3	1	直接交付
Net4	2	R4

修改R1发来的报文

【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 5, R1】
【Net3, 3, R1】
【Net4, 3, R1】

修改R4发来的报文

【Net0, 3, R4】
【Net1, 3, R4】
【Net2, 5, R4】
【Net3, 3, R4】
【Net4, 2, R4】

王道计算机考研
www.cskaoyan.com

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	4	R3
Net3	2	R3
Net4	1	直接交付

路由器损坏

咸鱼电信构建的自治系统AS

检查：①有没有发现新网络？
②旧网络的距离是否更新？

360 时刻：收到邻居的RIP报文后，更新自己的路由信息

R1 路由表		
目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	5	R3/R4
Net3	2	R3
Net4	2	R4

修改R3发来的报文
 【Net0, 3, R3】
 【Net1, 3, R3】
【Net2, 5, R3】
 【Net3, 2, R3】
 【Net4, 3, R3】

修改R4发来的报文
 【Net0, 3, R4】
 【Net1, 3, R4】
【Net2, 5, R4】
 【Net3, 3, R4】
 【Net4, 2, R4】

现象：每经过一轮路由信息交换，到达Net2的距离+1。
 多轮迭代后Net2距离增加到16，才能确定Net2不可达

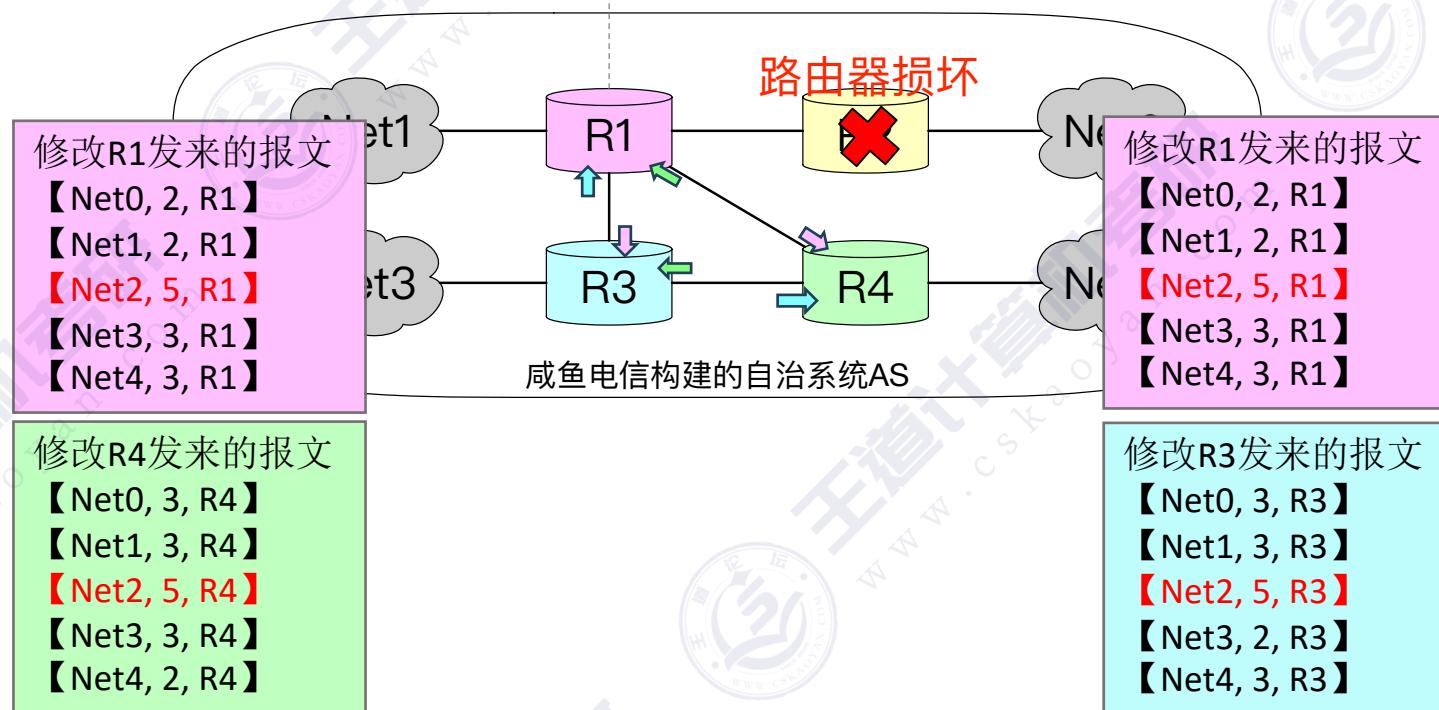
RIP的缺点：坏消息传播的慢（慢收敛现象）

R3 路由表		
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	5	R1/R4
Net3	1	直接交付
Net4	2	R4

修改R1发来的报文
 【Net0, 2, R1】
 【Net1, 2, R1】
【Net2, 5, R1】
 【Net3, 3, R1】
 【Net4, 3, R1】

修改R4发来的报文
 【Net0, 3, R4】
 【Net1, 3, R4】
【Net2, 5, R4】
 【Net3, 3, R4】
 【Net4, 2, R4】

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	5	R1/R3
Net3	2	R3
Net4	1	直接交付



检查：
 ①有没有发现新网络？
 ②旧网络的距离是否更新？

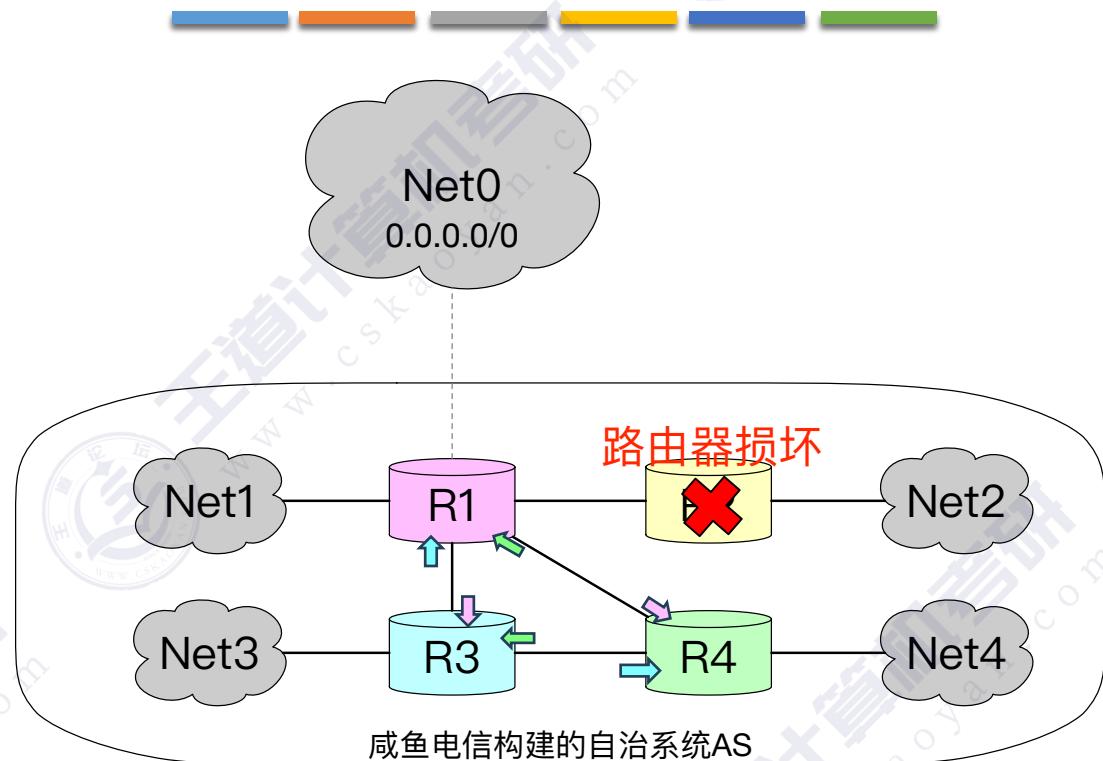
690 时刻：真不容易！终于收敛了！

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	16	NULL
Net3	2	R3
Net4	2	R4

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	1	直接交付
Net4	2	R4



现象：每经过一轮路由信息交换，到达Net2的距离+1。多轮迭代后Net2距离增加到16，才能确定Net2不可达

RIP的缺点：坏消息传播的慢（慢收敛现象）

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	2	R3
Net4	1	直接交付

本节内容

RIP

的工作过程示例
(反映RIP的优点)

RIP 的优缺点



RIP 的优点：

- 1) 实现简单、开销小、收敛过程较快。
- 2) 若一个路由器发现了更短的路由，则这种更新信息就传播得很快，在较短时间内便可被传至所有路由器，俗称“好消息传播得快”。

RIP 的缺点：

- 1) RIP 限制了网络的规模，它能使用的最大距离为 15（16 表示不可达）。
- 2) 路由器之间交换的是路由器中的完整路由表，因此网络规模越大，开销也越大。
- 3) 当网络出现故障时，路由器之间需反复多次交换信息才能完成收敛，要经过较长时间才能将故障消息传送到所有路由器（慢收敛现象），俗称坏消息传播得慢。

注：距离向量路由可能会出现环路的情况，RIP 规定路径上的最高跳数=16 是一种防止环路的简单粗暴的方式。

RIP 的工作过程（基于 距离向量算法）

周期=30秒

路由器刚开始工作时，知道自己到直接相连的几个网络的距离为 1。每个路由器仅和相邻路由器周期性地交换并更新路由信息。经过若干次交换和更新后，所有的路由器最终都会知道到达本自治系统内任何网络的最短距离和下一跳路由器的地址，称为收敛。

对每个相邻路由器发送来的 RIP 报文，执行如下步骤：

- 1) 对地址为 X 的相邻路由器发来的 RIP 报文，先修改该报文中的所有项目：
把“下一跳”字段中的地址都改为 X，并把所有“距离”字段的值加 1。

本质是距离向量路由算法

- 2) 对修改后的 RIP 报文中的每个项目，执行如下步骤：

IF (若原来的路由表中没有目的网络 N)

 则把该项目添加到路由表中（表明这是新的目的网络）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址是 X)

 用收到的项目替换原路由表中的项目（因为要以更新的消息为准）

ELSE IF (若原来的路由表中有目的网络 N，且下一跳路由器的地址不是 X)

 若收到的项目中的距离 d 小于路由表中的距离，则进行更新

ELSE 什么也不做。

- ①有没有发现新网络？
- ②旧网络的距离是否要更新？
 - 找到一条新路径，距离更近
 - 继续走老路，但老路距离有变化

- 3) 若 180 秒 (RIP 默认超时时间) 还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达的路由器，即把距离设置为 16 (表示不可达)
- 4) 返回

书接上回：690 时刻

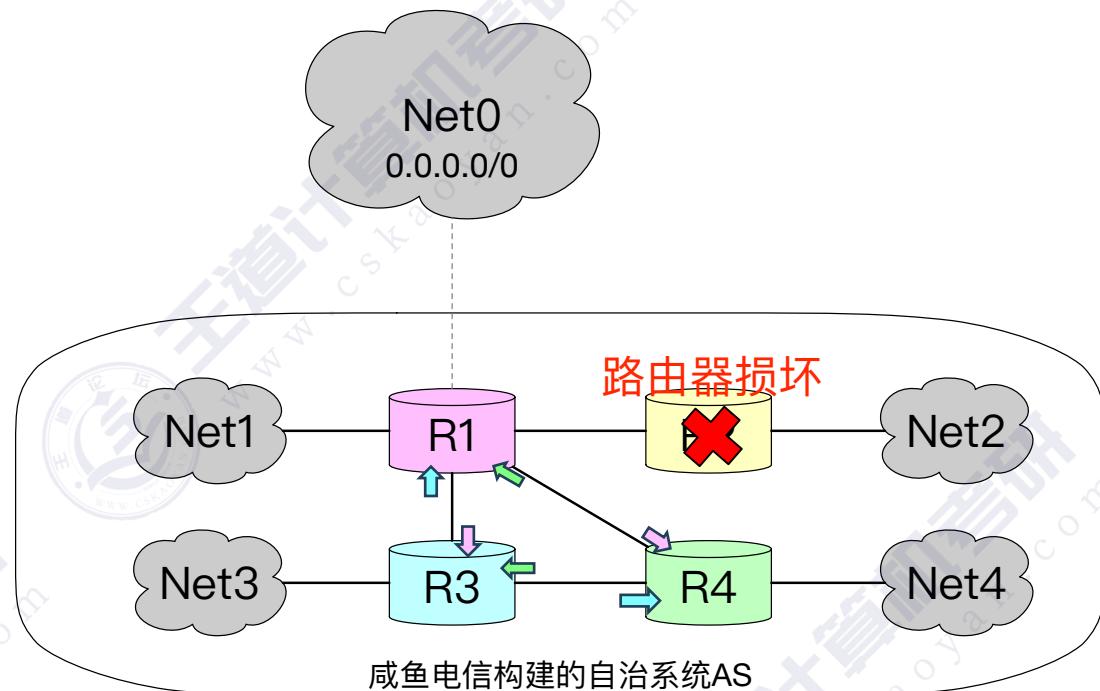
R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	16	NULL
Net3	2	R3
Net4	2	R4

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	1	直接交付
Net4	2	R4

我太难了



R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	2	R3
Net4	1	直接交付

自从R2损坏后，经过多轮迭代，所有路由器终于收敛！

700 时刻：路由器R2归来！

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	16	NULL
Net3	2	R3
Net4	2	R4

R3 路由表

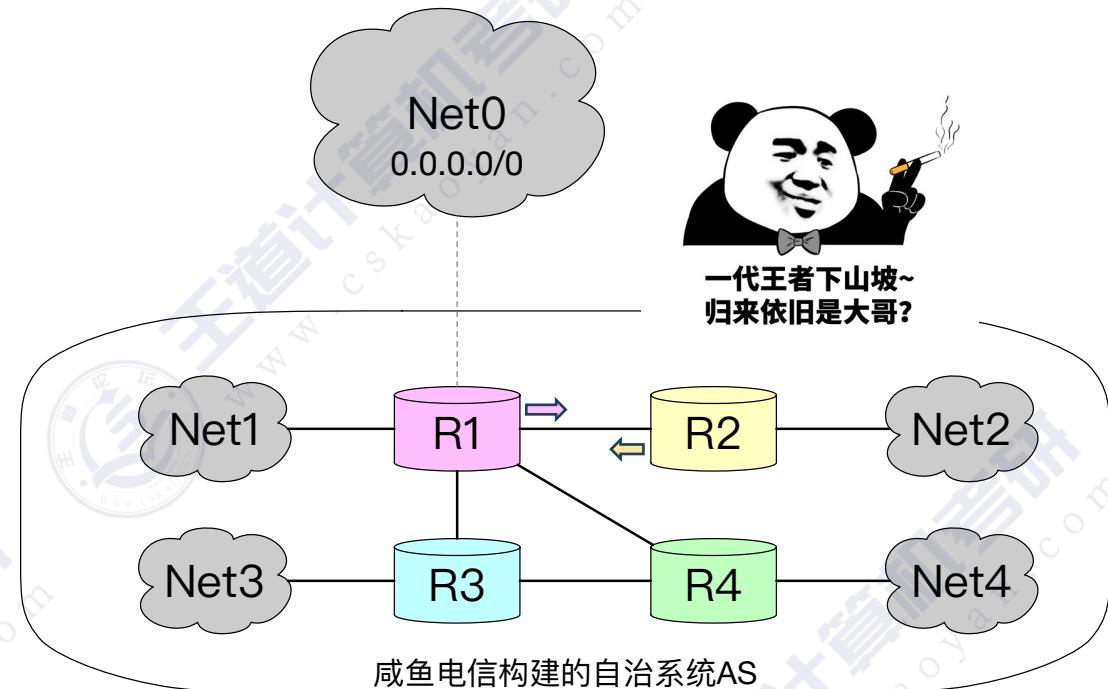
目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	1	直接交付
Net4	2	R4

R2 的初始路由表

目的网络	距离	下一跳路由器
Net2	1	直接交付

R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	2	R3
Net4	1	直接交付



- 路由器刚开始工作时，知道自己到直接相连的几个网络的距离为1
- 新路由器构建好初始路由表，就可以立即发送给邻居
- 新路由器，可以用RIP请求报文，迅速向邻居请求路由信息

700 时刻：路由器R2归来！

距离16已经到顶了，不必再+1

R1 路由表

目的网络	距离	下一跳路由器
Net0	1	外部路由
Net1	1	直接交付
Net2	16	NULL
Net3	2	R3
Net4	2	R4

修改R2发来的报文
【Net2, 2, R2】



修改R1发来的报文

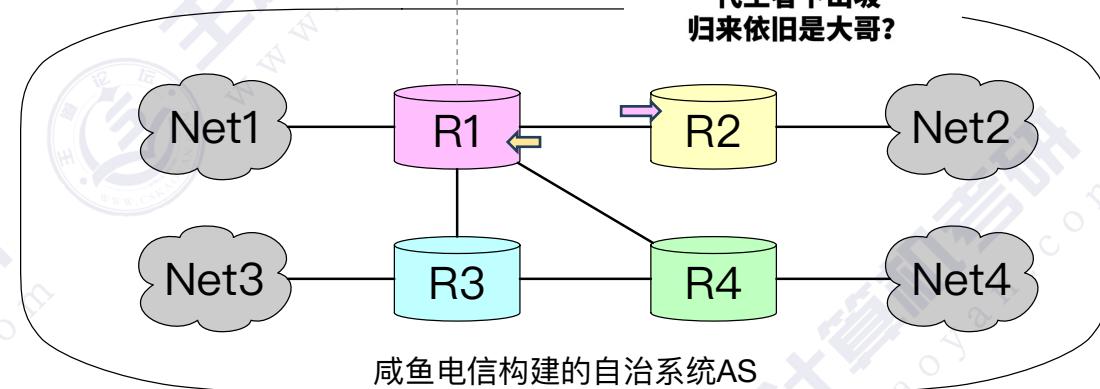
【Net0, 2, R1】
【Net1, 2, R1】
【Net2, 16, R1】
【Net3, 3, R1】
【Net4, 3, R1】

R2 的初始路由表

目的网络	距离	下一跳路由器
Net2	1	直接交付

R3 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	1	直接交付
Net4	2	R4

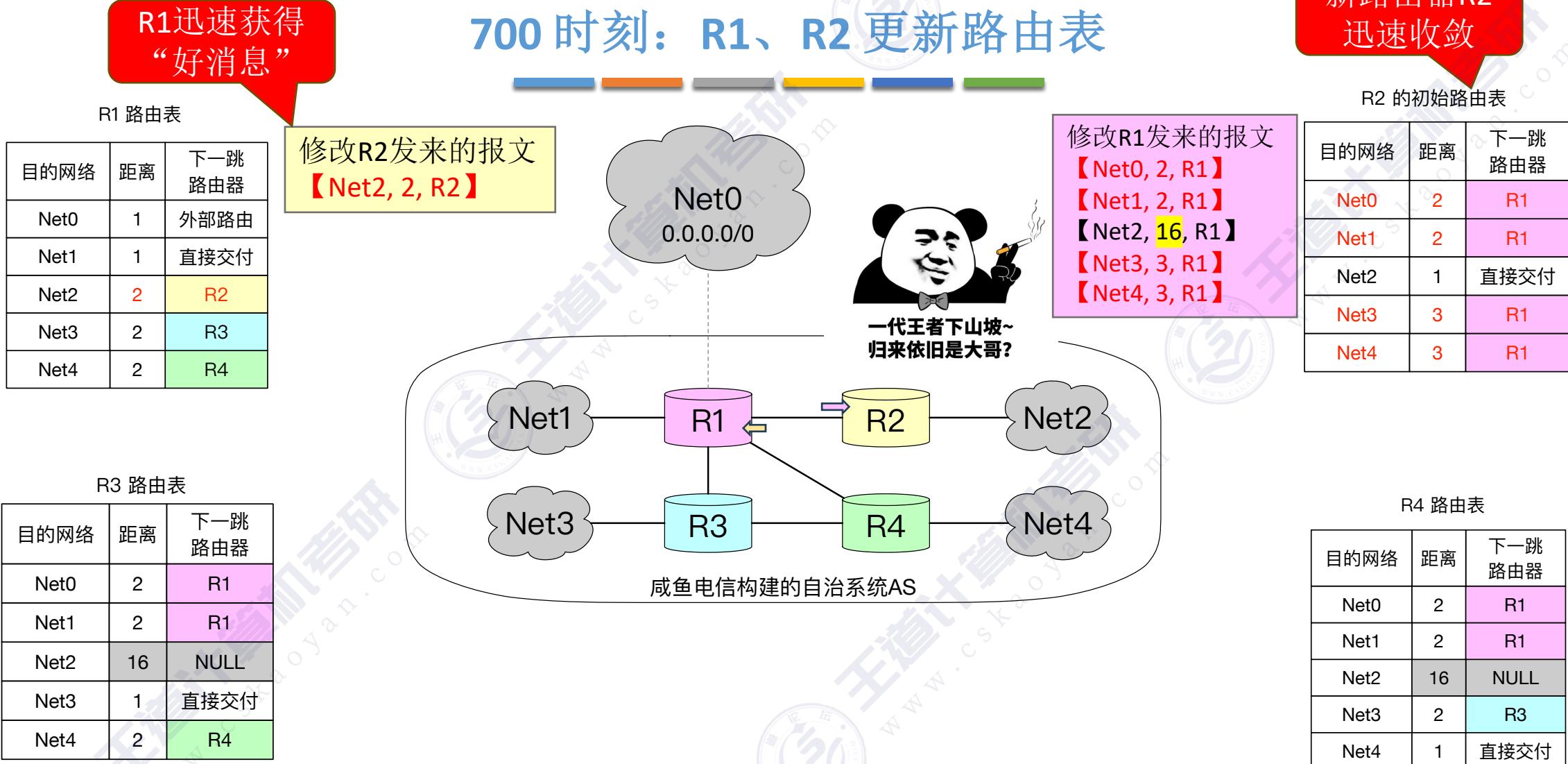


R4 路由表

目的网络	距离	下一跳路由器
Net0	2	R1
Net1	2	R1
Net2	16	NULL
Net3	2	R3
Net4	1	直接交付

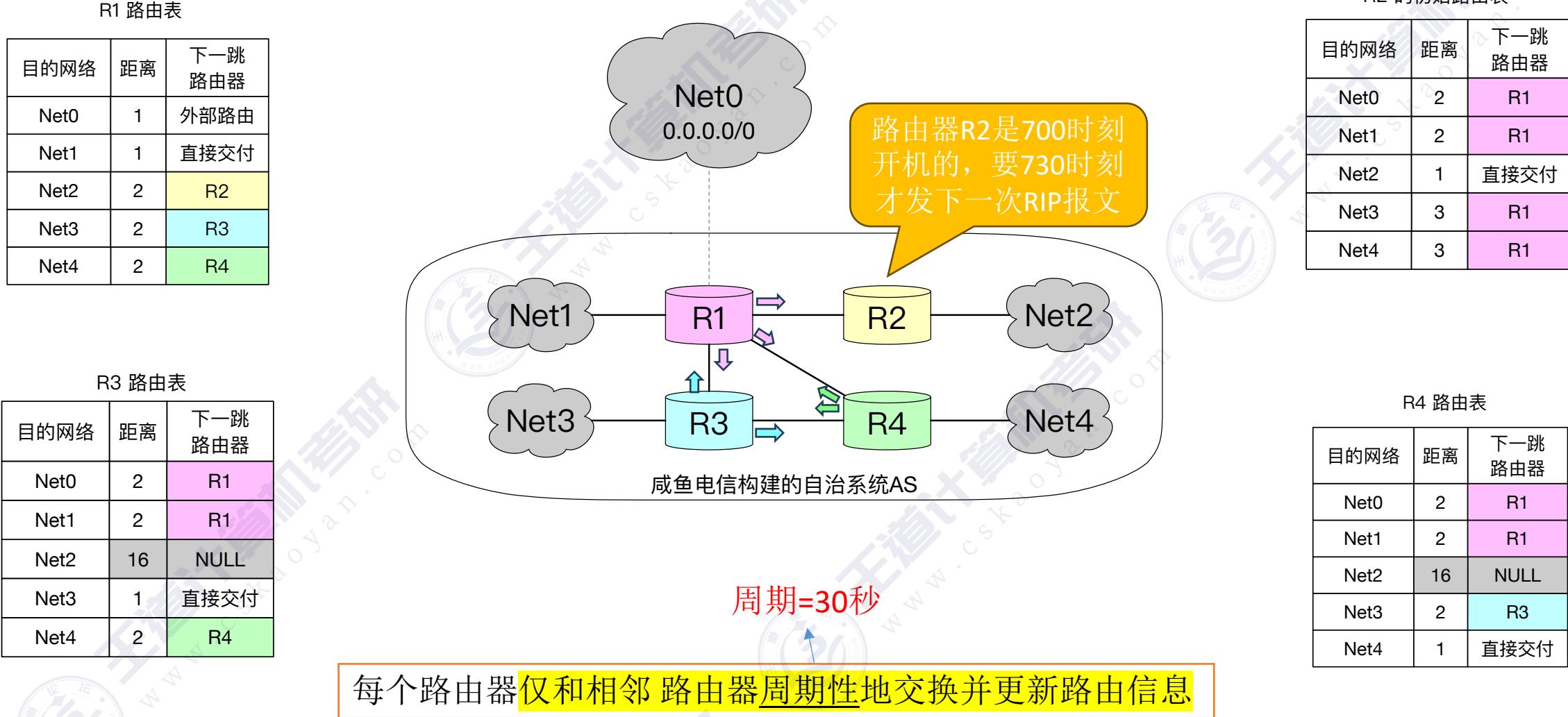
- 路由器刚开始工作时，知道自己到直接相连的几个网络的距离为1
- 新路由器构建好初始路由表，就可以立即发送给邻居
- 新路由器，可以用 RIP请求报文，迅速向邻居请求路由信息

700 时刻：R1、R2 更新路由表

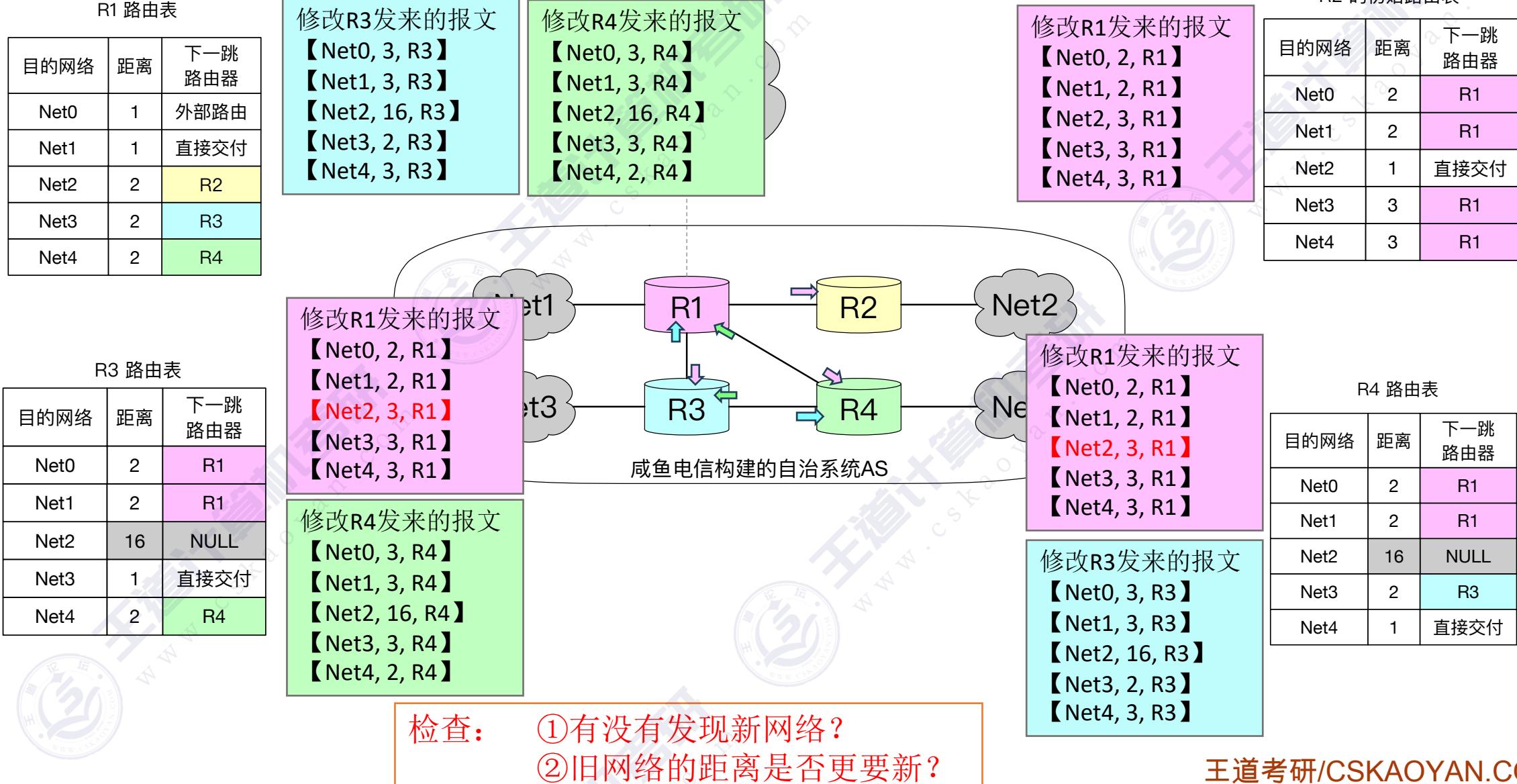


- 路由器刚开始工作时，知道自己到直接相连的几个网络的距离为 1
- 新路由器构建好初始路由表，就可以立即发送给邻居
- 新路由器，可以用 RIP 请求报文，迅速向邻居请求路由信息

720 时刻：R1、R3、R4 定时发送RIP报文



720 时刻：R1、R3、R4 定时发送RIP报文



720 时刻：R1、R3、R4 定时发送RIP报文

