

Quagga 使用手册

路由协议工具

PM0010020003 V1.01 Date: 2018/03/21

产品使用手册

类别	内容
关键词	Quagga、路由协议
摘 要	SylixOS 系统中 Quagga 的使用

修订历史

版本	日期	原因
V1.00	2018/03/21	创建文档

目 录

第 1 章 大规模网络路由技术概述	3
1.1 三层网络模型与路由技术	3
1.2 路由器在各层中的功能	4
1.2.1 核心层路由器	4
1.2.2 汇聚层路由器	4
1.2.3 接入层路由器	4
第 2 章 路由协议基础	5
2.1 路由分类	5
2.2 静态路由应用	5
2.3 动态路由协议	5
2.3.1 路由协议分类	5
2.3.2 路由协议的工作原理	6
2.4 路由选择原则	7
2.5 路由协议比较	8
第 3 章 RIP	10
3.1 RIP 协议概述	10
3.1.1 RIP 协议特点	10
3.1.2 RIP 协议基本原理	11
3.2 RIP 的路由表结构	11
3.3 RIP 简单用例	12
第 4 章 OSPF	14
4.1 OSPF 协议概述	14
4.1.1 OSPF 协议特点	14
4.1.2 OSPF 协议基本原理	14
4.2 分层结构	15
4.2.1 骨干区域与非骨干区域	15
4.2.2 OSPF 路由器类型	15
4.3 OSPF 网络类型	15
4.4 OSPF 报文和封装	16
4.5 LSDB 更新	16
4.6 OSPF 虚连接	17
4.7 OSPF 简单用例	18
4.7.1 OSPF 多区域连接用例	18
第 5 章 IS-IS	23
5.1 IS-IS 概述	23
5.2 IS-IS 路由级别	24
5.3 IS-IS 与 OSPF 的区别	24
5.4 IS-IS 协议报文类型及作用	24
5.5 IS-IS 网络类型	25
5.6 IS-IS 简单用例	26
5.6.1 基础用例	26
第 6 章 BGP	28

6.1	BGP 基本介绍	28
6.2	BGP 基本术语	28
6.3	BGP 的选路规则	29
6.4	BGP 简单用例	31
6.4.1	不同路由协议下通信用例（ospf-rip）	31
6.4.2	不同路由协议通信（ospf-rip-isis）	34

第1章 大规模网络路由技术概述

1.1 三层网络模型与路由技术

一个典型的大规模网络，根据功能可以划分为接入层、汇聚层、核心层 3 层，如图 1.1 所示。各层对路由的要求有所不同，所推荐使用的路由协议也有所不同。

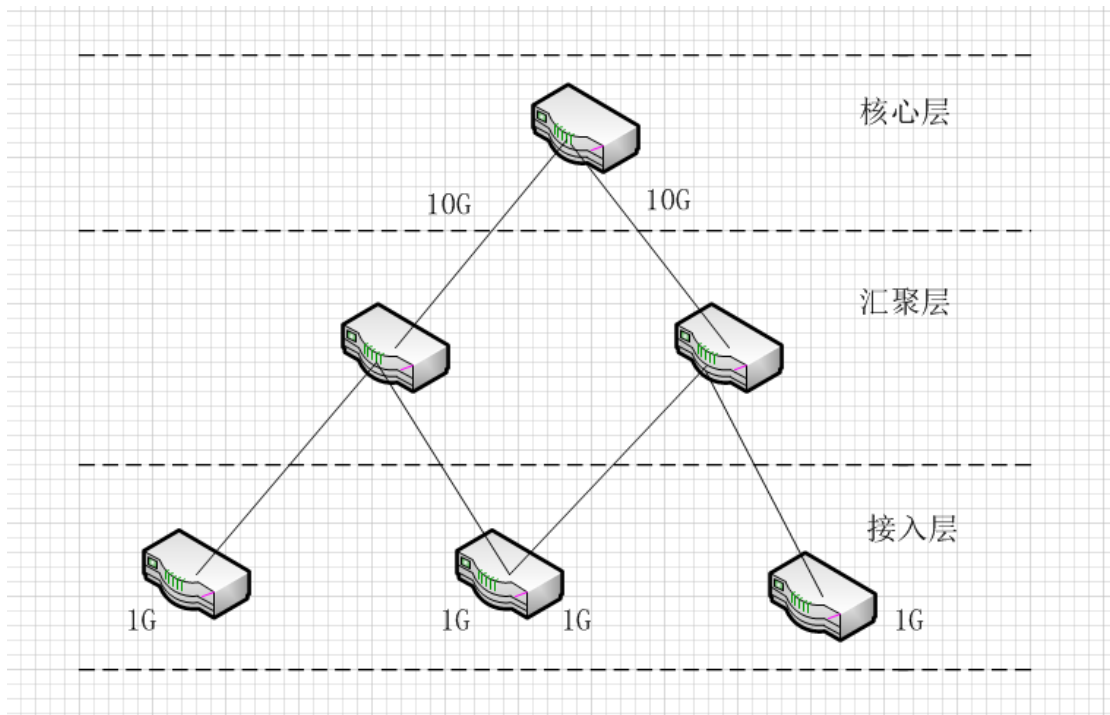


图 1.1 3 层网络模型

- **核心层：**核心层是网络的骨干，提供高速数据转发和路由快速收敛，具有较高的可靠性、稳定性和可扩展性。所以，通常核心层采用收敛速度快、扩展性好的路由协议，如 OSPF、ISIS 等。如果网络规模很大，如 ISP 网络，为了便于实现路由控制，确保高速路由转发，也会采用 BGP 协议作为核心层协议。
- **汇聚层：**汇聚层负责汇聚来自接入层的流量并执行复杂策略，所实现的路由功能包括路由聚合、路由策略、负载均衡、快速收敛等。所以，在路由层面上，汇聚层通常采用收敛速度快，支持路由聚合、负载分担，并且易于实现路由策略的路由协议，如 OSPF、ISIS 等。
- **接入层：**接入层负责提供网络的用户接入功能，所以通常接入层采用配置简单，占用系统资源少的路由协议，如 RIPv2、静态路由等。同时，也可以采用 OSPF 的 Stub 区域或者 ISIS 的 Level-1 区域，以减少区域中路由数量，并降低接入层路由变化对汇聚层的影响。

大规模路由网络通常使用 BGP 或静态路由接入 Internet。使用 BGP 接入可以更好的控制路由的发布与接收，而是用静态路由可以节省网络开销。

并不是所有的路由网络都必须划分为 3 层，层次越多，网络拓扑也就越复杂，所使用的协议可能也就越多，从而增加配置与维护的难度。

1.2 路由器在各层中的功能

1.2.1 核心层路由器

既然各层的功能不同，对相应的路由器的性能要求就有所不同。

核心层: 采用了控制控制平面和转发平面分离的系统架构具有强大的转发性能和很高的运行可靠性。这就要核心层设备本身具有强大的业务数据转发能力，且具有高速的业务接口。

1.2.2 汇聚层路由器

汇聚层: 该系列路由器产品是性能和特性均衡的产品，具有较强的转发性能、较高的运行可靠性和非常丰富的特性，如 ACL、路由过滤与策略、QoS、安全、语音等。

1.2.3 接入层路由器

接入层: 包括 MSR30、MSR20、MSR900 等系列路由器产品。MSR30、MSR20、MSR900 是接入层多业务路由器，具有非常丰富的特性，支持所有的常见的接口类型。其中，MSR30 系列路由器也可以作为中小型网络中的核心层或者汇聚层设备。

第2章 路由协议基础

2.1 路由分类

按照来源不同，路由可以分为以下几类：

- **直连路由**。直连路由是有由链路层协议发现的。直连路由无需配置，在接口存在 IP 地址时，由路由进程自动生成。它的特点是开销小，配置简单，无需人工维护，但只能发现本地接口所属网段的路由。
- **静态路由**。由管理员手工配置而成的路由成为静态路由。静态路由无开销，配置简单，适合简单的拓扑结构的网络。静态路由的缺点是无法自动根据网络拓扑变化改变。当一个网络故障发生后。静态路由不会自动修正，必须由管理员介入进行配置。
- **动态路由**。动态路由协议自动发现和维持的路由称为动态路由。动态路由的优点是无需人工配置具体路由表项，而由路由协议自动发现和计算。当网络拓扑结构复杂时，使用动态路由可减少管理员的配置工作，且减少配置错误。另外，动态路由协议支持路由备份，如果原油链路故障导致路由表项失效，协议可以自动计算和使用其他的路径，无需人工维护。系统启动动态路由协议后，系统之间交互协议报文，会占用一部分链路开销，并且动态路由协议配置复杂，需要管理员掌握一定的路由协议知识。

如果网络拓扑是星状，各节点之间没有冗余链路，则可以使用静态路由；如果网络中有冗余链路，如全互联或环形拓扑，可以使用动态路由，以增强路由可靠性。

如果网络是分层的，则通常在接入层使用静态路由来降低设备资源的消耗；而在汇聚层使用动态路由增强可靠性。

2.2 静态路由应用

恰当地设置和使用静态路由可以改进网络的性能，并可为重要的网络应用保证带宽。同时，在路由器上合理配置默认路由，能够减少路由表的表项数量，节省路由表空间，加快路由匹配速度。

在 SylixOS 系统中，有专门的命令实现静态路由的相关功能，可以通过 `help route` 命令查看。

配置静态路由时，要注意双向配置，避免出现单程路由。因为几乎所有的 Internet 应用如 HTTP、FTP 等，都是双向传输的，所以单程路由对用户的业务是没有意义的。

2.3 动态路由协议

常用的动态路由协议如表 2.1 所示。

表 2.1 常用动态路由协议

路由协议	协议算法	IGP/EGP	路由协议	协议算法	IGP/EGP
RIP	距离矢量	IGP	IS-IS	链路状态	IGP
OSPF	链路状态	IGP	BGP	路径矢量	EGP

2.3.1 路由协议分类

根据作用的范围不同，路由协议可以分为以下两种。

- **内部网关协议**（Internet Gateway Protocol, IGP），内部网关协议在一个自治系统内部运行。常见的 IGP 协议包括 RIP、OSPF 和 IS-IS 等；
- **外部网关协议**（Exterior Gateway Protocol, EGP）。外部网关协议运行于不同的自治系统之间。BGP 是目前最常用的 EGP。

根据使用的算法不同，路由协议分为以下两种。

- **距离矢量协议**（Distance-Vector）。常用的距离矢量协议包括 RIP 和 BGP。其中，BGP 又被称为路径矢量协议。
- **链路状态协议**（Link-State）。常用的链路状态协议包括 OSPF 和 IS-IS。

距离矢量协议和链路状态协议所采用的算法主要区别在发现和计算路由的方法不同。

2.3.2 路由协议的工作原理

各种动态路由协议工作的共同目的是计算和维护路由。通常，各种动态路由协议的工作过程大致相同，都包含以下四个阶段。

（1）邻居发现。运行了某种路由协议的路由器会主动的把自己介绍给网段内的其他路由器。路由器通过发誓不过广播报文或发送给指定的路由器邻居来做到这一点。

（2）交换路由信息。发现邻居后，每台路由器将自己已知的路由相关信息发给相邻的路由器，相邻的路由器又发送给下一台路由器。这样，最终所有的路由器都会收到网络中的所有路由信息。

（3）计算路由。每一台路由器都会运行某种算法，并计算出最终的路由来。实际上，需要计算的该条路由的下一跳和度量值。

（4）维护路由。为了能够观察到某台路由器突然失效等异常情况，路由协议规定两台路由器之间的协议报文应该周期性的发送。如果路由器有一段时间收不到邻居发来的协议报文，则可以认为该邻居失效了。

各个路由协议实现原理大致相同，但细节却会有所不同。

1. 距离矢量路由协议

距离矢量协议通常不维护邻居信息。在开始阶段，采用这种算法的路由器以广播或组播的形式发送给协议报文，请求邻居的路由信息；邻居路由器回应的协议报文中携带全部路由表，这样就完成路由表的初始化过程。

为了维护路由信息，路由器以一定的时间间隔向相邻的路由器发送路由更新，路由更新中携带本路由器的全部路由表。系统为路由表的表项设定超过时间，如果超过一定时间接收不到路由更新，系统会认为原有的路由失效，将其从路由表中删除。

距离矢量协议以到达目的地距离作为度量值，距离越大，路由越差。但是采用跳数作为度量值并不能完全反应链路带宽的实际状况，有时会造成协议选择次优路径。

距离矢量路由协议基于贝尔曼-福特算法。这种算法的缺点是计算路由时只考虑到目的网段的距离和方向。距离矢量路由协议完全信任邻居路由器，它不知道整个网络的拓扑环境，这样在环形拓扑网络中可能会造成环路。因此，采用 D-V 算法的路由器使用了一些避免环路的机制，如水平分割、路由毒化、毒性逆转等。

RIP 协议是一种典型的距离矢量路由协议。优点是配置简单，算法只占用较少的内存和 CPU 处理时间；缺点是算法本身不能完全杜绝路由自环，收敛相对较慢，周期性路由广播路由更新占用网络带宽较大，扩展性较差，并且最大跳数不能超过 16 跳。

2. 链路状态路由协议

链路状态路由协议基于 Dijkstra 算法，又被称为最短路径优先算法。

在开始阶段，采用这种算法的路由器以组播形式发送 Hello 报文，来发现邻居。收到 Hello 报文的邻居路由器会检查报文中所定义参数。如果双方一致就会形成邻居关系。有路由交换需求的邻居路由器会生成邻接关系，进而可以交换 LSA (Link State Advertisement, 链路状态通告)。

链路状态路由协议用 LSA 来描述路由器周边的网络拓扑和链路状态。邻接关系建立后，路由器会将自己的 LSA 发送给区域内的所有邻接路由器，同时也从邻接路由器接收 LSA。每台路由器都会收集其他路由的 LSA，所有的 LSA 放在一起便组成了 LSDB (链路状态数据库)。LSDB 是对整个自治系统的网络拓扑结构的描述。

链路状态路由协议以到达目的地的开销 (Cost) 作为度量值。路由器根据该接口的带宽自动计算到达邻居的权值。带宽与权值成反比，即带宽越高，权值越小，表示到邻居的路径越好。在使用最短路径优先算法计算最短路径时，将自己到各节点的路径上的权值相加，也就计算出了该节点到达各节点的开销，并将此开销作为度量值。

当网络拓扑发生变化时，路由器不发送路由表，只是发送含有链路变化信息的 LSA 在区域内扩散，被所有的路由器接收到。每个路由器更新

各自的 LSDB，再运行 SPF 算法，重新计算路由。这样好处是带宽占有小，路由收敛速度快。

OSPF 和 IS-IS 属于链路状态路由协议。它们能够完全杜绝协议内的路由自环，且采用增量更新方式来通告变化的 LSA，占用带宽较少。OSPF 和 IS-IS 采用路由分组和区域划分等机制能够支持大规模的网络，且扩展性比较好。但相对 RIP 来说，OSPF 和 IS-IS 的配置较复杂一些。

3. 路径矢量路由协议

路径矢量路由协议结合了距离矢量路由协议和链路状态路由协议的优点。

路径矢量路由协议采用单播方式与相邻路由器建立邻居关系。邻居关系建立后，根据预先配置的策略，路由器将全部或部分带有路由属性的路由表发送给邻居。邻居收到路由表后，根据预先配置的策略将全部或者部分的路由信息加入自己的路由表中。

当路由信息发生变化时，路径矢量路由协议只发送增量路由给路由，减少带宽的消耗。

邻居关系是以单播形式，通过 TCP 三方握手形式建立的，并且在建立后定时交换 Keepalive 报文，以维持邻居关系正常。

在拓扑发生变化时，路径矢量路由协议仅将变化的路由信息发送给邻居路由器，以逐跳的方式在全网络内扩散。

BGP 协议属于路径矢量路由协议。它采用了一些方法能够防止路由环路。BGP 协议吧 AS 间传递的路由都记录了传递的 AS 号码，这样路由器接收到路由时可以查看此条路由是不是自己发出的。在 AS 内，BGP 协议规定路由器不能把从邻居学到的路由再返回给邻居。

2.4 路由选择原则

每个路由协议都维护了自己的路由表，这种路由表成为协议路由表。协议路由表中只记录了本路由协议学习和计算的路由。

大多数路由协议都支持多进程。各个协议进程之间互不影响，相互独立。各个进程之间的交互相当于不同路由协议之间的路由交互。

各个路由协议的各个进程独立维护自己的路由表，然后统一汇总到 IP 路由表中。IP 路由表首先选择路由协议优先级高的路由使用。如果优先级一致，则再选择度量值最优的路由作为 IP 路由表的有效（Active）路由，指导 IP 报文转发。其余的路由进行备份，如果有效路由失效，在重新进行选择。

路由度量值只在同一种路由协议内才有比较意义，不同的路由协议之间的路由度量值没有可比性，也不存在换算关系。

对于相同的目的地，不同的路由协议（包括静态路由）可能会发现不同的路由，但这些路由并不都是最优的。事实上，在某一时刻，到某一目的地的当前路由仅能由唯一的路由协议来决定。为了判断最优路由，各路由协议（包括静态路由）都被赋予了一个优先级。当存在多个路由信息源时，具有较高优先级的路由协议发现的路由将成为当前路由。各种路由协议及其发现路由的默认优先级如表 2.2 所示。

表 2.2 路由协议及默认时的路由优先级

路由协议	优先级	路由协议	优先级
DIRECT	0	RIP	100
OSPF	10	IS-IS	15
UNKNOWN	256	BGP	255

表 2.2 所示中，0 表示直接连接的路由，256 表示任何来自不可信源端的路由。数值越小，表明优先级越高。除了直连路由外，各种路由的优先级都可由管理员手工进行配置。另外每条静态路由的优先级都可以不相同。

2.5 路由协议比较

目前常用的路由协议包括 RIP-1/2、OSPF、IS-IS、BGP4 种。本节对其协议特点进行全面的比较。

RIP 是最早的路由协议，其设计思想是为了小型网络中提供简单易用的动态路由，其算法简单，对 CPU 和内存资源要求低。RIP 采用广播（RIP-1）或组播（RIP-2）方式在邻居间传送协议报文，传输层采用 UDP 封装，端口号是 520。由于 UDP 是不可靠的传输层协议，所以 RIP 被设计成周期性广播所有路由表，即如果邻居超过三次无法收到路由更新，则认为路由失效。RIP-1 协议不支持验证，安全性较低；RIP-2 进行了改进，从而能够支持验证，安全性提高了。

OSPF 是目前应用最广泛的 IGP 协议。OSPF 的设计思想是为大中型网络提供分层次的、可划分区的路由协议。其算法复杂，但能够保证无域内环路。OSPF 采用 IP 来进行承载，所有的协议报文都有 IP 封装后进行传输，端口号是 89。IP 是网路层协议，本身是不可靠的。因此，为了保证协议报文传输的可靠性，OSPF 采用了确认机制：在邻居发现阶段和交互 LSA 阶段，OSPF 都采用确认机制来保证传输可靠。OSPF 支持验证，从而 OSPF 的安全性得到了保证。

IS-IS 是另外一种链路状态型的路由协议，其同样采用 SPF 算法，支持路由分组管理与划分区域，同样可应用在大中型网络中，可扩展性。与 OSPF 不同的是，IS-IS 直接运行在基本链路层，其所有协议报文通过链路层协议来承载。因此，IS-IS 也可以运行在无 IP 的网络中，如 OSI 网络中。为了保证协议报文传输的可靠性，IS-IS 同样设计了确认机制来保证协议报文，在传输过程中没有丢失。IS-IS 也支持验证，从而安全性得到了保证。

BGP 是唯一的 EGP 协议。与其他协议不同，BGP 采用 TCP 来保证协议传输的可靠性，端口号是 179。TCP 本身有三次握手的确认机制，运行 BGP 的路由器时，首先要建立可靠

的 TCP 链接；其次通过 TCP 连接来交互 BGP 报文。这样，BGP 不需要自己设计可靠性传输机制，降低了协议机制保温的复杂度和开销。另外，BGP 的安全性也可以由 TCP 来保证，这是因为 TCP 支持验证，通过验证的双方才能够建立 TCP 连接。

BGP 自己不学习路由，它的路由来源于 IGP 协议，如 OSPF 等。管理员手工指定那些 IGP 路由能够导入 BGP 中，并手工指定 BGP 能够与哪些邻居建立对等体关系从而交换路由信息。

将以上 4 中协议的协议端口、可靠性、安全性进行总结，如表 2.3 所示。

表 2.3 路由协议可靠性、安全性

协议	协议端口	可靠性	安全性（是否支持验证）
RIP_1	UDP520	低	否
RIP-2	UDP520	低	是
OSPF	IP89	高	是
IS-IS	基于链路层协议	高	是
BGP	TCP179	高	是

RIP-1 是早期的路由协议，所以不支持无类别（Classless）路由，只能支持按类自动聚合，并且不支持可变长子网掩码（VLSM），应用有一定限制。

除 RIP-1 外，其他路由协议都能够支持 VLSM 和手工聚合，这样能够对网络进行很细致的子网划分和汇聚，从而节省 IP 地址，减少路由表数量。

由于 RIP 和 BGP 的路由更新需要以逐跳的方式惊醒传播，所以路由收敛速度较慢。而链路状态型的路由协议采用 SPF 算法，根据自己的 LSDB 进行路由计算，所有收敛速度快。

所有的路由协议都需要采用定时器来维护邻居关系和路由信息。

第3章 RIP

3.1 RIP 协议概述

3.1.1 RIP 协议特点

(1) 路由信息更新特性:

路由器最初启动时只包含了其直连网络的路由信息，并且其直连网络的 metric 值为 1，然后它向周围的其他路由器发出完整路由表的 RIP 请求（该请求报文的“IP 地址”字段为 0.0.0.0）。路由器根据接收到的 RIP 应答来更新其路由表，具体方法是添加新的路由表项，并将其 metric 值加 1。如果接收到与已有表项的目的地址相同的路由信息，则分下面三种情况分别对待：第一种情况，已有表项的来源端口与新表项的来源端口相同，那么无条件根据最新的路由信息更新其路由表；第二种情况，已有表项与新表项来源于不同的端口，那么比较它们的 metric 值，将 metric 值较小的一个最为自己的路由表项；第三种情况，新旧表项的 metric 值相等，普遍的处理方法是保留旧的表项。

路由器一般每 30 秒发送一次自己的路由表（以 RIP 应答的方式广播出去）。针对某一条路由信息，如果 180 秒以后都没有接收到新的关于它的路由信息，那么将其标记为失效，即 metric 值标记为 16。

(2) RIP-1 对 RIP 报文中“版本”字段的处理:

0: 忽略该报文;

1: 版本 1 报文，检查报文中“必须为 0”的字段，若不符合规定，忽略该报文;

>1: 不检查报文中“必须为 0”的字段，仅处理 RFC 1058 中规定的有意义的字段。因此，运行 RIP 版本 1 的机器能够接收处理 RIP 版本 2 的报文，但会丢失其中的 RIP 版本 2 新规定的那些信息。

(3) RIP 版本 1 对地址的处理:

RIP 版本 1 不能识别子网网络地址，因为在其传送的路由更新报文中不包含子网掩码，因此 RIP 路由信息要么是主机地址，用于点对点链路的路由；要么是 A、B、C 类网络地址，用于以太网等的路由；另外，还可以是 0.0.0.0，即缺省路由信息。

(4) 计数到无穷大 (Counting to Infinity):

在 RIP 的局限性一部分提到了可能出现的计数到无穷大的现象，下面就来分析一下该现象的产生原因与过程。考察下面的简单网络:

c(目的网络) -----router A-----router B

在正常情况下，对于目标网络，A 路由器的 metric 值为 1，B 路由器的 metric 值为 2。当目标网络与 A 路由器之间的链路发生故障而断掉以后:

c(目的网络) --||--router A-----router B

A 路由器会将针对目标网络 C 的路由表项的 metric 值置为 16，即标记为目标网络不可达，并准备在每 30 秒进行一次的路由表更新中发送出去，如果在这条信息还未发出的时候，A 路由器收到了来自 B 的路由更新报文，而 B 中包含着关于 C 的 metric 为 2 的路由信息，根据前面提到的路由更新方法，路由器 A 会错误的认为有一条通过 B 路由器的路径可以到达目标网络 C，从而更新其路由表，将对于目标网络 C 的路由表项的 metric 值由 16 改为 3，

而对于的端口变为与 B 路由器相连接的端口。很明显，A 会将该条信息发给 B，B 将无条件更新其路由表，将 metric 改为 4；该条信息又从 B 发向 A，A 将 metric 改为 5……最后双发的路由表关于目标网络 C 的 metric 值都变为 16，此时，才真正得到了正确的路由信息。这种现象称为“计数到无穷大”现象，虽然最终完成了收敛，但是收敛速度很慢，而且浪费了网络资源来发送这些循环的分组。另外的 120 秒以后，如果仍然没有更新信息，该条失效信息被删除。

(5) RIP 协议的局限性：

- 协议中规定，一条有效的路由信息的度量（metric）不能超过 15，这就使得该协议不能应用于很大的网络，应该说正是由于设计者考虑到该协议只适合于小型网络所以才进行了这一限制。对于 metric 为 16 的目标网络来说，即认为其不可到达。
- 该路由协议应用到实际中时，很容易出现“计数到无穷大”的现象，这使得路由收敛很慢，在网络拓扑结构变化以后需要很长时间路由信息才能稳定下来。
- 该协议以跳数，即报文经过的路由器个数为衡量标准，并以此来选择路由，这一措施欠合理性，因为没有考虑网络延时、可靠性、线路负荷等因素对传输质量和速度的影响。

3.1.2 RIP 协议基本原理

路由器的关键作用是用于网络的互连，每个路由器与两个以上的实际网络相连，负责在这些网络之间转发数据报。在讨论 IP 进行选路和对报文进行转发时，我们总是假设路由器包含了正确的路由，而且路由器可以利用 ICMP 重定向机制来要求与之相连的主机更改路由。但在实际情况下，IP 进行选路之前必须先通过某种方法获取正确的路由表。在小型的、变化缓慢的互连网络中，管理者可以用手工方式来建立和更改路由表。而在大型的、迅速变化的环境下，人工更新的办法慢得不能接受。这就需要自动更新路由表的方法，即所谓的动态路由协议，RIP 协议是最简单的一种。

RIP 协议是基于 Bellman-Ford（距离向量）算法，此算法 1969 年被用于计算机路由选择，正式协议首先是由 Xerox 于 1970 年开发的，当时是作为 Xerox 的“Networking Services (NXS)”协议族的一部分。由于 RIP 实现简单，迅速成为使用范围最广泛的路由协议。

RIP 协议是一种内部网关协议（IGP），是一种动态路由协议，用于自治系统（AS）内的路由信息的传递。RIP 协议基于距离矢量算法（Distance Vector Algorithms），使用“跳数”（即 metric）来衡量到达目标地址的路由距离。这种协议的路由器只关心自己周围的世界，只与自己相邻的路由器交换信息，范围限制在 15 跳（15 度）之内，再远，它就不关心了。RIP 应用于 OSI 网络七层模型的网络层。

在路由实现时，RIP 作为一个系统常驻进程（daemon）而存在于路由器中，负责从网络系统的其它路由器接收路由信息，从而对本地 IP 层路由表作动态的维护，保证 IP 层发送报文时选择正确的路由。同时负责广播本路由器的路由信息，通知相邻路由器作相应的修改。RIP 协议处于 UDP 协议的上层，RIP 所接收的路由信息都封装在 UDP 协议的数据报中，RIP 在 520 号 UDP 端口上接收来自远程路由器的路由修改信息，并对本地的路由表做相应的修改，同时通知其它路由器。通过这种方式，达到全局路由的有效。

3.2 RIP 的路由表结构

使用 RIP 报文中列出的项，RIP 主机可以彼此之间交流路由信息。这些信息存储在路由表中，路由表为每一个知道的、可达的目的地保留一项。每个目的地表项是到达那个目的地的最低开销路由。

注意每个目的地的表项数可以随路由生产商的不同而变化。生产商可能选择遵守规范，也可以对标准进行他们认为合适的“强化”。所以，用户很可能会发现某个特殊商标的路由器为每一个网络中的目的地存储至多 4 条相同费用的路由。

每个路由表项包括以下各域：

- 目的 IP 地址域
- 距离-向量度量域
- 下一跳 IP 地址域
- 路由变化标志域
- 路由计时器域

注意虽然 RFC 1058 是一个开放式标准，能支持大量互连网络地址结构，然而它是由 IETF 设计用于 Internet 中自治系统内的协议。如此，使用这种形式 RIP 的自然是网络互联协议。

(1) 目的 IP 地址域

任何路由表中所包含的最重要信息是到所知目的地的 IP 地址。一旦一台 RIP 路由器收到一个数据报文，就会查找路由表中的目的 IP 地址以决定从哪里转发那个报文。

(2) 距离-向量度量域

路由表中的度量域指出报文从起始点到特定目的地的总耗费。路由表中的度量是从路由器到特定目的地之间网络链路的耗费总和。

(3) 下一跳 IP 地址域

下一跳 IP 地址域包括至目的地的网络路径上下一个路由器接口的 IP 地址。如果目的 IP 地址所在的网络与路由器不直接相连时，路由器表中才出现此项。

(4) 路由变化标志域

路由变化标志域用于指出至目的 IP 地址的路由是否在最近发生了变化。这个域是重要的，因为 RIP 为每一个目的 IP 地址只记录一条路由。

(5) 路由计时器域

有两个计时器与每条路由相联系，一个是超时计时器，一个是路由刷新计时器。这些计时器一同工作来维护路由表中存储的每条路由的有效性。路由表维护过程在 12.2.2 节中详细描述。

3.3 RIP 简单用例

在 SylixOS 操作系统中，支持 Quagga 配置网络路由，RIP 作为 Quagga 路由协议的一种最简单的动态路由协议，一般用于小型网络的路由配置。

以下用 Quagga 中 RIP 协议的简单用例作为参考。

测试拓扑图如图 3.1 所示。

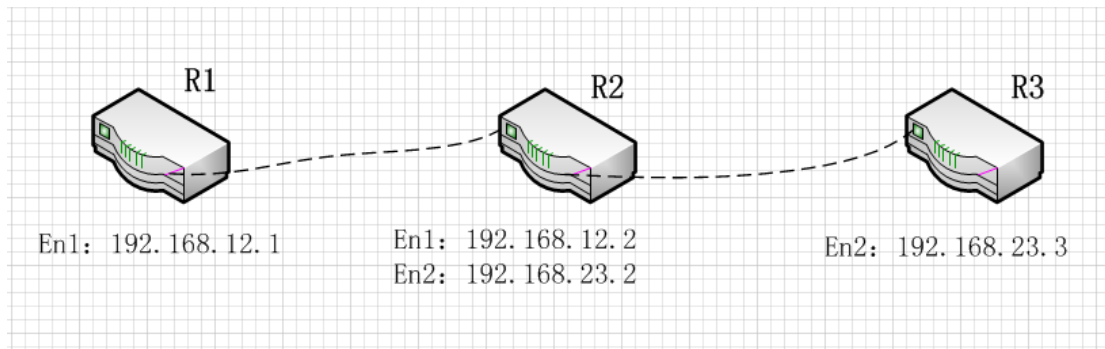


图 3.1 RIP 测试图

如图 3.1 所示，三台路由器 R1、R2、R3 彼此之间可以发现的邻居都是有直连路由的路由器，链路的两端可以互相到达，而 R1 想要到达 R3，可以通过配置 RIP 动态路由协议来实现。

首先分别在 R1、R2、R3 三台路由器开启 zebra 以及 ripd。

```
root# zebra -d
root# ripd -d
```

在 Quagga 中配置 RIP 指令如下：

在路由器 R1：

```
ripd(config)# router rip
ripd(config-router)# network 192.168.12.0/24
ripd(config-router)# end
```

在路由器 R2：

```
ripd(config)#router rip
ripd(config-router)#network 192.168.12.0/24
ripd(config-router)#network 192.168.23.0/24
ripd(config-router)#end
```

在路由器 R3：

```
ripd(config)# router rip
ripd(config-router)# network 192.168.23.0/24
ripd(config-router)# end
```

当 RIP 动态路由协议配置完成，三台路由器可以相互到达彼此（可以互相 ping 通）。

第4章 OSPF

4.1 OSPF 协议概述

4.1.1 OSPF 协议特点

OSPF 协议与 RIP 协议相比,存在很大的不同。OSPF 是一个典型的链路状态路由协议。它具有以下特点。

(1) 支持较大规模的网络。OSPF 协议无路由跳数限制,所以适用范围广,支持网络规模更大。在特定的组网环境下,OSPF 协议单区域甚至可支持几十台路由器。

(2) 组播触发式更新。OSPF 协议在收敛完成后,会以触发方式发送拓扑变化信息给其他路由器,从而占用较少的链路带宽;同时,在某些类型的链路上以组播方式发送协议报文,减少对其他设备的干扰。

(3) 收敛速度快。在网络拓扑结构发生变化后,OSPF 协议会立即发送更新报文,从而使拓扑变化很快扩散到整个自治系统;同时,OSPF 协议周期性发送 Hello 报文来维持邻居关系。

(4) 度量值为开销 (Cost)。OSPF 协议采用链路开销作为度量值,而链路开销与链路带宽成反比,即带宽越高,开销越小。OSPF 寻路主要基于带宽因素。

(5) 协议设计避免了路由环路。OSPF 根据 LSA 计算最短路径树算法计算路径,从算法根本保证了不会生成自环路由。

4.1.2 OSPF 协议基本原理

作为典型的链路状态路由协议,OSPF 协议的工作过程包含了邻居发现、路由交换、路由计算、路由维护等阶段。在这些过程中,主要涉及到三个表。

(1) 邻居表。邻居表记录了所有建立了邻居关系的路由器,包括相关描述和邻居状态。路由器会定时向邻居发送 Hello 报文,如果在一定周期内没有收到邻居的回应,就认为邻居路由器失效,从邻居表中删除。

(2) 链路状态数据库 (LSDB)。运行 OSPF 协议的路由器之间并不是交换路由表,而是交换彼此对于链路状态的描述信息。交换完成之后,所有同一区域的路由器的拓扑表中都具有当前区域的所有链路状态信息,并且都是一致的。

(3) 路由表。运行 OSPF 协议的路由器在获得完整的链路状态信息之后,运用 SPF 算法进行计算,并且将计算出来的最优路由加入 OSPF 路由表。

OSPF 协议生成路由的过程如下:

(1) 生成 LSA 描述自己的接口状态

每台运行 OSPF 协议的路由器都会根据自己周围的拓扑结构生成 LSA。LSA 包含了接口状态、链路开销、IP 地址/掩码等信息。

(2) 同步 OSPF 区域内每台路由器的 LSDB

由于一条 LSA 是一台路由器或一个网段拓扑结构的描述,整个 LSDB 就形成了对整个网络的拓扑结构的描述。OSPF 区域内的路由器都有一张完整且相同的图。

(3) 使用 SPF 算法计算出路由

OSPF 协议通过 SPF 算法根据 LSDB 算出最优路径树，这个树上，由根到各节点的开销最小。路由器将路由加入 OSPF 路由表。当 SPF 算法发现有两条目的网络相同的路由的开销相同时，会将这两条路由加入路由表，形成等价路由。

4.2 分层结构

4.2.1 骨干区域与非骨干区域

当一个大型网络的路由器都运行 OSPF 协议时，路由器的增多会导致 LSDB 异常庞大，占用大量的内存空间，SPF 算法的复杂度增加，导致 CPU 负担很重。

在网络规模增大后，拓扑结构发生变化的概率增加，网络会经常处于不稳定状态，造成网络中大量的 OSPF 报文在传递，降低了网络带宽的利用率。而且每一次拓扑结构变化都会导致网络中的所有路由器重新计算路由。

OSPF 协议通过将自治系统划分为不同的区域（Area）来解决上述问题，如图所示。区域是从逻辑上将路由器划分为不同的组，每个组用区域号（Area ID）来标识。区域的边界是路由器，而不是链路。一个链路只能属于一个区域，或者说每个运行 OSPF 的接口必须属于某个特定区域。

区域号为 0 的区域通常为骨干区域。骨干区域负责区域之间的路由，非骨干区域之间的路由信息必须通过骨干区域来转发。对此，OSPF 协议有以下两个规定。

- 所有非骨干区域必须与骨干区域保持连通；
- 骨干区域自身也必须保持连通。

区域划分的好处：

- 减少区域内的 LSA 数量。区域划分后，LSDB 就不需要维护所有的 LSA，只需要维护本区域内链路状态信息；
- 减少路由震荡的影响。OSPF 协议可以对部分区域进行特殊配置，或者在区域边缘设置路由聚合和路由过滤等策略，将路由震荡限制在区域内，从而减少对于自治系统内其他区域的影响，降低其他区域路由器 SPF 算法反复计算的次数。

4.2.2 OSPF 路由器类型

根据 OSPF 路由器在 AS 中位置的不同，分为以下几类。

- （1）区域内路由器（Internal Router）。该类路由器的所有接口都属于同一个 OSPF 区域。
- （2）区域边界路由器（Area Border Router, ABR）。该类路由器可以同时属于两个以上的区域，但其中一个必须是骨干区域。ABR 用来连接骨干区域和非骨干区域，它与骨干区域之间可以是物理上的连接，也可以是逻辑上的连接。
- （3）骨干路由器（Backbone Router）。该类路由器至少有一个接口属于骨干区域。所有的 ABR 和位于 Area0 的内部路由器都是骨干路由器。
- （4）自治系统边界路由器（Autonomous System Border Router, ASBR）。与其他 AS 交换路由信息的路由器为 ASBR。ASBR 不一定位于 AS 的边界，可能是区域内路由器，也有可能是 ABR。只要一台路由器引入了外部路由的信息，就成为了 ASBR。

4.3 OSPF 网络类型

OSPF 协议根据链路层协议类型将网络分为以下四种类型。

- （1）Broadcast：当链路层协议是 Ethernet、FDDI 时，OSPF 协议默认的网络类型是 Broadcast。通常以组播形式（224.0.0.5 和 224.0.0.6）发送协议报文。

(2) NBMA (Non-Broadcast Multi-Access, 非广播多点可达): 当链路层协议是帧中继、ATM 或 X.25 时, OSPF 默认的网络类型是 NBMA。在该类型的网络中, 以单播形式发送协议报文。

(3) P2MP (Point-to-MultiPoint, 点到多点): 没有一种链路层协议会被默认为 P2MP 类型。P2MP 必须是由其他的网络类型强制更改的。在该类型的网络中, 以组播形式(224.0.0.5) 发送协议报文。

(4) P2P (Point-to-Point, 点对点): 当链路层协议是 PPP、HDLC 时, OSPF 协议默认的网络类型是 P2P。在该类型的网络中, 以组播形式 (224.0.0.5) 发送协议报文。

对于接口的网络类型是 NBMA 需要进行一些特殊的配置。由于无法通过报文的形式发现相邻路由器, 必须手工为该接口指定相邻路由器的 IP 地址。

4.4 OSPF 报文和封装

OSPF 协议有 5 种类型的协议报文。

(1) Hello 报文: 周期性发送, 用来发现和维持 OSPF 协议邻居关系。包括一个定时器的数值、DR (Designated Router)、BDR (Backup Designated Router) 以及自己已知的邻居。

(2) DD (Database Description, 数据库描述) 报文: 描述了本地 LSDB 中每一条 LSA 的摘要信息, 用于两个路由器之间的数据库同步。

(3) LSR (Link State Request, 链路状态请求) 报文: 向对方请求所需的 LSA。两台路由器互相交换 DD 报文, 发现有哪些 LSA 是自己 LSDB 所没有的, 发送 LSR 请求获取对应的 LSA。

(4) LSU (Link State Update, 链路状态更新) 报文: 向对方发送其所需要的报文。

(5) LSAck (Link State Acknowledgment, 链路状态确认) 报文: 对收到的 LSA 进行确认。其内容是需要确认的 LSA 的 Header (一个报文可对多个 LSA 确认)。

OSPF 协议报文直接封装在 IP 报文之中, 其 IP 报文头的协议号为 89。OSPF 协议报文封装格式如图 4.1 所示。

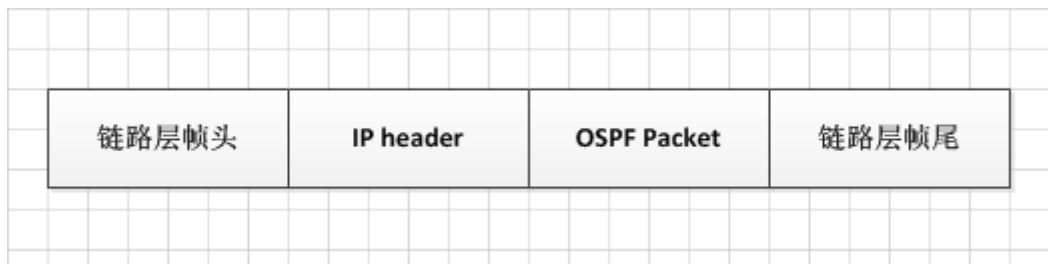


图 4.1 OSPF 协议报文封装格式

4.5 LSDB 更新

当网络拓扑发生变化时, 感知到变化的 OSPF 路由器会生成相应的 LSA 更新报文, 并发送到区域中, 如图 4.2 所示。

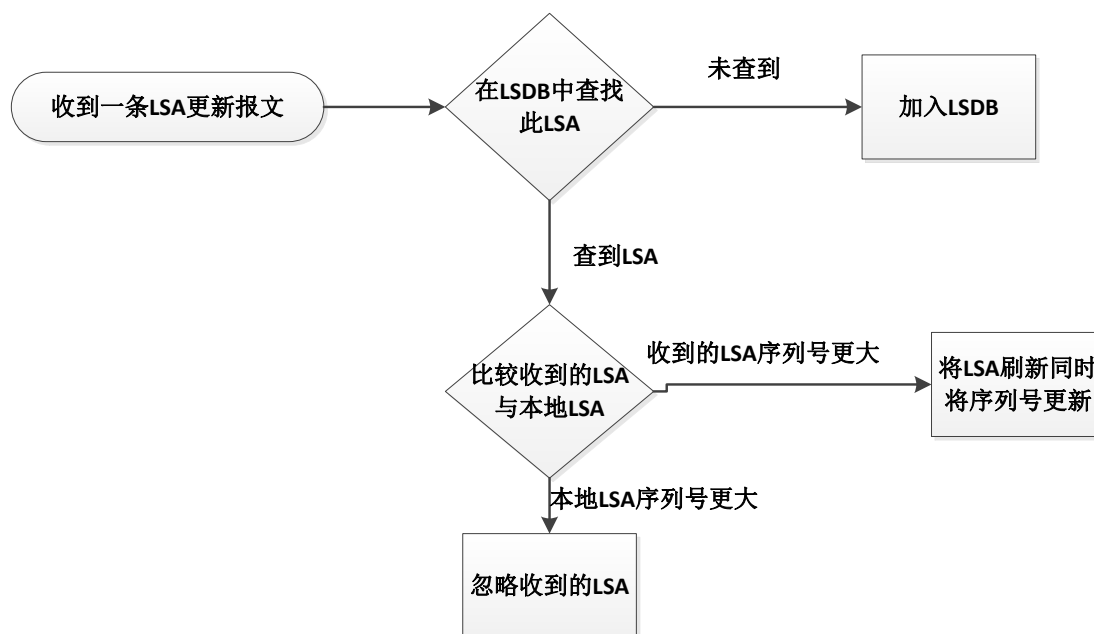


图 4.2 LSDB 更新

运行 OSPF 协议的路由器收到一条 LSA 更新报文的时候，其工作流程如下：

(1) 系统会首先在 LSDB 中查找此条 LSA。如果不能查到，就认为是一条新的 LSA，加入 LSDB。

(2) 如果查到了此条 LSA，那么比较这条 LSA 的序列号。如果收到的新 LSA 序列号更大，那么认为这条 LSA 有了更新，将这条 LSA 的计时器进行刷新，同时更新序列号。

(3) 如果收到的新 LSA 序列号等于或小于 LSDB 中 LSA 的序列号，那么就认为收到的 LSA 可能是由于网络拥塞或者重传的陈旧的 LSA，不会对 LSDB 的 LSA 做任何操作，并且将收到的 LSA 更新报文丢弃。

另外，为了保证 LSDB 及时刷新，LSDB 里面的 LSA 都设定了老化时间，默认为 1h。如果 1h 内 LSA 没有被更新，LSA 将会老化同时被移除。

在默认情况下，LSDB 每隔 0.5h 刷新一次所有的 LSA。此时，LSA 的序列号会加一，同时老化计时器会重置。

当路由器想将一条 LSA 从 LSDB 中删除，可以将老化时间设置为最大老化时间，然后向所有路由器发送更新。

为了节省网络带宽和降低路由器资源消耗，在广播和 NBMA 网络中的链路状态发生变化时，DR 路由器发送更新报文，通知其他路由器更新自己的 LSDB。

4.6 OSPF 虚连接

在 OSPF 网络中，通过划分区域能够减少区域中的 LSA 数量，降低拓扑变化导致的路由震荡。在区域划分时，为了保证路由学习正常，需要注意遵守以下两个规则。

- 骨干区域必须连续；
- 所有非骨干区域都必须和骨干区域相连接。

如果骨干区域不是连续的，则会导致骨干区域无法正常学习。而 OSPF 协议为了防止路由环路，规定 ABR 从骨干区域学到的路由不能再向骨干区域传播。如果非骨干区域没有

与骨干区域相连接，也会导致 OSPF 无法正常学习路由。OSPF 协议规定，所有非骨干区域的路由必须通过骨干区域进行。

如果出现骨干区域被分割，或者非骨干区域无法和骨干区域保持连通的问题时，可以通过配置 OSPF 虚连接（Virtual Link）予以解决。

虚连接是指在两台 ABR 之间通过一个非骨干区域而建立一条逻辑连接通道。它的两端必须是 ABR，而且必须在两端同时配置方可生效。为虚连接两端提供一条非骨干区域内部路由的区域为传输区域。

虚连接相当于在两个 ABR 之间形成了一个点到点的逻辑连接。虚连接和物理接口一样可以接口的各个参数，如发送 Hello 报文间隔等。虚连接建立后，两台 ABR 间通过单播方式直接传递 OSPF 协议报文。对于传输区域内的路由器来说，虚连接所传输的协议报文是透明的，只是当做普通的 IP 报文来转发。

虚连接的另外一个应用是提供一个冗余的备份链路。当骨干区域因链路故障不能保持连通时，通过虚连接仍然可以保持骨干区域在逻辑上的连通性。

4.7 OSPF 简单用例

4.7.1 OSPF 多区域连接用例

1. OSPF 协议多域

在大规模网络中，为了减少区域内的 LSA 数量需要进行区域划分，这样一来 LSDB 就不需要维护所有的 LSA，只需要维护本区域内链路状态信息，同时减少路由震荡的可能性。

如图 4.3 所示，在该网络中存在三个区域，骨干区域以及两个非骨干区域，非骨干区域必须与骨干区域连接才能与其他的非骨干区域进行交换信息。五台路由器通过交换机相互通信，从 R1 到 R5 分别对应接口 IP 为 192.168.1.81 到 192.168.1.85。

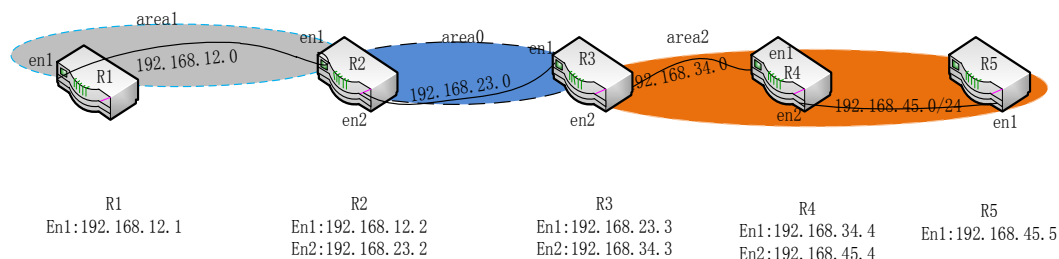


图 4.3 OSPF 多域

配置 OSPF 协议具体步骤如下：

首先分别将五台路由器开启 zebra 核心服务以及 ospf 协议。

```
root# zebra -d
root# ospfd -d
```

然后在相对应的协议配置文件中设置相应的输出信息，在 ospfd.conf 文件中配置如下：

```
hostname ospfd
password zebra
log stdout
```

根据路由器每个接口的 IP 配置 OSPF，通过指令进入 ospf 配置模式。

```
ospfd > enable
ospfd# configur terminal
ospfd(config)#
```

在路由器 R1 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.81
ospfd(config-router)#network 192.168.12.0/24 area 1
ospfd(config-router)#end
```

在路由器 R2 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.82
ospfd(config-router)#network 192.168.12.1/24 area 1
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#end
```

在路由器 R3 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.83
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#network 192.168.34.0/24 area 2
ospfd(config-router)#end
```

在路由器 R4 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.84
ospfd(config-router)#network 192.168.34.0/24 area 2
ospfd(config-router)#network 192.168.45.0/24 area 2
ospfd(config-router)#end
```

在路由器 R5 上：

```
ospfd(config)#router ospf
ospfd (config-router)#ospf router-id 192.168.1.85
ospfd(config-router)#network 192.168.45.0/24 area 1
ospfd(config-router)#end
```

配置结束后，非骨干区域 area1 和 area2 通过与骨干区域 area0 的信息交换来通信。各区域的路由器都可以相互到达。

2. OSPF 虚连接（非骨干区域与骨干区域不连接）

当有非骨干区域与骨干区域不连接时，会导致 OSPF 无法正常学习路由，而且非骨干区域之间的路由交换必须通过骨干区域进行，因此，需要设置虚连接。

如图 4.4 所示，非骨干区域通过传输区域建立虚连接与骨干区域连接。

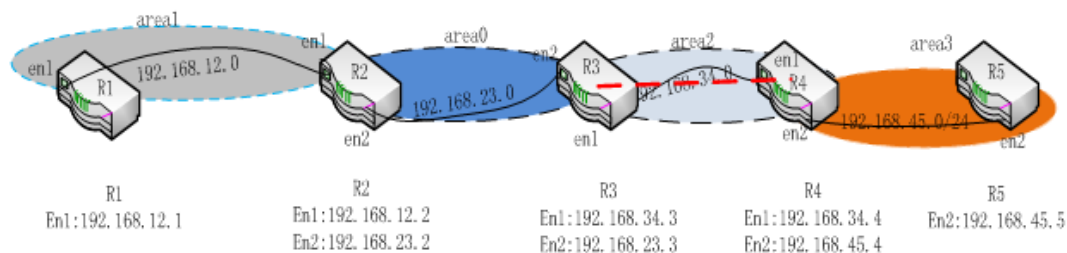


图 4.4 非骨干区域与骨干区域虚连接

配置 OSPF 协议具体步骤如下：

首先分别将五台路由器开启 zebra 核心服务以及 ospf 协议。

```
root# zebra -d
root# ospfd -d
```

然后在相对应的协议配置文件中设置相应的输出信息，在 ospfd.conf 文件中配置如下：

```
hostname ospfd
password zebra
log stdout
```

根据路由器每个接口的 IP 配置 OSPF，通过指令进入 ospf 配置模式。

```
ospfd > enable
ospfd# configur terminal
ospfd(config)#
```

在路由器 R1 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.81
ospfd(config-router)#network 192.168.12.0/24 area 1
ospfd(config-router)#end
```

在路由器 R2 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.82
ospfd(config-router)#network 192.168.12.0/24 area 1
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#end
```

在路由器 R3 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.83
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#network 192.168.34.0/24 area 2
```



```
ospfd(config-router)#area 2 virtual-link 192.168.1.84 // 配置虚链路
ospfd(config-router)#end
```

在路由器 R4 上:

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.84
ospfd(config-router)#network 192.168.34.0/24 area 2
ospfd(config-router)#network 192.168.45.0/24 area 3
ospfd(config-router)#area 2 virtual-link 192.168.1.83 //配置虚链路
ospfd(config-router)#end
```

在路由器 R5 上:

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.85
ospfd(config-router)#network 192.168.45.0/24 area 3
ospfd(config-router)#end
```

上述配置完成后，OSPF 协议配置结束。非骨干区域与骨干区域都建立了直接连接或者虚连接，使得每个路由器都可以从骨干区域及时的同步自己的 LSDB，形成最优路径树。也就是说每个路由器的 LSA 都可以到各个区域。

3. OSPF 虚连接（骨干区域被分割）

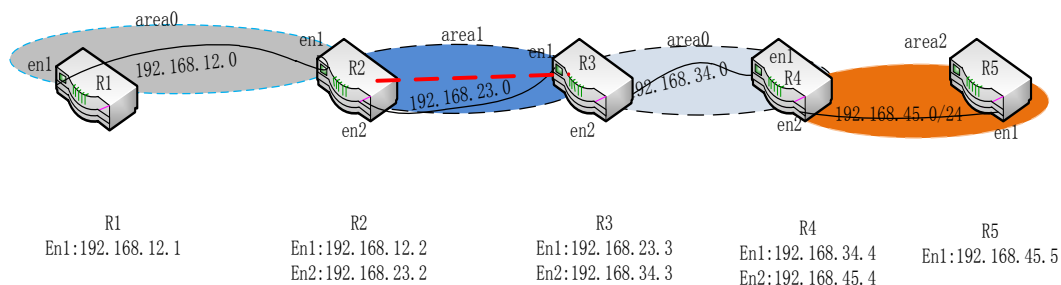


图 4.5 骨干区域被分割（虚连接）

如图 4.5 所示，骨干区域 area0 被 area1 分割，由于 OSPF 协议为了防止路由环路规定 ABR 从骨干区域学到的路由不能再向骨干区域传播。因此，area1 必须作为传输区域建立虚连接,使得两个 area0 可以相互学习。五台路由器通过交换机相互通信，从 R1 到 R5 分别对应接口 IP 为 192.168.1.81 到 192.168.1.85。

配置 OSPF 协议具体步骤如下：

首先分别将五台路由器开启 zebra 核心服务以及 ospf 协议。

```
root# zebra -d
root# ospfd -d
```

然后在相对应的协议配置文件中设置相应的输出信息，在 ospfd.conf 文件中配置如下：

```
hostname ospfd
```

```
password zebra
log stdout
```

根据路由器每个接口的 IP 配置 OSPF，通过指令进入 ospf 配置模式。

```
ospfd > enable
ospfd# configur terminal
ospfd(config)#
```

在路由器 R1 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.81
ospfd(config-router)#end
```

在路由器 R2 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.82
ospfd(config-router)#network 192.168.12.0/24 area 0
ospfd(config-router)#network 192.168.23.0/24 area 1
ospfd(config-router)#area1 virtual-link 192.168.1.83 // 配置虚链路
ospfd(config-router)#end
```

在路由器 R3 上

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.83
ospfd(config-router)#network 192.168.23.0/24 area 1
ospfd(config-router)#network 192.168.34.0/24 area 0
ospfd(config-router)#area 1 virtual-link 192.168.1.82 // 配置虚链路
ospfd(config-router)#end
```

在路由器 R4 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.84
ospfd(config-router)#network 192.168.34.0/24 area 0
ospfd(config-router)#network 192.168.45.0/24 area 2
ospfd(config-router)#end
```

在路由器 R5 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.85
ospfd(config-router)#network 192.168.45.0/24 area 3
ospfd(config-router)#end
```

上述配置完成后，OSPF 协议配置结束。非骨干区域与骨干区域都建立了直接连接或者虚连接，使得每个路由器都可以从骨干区域及时的同步自己的 LSDB，形成最优路径树。

第5章 IS-IS

5.1 IS-IS 概述

IS-IS (Intermediate System-to-Intermediate System intra-domain routing informationexchange protocol, 中间系统到中间系统的域内路由信息交换协议) 最初是 ISO 为它的无连接网络协议 (ConnectionLess Network Protocol, CLNP)。

ISO 定义了 OSI 七层参考模型作为计算机网络框架设计的标准, 并在此基础上定义了两种网络层服务: CONS (Connection-Oriented Network Service, 面向连接的网络服务) 和 CLNP (ConnectionLess Network Service, 无连接网络服务)。首先被定义的是 CONS, 它要求通信节点之间在进行数据传输前首先要建立连接。其后定义的 CLNP 则不需要通信节点在传输数据前预先建立端到端的连接, 而是允许每个数据包被沿途网络设备独立转发。

IS-IS 属于内部网管协议, 用于自治系统内部。IS-IS 是一种链路状态路由, 使用最短路径优先 (Shortest Path First, SPF) 算法进行路由计算。IS-IS 可支持大中型网络, 而且路由收敛速度快, 因此可以作为除 OSPF 协议外的另一选择。

OSPF 协议是专门为 IP 协议设计的路由选择协议, 而 IS-IS 是 OSI 专门为 CLNP 协议所设计的路由选择协议。

IS-IS 的基本术语如下:

- IS (Intermediate System, 中间系统): 相当于 TCP/IP 中路由器, 是 IS-IS 中生成路由和传播路由信息的基本单元。
- ES (End System, 终端系统): 相当于 TCP/IP 中主机系统。由于 ES 不参与 IS-IS 路由协议的处理, 所以 ISO 使用专门的 ES-IS 协议定义终端系统与中间系统的通信。
- RD (Routing Domain, 路由域): 在一个路由域中多个 IS 通过相同的路由协议来交换路由信息。
- Area (区域): 路由域的细分单元。类似于 OSPF 协议, IS-IS 允许将整个路由域划分为多个区域。
- ES-IS (End System to Intermediate System Routing Exchange Protocol, 终端系统到中间系统路由选择交换协议): 负责 ES 与 IS 之间的通信。

在路由域中, 路由器与终端系统间使用 ES-IS 协议来进行主机和网关发现, 而路由器之间使用 IS-IS 来进行邻居关系建立及路由信息交换。

在 ES-IS 中, ES 通过发送 ESH (End System Hello) 来告知 IS 自己的存在, 同时 IS 通过监听 ESH 来确定本网段是否存在 ES; IS 通过 ISH (Intermediate System Hello) 来通知 ES 自己的存在。ES 通过监听 ISH 确定网络中存在的 IS, 并选择其中一个 IS 作为自己的网关。ES 之间需要发送报文, 需要先发送给网关 IS, 由 IS 转发。

在 IS-IS 中, 路由器间通过 IIH (IS-to-IS Hello PDUs) 来建立邻居关系。每一个 IS 都会生成 LSPDU (Link State Protocol Data Unit, 链路状态协议数据单元, LSP), 此 LSP 包含了本 IS 的所有链路状态信息。通过 IS-IS 邻居间交换 LSP, 网络中每一个 IS 都生成了 LSDB (Link State DataBase, 链路状态数据库), 且所有的 IS 的 LSDB 中所包含的链路状态信息都是相同的。

5.2 IS-IS 路由级别

OSI 给路由定义了从 Level-0 到 Level-3 等 4 个路由级别：

(1) Level-0 路由。Level-0 路由存在于 ES 和 IS 之间，由 ES 和 IS 来完成。在 ES 发现了最近的 IS 后，主要完成的任务包括确定相连的区域地质，在 ES 和 IS 之间建立邻居关系以及完成网络地址到数据链路层地质的转换。

(2) Level-1 路由。Level-1 路由存在于同一区域内的不同 IS 间，所以又称为区域内路由。当 IS 要发送报文到另外一个 IS 时，查看报文中的目的地址，如果发现其位于同一区域的不同子网，IS 会选择最优的路径进行转发；如果目的地址不在同一区域，则 IS 把数据转发给本区域内最近的 Level-1-2 路由器，由它负责数据转发。

(3) Level-2 路由。Level-2 路由存在于同一路由域的区域间，所以又称区域间路由。当目的地址在不同区域时，IS 发送报文到最近的一个 Level-2 路由，由它负责转发到另外一个区域。

(4) Level-3 路由。Level-3 路由存在于路由域间。每个路由域相当于一个自治系统，彼此间通过 IDRP (Inter Domain Routing Protocol, 域间路由协议) 相连接。IS-IS 协议涉及 Level-1 和 Level-2 路由。

5.3 IS-IS 与 OSPF 的区别

IS-IS 与 OSPF 的区别：

(1) 区域边界不同。在 OSPF 中，区域的分界点在路由器上。一个路由器的不同可以属于不同区域。在集成化 IS-IS, 区域的分界点在链路上，一个路由器只能属于一个区域，因而没有 ABR 的概念。

(2) 骨干区域不同。在 OSPF 中，只有 Area0 是骨干区域, 其他区域均为非骨干区域。所有非骨干区域必须直接连接到骨干区域。骨干区域必须是连续的。

集成化 IS-IS 中，并没有规定哪个区域是骨干区域。所有 Level-2 路由器和 Level-1-2 路由器构成了 IS-IS 的骨干网，它们可以属于不同的区域，但必须是物理连接的。

OSPF 与 IS-IS 都是基于链路状态的动态路由协议，工作机制相似 (LSA 的通告、同步 LSDB、老化等)，它们的收敛速度都较快，它们都有区域划分的概念。

集成化 IS-IS 可同时支持 IP 和 OSI，这样在多协议网络中有优势；而 OSPF 协议是专为为 IP 设计的协议。

5.4 IS-IS 协议报文类型及作用

在链路层头部定义了所使用的 MAC 地址为 0180-c200-014 (Level-1 报文) 和 0180-c200-0015 (Level-2 报文)，在链路层上 IS-IS 协议报文的协议号是 0x83。

IS-IS 报文共有 4 类：IIH (IS-IS Hello PDUs)、LSP (Link State PDUs)、CSNP (Complete Sequence Numbers PDUs) 和 PSNP (Partial Sequence Numbers PDUs)。其中 PDU 是协议数据单元，表示这些报文直接封装在数据链路层的帧中。

- IIH 报文也是 Hello 报文用于建立和维持邻居关系。其中，广播网络中的 Level-1 路由器使用 Level-1 LAN IIH, 广播网络中的 Level-2 路由器使用 Level-2 LAN IIH。点到点网络中的路由器则使用 P2P IIH。
- LSP 报文(链路状态报文)用于交换链路状态。LSP 分为两种：Level-1 LSP 和 Level-2 LSP。Level-1 路由器传送 Level-1 LSP, Level-2 路由器传送 Level-2 LSP。

- SNP 报文也就是时序报文用于确认邻居之间最新接收的 LSP, 作用类似于确认报文, 但更有效。SNP 包括 CSNP (全时序报文) 和 PSNP (部分时序报文), 进一步又可分为 Level-1 CSNP、Level-2 CSNP、Level-1 PSNP 和 Level-2 PSNP。CSNP 包括了 LSDB 的所有 LSP 的摘要信息, 从而可以在相邻路由器间保持 LSDB 的同步。在广播网络上, CSNP 由 DIS 定期发送 (默认发送周期是 10s); 在点到点链路上, CSNP 只在第一次建立邻接关系时候发送。

5.5 IS-IS 网络类型

1. 网络类型

与 OSPF 协议不同, IS-IS 只支持广播类型和点到点类型网络。默认情况下, 路由器接口网络类型由物理链路决定。在广播型的链路 (如 Ethernet、Token-Ring) 上, 接口的默认类型是广播类型; 而在点对点链路 (如 PPP、HDLC), 接口的默认网络类型是点对点网络。对于 NBMA 网络, 需要对其配置子接口, 并将子接口类型配置到点到点网络或广播网络。IS-IS 不能再点到多点链路上运行。当网络类型为广播类型时, 需要选举 DIS 来同步 LSDB, 而在点到点网络中。

在 IS-IS 中, 路由器类型由 Level-1、Level-2 和 Level-1-2 三种, 所能够建立的邻居关系由 Level-1 和 Level-2 两种。属于不同区域的 Level-1 路由器不能形成邻居, 而 Level-2 路由器是否形成邻居关系跟区域无关。

2. 邻居关系的建立

在点对点网络中, 当 IS 的端口物理层 UP 后, 邻居的初始状态为 init。当收到对端发出的 P2P IIH 报文后, IS 检查报文中的相关参数, 如果参数一致, 则邻居状态转化为 UP。

在广播网络中, 两个路由器 R1 和 R2 之间邻居关系的建立需要三方握手过程。具体过程如下:

- (1) R1 的物理端口 UP, 发送 LAN IIH 报文, 报文中的 Neighbor 字段为空;
- (2) R2 收到 R1 的 Hello 报文后, 将 R1 的 MAC 地址放在邻居字段中, 发送出去, 邻居状态变为 Init。
- (3) R1 收到 R2 的 Hello 报文后, 发现自己的 MAC 地址在邻居列表中, 双向关系确认, 邻居状态变为 UP, 并将 R2 的接口 MAC 地址放在邻居列表中发送出去; R2 收到该 Hello 报文后, 发现自己的 MAC 地址也在对方的邻居列表中, 双向关系确认, 邻居状态 UP。

在点对点网络中, 只要邻居关系建立, 则邻接关系建立; 在广播网络中, 同一链路上的同一级别的路由器都会形成邻接关系, 包括所有的非 DIS 路由器之间也会形成邻接关系。

3. LSDB 的同步

邻接关系建立后, 邻居 IS 间进行 LSDB 的同步。同步过程主要由邻居间交互 LSP 和 SNP 来完成的。

LSP 用于描述链路状态信息。每一个 IS 产生一个 (或多个) LSP 来描述它与周围邻居 IS 的连接; LSP 报文中包含了发送者的 System ID 和序列号。如果 LSP 中包含的链路状态信息量太大, 报文太大, 则可以进行分片。Level-1 路由器产生 Level-1 LSP, 仅仅在区域内传播; Level-2 路由器产生 Level-2 LSP, 在骨干网内传播。

SNP 用于描述 LSDB 中 LSP 的摘要信息, 并对邻居之间最新接收的 LSP 进行确认。

在广播网络中，路由器之间的邻居状态变成 UP 后，为了加快链路状态数据库的同步过程，会先向邻居发送自己 LSDB 中的所有 LSP，从而使所有邻居的 LSDB 快速进入同步状态。

在同步状态下，DIS 会周期性的发送 CSNP 报文，以使区域中所有 IS 具有相同的 LSDB 数据库信息。如果某一个邻居路由器收到 CSNP 报文后，发现自己的 LSDB 与 DIS 没有达到同步状态，它会发送 PSNP 来请求 DIS 重新发送相应 LSP。DIS 收到请求后，重新发送相应的 LSP。

如果网络拓扑发生变化，则路由器直接发送描述该变化的 LSP。如果某种原因导致 DIS 未收到该 LSP，则在周期性的 CSNP 中没有该 LSP。路由器根据从 DIS 收到的 CSNP 发现没有该 LSP，则路由器会再次发送该 LSP，直到 DIS 发送的 CSNP 中包含该 LSP 为止。

在点对点网络中，路由器之间的邻居状态变为 UP 后，先向邻居发送自己 LSDB 中所有的 LSP 以快速进入同步状态。

与广播网络不同的是，CSNP 只是在邻居建立后发送一次，而不是周期性发送。此后，路由器之间的互相发送各自 LSDB 中 LSP，并使用 PSNP 对收到的 LSP 进行确认。

如果某种原因导致路由器所发送的 LSP 未到达的对端，则不会受到对端的 PSNP 报文。路由器会重新发送 LSP，直到收到对端的 PSNP 报文。

5.6 IS-IS 简单用例

5.6.1 基础用例

如图 5.1 所示，在两个路由器及一个终端使用 IS-IS 链路动态协议，使得该区域内互相可以到达，路由之间可以相互学习。

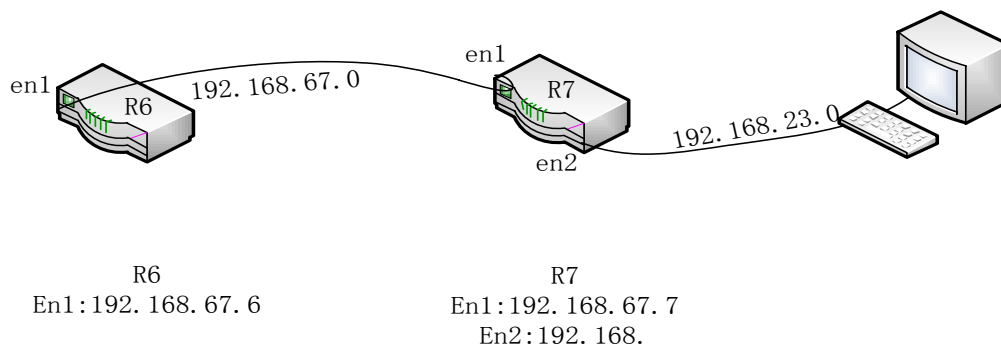


图 5.1 IS-IS 协议用例

配置 IS-IS 协议具体步骤如下：

首先分别将五台路由器开启 zebra 核心服务以及 is-is 协议。

```
root# zebra -d
root# isisd -d
```

然后在相对应的协议配置文件中设置相应的输出信息，在 isisd.conf 文件中配置如下：

```
hostname isisd
password zebra
log stdout
```

根据路由器每个接口的 IP 配置，通过指令进入相应配置模式：

```
isisd > enable
isisd# configur terminal
isisd(config)#
```

在路由器 R6 上:

```
isisd(config)#router isis test6           //启动 IS-IS 路由进程
isisd(cofig-router)#net 49.0001.1111.1111.00 //配置 NET 地址
isisd(config)#interface en1
isisd(cofnig-if)#ip router isis test6
```

在路由器 R7 上:

```
isisd(config)#router isis test7
isisd(cofig-router)#net 49.0001.2222.2222.00
isisd(cofnig-if)#ip router isis test7
isisd(cofnig-if)#exit
isisd(config)#interface en1
isisd(cofnig-if)#ip router isis test7
isisd(cofnig-if)#end
```

如上所示,配置 IS-IS 需要三条基本命令分别是创建 IS-IS 进程,配置 IS-IS 进程的网络实体名称,以及在指定接口上使能 IS-IS 路由进程的命令。配置完成后,路由器与终端都可以相互到达。

第6章 BGP

6.1 BGP 基本介绍

BGP 是一种外部网关协议 (EGP)，与 OSPF、RIP 等内部网关协议不同，其关心的重点不在于发现和计算路由，而在于 AS 之间传递路由信息以及控制优化路由信息。

BGP 是一种“路径矢量”路由协议，其路由信息携带了所经过的全部 AS 路径列表。这样，接收该路由信息的 BGP 路由器可以很明确的直到此路由是否源于自己的 AS。如果路由源于自己的 AS，BGP 路由会丢弃此条路由，这样就可以从根本上避免 AS 之间产生环路的可能性。

为了保证 BGP 协议的可靠传输，其使用 TCP 协议来承载，端口号为 179。TCP 协议天然的可靠传输、重传、排序等机制保证了 BGP 协议的可靠性。

BGP 能够支持 CIDR 和路由聚合，可以将一些连续的子网聚合成较大的子网，从而可以在一定程度上控制路由表的快速增长，并降低了路由查找的复杂度。

在邻居关系建立后，BGP 路由器会将自己的全部路由信息通告给邻居。此后，如果路由表有变化，则只将增量部分发送给邻居。这样可以大大减少 BGP 传播路由信息占用的带宽，有利于在 Internet 上传播大量的路由信息，并降低路由器 CPU 与内存资源的消耗。

与 IGP 不同，BGP 最重要的特性是丰富的路由属性以及强大的路由过滤和路由策略。通过使用路由策略来更改路由属性，或者根据路由更新信息中属性来实现路由过滤和路由策略，从而使 BGP 的使用者可以非常灵活的对路由进行选路以及控制。

路由策略是根据一些规则，使用某种策略改变规则中影响路由发布、接收或路由选择的参数而改变路由发现的结果，最终改变的是路由表的内容。是在路由发现的时候产生作用。

策略路由是尽管存在当前最优的路由，但是针对某些特别的主机(或应用、协议)不使用当前路由表中的转发路径而单独使用别的转发路径。在数据包转发的时候发生作用、不改变路由表中任何内容。

策略路由的优先级比路由策略高，当路由器接收到数据包，并进行转发的时候，会优先根据策略路由的规则进行匹配，如果能匹配上，则根据策略路由来转发，否则按照路由表中转发路径来进行转发。

总结一下，路由策略是路由发现规则，策略路由是数据包转发规则。其实将“策略路由”理解为“转发策略”，这样更容易理解与区分。由于转发在底层，路由在高层，所以转发的优先级比路由的优先级高，这点也能理解的通。

其实路由器中存在两种类型和层次的表，一个是路由表(routing-table)，另一个是转发表(forwording-table)。转发表是由路由表映射过来的，策略路由直接作用于转发表，路由策略直接作用于路由表。

6.2 BGP 基本术语

相关的 BGP 基本术语如下：

(1) BGP 发言者 (BGP Speaker)。发送 BGP 消息的路由器称为 BGP 发言者 (BGP Speaker)，它接收或产生新的路由信息，并发布给其他 BGP 发言者。

(2) Router ID。Router ID 是一个 32 位的无符号整数，用来在自治系统中唯一标识一台路由器。路由器如果要运行 BGP，则必须存在 Router ID 可由管理员手工指定或由协议自动选举。

(3) BGP 对等体 (BGP Peer)。相互交换信息的 BGP 发言者之间互称对等体。

(4) IBGP 对等体 (Internet BGP Peer)。如果 BGP 对等体位于同一自治系统内，称为 IBGP 对等体。IBGP 对等体不一定是物理直连，但是一定是 TCP 直达。

(5) EBGp 对等体 (External BGP Peer)。BGP 对等体处于不同的自治系统 (AS) 被称为 EBGp 对等体。尽管 BGP 是基于 TCP 的，通常情况下，协议要求建立 EBGp 连接的路由器之间具有直连的物理链路。如果路由器之间不是物理直达，则可以配置 BGP 以允许之间经过物理多跳而建立链路。

BGP 发言者从 EBGp 对等体获得路由后，会向所有的 BGP 对等体 (包括 IBGP 和 EBGp 对等体) 通告这些路由。与此同时，为了防止路由环路，他不会将学习到的路由再向原发布者发布。

为了防止路由环路，BGP 协议规定 BGP 发言者从 IBGP 获得的路由不向它的 IBGP 对等体发布。另外，为了防止产生路由黑洞，协议还规定 BGP 发言者从 IBGP 获得的路由是否发布给它的 EBGp 对等体与是否同步相关。

TCP 的可靠传输机制和滑动窗口机制可以确保承载于 TCP 之上的 BGP 可以可靠传递大量路由。但是，由于 TCP 连接是以点到点的单播方式来进行报文传输的，因而 BGP 连接只能是基于点到点的连接。在运行 BGP 协议的 AS 内，为了确保所有 BGP 路由器的路由信息相同，需要使所有的 IBGP 路由器保持全连接。

6.3 BGP 的选路规则

对于一个 BGP 路由器来说，其路由的来源有两种：从对等体接收和从 IGP 引入。

如图 6.1 所示，BGP 发言者从对等体接收到 BGP 路由后，其基本的操作过程为：接收路由过滤与属性设置以--路由聚合--路由优选--路由安装--发布策略—发布路由过滤、属性设置与路由聚合。

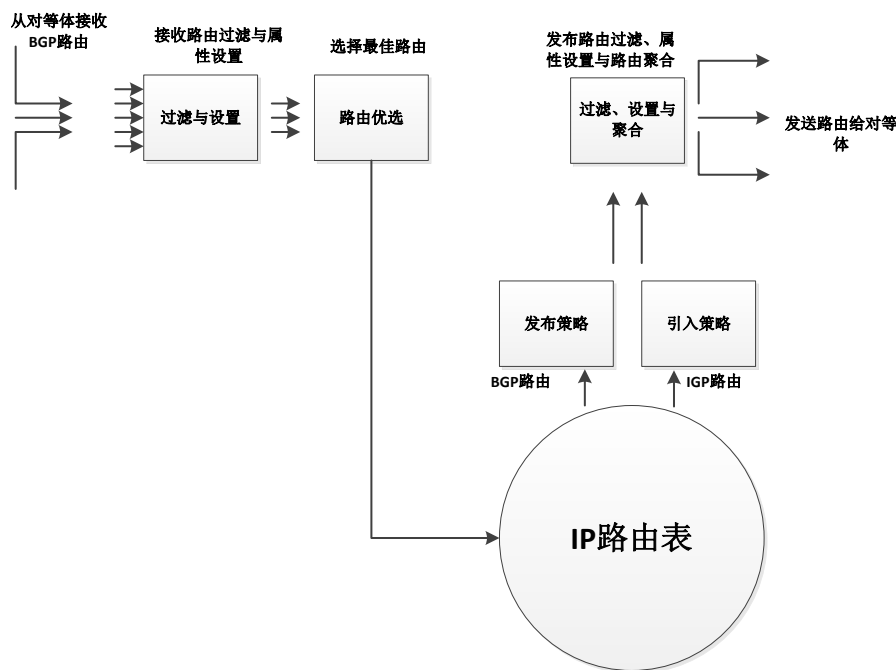


图 6.1 BGP 路由处理流程

(1) 接收路由过滤与属性设置是 BGP 发言者在对等体接收到路由后的第一步工作。BGP 接收到路由后，根据配置的接收策略对接收到的 BGP 进行匹配与过滤（根据路由携带的属性），并对其设置的相关属性。

(2) 完成了接收策略的匹配后，如果有需要的话，BGP 将对接收到的路由进行聚合，合并其中的路由，以此减少路由表的规模。

(3) 完成了接收策略的匹配具体路由合并后，BGP 将对接收到的路由优选。对于到达同一目的地址的多条 BGP 路由，BGP 发言者只选择最佳的路由给自己使用，并将此最佳路由安装到 IP 路由表中，成为有效路由。

(4) 在向 BGP 对等体发布路由的时候，BGP 发言者需要依据一定的发布策略，将对已经安装到自己 IP 路由表的部分有效路由进行发布。

(5) 同时，BGP 路由器需要执行发布路由过滤与属性设置，然后将通过过滤的 BGP 路由发送给自己的 BGP 对等体。如果有需要的话，BGP 发言者将对路由进行聚合，合并其中的具体路由，减少路由表的规模。

而对于 IGP 路由，则需要先经过引入策略的过滤和属性设置，将 IGP 路由表中的有效路由引入到 BGP 路由表中，然后才能进行发布路由过滤与属性设置，并将过滤后的路由发送给自己的 BGP 对等体。

在 MSR 上，BGP 会丢弃下一跳不可达的路由。如果到达目的有多条路由，则按照如下顺序选择最优路由。

- (1) 首先丢弃下一跳（NEXT_HOP）不可达的路由；
- (2) 优选 Preferred-value 值最大的路由；
- (3) 优选本地优先级（LOCAL_PREF）最高的路由；
- (4) 优选聚合路由；
- (5) 优选 AS 路径（AS_PATH）最短的路由；

- (6) 依次选择 ORIGIN 属性为 IGP、EGP、Incomplete 的路由；
- (7) 优选 MED 值最低的路由；
- (8) 依次选择从 EBGp、联盟、IBGP 学到的路由；
- (9) 优选下一跳度量值 (Metric) 最低的路由；
- (10) 优选 CLUSTER_LIST 长度最短的路由；
- (11) 优选 ORIGINATOR_ID 最小的路由；
- (12) 优选 Router ID 最小的路由器发布的路由；
- (13) 优选地址最小的对等体发布的路由。

BGP 协议本身可以选出唯一一条到达目的网段的最优路由，要实现 BGP 敷在分担，就需要在 MSR 路由器上配置允许 BGP 进行负载分担。

BGP 的负载分担与 IGP 的负载分担有所不同。

(1) IGP 通过协议定义的路由算法，对到达同一目的地址的不同路由进行计算。根据计算结果，将度量值相等的路由进行负载分担。

(2) BGP 本身并没有路由计算的算法，只是一个选路的路由协议。因此，BGP 不能根据一个明确的度量值决定是否对路由进行负载分担，但 BGP 有丰富的选路规则，可以在对路由进行的一定的选择后，有条件的进行负载分担，也就是将负载分担加入到 BGP 的选路规则中去。

在 BGP 中，由于协议本身的特殊性，它产生的路由的下一跳可能不是当前路由器直接连接的邻居。常见的原因就是 IBGP 之间发布路由信息时不改变下一跳，为了使报文能够正确的发送出去，路由器必须先找到一个可直接可达的地址（查找 IGP 建立的表项），通过这个地址到达路由表中指示的下一跳。在上述过程中，去往直接可达地址的路由成为依赖路由，BGP 路由依赖这些路由指导报文转发。根据下一跳的地址找到依赖路由的过程成为路由迭代。路由器支持基于迭代的 BGP 负载分担，即如果依赖路由本身是负载分担的，则 BGP 也会生成相同数量的下一跳地址来指导报文转发。需要说明的是，基于迭代的 BGP 负载分担并不需要命令配置。

BGP 发布路由时采用如下策略：

- (1) 存在多条有效路由时，BGP 发言者只将最优路由发布给对等体；
- (2) BGP 发言者只将自己使用的路由发布给对等体；
- (3) BGP 发言者从 EBGp 获得的路由会向它所有的 BGP 对等体发布；
- (4) BGP 发言者从 IBGP 获得路由不向它的对等体发布；
- (5) BGP 发言者从 IBGP 获得的路由发布给它的 EBGp 对等体；关闭 BGP 与 IGP 同步情况下，IBGP 路由被直接发布；开启 BGP 与 IGP 同步的情况下，该 IBGP 路由只有在 IGP 也发布了这条路由时才会被同步并发布给 EBGp 对等体
- (6) BGP 一旦建立连接，BGP 发言者将把自己的所有的 BGP 路由发布给新对等体。

6.4 BGP 简单用例

6.4.1 不同路由协议下通信用例 (ospf-rip)

如图 6.2 所示，R1、R2、R3 路由处于同一个自治系统 (AS) 中，所用动态协议为 OSPF 动态路由协议，R4 和 R5 处于同一个 AS 中，所有协议为 RIP 协议。两个自治系统之间需要通过 BGP 协议来通信，其中 R3 和 R4 为 BGP 发言者，互为 EBGp 对等体。

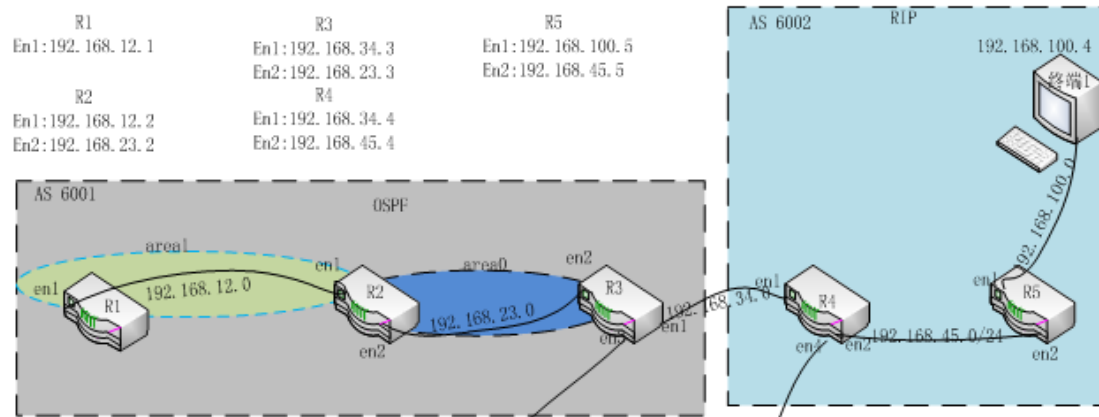


图 6.2 BGP 协议通信 (ospf-rip)

配置 BGP 协议具体步骤如下：

首先分别将五台路由器开启 zebra 核心服务以及 ospf、rip、bgp 协议。

```
root# zebra -d
root# ospfd -d
root# ripd -d
root# bgpd -d
```

然后在相对应的协议配置文件中设置相应的输出信息，在 bgpd.conf 文件中配置如下：

```
hostname bgpd
password zebra
log stdout
```

根据路由器每个接口的 IP 配置，通过指令进入相应配置模式：

```
bgpd > enable
bgpd# configur terminal
bgpd(config)#
```

由图 6.2 可知 OSPF 协议配置区域为 AS6001，在该区域配置路由器步骤如下：

在路由器 R1 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.81``
ospfd(config-router)#network 192.168.12.0/24 area 1
ospfd(config-router)#end
```

在路由器 R2 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.82
ospfd(config-router)#network 192.168.12.0/24 area 1
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#end
```

在路由器 R3 上:

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.83
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#end
```

而 RIP 协议配置区域为 AS6002, 在该区域内配置 RIP 协议步骤如下:

在路由器 R4 上:

```
ripd(config)#router rip
ripd(config-router)#network 192.168.45.0/24
ripd(config-router)#end
```

在路由器 R5 上:

```
ripd(config)#router rip
ripd(config-router)#network 192.168.45.0/24
ripd(config-router)#network 192.168.100.0/24
ripd(config-router)#end
```

BGP 协议需要配置在每一台路由器上, 配置步骤如下:

在路由器 R1 上:

```
bgpd(config)#router bgp 6001
bgpd(config-router)#bgp router-id 192.168.1.81
bgpd(config-router)#neighbor 192.168.12.2 remote-as 6001 //neighbor
bgpd(config-router)#neighbor 192.168.23.3 remote-as 6001
bgpd(config-router)#network 192.168.12.0/24 //route
bgpd(config-router)#end
```

在路由器 R2 上:

```
bgpd(config)#router bgp 6001
bgpd(config-router)#bgp router-id 192.168.1.82
bgpd(config-router)#neighbor 192.168.12.1 remote-as 6001
bgpd(config-router)#neighbor 192.168.23.3 remote-as 6001
bgpd(config-router)#network 192.168.12.0/24 //route
bgpd(config-router)#network 192.168.23.0/24 //route
bgpd(config-router)#end
```

在路由器 R3 上:

```
bgpd(config)#router bgp 6001
bgpd(config-router)#bgp router-id 192.168.1.83
bgpd(config-router)#neighbor 192.168.12.1 remote-as 6001
bgpd(config-router)#neighbor 192.168.23.2 remote-as 6001
```

```
bgpd(config-router)#neighbor 192.168.34.4 remote-as 6002
bgpd(config-router)#network 192.168.34.0/24
bgpd(config-router)#network 192.168.23.0/24
bgpd(config-router)#neighbor 192.168.23.2 next-hop-self
bgpd(config-router)#neighbor 192.168.12.1 next-hop-self
bgpd(config-router)#end
```

在路由器 R4 上:

```
bgpd(config)#router bgp 6002
bgpd(config-router)#bgp router-id 192.168.1.84
bgpd(config-router)#neighbor 192.168.34.3 remote-as 6001
bgpd(config-router)#neighbor 192.168.45.5 remote-as 6002
bgpd(config-router)#neighbor 192.168.45.5 next-hop-self
bgpd(config-router)#network 192.168.34.0/24
bgpd(config-router)#network 192.168.45.0/24
bgpd(config-router)#end
```

在路由器 R5 上:

```
bgpd(config)#router bgp 6002
bgpd(config-router)#bgp router-id 192.168.1.85
bgpd(config-router)#neighbor 192.168.45.4 remote-as 6002
bgpd(config-router)#network 192.168.45.0/24
bgpd(config-router)#network 192.168.100.0/24
bgpd(config-router)#end
```

R3 和 R4 都是 BGP 发言者，且互为 EBGP 对等体，需要告诉各自区域内的其他路由器想要到达另外的自治系统必须把 R3 或者 R4 当做下一跳。

6.4.2 不同路由协议通信 (ospf-rip-isis)

如图 6.3 BGP 协议用例 (ospf-rip-isis) 所示，有三个自治系统 (AS)。不同的 AS 运行不同的路由动态协议，AS6001 运行 OSPF 协议，AS6002 运行 RIP 协议，AS6003 运行 IS-IS 协议，三个自治系统通过外部网关协议 BGP 来通信。其中，R3、R4、R6 都是 BGP 发言者，且互为 EBGP 对等体。

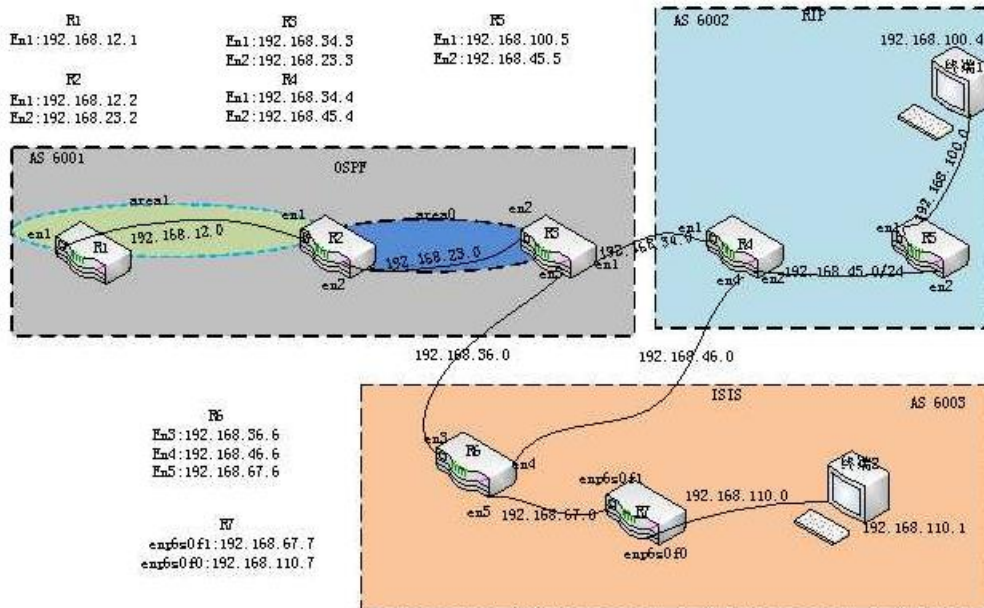


图 6.3 BGP 协议用例 (ospf-rip-isis)

配置 BGP 协议具体步骤如下：

首先分别将五台路由器开启 zebra 核心服务以及 ospf、rip、isis、bgp 协议。

```
root# zebra -d
root# ospfd -d
root# ripd -d
root# bgpd -d
root# isisd -d
```

然后在相对应的协议配置文件中设置相应的输出信息，在 bgpd.conf 文件中配置如下：

```
hostname bgpd
password zebra
log stdout
```

根据路由器每个接口的 IP 配置，通过指令进入相应配置模式：

```
bgpd > enable
bgpd# configur terminal
bgpd(config)#
```

在 AS6001 中，配置 OSPF 协议步骤如下：

在路由器 R1 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.81``
ospfd(config-router)#network 192.168.12.0/24 area 1
ospfd(config-router)#end
```

在路由器 R2 上：

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.82
ospfd(config-router)#network 192.168.12.0/24 area 1
```

```
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#end
```

在路由器 R3 上:

```
ospfd(config)#router ospf
ospfd(config-router)#ospf router-id 192.168.1.83
ospfd(config-router)#network 192.168.23.0/24 area 0
ospfd(config-router)#end
```

而 RIP 协议配置区域为 AS6002, 在该区域内配置 RIP 协议步骤如下:

在路由器 R4 上:

```
ripd(config)#router rip
ripd(config-router)#network 192.168.45.0/24
ripd(config-router)#end
```

在路由器 R5 上:

```
ripd(config)#router rip
ripd(config-router)#network 192.168.45.0/24
ripd(config-router)#network 192.168.100.0/24
ripd(config-router)#end
```

在 AS6003 内, 配置 IS-IS 协议步骤如下:

在路由器 R6 上:

```
isisd(config)#router isis test6 //启动 IS-IS 路由进程
isisd(config-router)#net 49.0001.1111.1111.1111.00 //配置 NET 地址
isisd(config)#interface en3
isisd(config-if)#ip router isis test6
isisd(config)#interface en4
isisd(config-if)#ip router isis test6
isisd(config)#interface en5
isisd(config-if)#ip router isis test6
```

在路由器 R7 上:

```
isisd(config)#router isis test7
isisd(config-router)#net 49.0001.2222.2222.2222.00
isisd(config)#interface enp6s0f1
isisd(config-if)#ip router isis test7
isisd(config-if)#exit
isisd(config)#interface enp6s0f0
isisd(config-if)#iprouter isis test7
isisd(config-if)#end
```

配置终端网卡:

```
#Ifconfig enp3s0 inet 192.168.110.1
#Ifconfig enp3s0 gateway 192.168.110.7
```

配置 BGP 协议步骤如下:

在路由器 R1 上:

```
bgpd(config)#router bgp 6001
bgpd(config-router)#bgp router-id 192.168.1.81
bgpd(config-router)#neighbor 192.168.12.2 remote-as 6001 //neighbor
bgpd(config-router)#neighbor 192.168.23.3 remote-as 6001
bgpd(config-router)#network 192.168.12.0/24 //route
bgpd(config-router)#end
```

在路由器 R2 上:

```
bgpd(config)#router bgp 6001
bgpd(config-router)#bgp router-id 192.168.1.82
bgpd(config-router)#neighbor 192.168.12.1 remote-as 6001
bgpd(config-router)#neighbor 192.168.23.3 remote-as 6001
bgpd(config-router)#network 192.168.12.0/24 //route
bgpd(config-router)#network 192.168.23.0/24 //route
bgpd(config-router)#end
```

在路由器 R3 上:

```
bgpd(config)#router bgp 6001
bgpd(config-router)#bgp router-id 192.168.1.83
bgpd(config-router)#neighbor 192.168.12.1 remote-as 6001
bgpd(config-router)#neighbor 192.168.23.2 remote-as 6001
bgpd(config-router)#neighbor 192.168.34.4 remote-as 6002
bgpd(config-router)#neighbor 192.168.36.6 remote-as 6003
bgpd(config-router)#network 192.168.36.0/24
bgpd(config-router)#network 192.168.34.0/24
bgpd(config-router)#network 192.168.23.0/24
bgpd(config-router)#neighbor 192.168.23.2 next-hop-self
bgpd(config-router)#neighbor 192.168.12.1 next-hop-self
bgpd(config-router)#end
```

在路由器 R4 上:

```
bgpd(config)#router bgp 6002
bgpd(config-router)#bgp router-id 192.168.1.84
bgpd(config-router)#neighbor 192.168.34.3 remote-as 6001
bgpd(config-router)#neighbor 192.168.45.5 remote-as 6002
bgpd(config-router)#neighbor 192.168.46.6 remote-as 6003
bgpd(config-router)#neighbor 192.168.45.5 next-hop-self
bgpd(config-router)#network 192.168.34.0/24
bgpd(config-router)#network 192.168.45.0/24
bgpd(config-router)#network 192.168.46.0/24
```

```
bgpd(config-router)#end
```

在路由器 R5 上:

```
bgpd(config)#router bgp 6002
bgpd(config-router)#bgp router-id 192.168.1.85
bgpd(config-router)#neighbor 192.168.45.4 remote-as 6002
bgpd(config-router)#network 192.168.45.0/24
bgpd(config-router)#network 192.168.100.0/24
bgpd(config-router)#end
```

在路由器 R6 上:

```
bgpd(config)#router bgp 6003
bgpd(config-router)#bgp router-id 192.168.1.86
bgpd(config-router)#neighbor 192.168.36.3 remote-as 6001
bgpd(config-router)#neighbor 192.168.46.4 remote-as 6002
bgpd(config-router)#neighbor 192.168.67.7 remote-as 6003
bgpd(config-router)#network 192.168.36.0/24
bgpd(config-router)#network 192.168.46.0/24
bgpd(config-router)#network 192.168.67.0/24
bgpd(config-router)#end
```

在路由器 R7 上:

```
bgpd(config)#router bgp 6003
bgpd(config-router)#bgp router-id 192.168.1.87
bgpd(config-router)#neighbor 192.168.67.6 remote-as 6003
bgpd(config-router)#network 192.168.67.0/24
bgpd(config-router)#network 192.168.110.0/24
bgpd(config-router)#end
```


销售与服务网络

北京翼辉信息技术有限公司

地址：北京市海淀区中关村翠湖科技园 12 号楼

电话：010-56082458

传真：010-56082457

邮箱：acoinfo@acoinfo.com

南京翼辉信息技术有限公司

地址：南京市雨花台区软件大道 180 号大数据产业基地 7 幢 6 楼

电话：025-83127300

传真：025-83127399

邮箱：nanjing@acoinfo.com



翼辉信息官网
www.acoinfo.com



SylixOS社区
www.sylixos.com



翼辉信息公众号
acoinfo

请您用以上方式联系我们，我们会为您安排产品现场演示，感谢您对我公司产品的关注！