

网络技术与信息安全



第6讲：路由器组网——动态路由

河南中医药大学信息技术学院
网络技术课程教学组

本章教学计划

- ▣ 动态路由的特点
- ▣ RIP路由协议
- ▣ OSPF路由协议
- ▣ BGP



一、路由算法

—— 路由表中动态路由的生成依靠动态路由协议，而路由协议的核心是路由算法。

1.1 认识路由算法

路由算法

1.1 认识路由算法

□ 什么是路由算法？

- 路由算法（Routing Algorithm），又名选路算法，是在给定一组路由器及连接路由器链路的情况下，找出一条从源节点到目标节点的最佳路径的算法。
- 通俗的讲，就是把路由器选择最佳路径的过程策略称为路由算法，是路由器的关键功能所在。
- 为了完成这项工作，路由器需要收集和保存着各种与传输路径的相关数据，如拓扑结构、跳数、延迟、端口度量、端口速率等，然后根据相应的路由算法生成一个个路由表（转发表）表项，在数据包转发时提供路由选择。

路由算法

1.1认识路由算法

□ 路由算法分类

■ 按是否自适应网络变化划分：

■ **静态路由算法**：不能根据网络流量和拓扑结构的变化更新路由表

- 静态路由一般不认为是真正的路由算法，它是由管理员在开始路由之前，在路由器上手工创建，不能对网络改变做出反映。

■ **动态路由算法**：可根据网络流量和拓扑结构的变化更新路由表

- 距离矢量路由算法
- 链路状态路由算法

■ 下面主要介绍两种常用的**动态路由算法**

1.2 距离矢量路由算法

路由算法

1.2 距离矢量路由算法

□ 距离矢量路由算法（也称为距离向量路由算法）

■ 距离和矢量的含义

- **距离**：指当前路由器到达目标网络的远近**度量**值。这个距离并不是物理上的距离，而是根据某种算法计算得出的一个数值
- **矢量**：从当前路由器到达目标网络的距离值是有方向的，这由其下一跳地址体现。
- **距离矢量**：将一条路由信息视为一个由目标和距离组成的矢量。此处“距离”也可理解为某种度量（Metric）值，用于衡量到达目的地的成本。

■ 距离矢量路由算法（Distance Vector Routing Algorithm，DV）核心是： **掌握达到每一个目标网络，下一跳怎么走。**

- 该算法主要在RIP协议中使用。Cisco的IGRP（内部网关路由协议）和EIGRP路由协议也采用DV这种路由算法

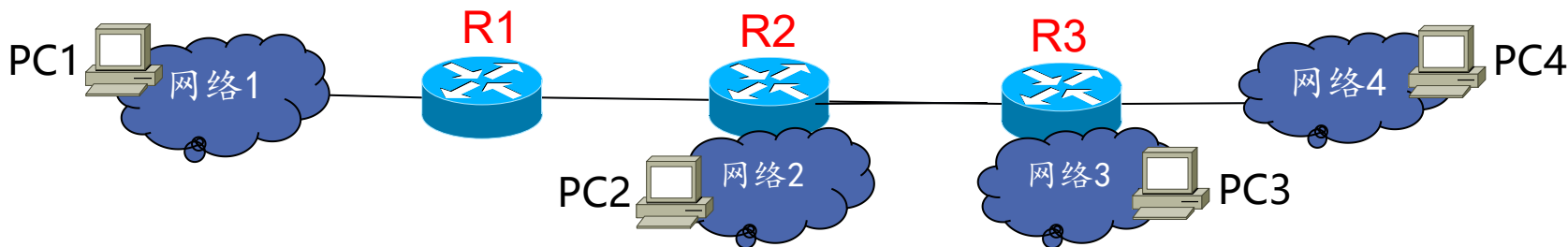
路由算法

1.2 距离矢量路由算法

□ 算法应用

■ 在一个相对复杂的路由环境中，由于有多个路由器的互联，从一个网络到另一个网络的路径可能就有多个。在应用距离矢量路由算法协议（例如RIP）的网络中，每个路由器要通过执行路由算法，从多个到达同一网络的路径中选择最优的写入到自己的路由表中。即，每个路由器都维护一张路由表，表中记录了到达网络中每个目标的最佳距离和所使用的线路（下一跳），并依据此来转发数据包。

➤ 如下图中，路由器R1、R2、R3的路由表中，都维护了从它自己到网1、网2、网3、网4的最佳距离记录。



路由算法

1.2 距离矢量路由算法

□ 如何衡量每条路径的好坏呢？

- 在相同的衡量标准下，度量值（Metric）最小的路径，就是最优的。
- 度量值代表距离。它们用来在寻找路由时确定最优路由。每一种路由算法在产生路由表时，会为每一条通过网络的路径产生一个数值（度量值），最小的值表示最优路径。度量值的计算可以只考虑路径的一个特性，但更复杂的度量值是综合了路径的多个特性产生的。
- 一些常用的度量值：跳数、带宽、时延、负载等
 - 举例：距离值

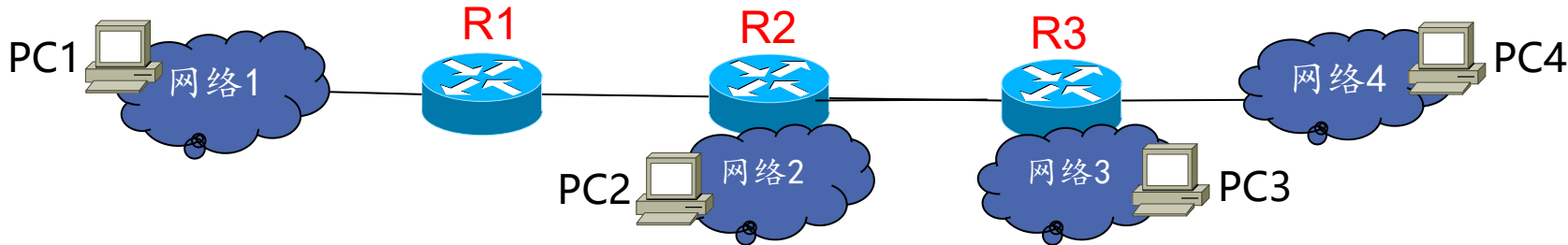
路由算法

1.2 距离矢量路由算法

□ 如何衡量每条路径的好坏呢？

■ “距离”举例

- 例如RIP协议中，距离通常指跳数（hop）。
- 从一路由器到直接连接的网络的距离定义为1，从本路由器开始，在到达目标主机的过程中，每经过一个路由器（或提供路由功能的三层交换机等），就加一跳，即距离值加1。RIP协议规定最大跳数为15，若为16则视为不可达。
- 举例：下图中，R1到PC1距离是1，R1到PC2的距离是2，R1到PC3和PC4的距离是3



路由算法

1.2 距离矢量路由算法

□ 距离矢量路由算法的工作原理

- **信息获取**：每台路由器和其邻居路由器交换路由信息，即告诉对方：从“我”这里到达网络A，其距离（度量值）是B。
- **信息叠加**：路由器在所收到的、邻居发来的每条路由信息上，叠加从自己到这个邻居路由器的距离矢量（即自己到邻居的Metric值）。这样，每条路由信息就变成了从当前路由器出发，到达目的网络的总距离及其下一跳地址。
- **路由表更新**：路由器根据叠加后的路由信息更新自己的路由表。如果新发现的路径比原有路径更优（即Metric值更小），则替换原有路径。

路由算法

1.2 距离矢量路由算法

□ 距离矢量路由算法的工作原理

- **信息交换**：路由器会定期或当路由信息发生变化时，将其路由表发送给邻居路由器。这样，邻居路由器也能根据最新的信息更新自己的路由表。
- **收敛**：随着路由信息的不断交换和路由表的不断更新，整个网络中的路由信息会逐渐趋于一致。这个过程称为路由收敛。在收敛状态下，网络中的每个路由器都拥有到达每个目标的最佳路径。

1.3 链路状态路由算法

路由算法

1.3 链路状态路由算法

□ 链路状态路由算法

- 链路状态路由算法（Link State Routing Algorithm），也被称为最短路径优先算法（SPF, Shortest Path First），是一种以图论为基础的网络路由算法。
- 其核心思想是通过掌握网络的全部拓扑结构信息，利用最短路径算法来计算网络间的最佳路由。
- 链路状态路由算法的一个典型应用实例是OSPF（Open Shortest Path First）协议。

路由算法

1.3 链路状态路由算法

□ 三个基本概念

- **链路状态广播**：每个路由器周期性地广播自己的链路状态信息，包括与其相连的链路和邻居路由器的状态。这些信息被称为链路状态数据包（LSA），用于构建整个网络的拓扑图。
- **拓扑数据库**：拓扑数据库是一个存储所有路由器链路状态信息的集合，用于描述网络拓扑结构。每个路由器都会维护一个本地的拓扑数据库，并使用它来计算最短路径。
- **最短路径计算**：基于拓扑数据库，每个路由器都可以使用最短路径算法（如Dijkstra算法）来计算从本地路由器到其他路由器的最短路径。这些计算结果将存储在路由表中，用于数据包的转发决策。

路由算法

1.3 链路状态路由算法

□ 链路状态路由算法的工作原理

■ 链路状态路由协议的工作原理主要包括以下几个步骤：

- **建立邻居关系**：通过交换Hello数据包来创建邻居关系。这些Hello数据包用于检测链路上的邻居路由器，并建立邻接关系。
- **传播LSA**：一旦建立了邻接关系，路由器将开始相互传播LSA。这些LSA包含了路由器的链路状态信息，如接口的IP地址、子网掩码、链路类型和带宽等。
- **泛洪LSA**：路由器将收到的LSA泛洪到其所有邻居，邻居再将LSA泛洪给它们的邻居，直到整个网络中的所有路由器都收到了这些LSA。
- **构建拓扑数据库**：每个路由器都会收集并存储所有邻居发来的LSA，形成一个完整的拓扑数据库。
- **计算最短路径**：基于拓扑数据库，每个路由器使用SPF算法计算出到每个目的地的最短路径，并将这些路径存储在路由表。

路由算法

1.3 链路状态路由算法

□ 链路状态路由算法的优点

■ 与距离矢量路由算法相比，链路状态路由算法具有以下优点：

- 快速收敛：由于LSA的发布是面向整个网络的，所有路由器都能迅速建立网络拓扑的准确视图，因此链路状态路由算法具有较快的收敛速度。
- 无路由环路：由于每个路由器都基于完整的拓扑数据库计算最短路径，因此可以避免路由环路的出现。
- 支持区域划分：链路状态路由协议支持将网络划分为多个区域，每个区域维护自己的链路状态数据库，这有助于减少网络开销并提高可扩展性。
- 负载平衡：链路状态路由协议可以根据链路的负载情况来选择最佳路径，实现负载平衡。

二、动态路由协议

动态路由协议

▣ 什么是动态路由协议？

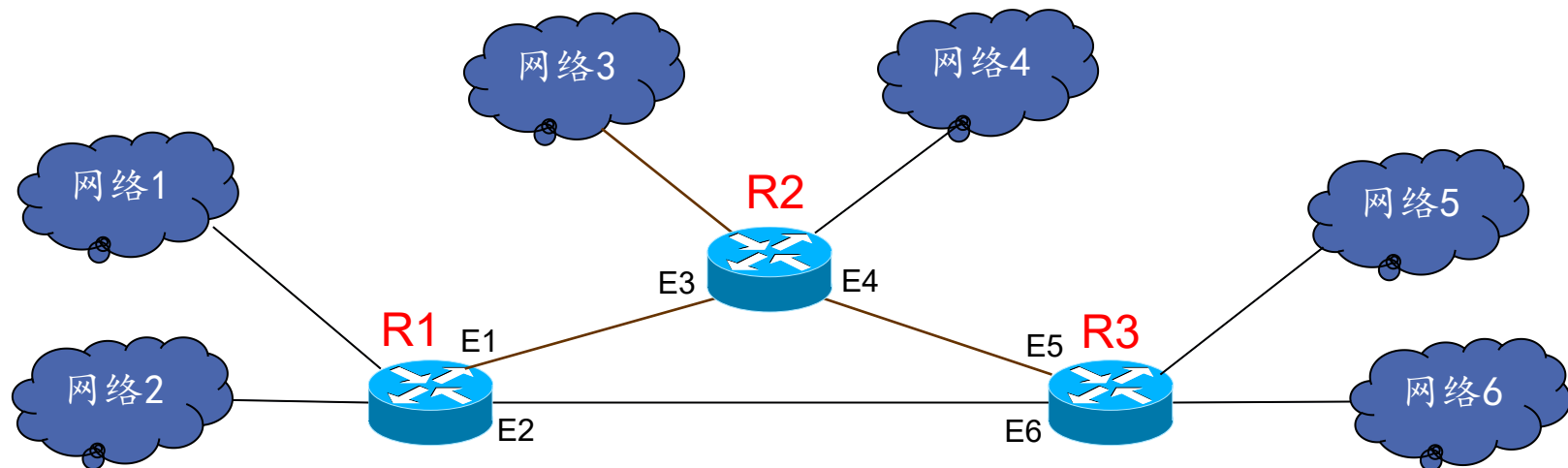
- 动态路由协议是用于路由器之间交换路由信息的协议。它基于某种路由算法（例如距离矢量算法），使网络中的路由器能够动态共享各自所具有的网络路由信息，并根据这些信息计算出从本路由器到达各个网络的最佳路径，然后将这些路径添加到各自的路由表中，即生成路由表。（也就是在《路由器基础》中所提到的“寻径”操作）
- 这一过程是自动化的，无需管理员手动配置每条路由，避免路由环路的出现。

动态路由协议

□ 动态路由协议的基本工作

- **路由发现与学习**：路由器通过动态路由协议向网络中的其他路由器发送和接收路由信息。每个路由器都会维护一个路由表，记录到达各个网络的路径信息。当网络拓扑发生变化时，如新增或删除网络设备、链路故障等，路由器会相互交换更新后的路由信息，以学习新的路由路径。
- **路由选择**：路由器根据收到的路由信息，结合自身的路由算法（如最短路径优先算法等），计算出到达每个目的网络的最优路径。最优路径的选择通常基于多种度量值，如跳数、带宽、延迟等。
- **路由维护**：路由器会定期发送路由更新消息，以确保路由表的准确性。当检测到网络拓扑变化时，路由器会立即发送更新消息，通知其他路由器进行相应的路由调整。
- **举例**

► 举例：动态路由的应用



R1路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络1 | — | 直连 |
| 网络2 | — | 直连 |

R2路由表

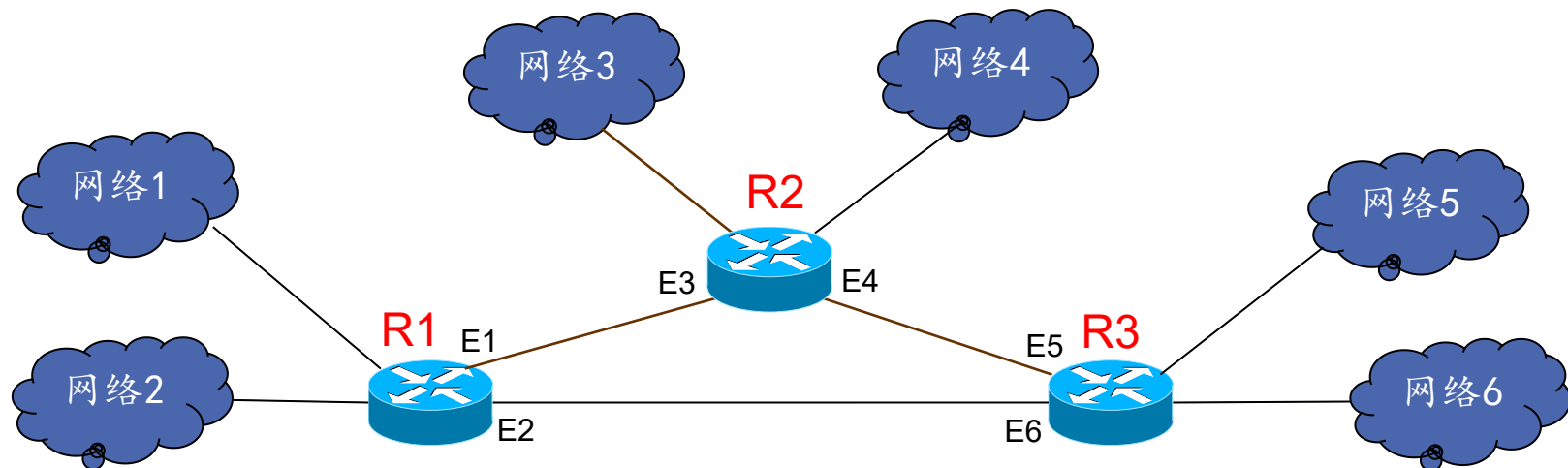
| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络3 | — | 直连 |
| 网络4 | — | 直连 |

R3路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络5 | — | 直连 |
| 网络6 | — | 直连 |

为节省篇幅，此处路由表中，省略了路由器之间的网络的直连路由

► 举例：动态路由的应用



R1路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络1 | — | 直连 |
| 网络2 | — | 直连 |

R2路由表

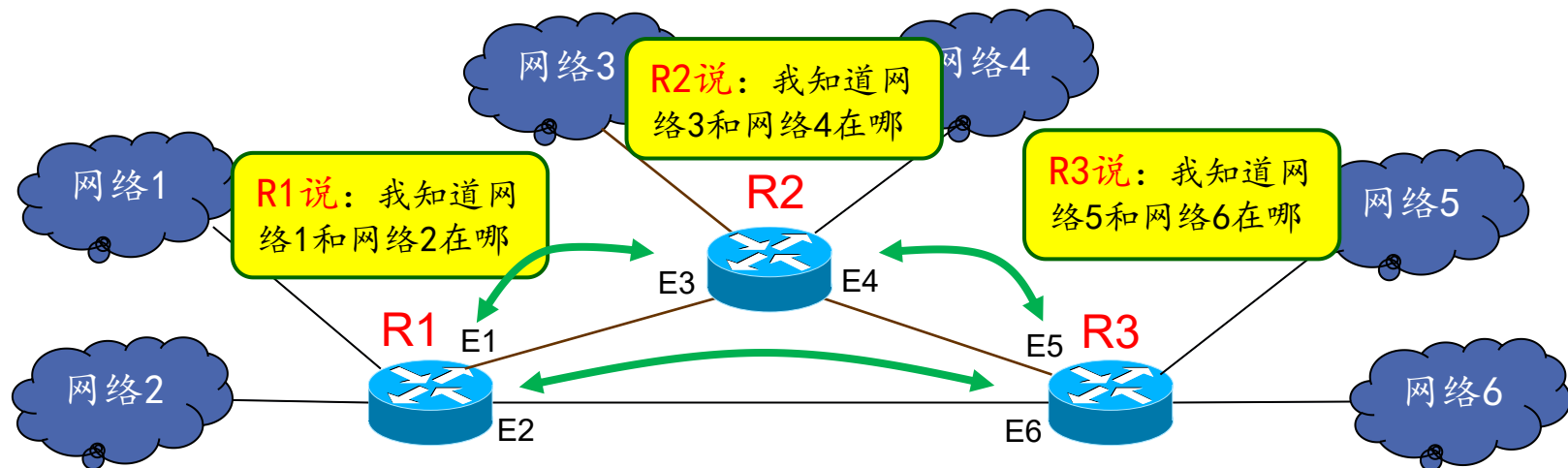
| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络3 | — | 直连 |
| 网络4 | — | 直连 |

R3路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络5 | — | 直连 |
| 网络6 | — | 直连 |

给路由器配置动态路由协议，使各路由器相互之间可以交换路由信息

► 举例：动态路由的应用



R1路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络1 | — | 直连 |
| 网络2 | — | 直连 |

R2路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络3 | — | 直连 |
| 网络4 | — | 直连 |

R3路由表

| 目的 | NEXT | 类型 |
|-----|------|----|
| 网络5 | — | 直连 |
| 网络6 | — | 直连 |

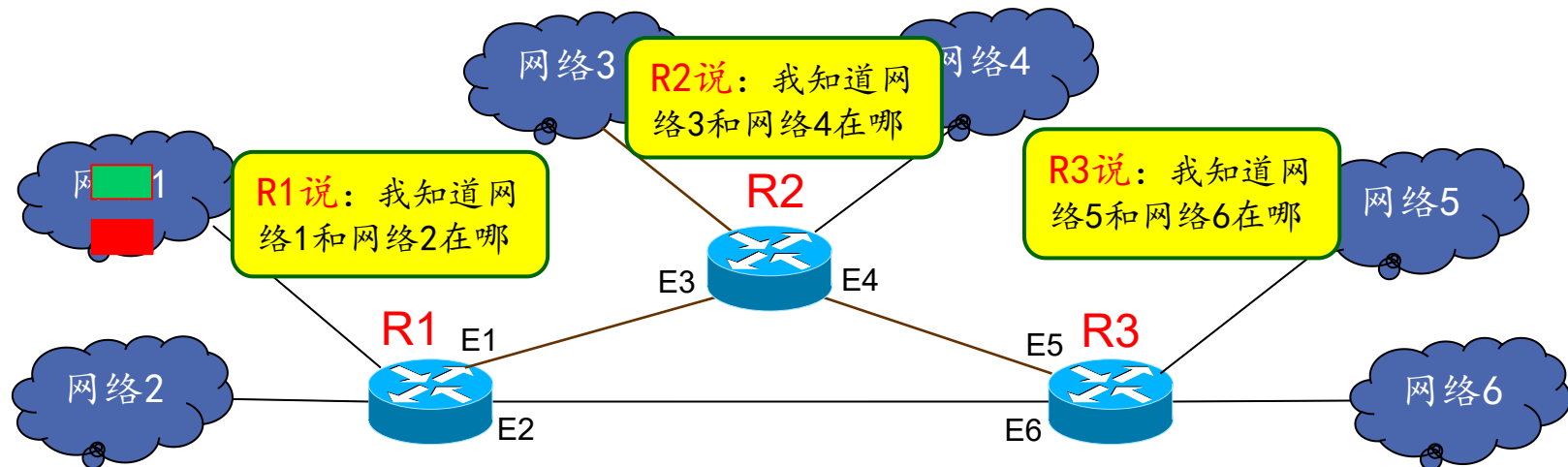
通过动态路由协议的“寻径”操作，使各个路由都知道了到达各个目的网络的路由信息

网络6 E6 动态

网络6 E5 动态

网络2 E2 动态

► 举例：动态路由的应用

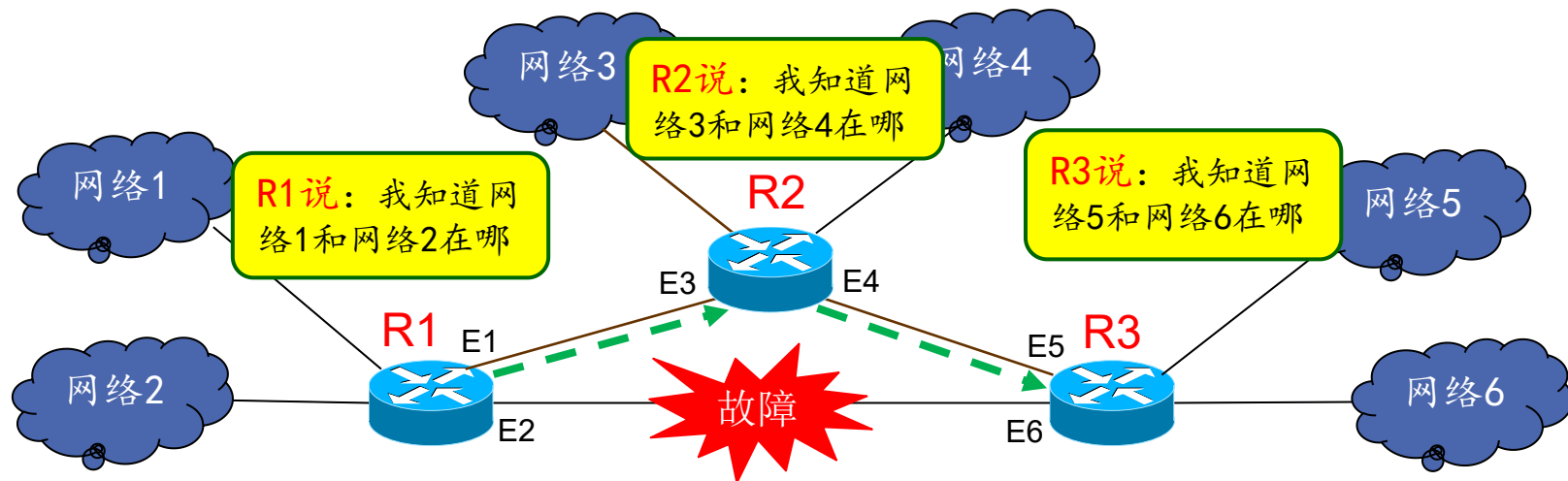


□ 问题：从网络1到达网络6，该如何走？

- 方案1：网络1→R1 →R2 →R3 →网络6（思考：方案1如何形成的？）
- 方案2：网络1→R1 →R3 →网络6（思考：方案2如何形成的？）
- 假设使用动态路由协议选择了方案2，认为这是最佳路径，并将相关信息写入R1路由表。
- 因此，当R1收到目的地是网络6的数据包时，就会转发至R3，而不是转发给R2。

通过动态路由协议的路由选择策略，可以确定“最佳”路径

► 举例：动态路由的应用



□ 问题：若R1和R3之间的链路出现故障，网络1访问网络6，该如何走？

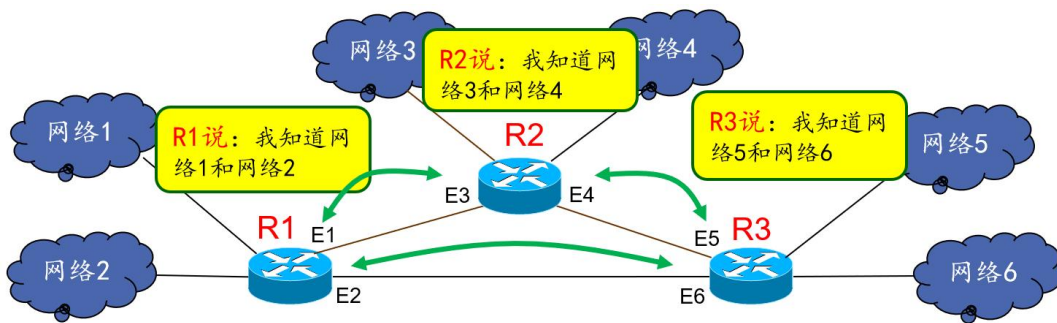
- 由于动态路由协议的作用，网络中的路由器会自动发现这一故障（即网络结构的变化），重新计算路由，并且自动更新路由器中的路由表。
- 此处，路由器R1中的动态路由协议，会根据网络结构变化，更新自己的路由表，将访问网络6的数据包，先转发至R2，即选择了方案1。这一过程不需要人工干预。
- 因此，当R1再转发目的地是网络6的数据包时，下一跳就是R2，而不是R3。

动态路由协议

□ 动态路由的特点

■ 自动通告

- 与静态路由不同，静态路由表项是一个路由器私有的，别的路由器并不知道。但是配置动态路由协议的路由器，可以通过宣告自己的直连网段的方式，将自己所知道的网络信息自动传送给相邻路由器。正是通过这种交互方式，使得路由器自动获取到整个网络内的路由信息。



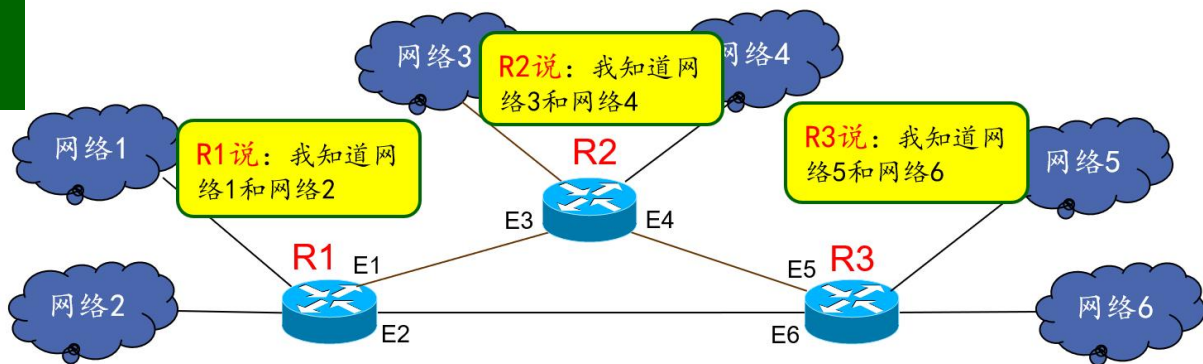
动态路由协议

□ 动态路由的特点

■ 自动生成

- 在“自动通告”的基础上，网络中各路由器利用从其他路由器那里收到的路由信息，生成自己的路由表，管理员无需一一手动创建。

R1中会自动生成到达网络3、网络4、网络5、网络6的动态路由记录。

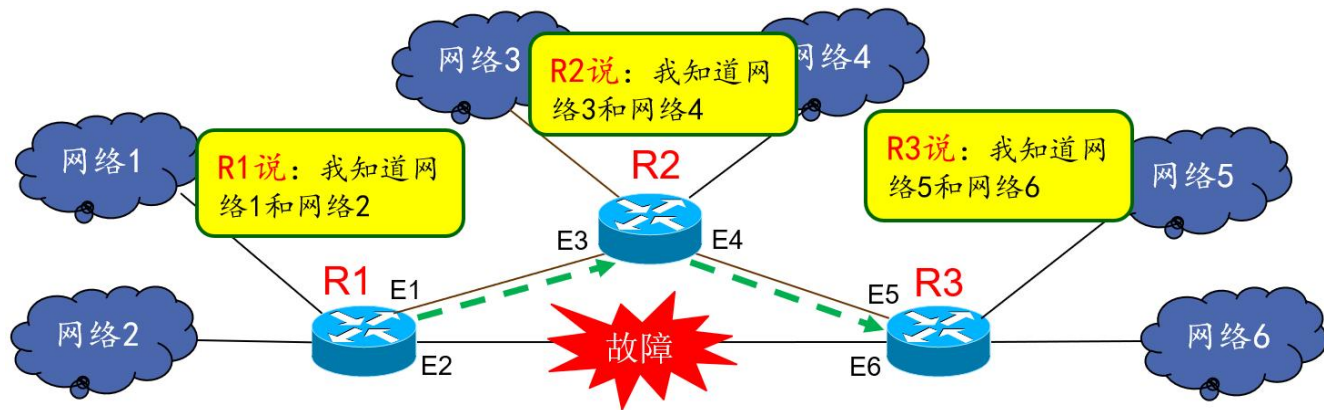


动态路由协议

□ 动态路由的特点

■ 自动更新

- 当网络结构发生改变，静态路由是无法自动跟着改变的。但动态路由协议可以根据网络拓扑结构的变化，自动重新计算网络内的路由信息，从而生成新的路由表项，同时还会自动删除无效的动态路由表项，更加方便路由管理。

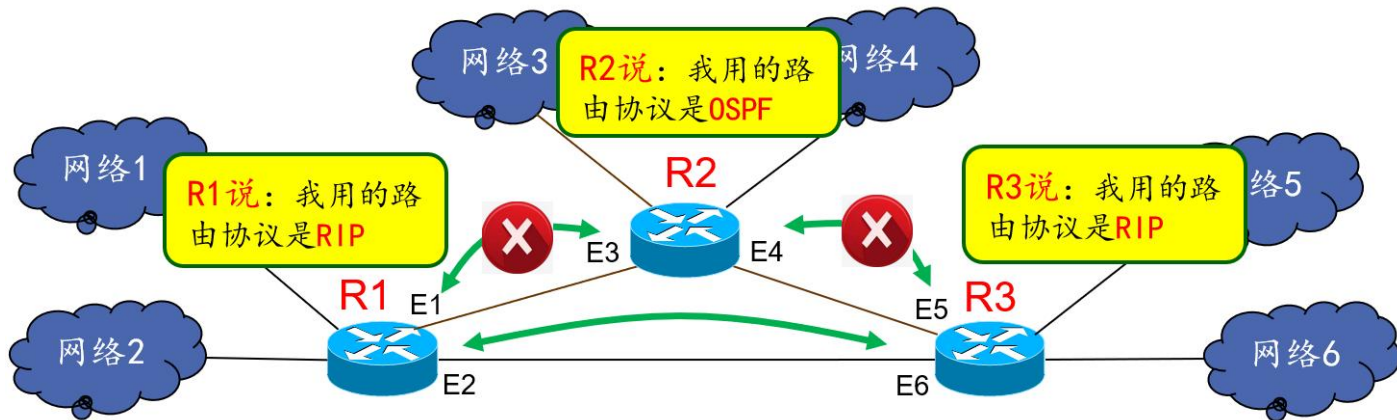


动态路由协议

□ 动态路由的特点

■ 不同动态路由协议不兼容

- 不同的动态路由协议适用的网络环境不一样，也是不兼容的。



动态路由协议

□ 内部网关协议和外部网关协议

■ 认识自治系统（AS）

- 互联网的规模非常大，如果让因特网中所有的路由器知道所有的网络应怎样到达，这样将使路由表变的非常庞大，不仅处理起来太花时间，而且路由器之间交换路由信息也需占用大量带宽。不仅如此，许多单位并不想让外界了解本单位网络建设的具体细节，包括网络结构以及所采用的路由选择协议，但同时还希望连接到因特网上。
- 为此，互联网将整个互联网划分为许多较小的自治系统（Autonomous Ssystem，简称 AS）。一个AS内部的路由器通常使用相同的路由选择协议和共同的度量，以确定分组在该AS内的路由。

动态路由协议

□ 内部网关协议和外部网关协议

■ 根据适用范围的不同，可以分为IGP和EGP

- **内部网关协议（IGP）**：用于自治系统（AS）内部的路由选择。常用的IGP协议包括RIP（路由信息协议）、OSPF（开放最短路径优先协议）等。这些协议允许自治系统内的路由器相互通信，共享路由信息，以确保数据包在自治系统内部的高效传输。
- **外部网关协议（EGP）**：用于不同自治系统之间的路由选择。目前，BGP（边界网关协议）是最常用的EGP协议。BGP允许不同自治系统的路由器相互通信，共享路由信息，以实现跨自治系统的数据包传输。路由器通常使用相同的路由选择协议和共同的度量，以确定分组在该AS内的路由。

三、 动态路由协议—— RIP

—— 路由表的生成、更新，基础配置

动态路由协议——RIP

□ RIP的基本概念

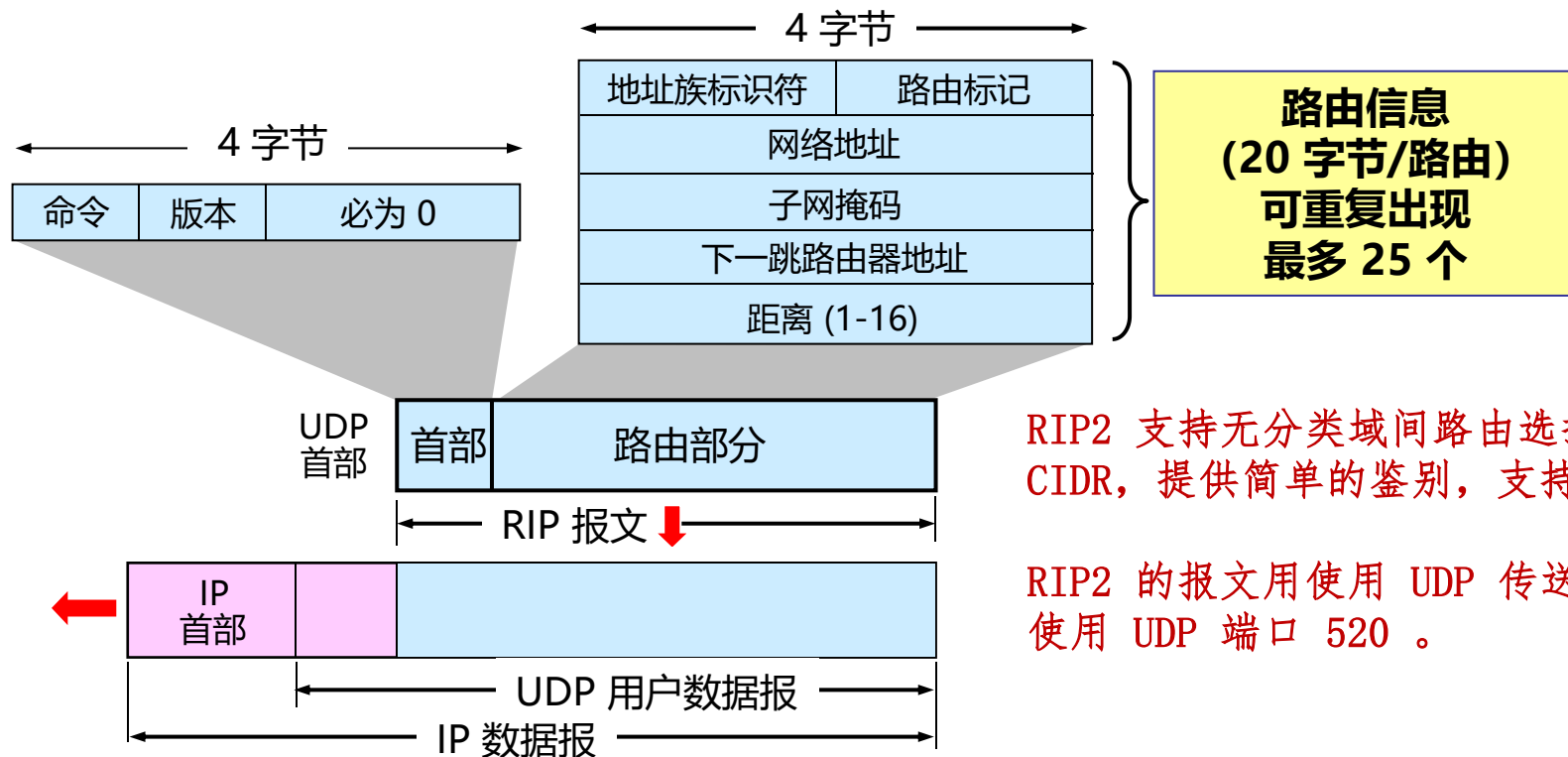
- 英文全称：Routing Information Protocol，路由信息协议
- RIP 是内部网关协议 IGP中最先得到广泛使用的协议。
- RIP 采用距离矢量路由算法，RIP协议中的“距离”也称为“跳数”，每经过一个路由器，跳数就加1。RIP认为一个好的路由就是它通过的路由器的数目少。RIP选择一个具有最少路由器的路由（即最短路由），哪怕还存在另一条高速（低时延）但路由器较多的路由。
- RIP允许一条路径最多只能包含15个路由器，当“距离”最大值为16时，即相当于不可达。
- RIP协议现在有两个版本，RIPv1和RIPv2。和RIPv1相比，RIPv2支持路由汇总（聚合）、无类别域间路由（CIDR）和可变长子网掩码（VLSM）

动态路由协议——RIP

□ RIP 协议的报文格式:

- 路由器通过RIP协议报文，携带路由信息，用以和相邻路由器进行交换。
- 现在较新的是RIP2版本，支持无分类域间路由选择（CIDR）
- 报文组成：首部（4个字节）+ 路由部分
- 路由部分：由若干个路由记录信息组成，每个路由记录信息共 20 个字节。
 - 地址族标识符（又称为地址类别）字段用来标志所使用的地址协议。
 - 路由标记填入自治系统的号码。
 - 后面为具体路由，指出某个网络地址、该网络的子网掩码、下一跳路由器地址以及到此网络的距离。
 - 一个 RIP 报文最多可包括 25 个路由，因而 RIP 报文的最大长度是
 $4（首部）+20（每个路由信息的长度）\times 25（最多包括25个路由）=504$ 字节
如超过，必须再用一个 RIP 报文来传送。

【RIP2协议的报文格式】



RIP2 支持无分类域间路由选择
CIDR，提供简单的鉴别，支持多播。

RIP2 的报文用使用 UDP 传送
使用 UDP 端口 520 。

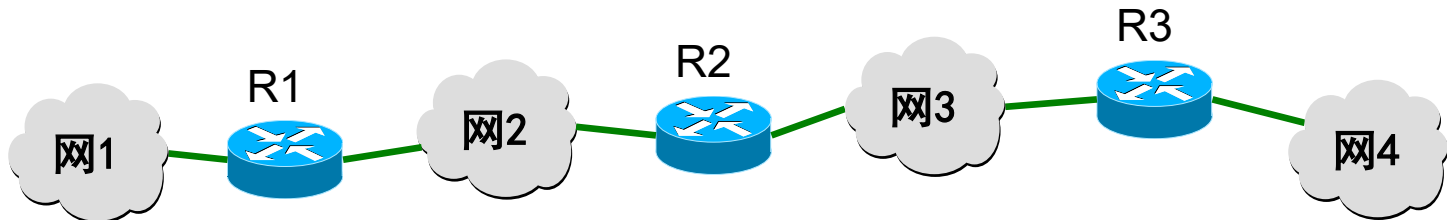
动态路由协议——RIP

□ 配置RIP协议的路由器的路由表建立过程

- 首先：路由器在刚刚开始工作时，路由表是空的。
- 然后：经过接口配置，路由器得出到直连网络的距离，距离定义为1。
- 接着：每一个路由器只和数目非常有限的相邻路由器交换并更新路由信息。
交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
 - 周期性更新：按固定的时间间隔交换路由信息，例如，每隔30秒；
 - 触发更新：除了周期性更新外，当网络拓扑发生变化时（如链路故障、新路由器加入等），RIP路由器还会立即发送触发更新，通知邻居路由器网络拓扑的变化。
触发更新只包含发生变化的路由条目，以减少带宽和资源的占用。
- 以后：经过若干次更新后，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。
- 举例

◆ 举例：路由表建立过程

t0时刻



R1的路由信息

| 目的 | 下一跳 | 跳数 |
|----|-----|----|
| 网1 | 直连 | 1 |
| 网2 | 直连 | 1 |
| | | |

R2的路由信息

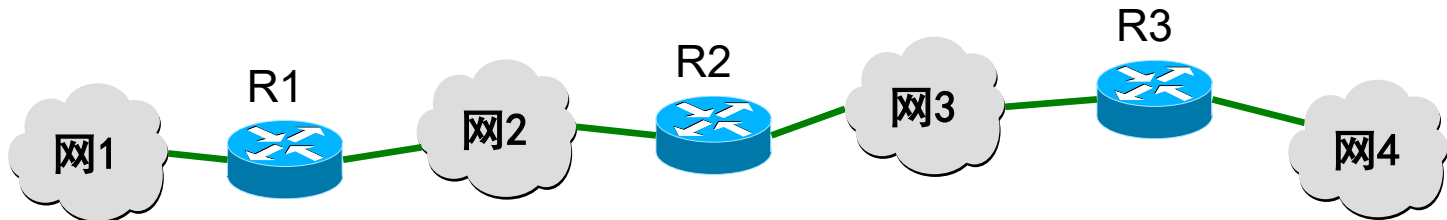
| 目的 | 下一跳 | 跳数 |
|----|-----|----|
| 网2 | 直连 | 1 |
| 网3 | 直连 | 1 |
| | | |

R3的路由信息

| 目的 | 下一跳 | 跳数 |
|----|-----|----|
| 网3 | 直连 | 1 |
| 网4 | 直连 | 1 |
| | | |

◆ 举例：路由表建立过程

t1时刻



R1的路由信息

| 目的 | 下一跳 | 跳数 |
|---------------|---------------|--------------|
| 网1 | 直连 | 1 |
| 网2 | 直连 | 1 |
| 网2 | R2 | 2 |
| 网3 | R2 | 2 |

R2的路由信息

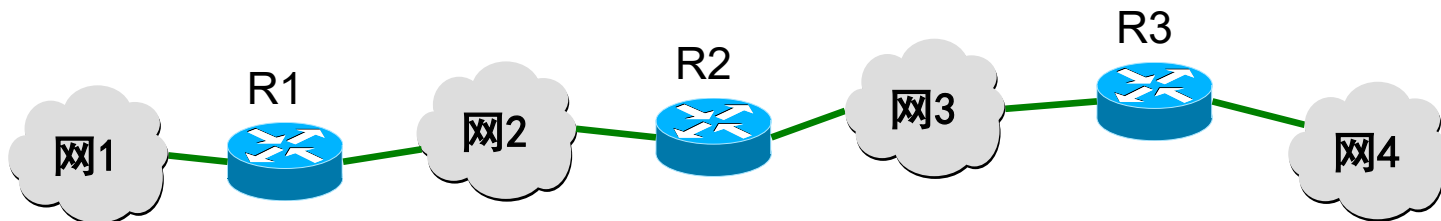
| 目的 | 下一跳 | 跳数 |
|---------------|---------------|--------------|
| 网2 | 直连 | 1 |
| 网3 | 直连 | 1 |
| 网1 | R1 | 2 |
| 网2 | R1 | 2 |
| 网3 | R3 | 2 |
| 网4 | R3 | 2 |

R3的路由信息

| 目的 | 下一跳 | 跳数 |
|---------------|---------------|--------------|
| 网3 | 直连 | 1 |
| 网4 | 直连 | 1 |
| 网2 | R2 | 2 |
| 网3 | R2 | 2 |

◆ 举例：路由表建立过程

t2时刻



R1的路由信息

| 目的 | 下一跳 | 跳数 |
|----|-----|----|
| 网1 | 直连 | 1 |
| 网2 | 直连 | 1 |
| 网3 | R2 | 2 |
| 网4 | R2 | 3 |

R2的路由信息

| 目的 | 下一跳 | 跳数 |
|----|-----|----|
| 网2 | 直连 | 1 |
| 网3 | 直连 | 1 |
| 网1 | R1 | 2 |
| 网4 | R3 | 2 |

R3的路由信息

| 目的 | 下一跳 | 跳数 |
|----|-----|----|
| 网3 | 直连 | 1 |
| 网4 | 直连 | 1 |
| 网2 | R2 | 2 |
| 网1 | R2 | 3 |

路由表更新完毕
(即收敛完毕)

动态路由协议——RIP

□ 路由器更新路由表的具体算法

■ 本路由器是Y，假设相邻路由器（地址为 X）发送过来 RIP 报文，则Y：

1. 修改 RIP 报文中的所有项目（即路由）：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。
2. 对修改后的 RIP 报文中的每一个项目，重复以下步骤：
 - ① 若路由表中没有目的网络N，则把该项目添加到路由表中。否则
 - ② 若路由表中网络 N 的下一跳也是X，则用收到的项目替换原路由表中的项目。否则
 - ③ 若收到项目中的距离小于路由表中的距离，则用收到项目更新原路由表中的项目。否则
 - ④ 什么也不做。
3. 若 3 分钟还未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路由器，即将距离置为 16（表示不可达）。
4. 返回。

【RIP协议举例】已知路由器 R6 的路由表如表1所示。现在收到相邻路由器 R4 发来的路由更新信息（表2）。试填写更新后的R6 的路由表（见表3）

表 1：路由器 R6的路由表

| 目的网络 | 距离 | 下一跳路由器 |
|------|-----|----------------|
| Net2 | 3 | R ₄ |
| Net3 | 4 | R ₅ |
| ... | ... | ... |

表 2： R4 发来的路由更新信息

| 目的网络 | 距离 | 下一跳路由器 |
|------|----|----------------|
| Net1 | 3 | R ₁ |
| Net2 | 4 | R ₂ |
| Net3 | 1 | 直接交付 |

计算
更新

① 距离+1，修改下一跳地址

表 3：路由器 R6 更新后的路由表

| 目的网络 | 距离 | 下一跳路由器 |
|------|-----|----------------|
| Net1 | 4 | R ₄ |
| Net2 | 5 | R ₄ |
| Net3 | 2 | R ₄ |
| ... | ... | ... |

表 2'：修改后的表2

| 目的网络 | 距离 | 下一跳路由器 |
|------|----|----------------|
| Net1 | 4 | R ₄ |
| Net2 | 5 | R ₄ |
| Net3 | 2 | R ₄ |

动态路由协议——RIP

□ RIP的收敛机制

■ 收敛的基本概念

- 收敛：是指在网络拓扑发生变化后，路由协议使所有路由器都能及时感知到这种变化，并重新计算路由，以达到新的稳定状态的过程。
- 收敛时间：从网络拓扑发生变化到网络中所有路由器都知道这个变化并重新计算路由的时间。收敛时间越短，网络的稳定性和性能就越好。

动态路由协议——RIP

□ RIP的收敛机制

RIP协议通过以下方式实现路由收敛

1. 周期性更新：

- RIP协议采用周期性更新的方式，默认情况下，每30秒向邻居路由器发送一次完整的路由表更新。这种更新方式可以确保在网络拓扑稳定时，路由器能够定期交换路由信息，保持路由表的一致性。
- 需要注意的是，RIP的更新周期并不是严格的30秒，发送前会附加一段随机延迟（0到5秒），以避免大量路由器同时发送路由表而造成网络拥塞。

动态路由协议——RIP

□ RIP的收敛机制

2. 触发更新：

- 当网络拓扑发生变化时（如链路故障、路由器宕机等），RIP协议会立即触发更新，将变化的信息广播给所有直连的邻居路由器。这种更新方式可以加速收敛过程，使路由器更快地感知到网络拓扑的变化。
- 触发更新消息只包含更新的路由条目，以减少带宽和资源占用。

动态路由协议——RIP

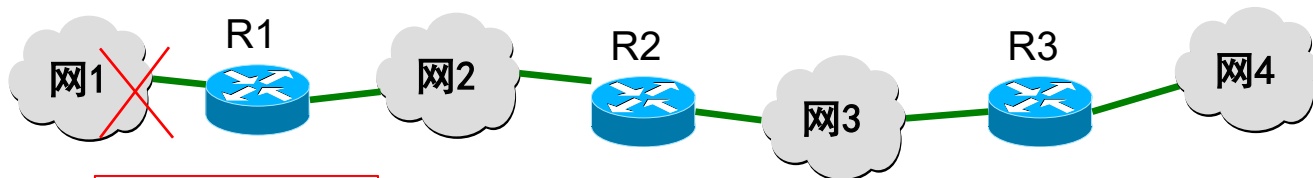
□ RIP的收敛机制

3. 防止环路的问题

- 在RIP的收敛机制中，还存在一个特殊的问题：由于每台路由器不能同时或接近同时地完成路由表更新，因此，有可能出现路由环路，从而造成“坏消息传播的慢”，进而影响了正常的收敛过程（使收敛时间过长）。

- 举例

◆ 举例：“路由环路”造成“坏消息传播的慢”



更新自己 网1, 16, -

更新 网1, 3, R2 →

更新 网1, 5, R2 →

更新 →

更新 网1, 15, R2 →

更新 网1, 17, R2 →

.....

← 网1, 2, R1

← 网1, 4, R1

← 网1, 6, R1

←

← 网1, 16, R1

.....

假设在R1告知之前，R2先告知R1

更新

更新

更新

更新

“坏消息传播的慢”

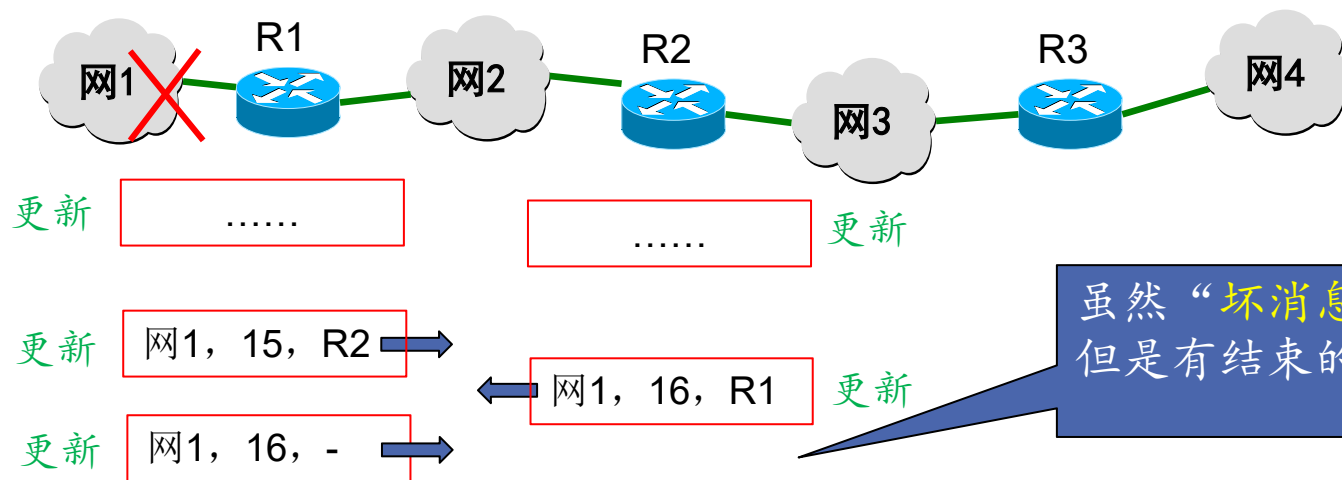
这种情况有可能持续下去，造成无法正常收敛！

动态路由协议——RIP

□ RIP的收敛机制

3. 防止环路（措施）

- **措施1：最大跳数限制。** RIP 协议规定，一条路由的最大跳数不能超过15。如果某条路由的跳数达到16，RIP 就认为该路由是不可达的，并将其从路由表中删除。这个机制有助于防止路由环路在全网范围内无限循环。

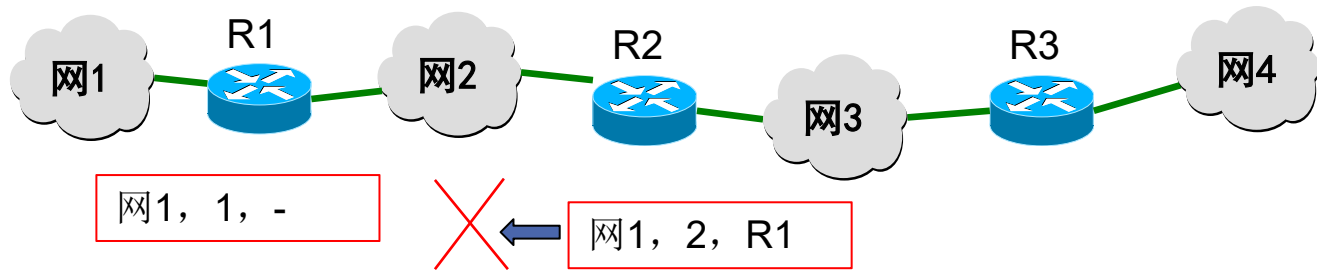


动态路由协议——RIP

□ RIP的收敛机制

3. 防止环路（措施）

- **措施2：水平分割。** 水平分割是一种防止路由信息回流到原始发送者的技术。RIP路由器默认不会向它学习到某条路由的接口发送该路由的更新信息。这有助于减少路由信息的冗余和环路的可能性。



从R1获取到的路由信息，
就不要再送回到R1了！

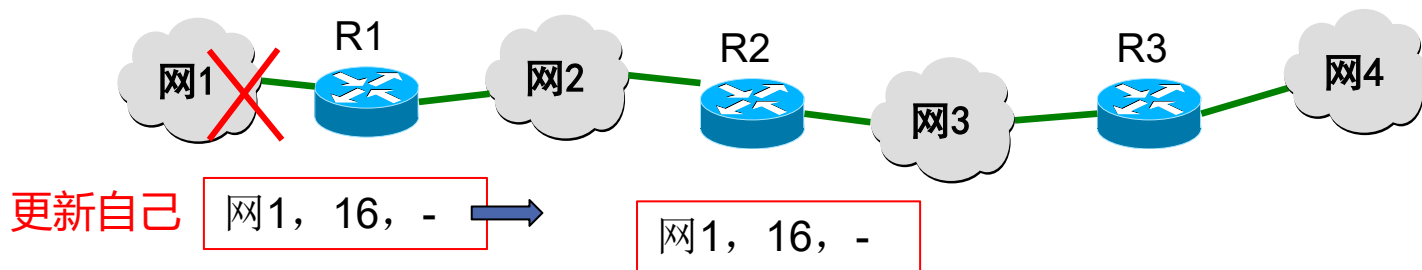
动态路由协议——RIP

□ RIP的收敛机制

3. 防止环路（措施）

- **措施3：毒性反转。**当RIP 路由器检测到某条路由不可达时（如通过其他机制得知该路由的下一跳已经故障），它不会立即从路由表中删除该路径信息，而是将该路径的度量值（如跳数）设置为一个特殊的值16（在RIP中为16，表示不可达），并将此信息广播给相邻的路由器。这样，邻居路由器就会知道该路由已经不可达，并立即相应地更新自己的路由表。
- 毒性反转机制有助于加速路由信息的传播和收敛过程。

◆ 举例：毒性反转加快网络收敛



不采用“毒性反转”：R1检测到网1不可达，直接从自己的路由表中删除到达网1的路由记录。接下来，R1传递给R2的路由信息中，不包含网1的信息，因此，R2中到达网1的信息不会改变。R2以为还可以通过R1到达网1，从而引起错误转发，形成网络震荡。

采用“毒性反转”：R1把毒性信息及时发给R2，使得R1和R2都及时更新自己的路由表，有助于快速收敛。

动态路由协议——RIP

▣ RIP的配置（以华为路由器为例） 【结合实验自主学习】

1. 配置接口IP地址

- 首先，需要在路由器上配置各接口的IP地址，确保路由器之间能够相互通信。这通常包括物理接口（如GigabitEthernet、Serial等）和逻辑接口（如LoopBack）的IP地址配置。

2. 启动RIP进程

- 进入系统视图。
- 执行rip [process-id]命令启动RIP进程，并进入RIP视图。如果不指定process-id，则默认取值为1。process-id在同一网络中的不同路由器上必须相同，但在不同网络中可以不同。

动态路由协议——RIP

▣ RIP的配置（以华为路由器为例）

3. 宣告网段信息

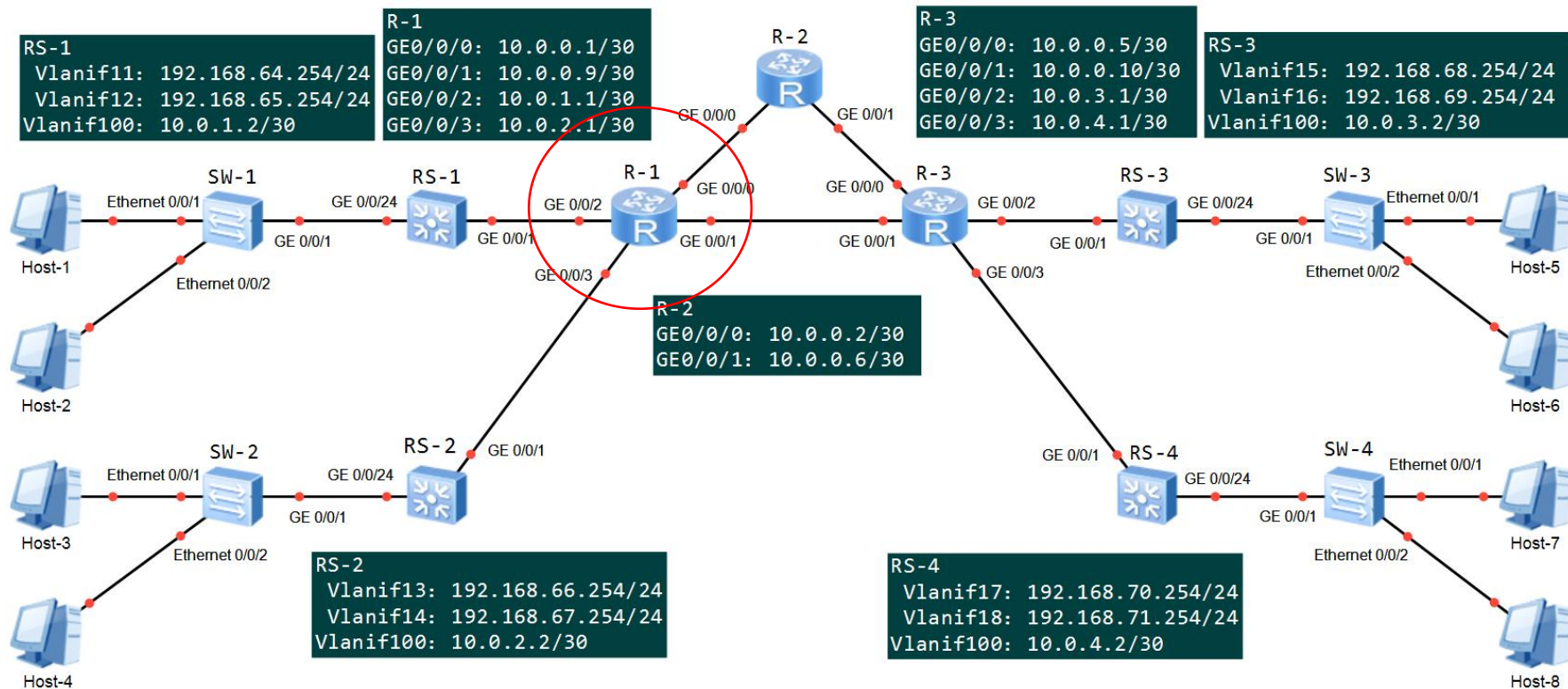
- 在RIP视图中，通过`network network-address`命令宣告路由器直连的网段。
- 这里的`network-address`必须是自然网段的网络地址（按照分类IP地址区分网段，例如10.0.1.0/24和10.1.0.0/24会被认为是一个自然网段，即10.0.0.0/8），且路由器上至少有一个接口属于该网段。RIP路由器只会将宣告的网段信息发送给邻居路由器。

4. 配置RIP版本（可选）

- RIP有两个版本：RIPv1和RIPv2。RIPv2相比RIPv1提供了更多的功能，如无类别域间路由（CIDR）支持、认证机制等。如果需要使用RIPv2，可以在RIP视图中通过`version 2`命令来指定。

■ 举例

◆ 举例：RIP协议的配置



◆ 举例：RIP协议的配置

[R-1]配置接口IP地址（见图）

//创建RIP进程1

[R-1]rip 1

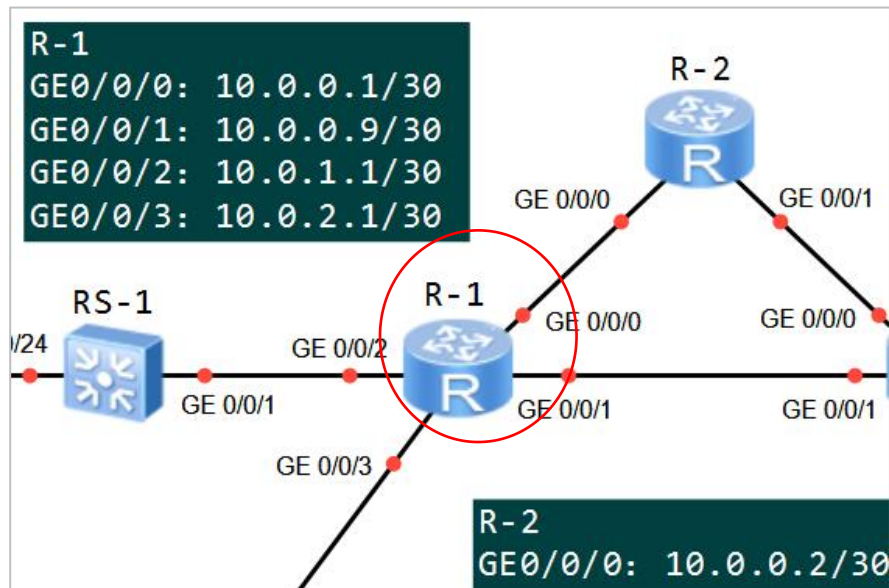
//启用RIP版本2

[R-1-rip-1]version 2

//宣告直连网络

[R-1-rip-1]network 10.0.0.0

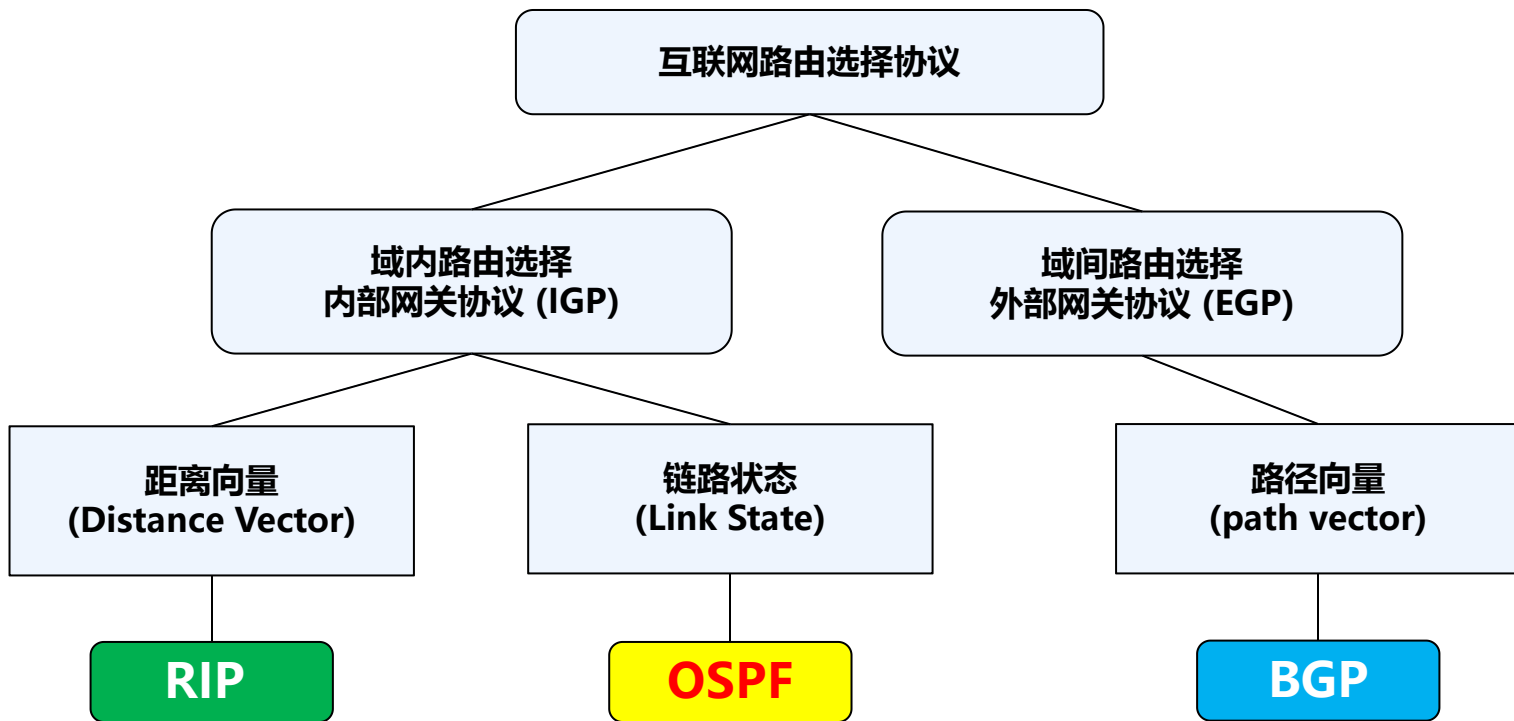
[R-1-rip-1]quit



- 注意，此处R-1有四个直连网段，分别是10.0.0.1/30、10.0.0.8/30、10.0.1.0/30、10.0.2.0/30，但是，对于RIP协议来讲，路由器只能宣告自然网段的网络地址，即只宣告10.0.0.0。
- 但是，由于启用了RIPv2版本，支持CIDR，并且每个接口都配置了IP地址，所以会生成4条直连路由，可通过display命令查看。

四、动态路由协议—— OSPF

动态路由协议——OSPF



动态路由协议——OSPF

□ 开放最短路径优先（Open Shortest Path First, OSPF）

- 由于RIP协议主要用于小型网络，在20世纪80年代中期就已不能适应大规模异构网络的互连，一种互连功能更强大的路由协议——OSPF，就随之产生了。
- OSPF只是协议名字，并不表示其他路由选择协议不是“最短路径优先”。实际上，所有的在自治系统内部使用的路由选择协议都是要寻找最短路径的。
- OSPF协议最主要的特征就是使用链路状态路由算法，而不是像RIP那样的距离矢量路由算法。

动态路由协议——OSPF

▣ OSPF的特点（和RIP对比）

- **和谁交换路由信息：**采用洪泛法（flooding），路由器向所有相邻路由器发送信息，而每一个相邻路由器再将此信息发往其所有的相邻路由器，最终本自治系统中所有路由器都得到了这个信息的一个副本。（RIP仅仅向自己相邻的几个路由器发送信息）
- **交换什么信息：**发送的是与本路由器相邻的所有路由器的**链路状态**。链路状态用来说明本路由器都和哪些路由器相邻，以及该链路的度量（metric）。
 - RIP协议中，本路由器向相邻的路由器发送的是自己的路由表（到各网络的距离和下一跳路由）
- **何时交换：**当链路状态发生变化或每隔一段时间（如30分钟），路由器才用洪泛法向所有路由器发送此信息。（与RIP相似）

动态路由协议——OSPF

▣ OSPF的特点（和RIP对比）

- **最终结果如何：**OSPF会构建起**链路状态数据库（Link-state Database）**，由于各路由器之间频繁地交换链路状态信息，因此所有的路由器最终都能建立一个链路状态数据库，这个数据库实际上就是全网的拓扑结构图，它在全网范围内是一致的（这称为链路状态数据库的同步）。
 - 所以，配置OSPF的每个路由器都知道全网共有多少个路由器，以及哪些路由器是相连的，其代价是多少。即各路由器的链路状态数据库是相同的。**每个路由器使用链路状态数据库中的数据构造自己的路由表**
 - 再看RIP，RIP中的每一个路由器虽然知道到达所有的网络的距离以及下一跳路由器，但却不知道全网的拓扑结构（只有到了下一跳，才能知道接下来怎样走）。

动态路由协议——OSPF

▣ OSPF的区域化设计

- 这也是OSPF和RIP的一个重要区别！
- OSPF支持层次化设计，可以通过区域（Area）的概念来组织大型网络。这有助于减少路由器的内存消耗和计算负担，因为路由器只需要了解它们所在区域内部的详细信息，而不需要知道整个网络的信息。
- 接下来谈谈OSPF区域

动态路由协议——OSPF

▣ OSPF区域 (area)

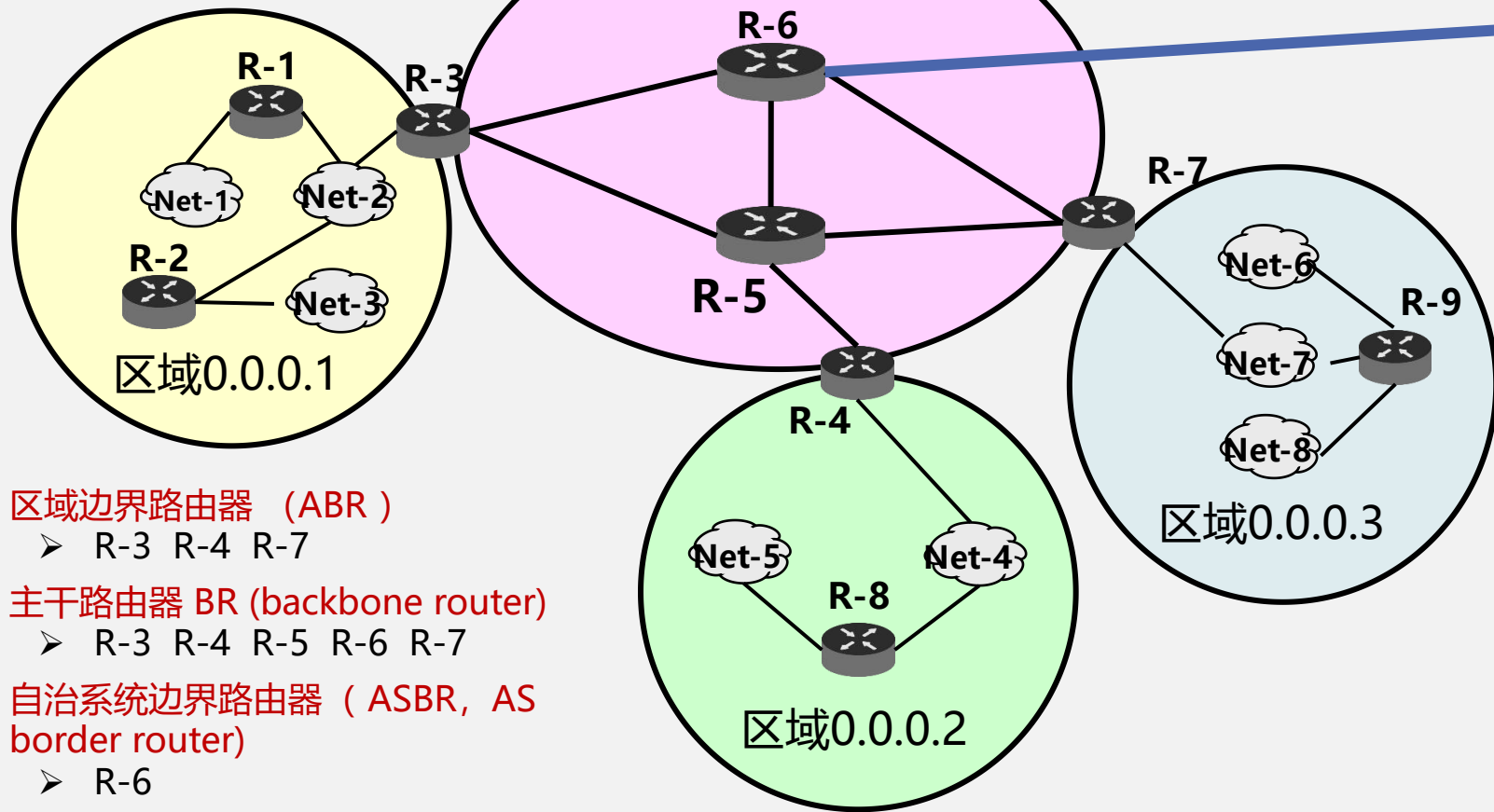
- 为了使OSPF能够用于规模很大的网络，OSPF 将一个自治系统再划分为若干个更小的范围，叫作区域。
- 划分区域的好处就是**将**利用洪泛法交换链路状态信息的**范围**局限于每一个区域而不是整个的自治系统，这就减少了整个网络上的通信量。
- 所以，在一个区域内部的路由器只需要知道本区域的完整网络拓扑，而不需知道其他区域的网络拓扑的情况。
- 那么，如何实现不同区域之间的路由通信呢？

动态路由协议——OSPF

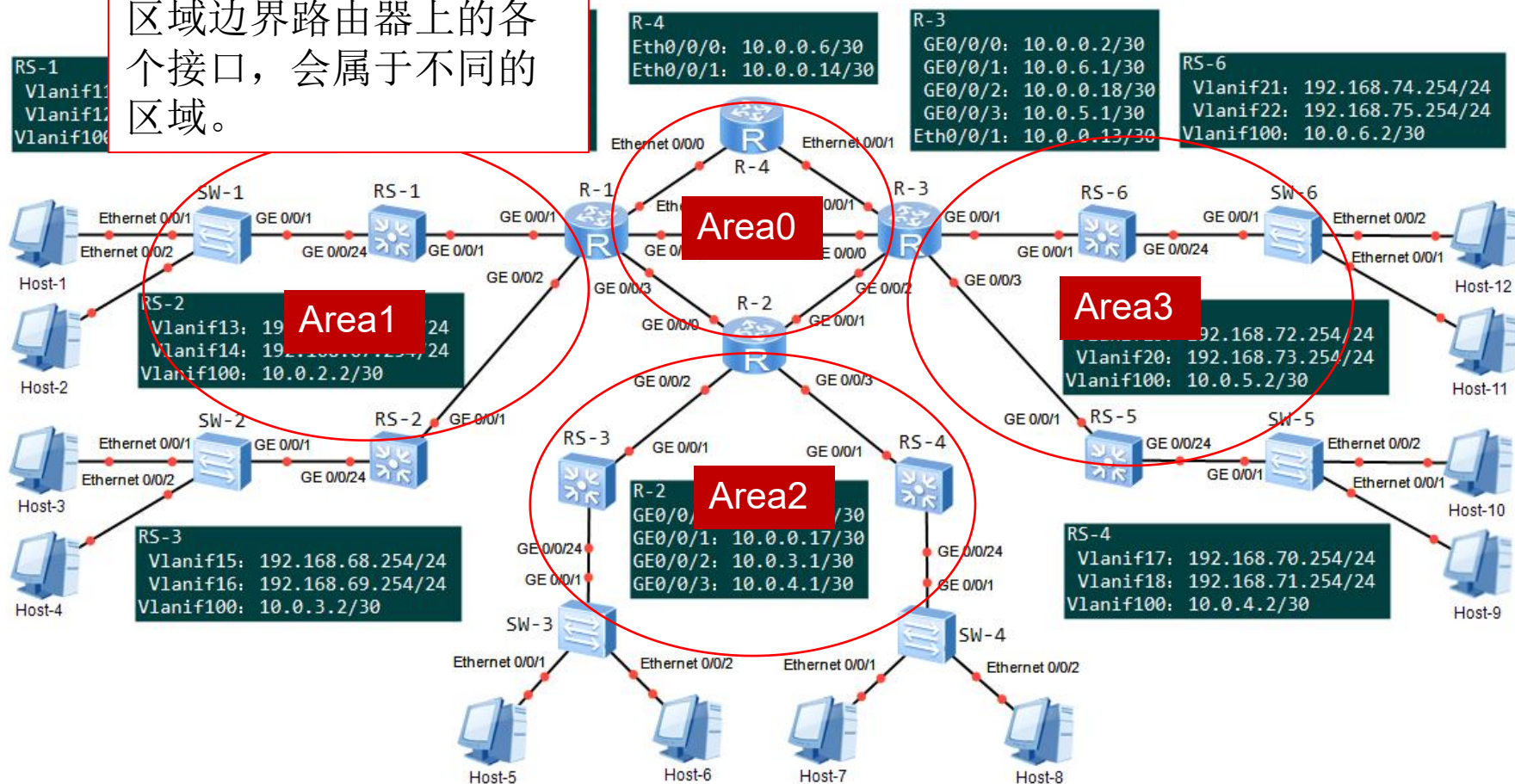
□ OSPF区域 (area)

- 为了使每个区域能够和本区域以外的区域进行通信，OSPF 使用了**主干区域** (backbone area) 和**普通区域**的概念。
- 在规划一个AS中的区域时，OSPF要设置一个主干区域和若干普通区域，并且，所有普通区域必须与主干区域直接相连。
- 每个区域有一个或若干个边界路由器，边界路由器将本区域中的路由信息进行概括后，只将基本信息发给其他区域（例如**普通2→主干→普通3**），从而既实现了不同区域之间路由信息交换，又减少了区域之间的通信量。
- 每一个区域都有一个 32 位的区域标识符，用点分十进制表示。例如主干区域的标识符规定为0.0.0.0。
- **举例**

自治系统 AS100



区域边界路由器上的各个接口，会属于不同的区域。



➤ 例1：在RS-1上配置OSPF

```
[RS-1]ospf 1
```

//创建并进入**OSPF**区域，此处是区域1（因为**RS-1**的接口都属于**area 1**）

```
[RS-1-ospf-1]area 1
```

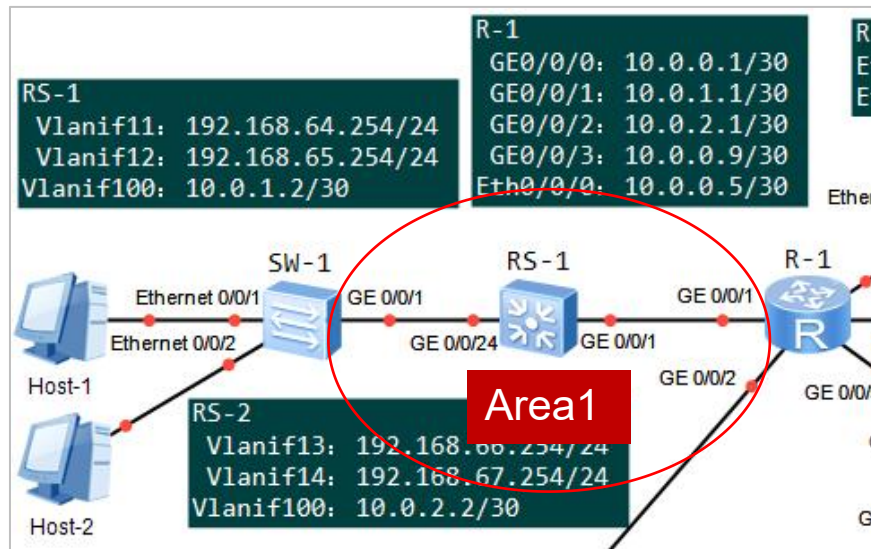
//宣告当前区域中的直连网络，注意需要配置子网掩码（与**RIP**不同）

```
[RS-1-ospf-1-area-0.0.0.1]network 192.168.64.0 0.0.0.255
```

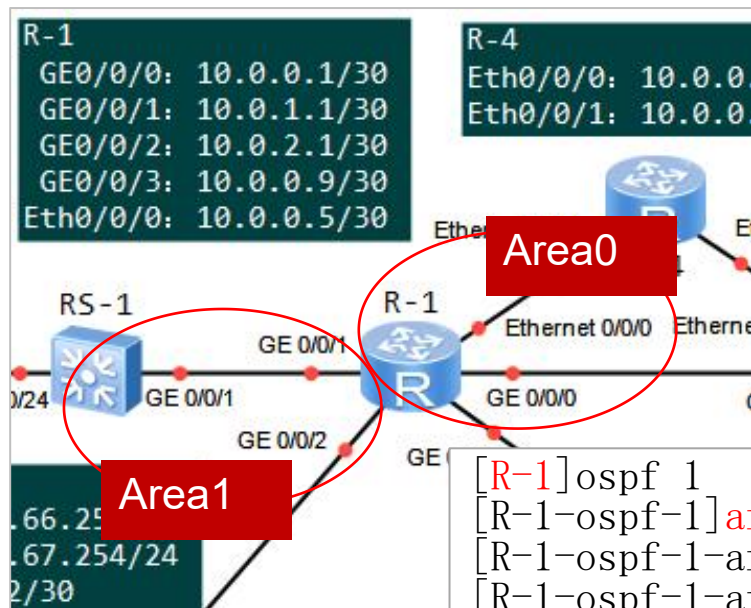
```
[RS-1-ospf-1-area-0.0.0.1]network 192.168.65.0 0.0.0.255
```

```
[RS-1-ospf-1-area-0.0.0.1]network 10.0.1.0 0.0.0.3
```

注意，ospf在宣告直连网络时，需要加上子网掩码，且以反码形式



► 例2：在R-1上配置OSPF



注意，R-1是区域边界路由器，有5个直连网络，其中3个属于area0, 2个属于area1，需要分别宣告。

```

[R-1]ospf 1
[R-1-ospf-1]area 0
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.0 0.0.0.3
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.4 0.0.0.3
[R-1-ospf-1-area-0.0.0.0]network 10.0.0.8 0.0.0.3
[R-1-ospf-1-area-0.0.0.0]quit
[R-1-ospf-1]area 1
[R-1-ospf-1-area-0.0.0.1]network 10.0.1.0 0.0.0.3
[R-1-ospf-1-area-0.0.0.1]network 10.0.2.0 0.0.0.3
[R-1-ospf-1-area-0.0.0.1]quit
  
```

动态路由协议——OSPF

□ OSPF 的五种分组（消息）类型：

■ 类型1：Hello报文（问候信息）：

- 周期性发送，用来发现和维护OSPF邻居关系（可达性）

■ 类型2：Database Description分组（链路状态数据库描述信息）

- 描述本地LSDB的摘要信息，用于两台设备进行数据库同步

■ 类型3：Link State Request分组（链路状态请求信息）。

- 用于向对方请求所需要的LSA（链路状态），设备只有在OSPF邻居双方成功交换DD报文后才会向对方发出LSR报文

■ 类型4：链路状态更新（Link State Update）分组。

- 用洪泛法对全网更新链路状态，向对方发送其所需要的LSA。

■ 类型5：链路状态确认（Link State Acknowledgment）分组。

- 用来对收到的LSA进行确认

动态路由协议——OSPF

▣ OSPF区域的基本工作流程

1. Hello协议：用于发现和维护邻居关系（确认邻站可达）。

- 相邻路由器每隔 10 秒钟要交换一次问候分组（Hello报文）。
- 若有 40 秒钟没有收到某个相邻路由器发来的问候分组，则可认为该相邻路由器是不可达的，立即修改链路状态数据库，重新计算路由表。

2. 同步链路状态数据库

- 同步：指不同路由器的链路状态数据库的内容是一样的。
- 通过DD（数据库描述信息）分组和LSR（链路状态请求）分组，使各路由器之间频繁地交换链路状态信息，因此所有的路由器最终都能建立一个同步的链路状态数据库，

动态路由协议——OSPF

▣ OSPF区域的基本工作流程

3. 泛洪（Flooding）：交换路由信息

- 当路由器链路状态发生变化，路由器就使用链路状态更新分组（LSA），采用可靠的洪泛法向全网更新链路状态。直到所有路由器都获得最新的链路状态信息。
- 为确保链路状态数据库与全网的状态保持一致，OSPF 还规定：每隔一段时间，如 30 分钟，要刷新一次数据库中的链路状态。
- OSPF 链路状态只涉及相邻路由器，与整个互联网的规模并无直接关系，因此当互联网规模很大时，OSPF 协议要比距离向量协议 RIP 好得多。

动态路由协议——OSPF

▣ OSPF区域的基本工作流程

4. 计算最短路径树：

- 使用Dijkstra算法，各路由器根据链路状态数据库中的信息来计算到所有可达网络的最短路径。

5. 生成路由表：

- 基于最短路径树，各路由器会生成自己的路由表，用于指导数据包的转发。

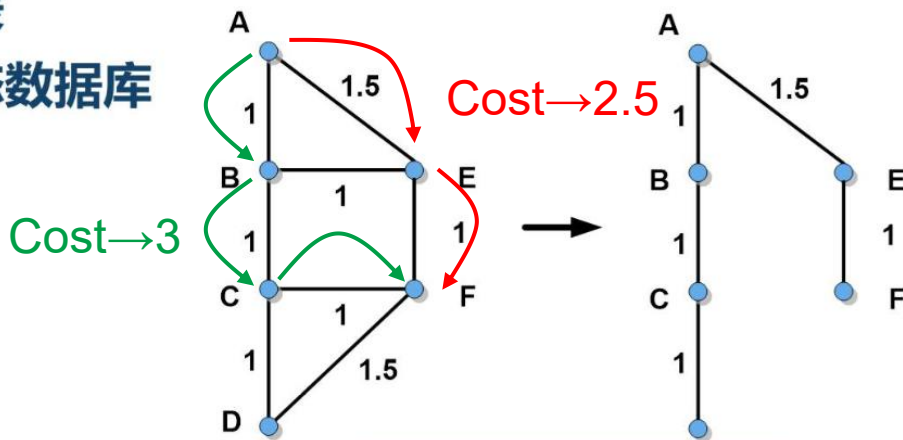
动态路由协议——OSPF

OSPF区域的基本工作流程 —— 示例（图）

邻居列表

链路状态数据库

路由表



以A为例，生成路由表的过程

OSPF五种分组

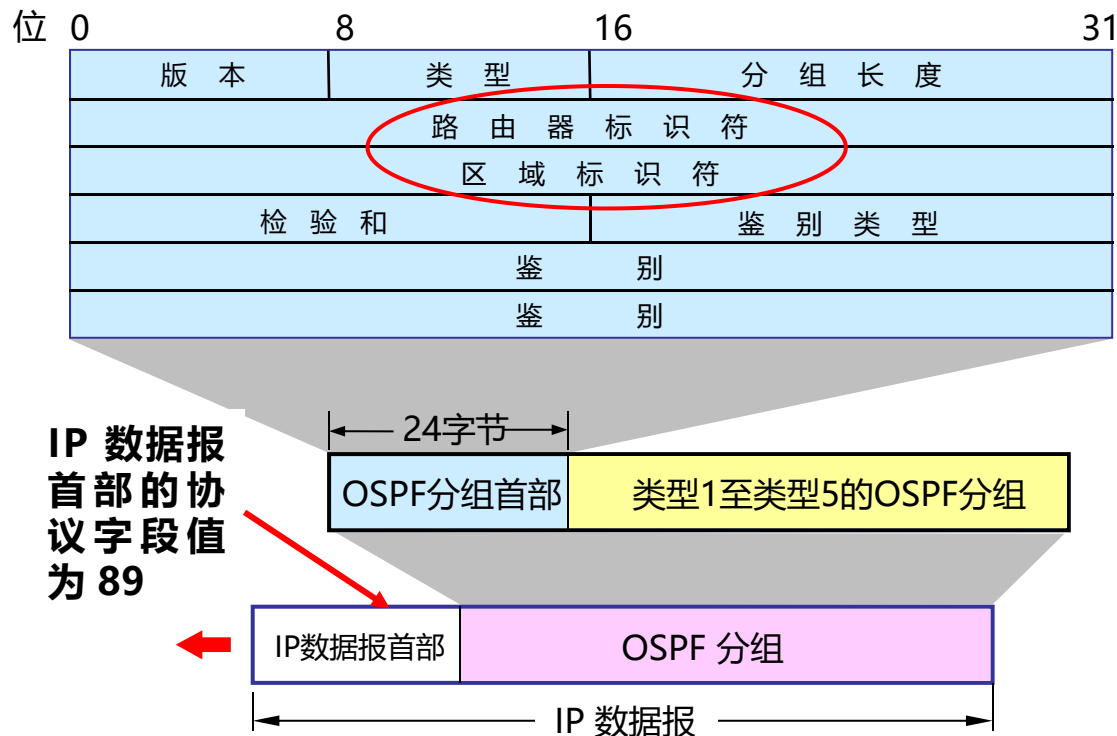


https://blog.csdn.net/weixin_51616021

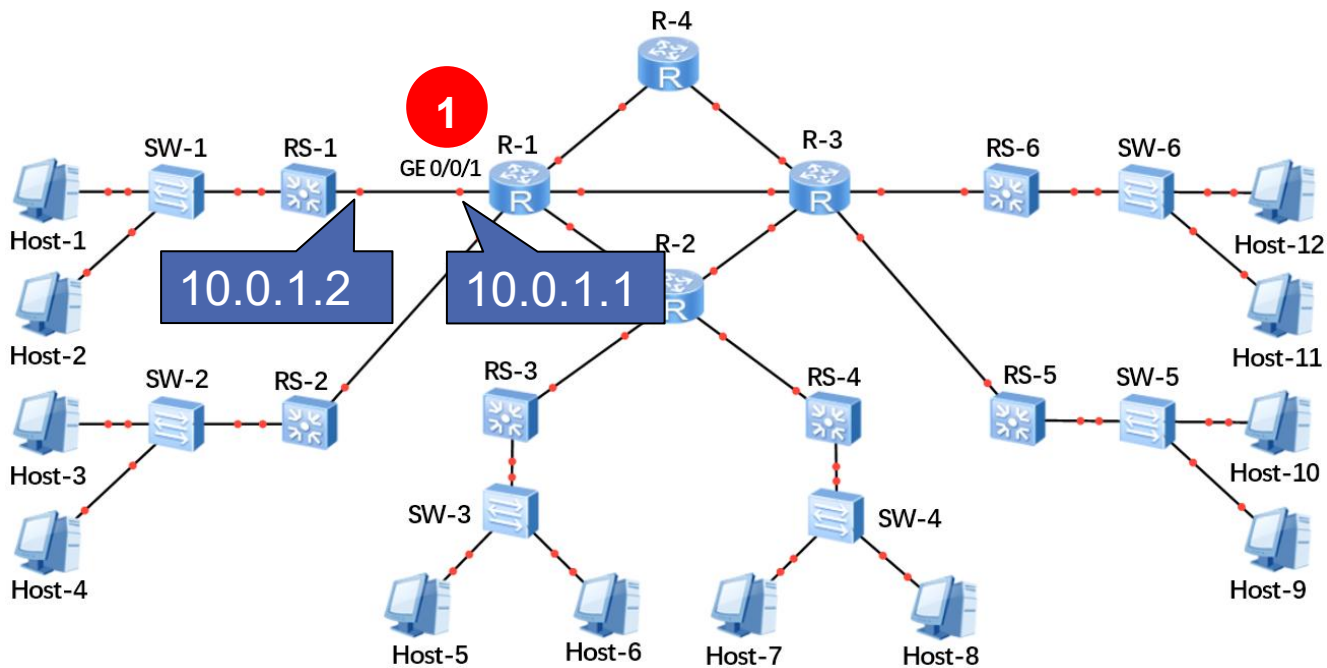
动态路由协议——OSPF

□ OSPF报文

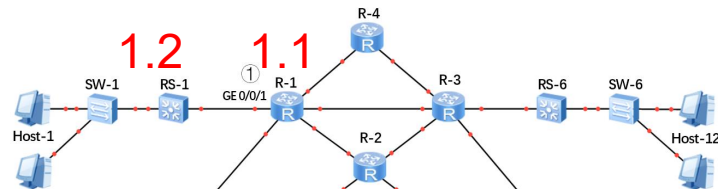
- OSPF 分组是作为IP数据报的数据部分来传送的
- OSPF 不用 UDP，而是直接用 IP 数据报传送



◆ 举例：OSPF更新链路状态



◆ 举例：OSPF更新链路状态



The figure shows a Wireshark packet capture window. The top menu bar includes: 文件(F), 编辑(E), 视图(V), 跳转(G), 捕获(C), 分析(A), 统计(S), 电话(V), 无线(W), 工具(T), 帮助(H). The toolbar contains various icons for file operations, packet selection, and analysis. The packet list pane on the left shows a single packet selected, labeled 'ospf'. The packet details pane on the right shows the following information:

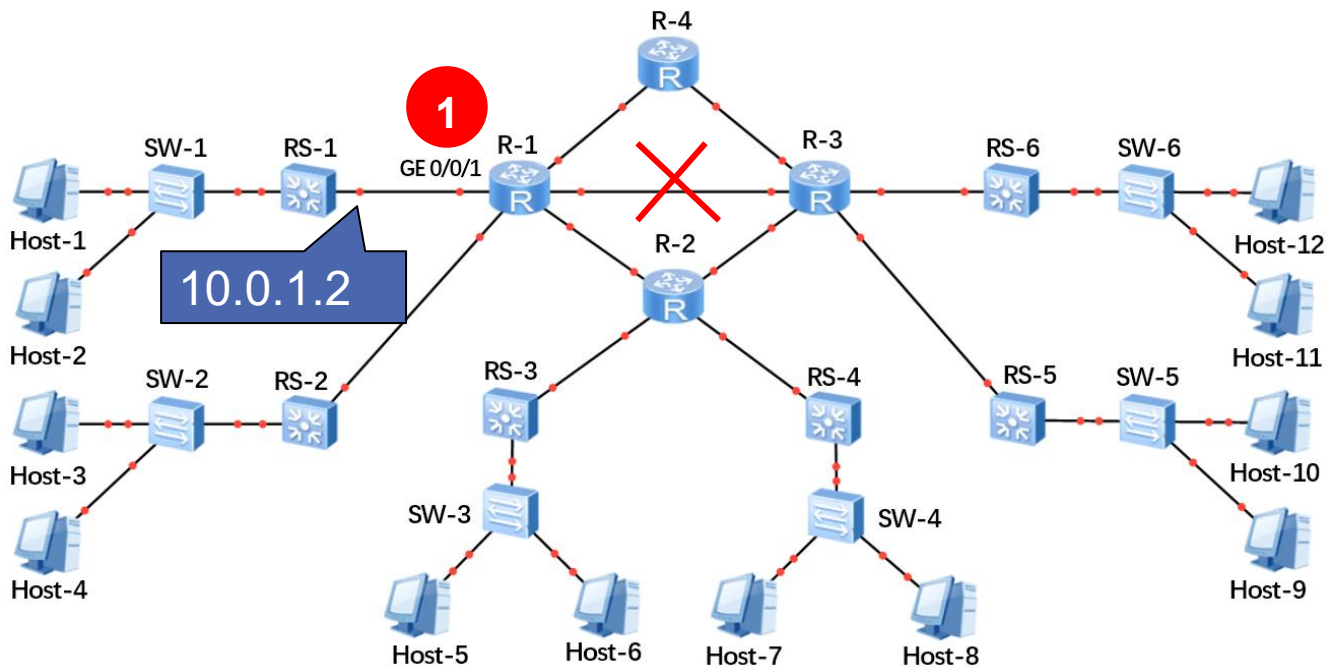
| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|-----------------|------------------|----------|---------------------|
| 2 | 1.578000 | <u>10.0.1.2</u> | <u>224.0.0.5</u> | OSPF | <u>Hello Packet</u> |
| 5 | 6.515000 | <u>10.0.1.1</u> | <u>224.0.0.5</u> | OSPF | <u>Hello Packet</u> |
| 8 | 11.125000 | 10.0.1.2 | 224.0.0.5 | OSPF | Hello Packet |
| 11 | 15.984000 | 10.0.1.1 | 224.0.0.5 | OSPF | Hello Packet |

Below the packet list, the packet details for the selected packet (Frame 2) are shown:

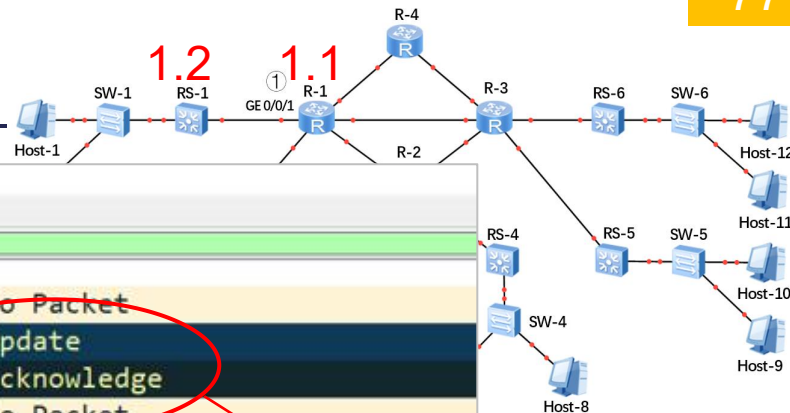
- > Frame 2: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
- > Ethernet II, Src: HuaweiTe_41:5f:7d (4c:1f:cc:41:5f:7d), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
- > Internet Protocol Version 4, Src: 10.0.1.2 (10.0.1.2), Dst: ospf-all.mcast.net (224.0.0.5)
- > Open Shortest Path First

周期性发送，用来发现和**维护OSPF邻居关系**（可达性）

◆ 举例：OSPF更新链路状态



◆ 举例：OSPF更新链路状态



| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|----------|-------------|----------|----------------|
| 33 | 28.656000 | 10.0.1.1 | 224.0.0.5 | OSPF | Hello Packet |
| 35 | 30.093000 | 10.0.1.1 | 224.0.0.5 | OSPF | LS Update |
| 38 | 31.015000 | 10.0.1.2 | 224.0.0.5 | OSPF | LS Acknowledge |
| 49 | 36.156000 | 10.0.1.2 | 224.0.0.5 | OSPF | Hello Packet |

> Ethernet II, Src: HuaweiTe_81:31:65 (54:89:98:81:31:65), Dst: IPv4mcast_05 (01:00:
 > Internet Protocol Version 4, Src: 10.0.1.1 (10.0.1.1), Dst: ospf-all.mcast.net (224.0.0.5)
 > Open Shortest Path First
 > OSPF Header
 > LS Update Packet
 Number of LSAs: 7
 > LSA-type 3 (Summary-LSA (IP network)), len 28
 > LSA-type 3 (Summary-LSA (IP network)), len 28
 > LSA-type 3 (Summary-LSA (IP network)), len 28
 > LSA-type 3 (Summary-LSA (IP network)), len 28
 > LSA-type 3 (Summary-LSA (IP network)), len 28
 > LSA-type 3 (Summary-LSA (IP network)), len 28
 > LSA-type 3 (Summary-LSA (IP network)), len 28

LSU: 链路状态数据包
 LSAck: 链路状态确认数据包
 LSU报文中的LSA信息
 LSA: 链路状态数据包

LSU: 链路状态更新分组
LSAck: 链路状态确认分组

LSU报文中的LSA信息

LSA: 链路状态数据包

链路状态更新与链路状态确认分组

第7章 动态路由协议

完