

## 4.2 路由算法

### 4.2.1 静态路由与动态路由

路由器转发分组是通过路由表转发的，而路由表是通过各种算法得到的。从能否随网络的通信量或拓扑自适应地进行调整变化来划分，路由算法可以分为如下两大类。

**静态路由算法**（又称 **非自适应路由算法**）。指由网络管理员手工配置的路由信息。当网络的拓扑结构或链路的状态发生变化时，网络管理员需要手工去修改路由表中相关的静态路由信息。它不能及时适应网络状态的变化，对于简单的小型网络，可以采用静态路由。

**动态路由算法**（又称 **自适应路由算法**）。指路由器上的路由表项是通过相互连接的路由器之间彼此交换信息，然后按照一定的算法优化出来的，而这些路由信息会在一定时间间隙里不断更新，以适应不断变化的网络，随时获得最优的寻路效果。

静态路由算法的特点是简便和开销较小，在拓扑变化不大的小网络中运行效果很好。动态路由算法能改善网络的性能并有助于流量控制；但算法复杂，会增加网络的负担，有时因对动态变化的反应太快而引起振荡，或反应太慢而影响网络路由的一致性，因此要仔细设计动态路由算法，以发挥其优势。常用的动态路由算法可分为两类：距离—向量路由算法和链路状态路由算法。

### 4.2.2 距离—向量路由算法

在距离—向量路由算法中，所有结点都定期地将它们的整个路由选择表传送给所有与之直接相邻的结点。这种路由选择表包含：

- 每条路径的目的地（另一结点）。
- 路径的代价（也称 **距离**）。

**注意：**这里的距离是一个抽象的概念，如 RIP 就将距离定义为「跳数」。跳数指从源端口到达目的端口所经过的路由器个数，每经过一个路由器，跳数加 1。

在这种算法中，所有结点都必须参与距离向量交换，以保证路由的有效性和一致性，也就是说，所有的结点都监听从其他结点传来的路由选择更新信息，并在下列情况下更新它们的路由选择表：

1. 被通告一条新的路由，该路由在本结点的路由表中不存在，此时本地系统加入这条新的路由。
2. 发来的路由信息中有一条到达某个目的地的路由，该路由与当前使用的路由相比，有较短的距离（较小的代价）。此种情况下，就用经过发送路由信息的结点的新路由替换路由表中到达那个目的地的现有路由。

距离—向量路由算法的实质是，迭代计算一条路由中的站段数或延迟时间，从而得到到达一个目标的最短（最小代价）通路。它要求每个结点在每次更新时都将它的全部路由表发送给所有相邻的结点。显然，更新报文的大小与通信子网的结点数成正比，大的通信子网将导致很大的更新报文。由于更新报文发给直接邻接的结点，所以所有结点都将参加路由选择信息交换。基于这些原因，在通信子网上传送的路由选择信息的数量很容易变得非常大。

最常见的距离—向量路由算法是 RIP 算法，它采用「跳数」作为距离的度量。

### 4.2.3 链路状态路由算法

链路状态路由算法要求每个参与该算法的结点都具有完全的网络拓扑信息，它们执行下述两项任务。第一，主动测试所有邻接结点的状态。两个共享一条链接的结点是相邻结点，它们连接到同一条链路，或者连接到同一广播型物理网络。第二，定期地将链路状态传播给所有其他结点（或称路由结点）。典型的链路状态算法是 OSPF 算法。

在一个链路状态路由选择中，一个结点检查所有直接链路的状态，并将所得的状态信息发送网，上的所有其他结点，而不是仅送给那些直接相连的结点。每个结点都用这种方式从网上所其他的结点接收包含直接链路状态的路由选择信息。

每当链路状态报文到达时，路由结点便使用这些状态信息去更新自己的网络拓扑和状态「视野图」，一旦链路状态发生变化，结点就对更新的网络图利用 Dijkstra 最短路径算法重新计算由，从单一的源出发计算到达所有目的结点的最短路径。

链路状态路由算法主要有三个特征：

1. 向本自治系统中所有路由器发送信息，这里使用的方法是泛洪法，即路由器通过所有端口向所有相邻的路由器发送信息。而每个相邻路由器又将此信息发往其所有相邻路由器（但不再发送给刚刚发来自信息的那个路由器）。
2. 发送的信息是与路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。所谓「链路状态」，是指说明本路由器与哪些路由器相邻及该链路的「度量」。对于 OSPF 算法，链路状态的「度量」主要用来表示费用、距离、时延、带宽等。
3. 只有当链路状态发生变化时，路由器才向所有路由器发送此信息。

由于一个路由器的链路状态只涉及相邻路由器的连通状态，而与整个互联网的规模并无直接关系，因此链路状态路由算法可以用于大型的或路由信息变化聚敛的互联网环境。

链路状态路由算法的主要优点是，每个路由结点都使用同样的原始状态数据独立地计算路径，而不依赖中间结点的计算；链路状态报文不加改变地传播，因此采用该算法易于查找故障。当一个结点从所有其他结点接收到报文时，它可以在本地立即计算正确的通路，保证一步汇聚。最后，由于链路状态报文仅运载来自单个结点关于直接链路的信息，其大小与网络中的路由结点数目无关，因此链路状态算法比距离—向量算法有更好的规模可伸展性。

距离—向量路由算法与链路状态路由算法的比较：在距离—向量路由算法中，每个结点仅与它的直接邻居交谈，它为它的邻居提供从自己到网络中所有其他结点的最低费用估计。在链路状态路由算法中，每个结点通过广播的方式与所有其他结点交谈，但它仅告诉它们与它直接相连的链路费用。相较之下，距离—向量路由算法有可能遇到路由环路等问题。

## 4.2.4 层次路由

当网络规模扩大时，路由器的路由表成比例地增大。这不仅会消耗越来越多的路由器缓冲区空间，而且需要用更多 CPU 时间来扫描路由表，用更多的带宽来交换路由状态信息。因此路由选择必须按照层次的方式进行。

因特网将整个互联网划分为许多较小的自治系统（注意一个自治系统中包含很多局域网），每个自治系统有权自主地决定本系统内应采用何种路由选择协议。如果两个自治系统需要通信，那么就需要一种在两个自治系统之间的协议来屏蔽这些差异。据此，因特网把路由选择协议划分为两大类：

1. 一个自治系统内部所使用的路由选择协议称为 **内部网关协议（IGP）**，也称 **域内路由选择**，具体的协议有 RIP 和 OSPF 等。
2. 自治系统之间所使用的路由选择协议称为 **外部网关协议（EGP）**，也称域间路由选择，用在不同自治系统的路由器之间交换路由信息，并负责为分组在不同自治系统之间选择最优的路径。具体的协议有 BGP。

使用层次路由时，OSPF 将一个自治系统再划分为若干区域（Area），每个路由器都知道在本区域内如何把分组路由到目的地的细节，但不用知道其他区域的内部结构。

采用分层次划分区域的方法虽然会使交换信息的种类增多，但也会使 OSPF 协议更加复杂。但这样做却能使每个区域内部交换路由信息的通信量大大减小，因而使 OSPF 协议能够用于规模很大的自治系统中。