该过程将要继续多久呢？你应认识到该循环将持续44次迭代（在y与z之间交换报文），即直到z最终算出它经由y的路径开销大于50为止。此时，z将（最终）确定它到x的最低开销路径是经过它到x的直接连接。y将经由z路由选择到x。关于链路开销增加的坏消息的确传播得很慢！如果链路开销$c(y, x)$从4变为10 000且开销$c(z, x)$为9999时将发生什么样的现象呢？由于这种情况，我们所见的问题有时被称为无穷计数（count-to-infinity）问题。

2. 距离向量算法：增加毒性逆转

刚才描述的特定循环的场景可以通过使用一种称为*毒性逆转*（poisoned reverse）的技术而加以避免。其思想较为简单：如果z通过y路由选择到目的地x，则z将通告y，它（即z）到x的距离是无穷大，也就是z将向y通告$D_z(x)=\infty$（即使z实际上知道$D_z(x)=5$）。只要z经y路由选择到x，z就持续地向y讲述这个善意的小谎言。因为y相信z没有到x的路径，故只要z继续经y路由选择到x（并这样去撒谎），y将永远不会试图经由z路由选择到x。

我们现在看一下毒性逆转如何解决我们前面在图5-7b中遇到的特定环路问题。作为毒性逆转的结果，y的距离表指示了$D_z(x)=\infty$。当(x, y)链路的开销在t_0时刻从4变为60时，y更新其表，虽然开销高达60，仍继续直接路由选择到x，并将到x的新开销通知z，即$D_y(x)=60$。z在t_1时刻收到更新后，便立即将其到x的路由切换到经过开销为50的直接(z, x)链路。因为这是一条新的到x的最低开销路径，且因为路径不再经过y，z就在t_2时刻通知y现在$D_z(x)=50$。在收到来自z的更新后，y便用$D_y(x)=51$更新其距离表。另外，因为z此时位于y到x的最低开销路径上，所以y通过在t_3时刻通知z其$D_y(x)=\infty$（即使y实际上知道$D_y(x)=51$）毒化从z到x的逆向路径。

毒性逆转解决了一般的无穷计数问题吗？没有。你应认识到涉及3个或更多节点（而不只是两个直接相连的邻居节点）的环路将无法用毒性逆转技术检测到。

3. LS与DV路由选择算法的比较

DV和LS算法采用互补的方法来解决路由选择计算问题。在DV算法中，每个节点仅与它的直接相连的邻居交谈，但它为其邻居提供了从它自己到网络中（它所知道的）所有其他节点的最低开销估计。LS算法需要全局信息。因此，当在每台路由器中实现时，例如像在图4-2和图5-1中那样，每个节点（经广播）与所有其他节点通信，但仅告诉它们与它直接相连链路的开销。我们通过快速比较它们各自的属性来总结所学的链路状态与距离向量算法。记住N是节点（路由器）的集合，而E是边（链路）的集合。

- *报文复杂性*。我们已经看到LS算法要求每个节点都知道网络中每条链路的开销。这就要求要发送$O(|N||E|)$个报文。而且无论何时一条链路的开销改变时，必须向所有节点发送新的链路开销。DV算法要求在每次迭代时，在两个直接相连邻居之间交换报文。我们已经看到，算法收敛所需时间依赖于许多因素。当链路开销改变时，DV算法仅当在新的链路开销导致与该链路相连节点的最低开销路径发生改变时，才传播已改变的链路开销。
- *收敛速度*。我们已经看到LS算法的实现是一个要求$O(|N||E|)$个报文的$O(|N|^2)$算法。DV算法收敛较慢，且在收敛时会遇到路由选择环路。DV算法还会遭遇无穷计数的问题。
- *健壮性*。如果一台路由器发生故障、行为错乱或受到蓄意破坏时情况会怎样呢？对

于LS算法，路由器能够向其连接的链路（而不是其他链路）广播不正确的开销。作为LS广播的一部分，一个节点也可损坏或丢弃它收到的任何LS广播分组。但是一个LS节点仅计算自己的转发表；其他节点也自行执行类似的计算。这就意味着在LS算法下，路由计算在某种程度上是分离的，提供了一定程度的健壮性。在DV算法下，一个节点可向任意或所有目的节点通告其不正确的最低开销路径。（在1997年，一个小ISP的一台有故障的路由器的确向美国的主干路由器提供了错误的路由选择信息。这引起了其他路由器将大量流量引向该故障路由器，并导致因特网的大部分中断连接达数小时［Neumann 1997］。）更一般地，我们会注意到每次迭代时，在DV算法中一个节点的计算会传递给它的邻居，然后在下次迭代时再间接地传递给邻居的邻居。在此情况下，DV算法中一个不正确的节点计算值会扩散到整个网络。

总之，两个算法没有一个是明显的赢家，它们的确都在因特网中得到了应用。

5.3　因特网中自治系统内部的路由选择：OSPF

在我们至今为止的算法研究中，我们将网络只看作一个互联路由器的集合。从所有路由器执行相同的路由选择算法以计算穿越整个网络的路由选择路径的意义上来说，一台路由器很难同另一台路由器区别开来。在实践中，该模型和这种一组执行同样路由选择算法的同质路由器集合的观点有一点简单化，有以下两个重要原因：

- 规模。随着路由器数目变得很大，涉及路由选择信息的通信、计算和存储的开销将高得不可实现。当今的因特网由数亿台主机组成。在这些主机中存储的路由选择信息显然需要巨大容量的内存。在所有路由器之间广播连通性和链路开销更新所要求的负担将是巨大的！在如此大量的路由器中迭代的距离向量算法将肯定永远无法收敛！显然，必须采取一些措施以减少像因特网这种大型网络中的路由计算的复杂性。
- 管理自治。如在1.3节描述的那样，因特网是ISP的网络，其中每个ISP都有它自己的路由器网络。ISP通常希望按自己的意愿运行路由器（如在自己的网络中运行它所选择的某种路由选择算法），或对外部隐藏其网络的内部组织面貌。在理想情况下，一个组织应当能够按自己的愿望运行和管理其网络，还要能将其网络与其他外部网络连接起来。

这两个问题都可以通过将路由器组织进**自治系统**（Autonomous System，AS）来解决，其中每个AS由一组通常处在相同管理控制下的路由器组成。通常在一个ISP中的路由器以及互联它们的链路构成一个AS。然而，某些ISP将它们的网络划分为多个AS。特别是，某些一级ISP在其整个网络中使用一个庞大的AS，而其他ISP则将它们的ISP拆分为数十个互联的AS。一个自治系统由其全局唯一的AS号（ASN）所标识［RFC 1930］。就像IP地址那样，AS号由ICANN区域注册机构所分配［ICANN 2016］。

在相同AS中的路由器都运行相同的路由选择算法并且有彼此的信息。在一个自治系统内运行的路由选择算法叫作**自治系统内部路由选择协议**（intra-autonomous system routing protocol）。

开放最短路优先（OSPF）

OSPF路由选择及其关系密切的协议IS-IS都被广泛用于因特网的AS内部路由选

择。OSPF 中的开放（open）一词是指路由选择协议规范是公众可用的（与之相反的是 Cisco 的 EIGRP 协议，该协议在最近才成为开放的［Savage 2015］，它作为 Cisco 专用协议大约有 20 年时间）。OSPF 的最新版本是版本 2，由［RFC 2328］这个公用文档所定义。

OSPF 是一种链路状态协议，它使用洪泛链路状态信息和 Dijkstra 最低开销路径算法。使用 OSPF，一台路由器构建了一幅关于整个自治系统的完整拓扑图（即一幅图）。于是，每台路由器在本地运行 Dijkstra 的最短路径算法，以确定一个以自身为根节点到所有子网的最短路径树。各条链路开销是由网络管理员配置的（参见“实践原则：设置 OSPF 链路权值”）。管理员也许会选择将所有链路开销设为 1，因而实现了最少跳数路由选择，或者可能会选择将链路权值按与链路容量成反比来设置，从而不鼓励流量使用低带宽链路。OSPF 不强制使用设置链路权值的策略（那是网络管理员的任务），而是提供了一种机制（协议），为给定链路权值集合确定最低开销路径的路由选择。

使用 OSPF 时，路由器向自治系统内所有其他路由器广播路由选择信息，而不仅仅是向其相邻路由器广播。每当一条链路的状态发生变化时（如开销的变化或连接/中断状态的变化），路由器就会广播链路状态信息。即使链路状态未发生变化，它也要周期性地（至少每隔 30 min 一次）广播链路状态。RFC 2328 中有这样的说明：“链路状态通告的这种周期性更新增加了链路状态算法的健壮性。”OSPF 通告包含在 OSPF 报文中，该 OSPF 报文直接由 IP 承载，对 OSPF 其上层协议的值为 89。因此 OSPF 协议必须自己实现诸如可靠报文传输、链路状态广播等功能。OSPF 协议还要检查链路正在运行（通过向相连的邻居发送 HELLO 报文），并允许 OSPF 路由器获得相邻路由器的网络范围链路状态的数据库。

OSPF 的优点包括下列几方面：

- 安全。能够鉴别 OSPF 路由器之间的交换（如链路状态更新）。使用鉴别，仅有受信任的路由器能参与一个 AS 内的 OSPF 协议，因此可防止恶意入侵者（或正在利用新学的知识到处试探的网络专业的学生）将不正确的信息注入路由器表内。在默认状态下，路由器间的 OSPF 报文是未被鉴别的并能被伪造。能够配置两类鉴别，即简单的和 MD5 的（参见第 8 章有关 MD5 和鉴别的一般性讨论）。使用简单的鉴别，每台路由器配置相同的口令。当一台路由器发送一个 OSPF 分组，它以明文方式包括了口令。显然，简单鉴别并不是非常安全。MD5 鉴别基于配置在所有路由器上的共享秘密密钥。对发送的每个 OSPF 分组，路由器对附加了秘密密钥的 OSPF 分组内容计算 MD5 散列值（参见第 8 章中报文鉴别码的讨论）。然后路由器将所得的散列值包括在该 OSPF 分组中。接收路由器使用预配置的秘密密钥计算出该分组的 MD5 散列值，并与该分组携带的散列值进行比较，从而验证了该分组的真实性。在 MD5 鉴别中也使用了序号对重放攻击进行保护。
- 多条相同开销的路径。当到达某目的地的多条路径具有相同的开销时，OSPF 允许使用多条路径（这就是说，当存在多条相等开销的路径时，无须仅选择单一的路径来承载所有的流量）。
- 对单播与多播路由选择的综合支持。多播 OSPF（MOSPF）［RFC 1584］提供对 OSPF 的简单扩展，以便提供多播路由选择。MOSPF 使用现有的 OSPF 链路数据库，并为现有的 OSPF 链路状态广播机制增加了一种新型的链路状态通告。

- 支持在单个 AS 中的层次结构。一个 OSPF 自治系统能够层次化地配置多个区域。每个区域都运行自己的 OSPF 链路状态路由选择算法，区域内的每台路由器都向该区域内的所有其他路由器广播其链路状态。在每个区域内，一台或多台区域边界路由器负责为流向该区域以外的分组提供路由选择。最后，在 AS 中只有一个 OSPF 区域配置成主干区域。主干区域的主要作用是为该 AS 中其他区域之间的流量提供路由选择。该主干总是包含本 AS 中的所有区域边界路由器，并且可能还包含了一些非边界路由器。在 AS 中的区域间的路由选择要求分组先路由到一个区域边界路由器（区域内路由选择），然后通过主干路由到位于目的区域的区域边界路由器，进而再路由到最终目的地。

OSPF 是一个相当复杂的协议，而我们这里的讨论是十分简要的，［Huitema 1998；Moy 1998；RFC 2328］提供了更多的细节。

实 践 原 则

设置 OSPF 链路权值

我们有关链路状态路由选择的讨论隐含地假设了下列事实：链路权重已经设置好了，运行诸如 OSPF 这样的路由选择算法，流量根据由 LS 算法计算所得的路由选择表流动。就因果而言，给定链路权重（即它们先发生），结果得到（经 Dijkstra 算法）最小化总体开销的路由选择路径。从这个角度看，链路权重反映了使用一条链路的开销（例如，如果链路权重与容量成反比，则使用高容量链路将具有较小的权重并因此从路由选择的角度更有吸引力），并且使用 Dijkstra 算法使得总开销为最小。

在实践中，链路权重和路由选择路径之间的因果关系也许是相反的，网络操作员配置链路权重，以获取某些流量工程目标的路由选择路径［Fortz 2000；Fortz 2002］。例如，假设某网络操作员具有在每个入口点进入和发向每个出口点的该网络的流量估计。该操作员接下来可能要设置特定入口到出口的流路由选择，以最小化经所有网络链路的最大利用率。但使用如 OSPF 这样的路由选择算法，操作员调节网络流的路由选择的主要手段就是链路权重。因此，为了取得最小化最大链路利用率的目标，操作员必须找出取得该目标的链路权重集合。这是一种相反的因果关系，即所希望的流路由选择已知，必须找到 OSPF 链路权重，使得该 OSPF 路由选择算法导致这种希望的流路由选择。

5.4　ISP 之间的路由选择：BGP

我们刚才学习了 OSPF 是一个 AS 内部路由选择协议。当在相同 AS 内的源和目的地之间进行分组选路时，分组遵循的路径完全由 AS 内路由选择协议所决定。然而，当分组跨越多个 AS 进行路由时，比如说从位于马里廷巴克图的智能手机到位于美国硅谷数据中心的一台服务器，我们需要一个**自治系统间路由选择协议**（inter-autonomous system routing protocol）。因为 AS 间路由选择协议涉及多个 AS 之间的协调，所以 AS 通信必须运行相同的 AS 间路由选择协议。在因特网中，所有的 AS 运行相同的 AS 间路由选择协议，称为**边界网关协议**（Broder Gateway Protocol，BGP）［RFC 4271；Stewart 1999］。

BGP 无疑是所有因特网协议中最为重要的（唯一竞争者可能是我们已经在 4.3 节中学

习的 IP 协议），因为正是这个协议将因特网中数以千计的 ISP 黏合起来。如我们将看到的那样，BGP 是一种分布式和异步的协议，与 5.2.2 节中描述的距离向量路由选择协议一脉相承。尽管 BGP 是一种复杂和富有挑战性的协议，但为了深层次理解因特网，我们需要熟悉它的基础结构和操作。我们专注于学习 BGP 的时间将是物有所值的。

5.4.1　BGP 的作用

为了理解 BGP 的职责所在，考虑一个 AS 和在该 AS 中的任意一个路由器。前面讲过，每台路由器具有一张转发表，该转发表在将到达分组转发到出路由器链路的过程中起着主要作用。如我们已经学习过的那样，对于位于相同 AS 中的目的地而言，在路由器转发表中的表项由 AS 内部路由选择协议所决定。而对于位于该 AS 外部的目的地而言情况如何呢？这正是 BGP 用武之地。

在 BGP 中，分组并不是路由到一个特定的目的地址，相反是路由到 CIDR 化的前缀，其中每个前缀表示一个子网或一个子网的集合。在 BGP 的世界中，一个目的地可以采用 138.16.68/22 的形式，对于这个例子来说包括 1024 个 IP 地址。因此，一台路由器的转发表将具有形式为（x，I）的表项，其中 x 是一个前缀（例如 138.16.68/22），I 是该路由器的接口之一的接口号。

作为一种 AS 间的路由选择协议，BGP 为每台路由器提供了一种完成以下任务的手段：

1）从邻居 AS 获得前缀的可达性信息。特别是，BGP 允许每个子网向因特网的其余部分通告它的存在。一个子网高声宣布“我存在，我在这里”，而 BGP 确保在因特网中的所有 AS 知道该子网。如果没有 BGP 的话，每个子网将是隔离的孤岛，即它们孤独地存在，不为因特网其余部分所知和所达。

2）确定到该前缀的“最好的”路由。一台路由器可能知道两条或更多条到特定前缀的不同路由。为了确定最好的路由，该路由器将本地运行一个 BGP 路由选择过程（使用它经过相邻的路由器获得的前缀可达性信息）。该最好的路由将基于策略以及可达性信息来确定。

我们现在钻研 BGP 如何执行这两个任务。

5.4.2　通告 BGP 路由信息

考虑图 5-8 中显示的网络。如我们看到的那样，这个简单的网络具有 3 个自治系统：AS1、AS2 和 AS3。如显示的那样，AS3 包括一个具有前缀 x 的子网。对于每个 AS，每台路由器要么是一台**网关路由器**（gateway router），要么是一台**内部路由器**（internal router）。网关路由器是一台位于 AS 边缘的路由器，它直接连接到在其他 AS 中的一台或多台路由器。内部路由器仅连接在它自己 AS 中的主机和路由器。例如，在 AS1 中路由器 1c 是网关路由器；路由器 1a、1b 和 1d 是内部路由器。

我们考虑这样一个任务：向图 5-8 中显示的所有路由器通告对于前缀 x 的可达性信息。在高层次上，这是简明易懂的。首先，AS3 向 AS2 发送一个 BGP 报文，告知 x 存在并且位于 AS3 中；我们将该报文表示为“AS3 x”。然后 AS2 向 AS1 发送一个 BGP 报文，告知 x 存在并且能够先通过 AS2 然后进入 AS3 进而到达 x；我们将该报文表示为“AS2 AS3 x”。以这种方式，每个自治系统不仅知道 x 的存在，而且知道通向 x 的自治系统的路径。

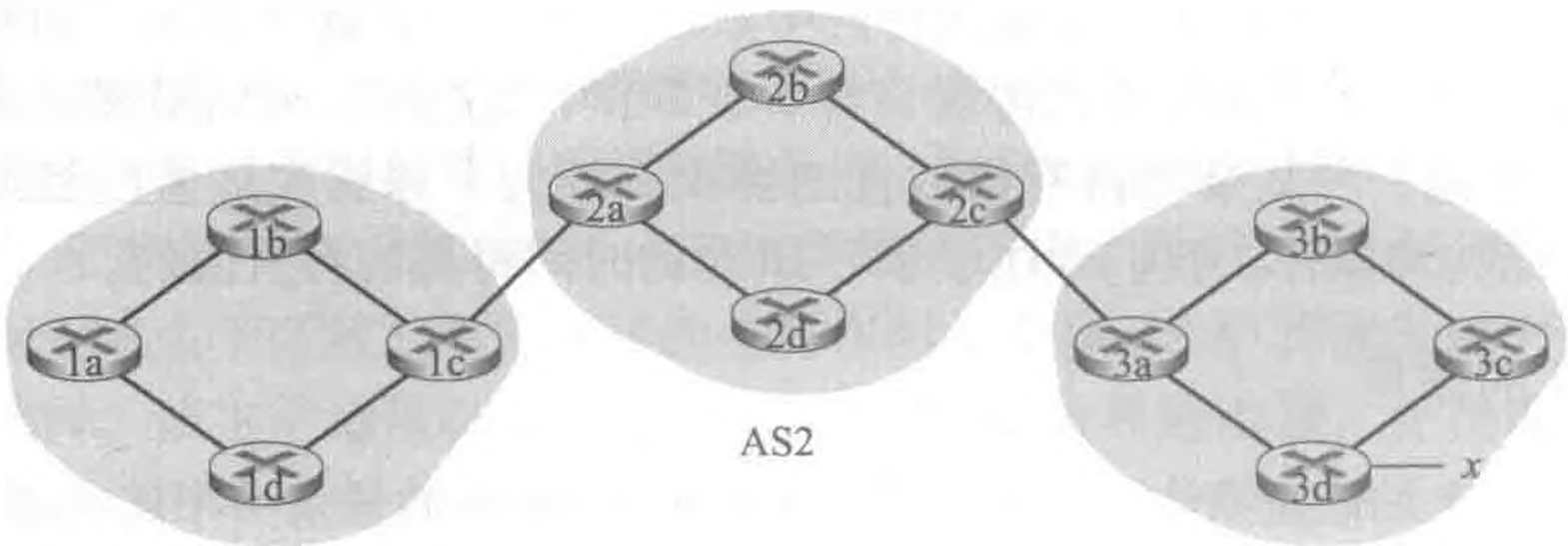

图 5-8　具有 3 个自治系统的网络。AS3 包括一个具有前缀 *x* 的子网

虽然在图 5-8 中有关通告 BGP 可达性信息的讨论能得到路径穿越的大意，但就自治系统彼此并未实际发送报文而言，它并不是准确的，相反是路由器在发送报文。为了理解这一点，我们现在重温图 5-8 中的例子。在 BGP 中，每对路由器通过使用 179 端口的半永久 TCP 连接交换路由选择信息。每条直接连接以及所有通过该连接发送的 BGP 报文，称为 **BGP 连接**（BGP connection）。此外，跨越两个 AS 的 BGP 连接称为**外部 BGP**（eBGP）连接，而在相同 AS 中的两台路由器之间的 BGP 会话称为**内部 BGP**（iBGP）连接。图 5-8 所示网络的 BGP 连接的例子显示在图 5-9 中。对于直接连接在不同 AS 中的网关路由器的每条链路而言，通常有一条 eBGP 连接；因此，在图 5-9 中，在网关路由器 1c 和 2a 之间有一条 eBGP 连接，而在网关路由器 2c 和 3a 之间也有一条 eBGP 连接。

在每个 AS 中的路由器之间还有多条 iBGP 连接。特别是，图 5-9 显示了一个 AS 内部的每对路由器之间的一条 BGP 连接的通常配置，在每个 AS 内部产生了网状的 TCP 连接。在图 5-9 中，eBGP 会话显示为长虚线，iBGP 显示为短虚线。注意到 iBGP 连接并不总是与物理链路对应。

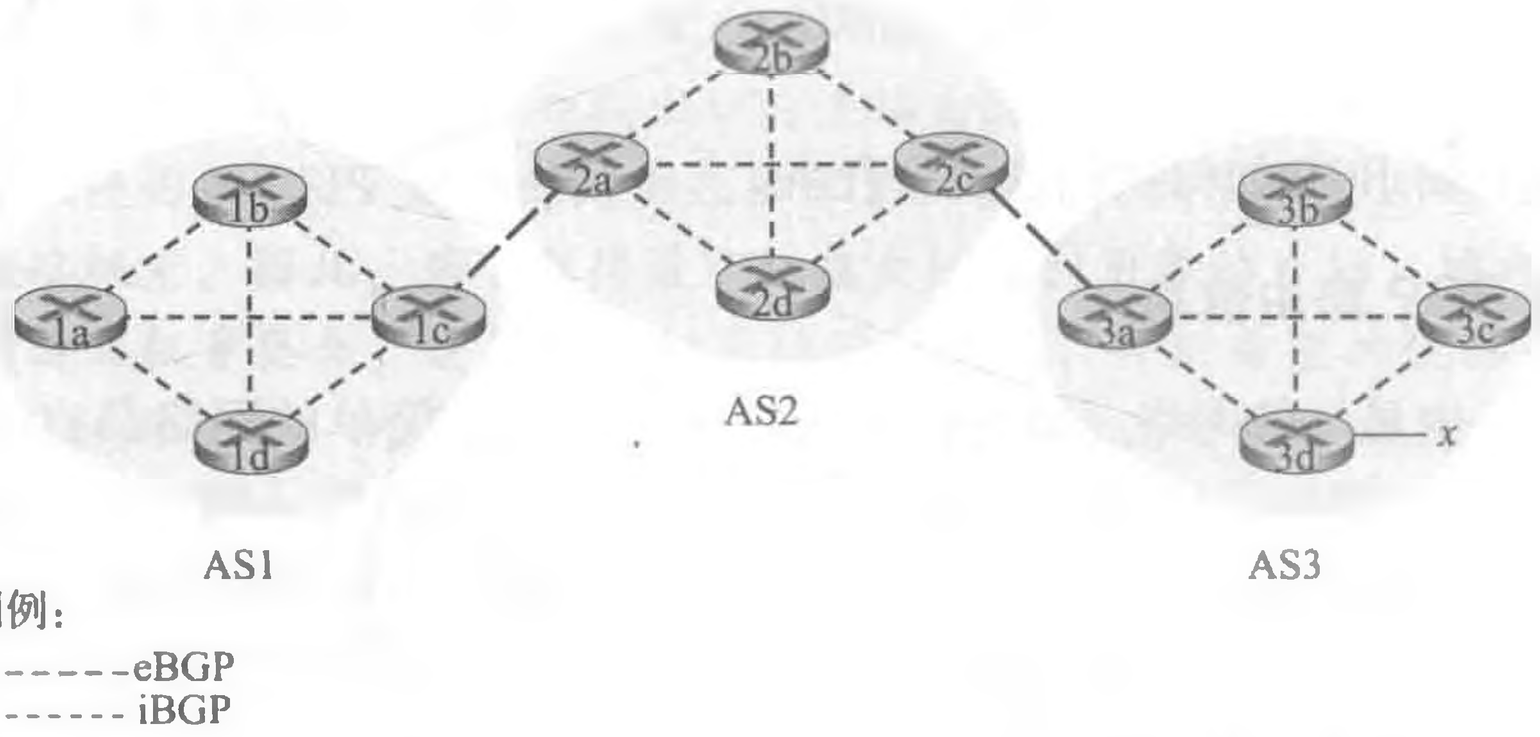

图 5-9　eBGP 和 iBGP 连接

为了传播可达性信息，使用了 iBGP 和 eBGP 会话。再次考虑向 AS1 和 AS2 中的所有路由器通告前缀 *x* 的可达性信息。在这个过程中，网关路由器 3a 先向网关路由器 2c 发送一个 eBGP 报文“AS3 *x*”。网关路由器 2c 然后向 AS2 中的所有其他路由器（包括网关路由器 2a）发送 iBGP 报文“AS3 *x*”。网关路由器 2a 接下来向网关路由器 1c 发送一个 eBGP 报文“AS2 AS3 *x*”。最后，网关路由器 1c 使用 iBGP 向 AS1 中的所有路由器发送报文“AS2 AS3 *x*”。在这个过程完成后，在 AS1 和 AS2 中的每个路由器都知道了 *x* 的存在并且也都知道了通往 *x* 的 AS 路径。

当然，在真实的网络中，从某个给定的路由器到某个给定的目的地可能有多条不同的路径，每条通过了不同的 AS 序列。例如，考虑图 5-10 所示的网络，它是在图 5-8 那个初始网络基础上，从路由器 1d 到路由器 3d 附加了一条物理链路。在这种情况下，从 AS1 到 x 有两条路径：经过路由器 1c 的路径“AS2 AS3 x”；以及经过路由器 1d 的新路径“AS3 x”。

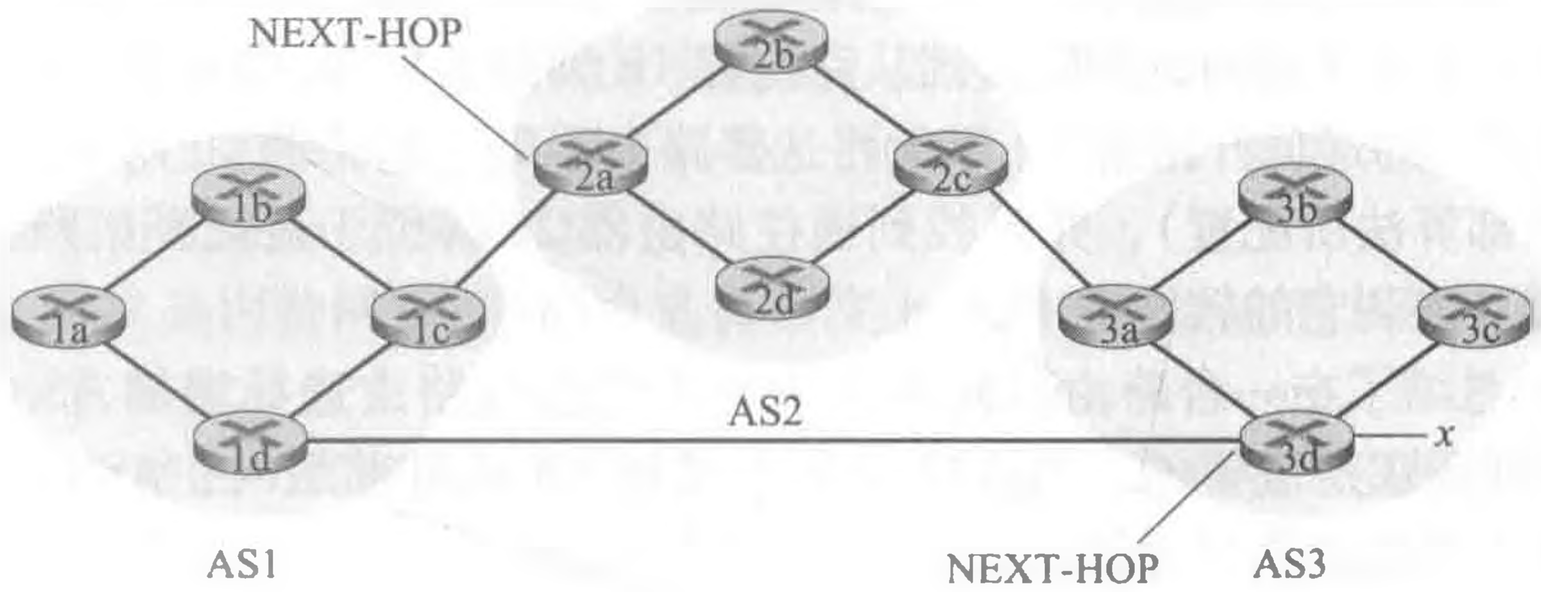

图 5-10　在 AS1 和 AS3 之间增加对等链路后的网络

5.4.3　确定最好的路由

如我们刚才学习到的那样，从一个给定的路由器到一个目的子网可能有多条路径。事实上，因特网中的路由器常常接收到很多不同的可能路径的可达性信息。一台路由器如何在这些路径之间进行选择（并且再相应地配置它的转发表）呢？

在处理这个关键性问题之前，我们需要引入几个 BGP 术语。当路由器通过 BGP 连接通告前缀时，它在前缀中包括一些 **BGP 属性**（BGP attribute）。用 BGP 术语来说，前缀及其属性称为**路由**（route）。两个较为重要的属性是 AS-PATH 和 NEXT-HOP。AS-PATH 属性包含了通告已经通过的 AS 的列表，如我们在前面的例子中所见。为了生成 AS-PATH 的值，当一个前缀通过某 AS 时，该 AS 将其 ASN 加入 AS-PATH 中的现有列表。例如，在图 5-10 中，从 AS1 到子网 x 有两条路：其中一条使用 AS-PATH“AS2 AS3”；而另一条使用 AS-PATH“AS3”。BGP 路由器还使用 AS-PATH 属性来检测和防止通告环路；特别是，如果一台路由器在路径列表中看到包含了它自己的 AS，它将拒绝该通告。

在 AS 间和 AS 内部路由选择协议之间提供关键链路方面，NEXT-PATH 属性具有敏感而重要的作用。NEXT-HOP 是 AS-PATH 起始的路由器接口的 IP 地址。为了深入理解该属性，我们再次参考图 5-10。如图 5-10 中所指示的那样，对于从 AS1 通过 AS2 到 x 的路由“AS2 AS3 x”，其属性 NEXT-HOP 是路由器 2a 左边接口的 IP 地址。对于从 AS1 绕过 AS2 到 x 的路由“AS3 x”，其 NEXT-HOP 属性是路由器 3d 最左边接口的 IP 地址。总的说来，在这个假想的例子中，AS1 中的每台路由器都知道了到前缀 x 的两台 BGP 路由：

路由器 2a 的最左侧接口的 IP 地址：AS2 AS3；x

路由器 3d 的最左侧接口的 IP 地址：AS3；x

这里，每条 BGP 路由包含 3 个组件：NEXT-HOP；ASPATH；目的前缀。在实践中，一条 BGP 路由还包括其他属性，眼下我们将暂且忽略它。注意到 NEXT-HOP 属性是不属于 AS1 的某路由器的 IP 地址；然而，包含该 IP 地址的子网直接连接到 AS1。

1. 热土豆路由选择

终于到了以精确的方式来讨论 BGP 路由选择算法的时刻了。我们将以一个最简单的路由选择算法开始，即**热土豆路由选择**（hot potato routing）。

考虑在图5-10网络中的路由器1b。如同刚才所述，这台路由器将学习到达前缀 x 的两条BGP路由。使用热土豆路由选择，（从所有可能的路由中）选择的路由到开始该路由的NEXT-HOP路由器具有最小开销。在这个例子中，路由器1b将查阅它的AS内部路由选择信息，以找到通往NEXT-HOP路由器2a的最低开销AS内部路径以及通往NEXT-HOP路由器3d的最低开销AS间路径，进而选择这些最低开销路径中具有最低开销的那条。例如，假设开销定义为穿越的链路数。则从路由器1b到路由器2a的最低开销是2，从路由器1b到路由器2d的最低开销是3，因此将选择路由器2a。路由器1b则将查阅它的转发表（由它的AS内部算法所配置），并且找到通往路由器2a的位于最低开销路径上的接口 I。1b则把（x，I）加到它的转发表中。

图5-11中总结了在一台路由器转发表中对于热土豆路由选择增加AS向外前缀的步骤。注意到下列问题是重要的：当在转发表中增加AS向外前缀时，AS间路由选择协议（BGP）和AS内部路由选择协议（如OSPF）都要用到。

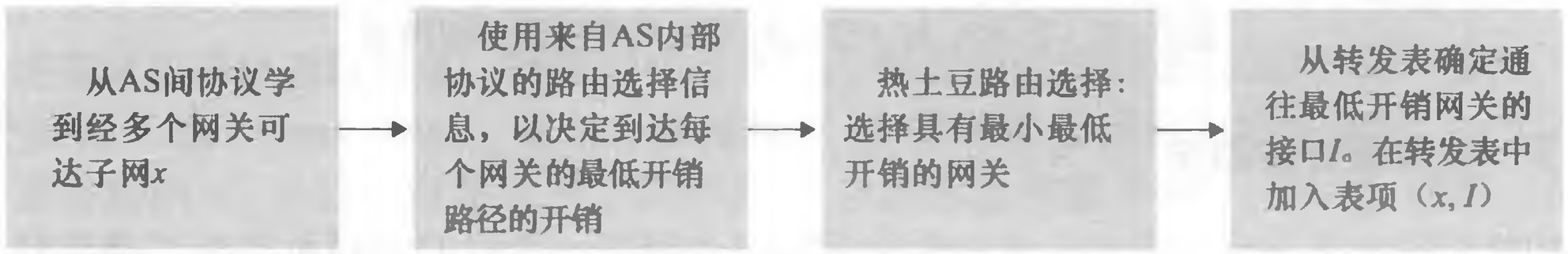

图5-11　在路由器转发表中增加AS外部目的地的步骤

热土豆路由选择依据的思想是：对于路由器1b，尽可能快地将分组送出其AS（更明确地说，用可能的最低开销），而不担心其AS外部到目的地的余下部分的开销。就“热土豆路由选择”名称而言，分组被类比为烫手的热土豆。因为它烫手，你要尽可能快地将它传给另一个人（另一个AS）。热土豆路由选择因而是自私的算法，即它试图减小在它自己AS中的开销，而忽略在其AS之外的端到端开销的其他部分。注意到使用热土豆路由选择，对于在相同AS中的两台路由器，可能对相同的前缀选择两条不同的AS路径。例如，我们刚才看到路由器1b到达 x 将通过AS2发送分组。而路由器1d将绕过AS2并直接向AS3发送分组到达 x。

2. 路由器选择算法

在实践中，BGP使用了一种比热土豆路由选择更为复杂但却结合了其特点的算法。对于任何给定的目的地前缀，进入BGP的路由选择算法的输入是到某前缀的所有路由的集合，该前缀是已被路由器学习和接受的。如果仅有一条这样的路由，BGP则显然选择该路由。如果到相同的前缀有两条或多条路由，则顺序地调用下列消除规则直到余下一条路由：

1）路由被指派一个**本地偏好**（local preference）值作为其属性之一（除了AS-PATH和NEXT-HOP以外）。一条路由的本地偏好可能由该路由器设置或可能由在相同AS中的另一台路由器学习到。本地偏好属性的值是一种策略决定，它完全取决于该AS的网络管理员（我们随后将更为详细地讨论BGP策略问题）。具有最高本地偏好值的路由将被选择。

2）从余下的路由中（所有都具有相同的最高本地偏好值），将选择具有最短AS-PATH的路由。如果该规则是路由选择的唯一规则，则BGP将使用距离向量算法决定路径，其中距离测度使用AS跳的跳数而不是路由器跳的跳数。

3）从余下的路由中（所有都具有相同的最高本地偏好值和相同的AS-PATH长度），

使用热土豆路由选择，即选择具有最靠近 NEXT-HOP 路由器的路由。

4）如果仍留下多条路由，该路由器使用 BGP 标识符来选择路由，参见［Stewart 1999］。

举一个例子，我们再次考虑图 5-10 中的路由器 1b。前面讲过到前缀 x 确切地有两条 BGP 路由，一条通过 AS2 而另一条绕过 AS2。前面也讲过如果它使用自己的热土豆路由选择，则 BGP 将通过 AS2 向前缀 x 路由分组。但在上面的路由选择算法中，在规则 3 之前应用了规则 2，导致 BGP 选择绕过 AS2 的那条路由，因为该路由具有更短的 AS-PATH。因此我们看到使用上述路由选择算法，BGP 不再是一种自私的算法，即它先查找具有短 AS 路径的路由（因而很可能减小端到端时延）。

如上所述，BGP 是因特网 AS 间路由选择事实上的标准。要查看从第 1 层 ISP 中提取的各种 BGP 路由选择表（庞大!），可参见 http://www. routeviews. org。BGP 路由选择表通常包含超过 50 万条路由（即前缀和相应的属性）。BGP 路由选择表的规模和特征的统计可在［Potaroo 2016］中找到。

5.4.4　IP 任播

除了作为因特网的 AS 间路由选择协议外，BGP 还常被用于实现 IP 任播（anycast）服务［RFC 1546，RFC 7094］，该服务通常用于 DNS 中。为了说明 IP 任播的动机，考虑在许多应用中，我们对下列情况感兴趣：①在许多分散的不同地理位置，替换不同服务器上的相同内容；②让每个用户从最靠近的服务器访问内容。例如，一个 CDN 能够更换位于不同国家、不同服务器上的视频和其他对象。类似地，DNS 系统能够在遍及全世界的 DNS 服务器上复制 DNS 记录。当一个用户要访问该复制的内容，可以将用户指向具有该复制内容的“最近的”服务器。BGP 的路由选择算法为做这件事提供了一种最为容易和自然的机制。

为使我们的讨论具体，我们描述 CDN 可能使用 IP 任播的方式。如图 5-12 所示，在 IP 任播配置阶段，CDN 公司为它的多台服务器指派相同的 IP 地址，并且使用标准的 BGP 从这些服务器的每台来通告该 IP 地址。当某台 BGP 路由器收到对于该 IP 地址的多个路由通告，它将这些通告处理为对相同的物理位置提供不同的路径（事实上，这时这些通告对不同的物理位置是有不同路径的）。当配置其路由选择表时，每台路由器将本地化地使用 BGP 路由选择算法来挑选到该 IP 地址的“最好的”（例如，由 AS 跳计数确定的最近的）路由。例如，如果一个 BGP 路由（对应于一个位置）离该路由器仅一 AS 跳的距离，并且所有其他 BGP 路由（对应于其他位置）是两 AS 跳和更多 AS 跳，则该 BGP 路由器将选择把分组路由到一跳远的那个位置。在这个初始 BGP 地址通告阶段后，CDN 能够进行其分发内容的主要任务。当某客户请求视频时，CDN 向该客户返回由地理上分散的服务器所使用的共同 IP 地址，而无论该客户位于何处。当该客户想向那个 IP 地址发送一个请求时，因特网路由器则向那个“最近的”服务器转发该请求分组，最近的服务器是由 BGP 路由选择算法所定义的。

尽管上述 CDN 的例子很好地诠释了能够如何使用 IP 任播，但实践中 CDN 通常选择不使用 IP 任播，因为 BGP 路由选择变化能够导致相同的 TCP 连接的不同分组到达 Web 服务器的不同实例。但 IP 任播被 DNS 系统广泛用于将 DNS 请求指向最近的根 DNS 服务器。2.4 节讲过，当前根 DNS 服务器有 13 个 IP 地址。但对应于这些地址的每一个，有多个 DNS 根服务器，其中有些地址具有 100 多个 DNS 根服务器分散在世界的各个角落。当一个 DNS 请求向这 13 个 IP 地址发送时，使用 IP 任播将该请求路由到负责该地址的最近的那个 DNS 根服务器。

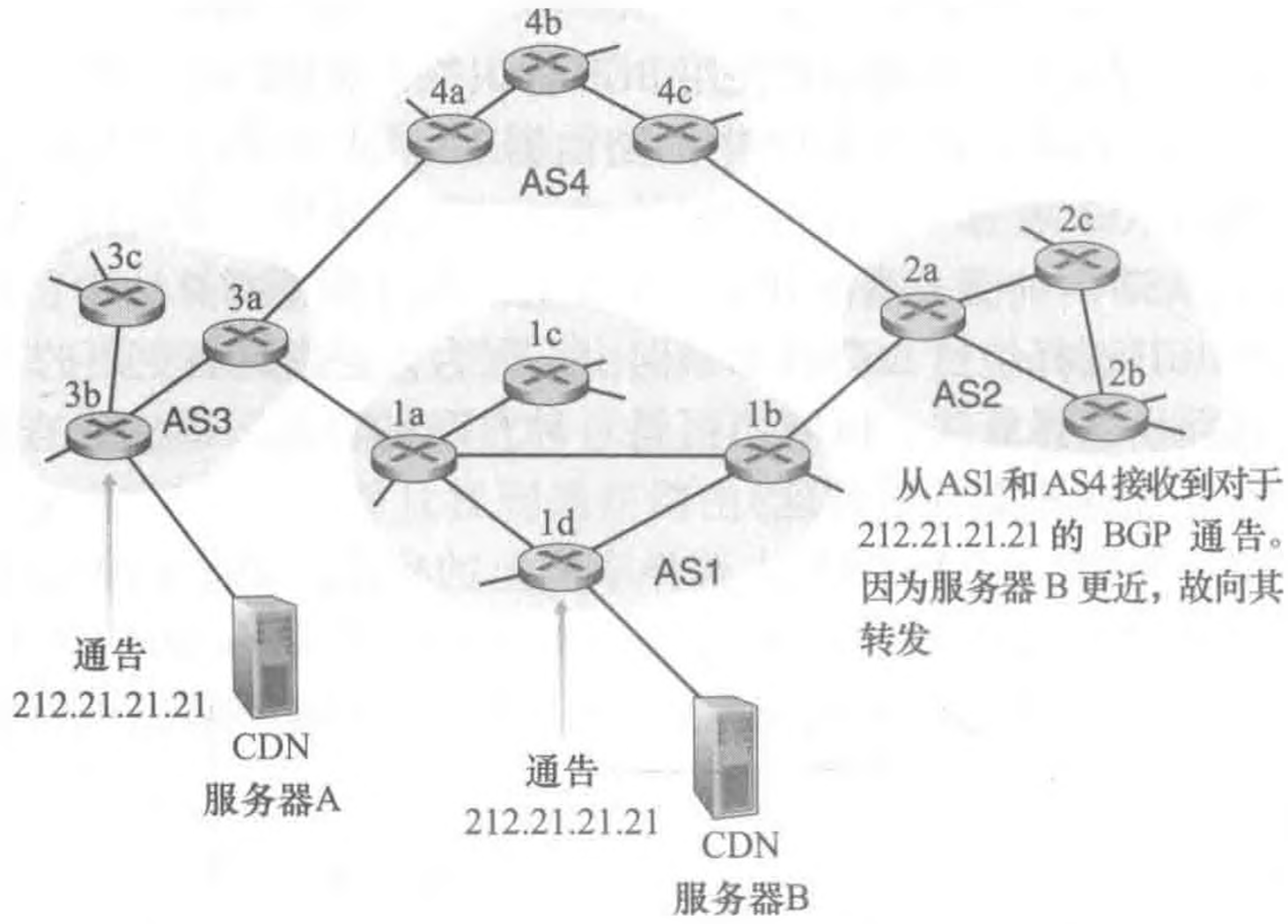

图 5-12　使用 IP 任播将用户引向最近的 CDN 服务器

5.4.5　路由选择策略

当某路由器选择到目的地的一条路由时，AS 路由选择策略能够胜过所有其他考虑，例如最短 AS 路径或热土豆路由选择。在路由选择算法中，实际上首先根据本地偏好属性选择路由，本地偏好值由本地 AS 的策略所确定。

我们用一个简单的例子说明 BGP 路由选择策略的某些基本概念。图 5-13 显示了 6 个互联的自治系统：A、B、C、W、X 和 Y。重要的是注意到 A、B、C、W、X 和 Y 是 AS，而不是路由器。假设自治系统 W、X 和 Y 是接入 ISP，而 A、B 和 C 是主干提供商网络。我们还要假设 A、B 和 C 直接向彼此发送流量，并向它们的客户网络提供全部的 BGP 信息。所有进入一个接入 ISP 网络的流量必定是以该网络为目的地，所有离开一个接入 ISP 网络的流量必定源于该网络。W 和 Y 显然是接入 ISP。X 是一个**多宿接入 ISP**（multi-homed stub network），因为它是经由两个不同的提供商连到网络的其余部分（这种方法在实践中变得越来越普遍）。然而，就像 W 和 Y 一样，X 自身必定是进入/离开 X 的所有流量的源/目的地。但这种桩网络的行为是如何实现和强制实现的呢？X 如何防止转发 B 与 C 之间的流量呢？这能够通过控制 BGP 路由的通告方式容易地实现。特别是，X 如果（向其邻居 B 和 C）通告它没有通向（除自身以外）任何其他目的地的路径，那么它将起到一个接入 ISP 的作用。这就是说，即使 X 可能知道一条路径（比如说 XCY）能到达网络 Y，它也将不把该条路径通告给 B。由于 B 不知道 X 有一条路径到 Y，B 绝不会经由 X 转发目的为 Y（或 C）的流量。这个简单的例子说明了如何使用一条选择的路由通告策略来实现客户/提供商路由选择关系。

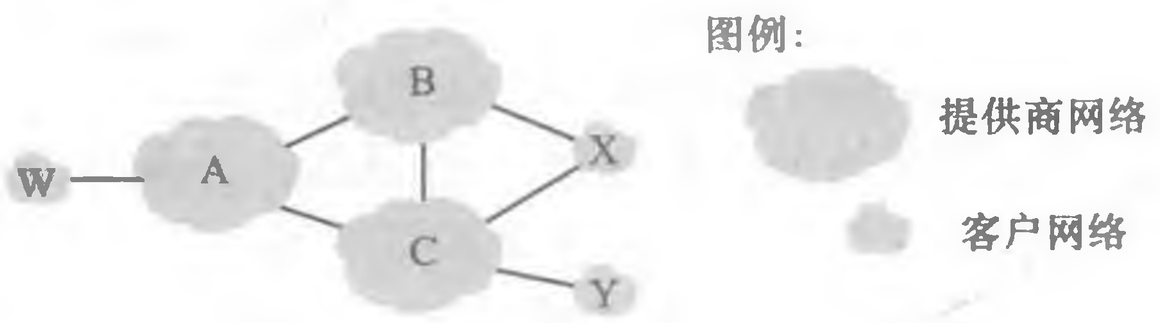

图 5-13　一个简单的 BGP 策略场景

我们接下来关注一个提供商网络，比如自治系统 B。假定 B 已经（从 A 处）知道了 A 有一条到 W 的路径 AW。B 因此能将路由 AW 安装到其路由信息库中。显然，B 也想向它的客户 X 通告路径 BAW，这样 X 知道它能够通过 B 路由到 W。但是，B 应该将路径 BAW

通告给 C 吗？如果它这样做，则 C 可以经由 BAW 将流量引导到 W。如果 A、B 和 C 都是主干提供商，而 B 也许正好觉得它不应该承担在 A 与 C 之间传送流量的负担（和开销）。B 可能有理由认为，确保 C 能经过 A 和 C 之间的直接连接引导 A 客户的来去流量是 A 和 C 的工作（和开销）。目前还没有强制主干 ISP 之间如何路由选择的官方标准。然而，商业运行的 ISP 们都遵从的一个经验法则是：任何穿越某 ISP 主干网的流量必须是其源或目的（或两者）位于该 ISP 的某个客户网络中；不然的话这些流量将会免费搭车通过该 ISP 的网络。各个对等协定（用于解决前面提到的问题）通常都是 ISP 双方进行协商，而且经常是对外保密的；[Huston 1999a] 提供了关于对等协定的有趣讨论。路由选择策略如何反映 ISP 之间的商业关系的详细描述参见 [Gao 2001；Dmitiropoulos 2007]。从 ISP 的立场出发，有关 BGP 路由选择策略的讨论参见 [Caesar 2005b]。

我们完成了对 BGP 的简要介绍。理解 BGP 是重要的，因为它在因特网中起着重要作用。我们鼓励你阅读参考文献 [Griffin 2012；Stewart 1999；Labovitz 1997；Halabi 2000；Huitema 1998；Gao 2001；Feamster 2004；Caesar 2005b；Li 2007]，以学习更多的 BGP 知识。

实践原则

为什么会有不同的 AS 间和 AS 内部路由选择协议？

学习了目前部署在因特网中的特定的 AS 间和 AS 内部路由选择协议的细节后，我们可通过思考对这些协议首先会问的也许最为根本性的问题来得到结论（希望你已经在思考该问题，并且不致因技术细节而不能把握全局）：为什么所使用的 AS 间和 AS 内部路由选择协议是不同的？

对该问题的答案触及了 AS 内与 AS 间的路由选择目标之间差别的本质：

- 策略。在 AS 之间，策略问题起主导作用。一个给定 AS 产生的流量不能穿过另一个特定的 AS，这可能非常重要。类似地，一个给定 AS 也许想很好地控制它承载的其他 AS 之间穿越的流量。我们已看到，BGP 承载了路径属性，并提供路由选择信息的受控分布，以便能做出这种基于策略的路由选择决策。在一个 AS 内部，一切都是在相同的管理控制名义下进行的，因此策略问题在 AS 内部选择路由中起着微不足道的作用。
- 规模。扩展一个路由选择算法及其数据结构以处理到大量网络或大量网络之间的路由选择的这种能力，是 AS 间路由选择的一个关键问题。在一个 AS 内，可扩展性不是关注的焦点。首先，如果单个 ISP 变得太大时，总是能将其分成两个 AS，并在这两个新的 AS 之间执行 AS 间路由选择。（前面讲过，OSPF 通过将一个 AS 分成区域而建立这样的层次结构。）
- 性能。由于 AS 间路由选择是面向策略的，因此所用路由的质量（如性能）通常是次要关心的问题（即一条更长或开销更高但能满足某些策略条件的路由也许被采用了，而更短但不满足那些条件的路由却不会被采用）。我们的确看到了在 AS 之间，甚至没有与路由相关的开销（除了 AS 跳计数外）概念。然而在一个 AS 内部，这种对策略的关心就不重要了，可以使路由选择更多地关注一条路由实现的性能级别。

5.4.6　拼装在一起：在因特网中呈现

尽管本小节不是有关 BGP 本身的，但它将我们到此为止看到的许多协议和概念结合到一起，包括 IP 地址、DNS 和 BGP。

假定你只是创建了一个具有若干服务器的小型公司网络，包括一台描述公司产品和服务的公共 Web 服务器，一台从你的雇员获得他们的电子邮件报文的电子邮件服务器和一台 DNS 服务器。你当然乐意整个世界能够访问你的 Web 站点，以得知你的现有产品和服务。此外，你将乐意你的雇员能够向遍及世界的潜在客户发送和接收电子邮件。

为了满足这些目标，你首先需要获得因特网连接，要做到这一点，需要与本地 ISP 签订合同并进行连接。你的公司将有一台网关路由器，该路由器将与本地 ISP 的一台路由器相连。该连接可以是一条通过现有电话基础设施的 DSL 连接、一条到 ISP 路由器的租用线，或者是第 1 章描述的许多其他接入解决方案之一。你的本地 ISP 也将为你提供一个 IP 地址范围，例如由 256 个地址组成的一个/24 地址范围。一旦你有了自己的物理连接和 IP 地址范围，你将在该地址范围内分配 IP 地址：一个给你的 Web 服务器，一个给你的电子邮件服务器，一个给你的 DNS 服务器，一个给你的网关路由器，并将其他 IP 地址分配给公司网络中的其他服务器和联网设备。

除了与一个 ISP 签订合同外，你还需要与一个因特网注册机构签订合同，以便为你的公司获得一个域名，如在第 2 章中所描述的那样。例如，如果你的公司名称比如说是 Xanadu Inc.，你自然希望获得域名 xanadu. com。你的公司还必须呈现在 DNS 系统中。具体而言，因为外部世界将要联系你的 DNS 服务器以获得该服务器的 IP 地址，所以你还需要为注册机构提供你的 DNS 服务器的 IP 地址。该注册机构则在 . com 顶级域名服务器中为你的 DNS 服务器设置一个表项（域名和对应的 IP 地址），如第 2 章所述。在这个步骤完成后，任何知道你的域名（例如 xanadu. com）的用户将能够经过 DNS 系统获得你 DNS 服务器的 IP 地址。

为了使人们能够发现你的 Web 服务器的 IP 地址，你需要在你的 DNS 服务器中包括一个将你的 Web 服务器的主机名（例如 www. xanadu. com）映射到它的 IP 地址的表项。你还要为公司中其他公共可用的服务器设置类似的表项，包括你的电子邮件服务器。如此一来，如果 Alice 要浏览你的 Web 服务器，DNS 系统将联系你的 DNS 服务器，找到你的 Web 服务器的 IP 地址，并将其给 Alice。Alice 则能与你的 Web 服务器创建一个直接的 TCP 连接。

然而，允许来自世界各地的外部人员访问你的 Web 服务器，仍然还有一个必要的、决定性的步骤。考虑当 Alice 做下列事情发生的状况：Alice 知道你的 Web 服务器的 IP 地址，她向该 IP 地址发送一个 IP 数据报（例如一个 TCP SYN 报文段）。该数据报将通过因特网进行路由，经历了在许多不同的自治系统中的一系列路由器，最终到达你的 Web 服务器。当任何一个路由器收到该数据报时，将去它的转发表中寻找一个表项来确定转发该数据报的外出端口。因此，每台路由器需要知道你公司的/24 前缀（或者某些聚合项）。一台路由器如何知道你公司的前缀呢？如我们刚才看到的那样，它从 BGP 知道了该前缀。具体而言，当你的公司与本地 ISP 签订合同并且获得了分配的前缀（即一个地址范围），你的本地 ISP 将使用 BGP 向与之连接的 ISP 通告你的前缀。这些 ISP 将依次使用 BGP 来传播该通告。最终，所有的因特网路由器将得知了你的前缀（或者包括你的前缀的某个聚合项），因而能够将数据报适当地转发到适当的 Web 和电子邮件服务器。

5.5　SDN 控制平面

在本节中，我们将深入 SDN 控制平面，即控制分组在网络的 SDN 使能设备中转发的网络范围逻辑，以及这些设备和它们的服务的配置与管理。这里的学习建立在前面 4.4 节中一般化 SDN 转发讨论的基础上，因此你在继续学习前需要先回顾一下那一节，以及本章的 5.1 节。如同 4.4 节中一样，我们将再次采用在 SDN 文献中所使用的术语，将网络的转发设备称之为“分组交换机”（或直接称为交换机，理解时带上“分组”二字），因为能够根据网络层源/目的地址、链路层源/目的地址以及运输层、网络层和链路层中分组首部字段做出转发决定。

SDN 体系结构具有 4 个关键特征［Kreutz 2015］：

- *基于流的转发。*SDN 控制的交换机的分组转发工作，能够基于运输层、网络层或链路层首部中任意数量的首部字段值进行。在 4.4 节中，我们看到了 OpenFlow 1.0 抽象允许基于 11 个不同的首部字段值进行转发。这与我们 5.2～5.4 节中学习的基于路由器转发的传统方法形成了鲜明的对照，传统方法中 IP 数据报的转发仅依据数据报的目的 IP 地址进行。回顾图 5-2，分组转发规则被精确规定在交换机的流表中；SDN 控制平面的工作是计算、管理和安装所有网络交换机中的流表项。
- *数据平面与控制平面分离。*这种分离明显地显示在图 5-2 和图 5-14 中。数据平面由网络交换机组成，交换机是相对简单（但快速）的设备，该设备在它们的流表中执行“匹配加动作”的规则。控制平面由服务器以及决定和管理交换机流表的软件组成。
- *网络控制功能：位于数据平面交换机外部。*考虑到 SDN 中的“S”表示“软件”，也许 SDN 控制平面由软件实现并不令人惊讶。然而，与传统的路由器不同，这个软件在服务器上执行，该服务器与网络交换机截然分开且与之远离。如在图 5-14 中所示，控制平面自身由两个组件组成：一个 SDN 控制器（或网络操作系统［Gude 2008］），以及若干网络控制应用程序。控制器维护准确的网络状态信息（例如，远程链路、交换机和主机的状态）；为运行在控制平面中的网络控制应用程序提供这些信息；提供方法，这些应用程序通过这些方法能够监视、编程和控制下面的网络设备。尽管在图 5-14 中的控制器显示为一台单一的服务器，但实践中控制器仅是逻辑上集中的，通常在几台服务器上实现，这些服务器提供协调的、可扩展的性能和高可用性。
- *可编程的网络。*通过运行在控制平面中的网络控制应用程序，该网络是可编程的。这些应用程序代表了

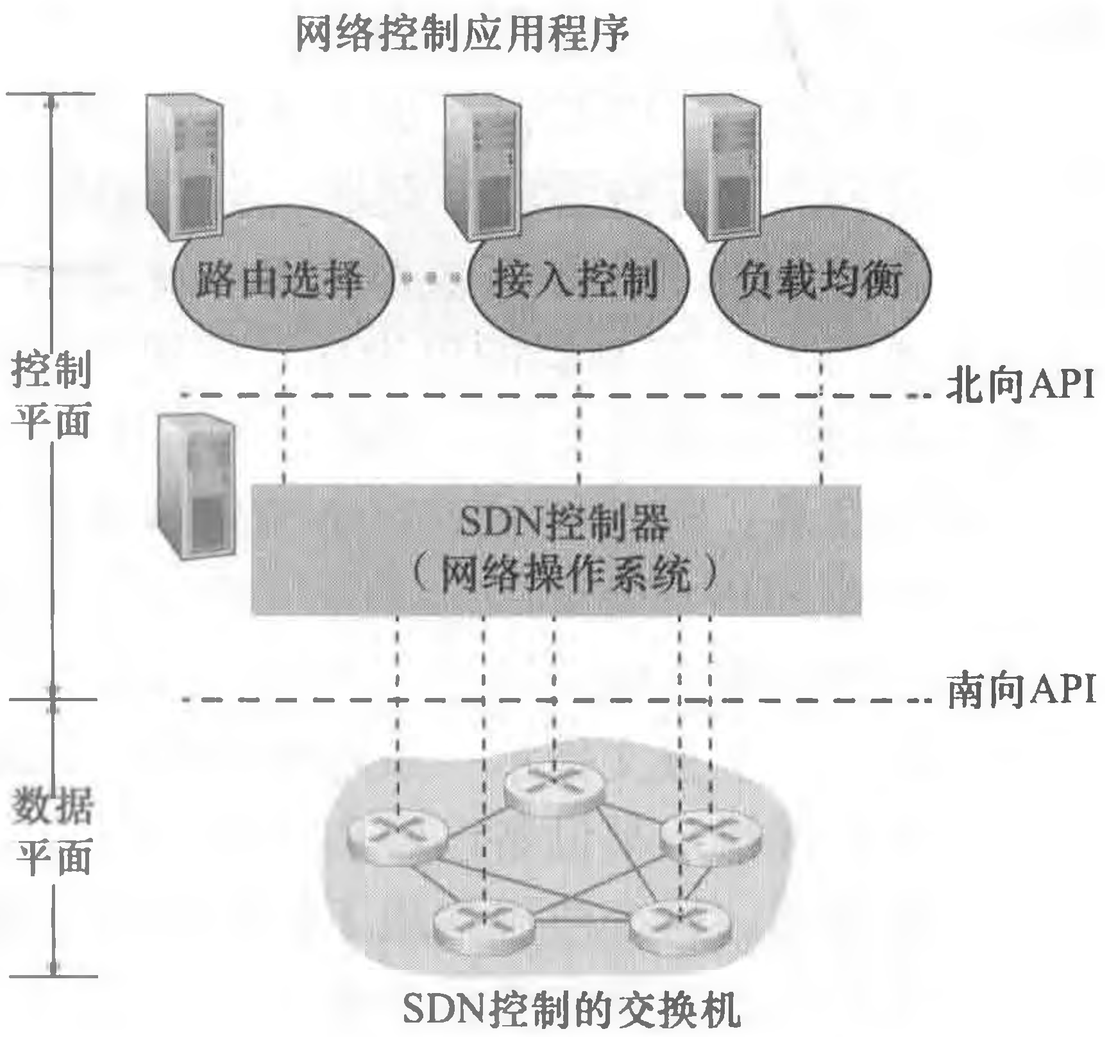

图 5-14　SDN 体系结构的组件：SDN 控制的交换机、SDN 控制器和网络控制应用程序

SDN 控制平面的“智力”，使用了由 SDN 控制器提供的 API 来定义和控制网络设备中的数据平面。例如，一个路由选择网络控制应用程序可以决定源和目的地之间的端到端路径（例如，通过使用由 SDN 控制器维护的节点状态和链路状态信息，执行 Dijkstra 算法）。另一个网络应用程序可以执行访问控制，即决定交换机阻挡哪个分组，如 4.4.3 节中的第三个例子那样。还有一个应用程序可以用执行服务器负载均衡的方式转发分组（4.4.3 节中我们考虑的第二个例子）。

从讨论中我们可见，SDN 表示了一种意义重大的网络功能的“分类”，即数据平面交换机、SDN 控制器和网络控制应用程序是分离的实体，该实体可以由不同的厂商和组织机构所提供。这与 SDN 之前模式形成了鲜明对照，在 SDN 之前模式中，交换机/路由器（连同其嵌入的控制平面软件和协议实现）是一个整体，它是垂直、综合的，并且由单一的厂商所销售。在 SDN 中的这种网络功能分类，可以与大型计算机到个人计算机的早期演化相比拟，前者的硬件、系统软件和应用程序是由单一厂商所提供的，而后者具有各自的硬件、操作系统和应用程序。计算硬件、系统软件和应用程序的分类，无疑已经在所有这三个领域的创新驱动下导致了丰富、开放的生态系统。对 SDN 的希望是，它也将导致如此丰富的创新。

给出了我们对图 5-14 的 SDN 体系结构的理解，自然会产生许多问题。流表是如何实际计算的，以及在哪里进行的？响应 SDN 控制的设备产生的事件时（例如，一条附属链路的激活/关闭），这些流表是如何更新的呢？在协作的多台交换机中，流表项是如何以一种导致和谐、一致的网络范围功能的方式进行协作的呢（例如，用于转发分组的从源到目的地的端到端路径，或与分布式防火墙协作）？提供这些以及许多其他能力是 SDN 控制平面的作用。

5.5.1　SDN 控制平面：SDN 控制器和 SDN 网络控制应用程序

我们通过考虑控制平面必须提供的一般能力开始抽象地讨论 SDN 控制平面。如我们所见，这种抽象即“基本原理”方法将我们引向一个总体的体系结构，该体系结构反映了 SDN 控制平面如何在实际中实现。

如上所述，SDN 控制平面大体划分为两个部分，即 SDN 控制器和 SDN 网络控制应用程序。我们先来仔细考察控制器。自从最早的 SDN 控制器［Gude 2008］开发以来，已经研制了多种 SDN 控制器，文献［Kreutz 2015］极其全面地综述了最新进展。图 5-15 提供了一个通用 SDN 控制器的更为详尽的视图。控制器的功能可大体组织为 3 个层次。我们以一种非典型的自底向上方式考虑这些层次：

- 通信层：SDN 控制器和受控网络设备之间的通信。显然，如果 SDN 控制器要控制远程 SDN 使能的交换机、主机或其他设备的运行，需要一个协议来传送控制器与这些设备之间的信息。此外，设备必须能够向控制器传递本地观察到的事件（例如，一个报文指示一条附属链路已经激活或停止，一个设备刚刚加入了网络，或一个心跳指示某设备已经启动和运行）。这些事件向 SDN 控制器提供该网络状态的最新视图。这个协议构成了控制器体系结构的最底层，如图 5-15 中所示。控制器和受控设备之间的通信跨越了一个接口，它现在被称为控制器的“南向”接口。在 5.5.2 节中，我们将学习 OpenFlow，它是一种提供这种通信功能的特定协议。OpenFlow 在大多数 SDN 控制器中得到了实现（即使不是全部）。
- 网络范围状态管理层。由 SDN 控制平面所做出的最终控制决定（例如配置所有交

换机的流表以取得所希望的端到端转发，实现负载均衡，或实现一种特定的防火墙能力），将要求控制器具有有关网络的主机、链路、交换机和其他 SDN 控制设备的最新状态信息。交换机的流表包含计数器，其值也可以由网络控制应用程序很好地使用；因此这些值应当为应用程序所用。既然控制平面的终极目标是决定用于各种受控设备的流表，控制器也就可以维护这些表的拷贝。这些信息都构成了由 SDN 控制器维护的网络范围“状态”的例子。

- 对于网络控制应用程序层的接口。控制器通过它的“北向”接口与网络控制应用程序交互。该 API 允许网络控制应用程序在状态管理层之间读/写网络状态和流表。当状态改变事件出现时，应用程序能够注册进行通告。可以提供不同类型的 API，我们将看到两种流行 SDN 控制器使用 REST［Fielding 2000］请求响应接口与它们的应用程序进行通信。

我们已经提到过几次，SDN 控制器被认为是“逻辑上集中”的，即该控制器可以被外部视为一个单一、整体的服务（例如，从 SDN 控制设备和外部的网络控制应用程序的角度看）。然而，出于故障容忍、高可用性或性能等方面的考虑，在实践中这些服务和用于保持状态信息的数据库一般通过分布式服务器集合实现。在服务器集合实现控制器功能时，必须考虑控制器的内部操作（例如维护事件的逻辑时间顺序、一致性、意见一致等）的语义［Panda 2013］。这些关注点在许多不同的分布式系统中都是共同的，这些挑战的简洁解决方案可参见［Lamport 1989；Lampson 1996］。诸如 OpenDaylight 和 ONOS 这样的现代控制器已经将很大的注意力放在构建一种逻辑上集中但物理上分布的控制器平台上，该平台提供可扩展的服务和对受控设备以及网络控制应用程序的高可用性。

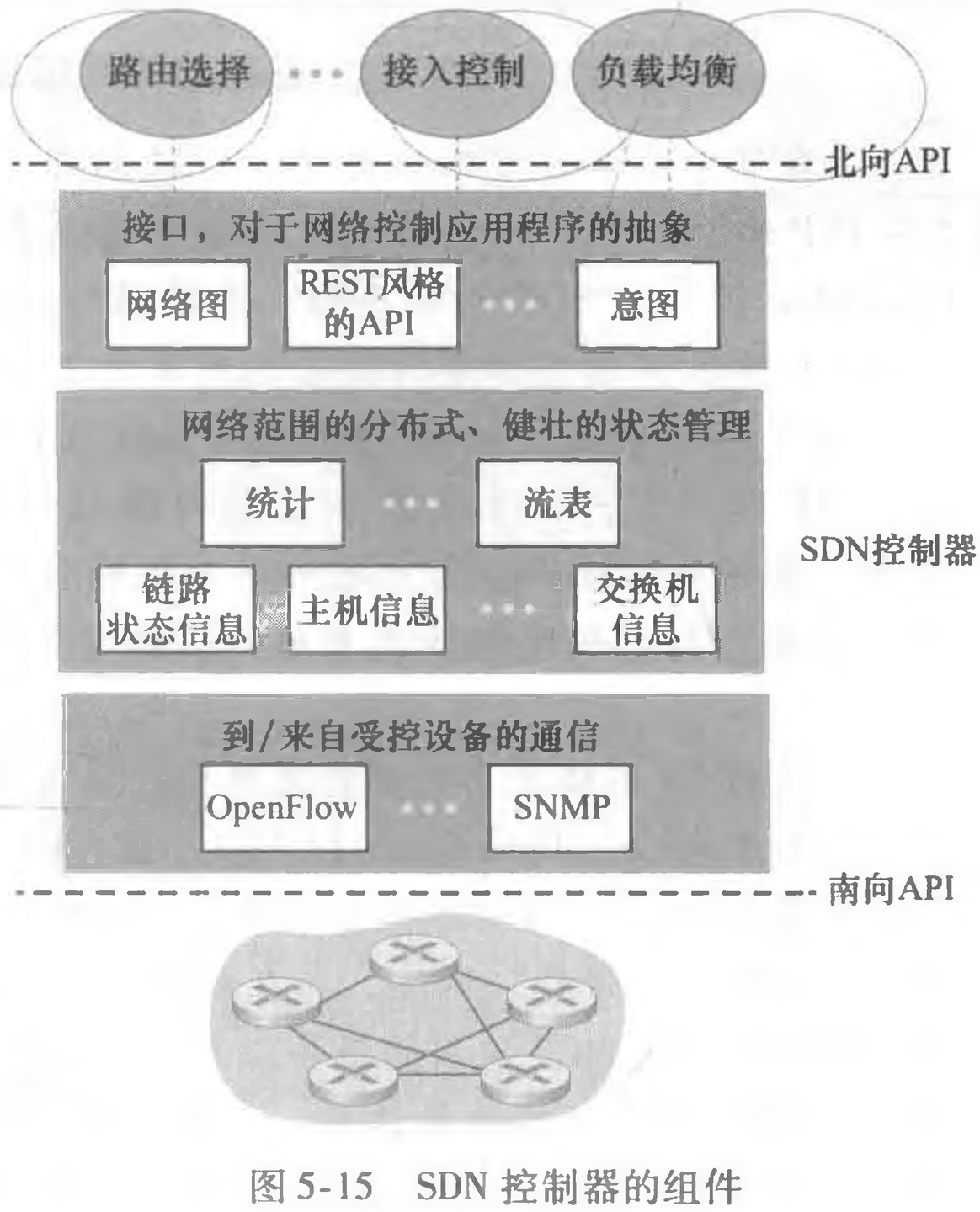

图 5-15　SDN 控制器的组件

在图 5-15 中描述的体系结构与 2008 年最初提出的 NOX 控制器［Gude 2008］以及今天的 OpenDaylight［OpenDaylight Lithium 2016］和 ONOS［ONOS 2016］SDN 控制器（参见插入材料）的体系结构极为相似。我们将在 5.5.3 节介绍一个控制器操作的例子。然而，我们先来仔细审视 OpenFlow 协议，该协议位于控制器的通信层中。

5.5.2　OpenFlow 协议

OpenFlow 协议［OpenFlow 2009；ONF 2016］运行在 SDN 控制器和 SDN 控制的交换机或其他实现 OpenFlow API 的设备之间（OpenFlow API 我们在 4.4 节学习过）。OpenFlow 协议运行在 TCP 之上，使用 6653 的默认端口号。从控制器到受控交换机流动的重要报文有下列这些：