

第13章 开放式最短路径优先

作者：Mark A. Sportack

本章内容包括：

- OSPF起源
- 理解RFC 2328 OSPF，版本2
- 研究OSPF数据结构
- 计算路由

在20世纪80年代即将结束时，距离-向量路由协议的不足变得越来越明显。一种试图改善网络可扩展性的努力是使用基于链路-状态来计算路由，而不是靠跳步数或其他的距离向量。链路是网络中两个路由器之间的连接。链路状态包括传输速度和延迟级等属性。

这一章深入考查了 Internet 工程任务组 (IETF) 的链路-状态，内部网关路由协议类型：开放式最短路径优先 (Open Shortest Path First, OSPF)。OSPF 最先体现在 RFC 1131 中。这个暂时的规范很快被 RFC 1247 淘汰。两个 OSPF 之间有实质性差别以至于 RFC 1247 OSPF 称为 OSPF 版本 2。OSPF 版本 2 不断成熟和演进。之后的一些改变出现在 RFC 1583，2178 和 2328 (最新版本) 中。由于 Internet 和 IP 都是高度动态的，因此 OSPF 很可能会继续发展以跟上 Internet 和 IP 的进步。

13.1 OSPF 起源

IETF——为了满足建造越来越大基于 IP 网络的需要，形成了一个工作组，专门用于开发开放式的、链路-状态路由协议，以便用在大型、异构的 IP 网络中。新的路由协议以已经取得一些成功的一系列私人的、和生产商相关的、最短路径优先 (SPF) 路由协议为基础，SPF 在市场上广泛使用。包括 OSPF 在内，所有的 SPF 路由协议基于一个数学算法——Dijkstra 算法。这个算法能使路由选择基于链路-状态，而不是距离向量。

OSPF 由 IETF 在 20 世纪 80 年代末期开发，OSPF 是 SPF 类路由协议中的开放式版本。最初的 OSPF 规范体现在 RFC 1131 中。这个第 1 版 (OSPF 版本 1) 很快被进行了重大改进的版本所代替，这个新版本体现在 RFC 1247 文档中。RFC 1247 OSPF 称为 OSPF 版本 2 是为了明确指出其在稳定性和功能性方面的实质性改进。这个 OSPF 版本有许多更新文档，每一个更新都是对开放标准的精心改进。接下来的一些规范出现在 RFC 1583、2178 和 2328 中。

OSPF 版本 2 的最新版体现在 RFC 2328 中。最新版只会和由 RFC 2138、1583 和 1247 所规范的版本进行互操作。本章对当前开放式 OSPF 标准的循环开发过程不作讨论，而是集中于讨论 RFC 2328 中规范的最新版 OSPF 的功能、特点及使用。

13.2 理解 RFC 2328 OSPF，版本 2

OSPF 是专门设计用于自治系统之内的 IP 路由协议。如此看来，它不能传输其他可路由网络如 IPX 或 AppleTalk 的报文。如果用户的网络必须适用多种可路由协议，就要考虑使用别的

路由协议而不是 OSPF。

SPF 基于 IP 数据报头中的目的 IP 地址来计算路由，并不提供对非一 IP 目的地的路由计算。而且，各种 OSPF 信息直接封装在 IP 中：无需其他协议 (TCP、UDP 等) 来传输。

OSPF 也被设计用于快速地检测自治系统内的拓扑变化，并且在发现变化之后收敛到新的拓扑。路由决定以自治系统内互联的路由器之间的链路状态为基础。这些路由器各自都维护一个相同的数据库，其中记录了网络的链路状态。这个数据库中包含的是路由器状态，其中有可用的接口、可以到达的相邻路由器及链路、状态信息。

路由表更新，也就是链路-状态广播 (Link-state advertisement, LSA)，直接发送给路由器区内的所有相邻路由器。这种更新过程的技术术语称为洪泛 (flood)，这个术语会给人负面印象，造成对 OSPF 性能特点的错误认识。

实际上，OSPF 网络可以非常快地收敛。网络中的所有路由器运行相同的路由算法并且直接在彼此之间传送路由表更新报文。这些信息用于建立网络和链路的视图。每个路由器的网络视图使用类-UNIX 的树结构，并以自身作为根。这个树称为最短路径树 (shortest-path tree)，记录了到达自治系统内每个目的地的最短路径。自治系统之外的目的地可以通过到外部网络的边界网关来获得，边界网关出现在最短路径树结构的叶上。这样的目的地和 / 或网络的链路-状态数据不被维护，因为它们位于 OSPF 网络之外。因此，它们不能出现在最短路径树的枝上。

13.2.1 OSPF 区

OSPF 快速收敛的一个关键原因是它使用了区。记住，IETF 使用 OSPF 想达到的两个主要目标是：

- 改善网络的可扩展性。
- 快速收敛。

取得两个目标的关键是把网络分成更小的区。一个区是一些网络端系统、路由器以及传输线路的集合体。每个区由一个惟一的区号定义，这个区号配置在每一个路由器内。定义了相同区号的路由器接口成为相同区的组成部分。理想情况下，这些区号不是任意定义的。相反，应该选择区的边界以使不同区之间的流量最小。每个区应反映实际的交通模式而非地理或政治边界。当然，这是理论上的理想情况，在特定环境下可能是不实用的。

OSPF 网络中能支持的区数量受限于区 ID 的大小。这个域是 32 位的二进制数。因此，32 位二进制数的理论最大值应是每一位为 1，其对应的十进制数为 4 294 967 295。显然，能支持的实际最大数比这个理论上的最大数小得多。实际上，网络设计的好坏将决定能在其中支持的最大区数。图 13-1 显示了一个相当简单的 OSPF 网络，其中只有 3 个区，编号为 0、1 和 2。

1. 路由器类型

需要重点记住的是 OSPF 是一个链路-状态协议。因此，链路以及与链路相接的路由器端口定义为区号。基于区成员关系，OSPF 网络中有三种不同类型的路由器：

- 内部路由器。
- 区边界路由器。
- 骨干路由器。

图 13-2 使用图 13-1 给出的网络图来标识三种不同类型的路由器。

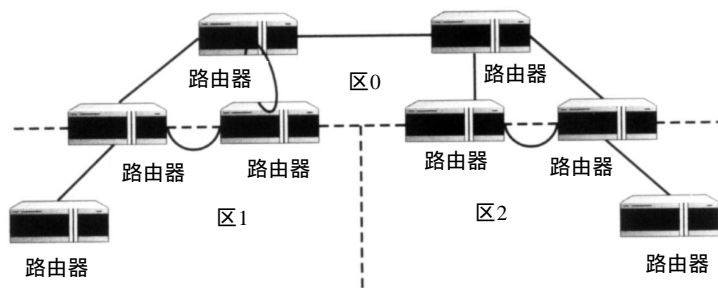


图13-1 具有3个区的小型OSPF网络

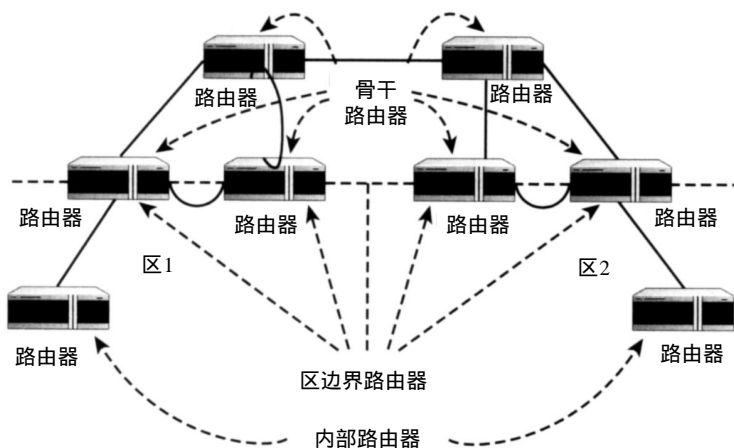


图13-2 OSPF网络中的区边界路由器、内部路由器、骨干路由器

如图13-2所示，具有多个接口的路由器可以属于两个或多个区。这样的路由器成为区边界路由器。也就是说，它们把自身的区号与骨干互联起来。骨干路由器是至少有一个接口定义为属于区0的路由器。一个区边界路由器也可能是一个骨干路由器。任何一个与区0互联的区边界路由器也将成为骨干路由器。

内部路由器使其所有定义接口属于同一区，但这个区不是0区。使用这三种基本的路由器，可以建造高效且可扩展的OSPF网络。

2. 路由类型

考虑图13-2中显示的三种不同类型的OSPF路由器，需要重点注意的是OSPF支持两种不同类型的路由：

- 区内路由
- 区间路由

它们的名字含义相当明显。区内路由是自含的，只限于一个区内部的路由器之间的路由。使用显示在图13-1中的例子网络，图13-3给出了OSPF网络内的区内通信。

区间路由需要在不同的区之间交换数据。所有的区间路由必须经过区0传输，不允许非0区直接和其他区通信。这个层次限制确保了OSPF具有良好的可扩展性，而不会导致链路和路由器的混乱。

图13-4显示了恰当地使用区0能使OSPF网络中的区间通信容易进行。

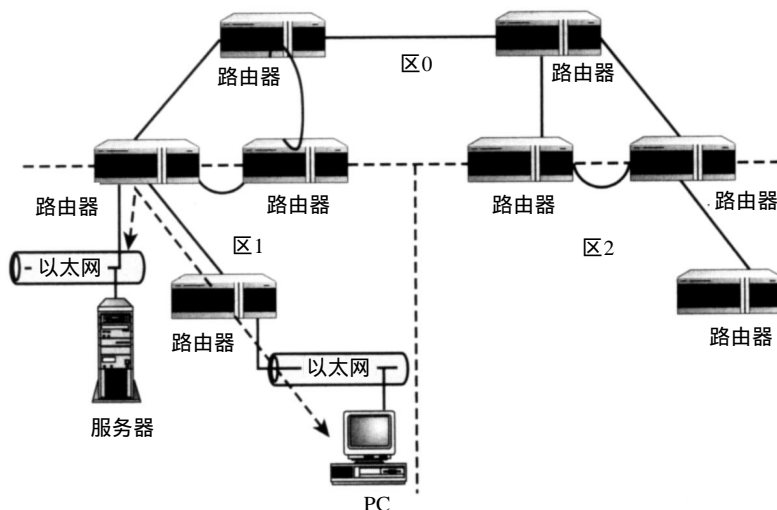


图13-3 OSPF网络中的区内通信

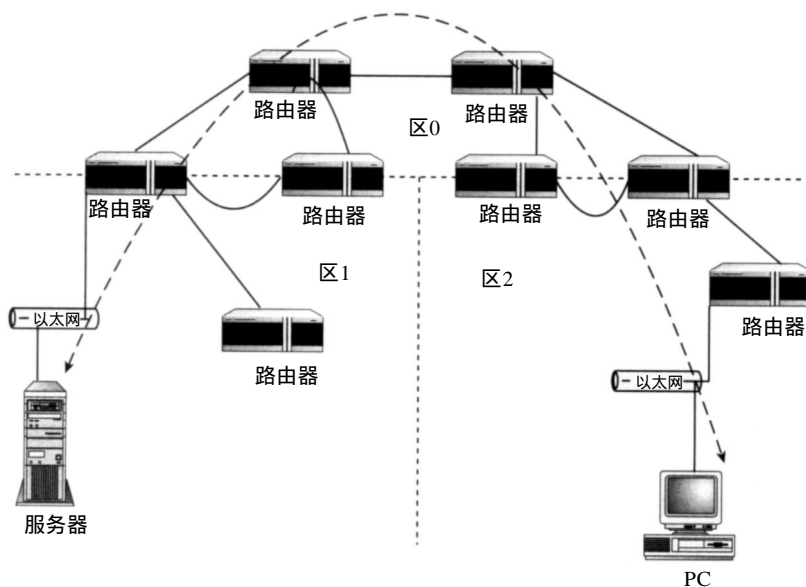


图13-4 OSPF网络中的区间路由

前面的例子显示了在高层，一个OSPF网络中通信是如何工作的。然而，OSPF也能用于在OSPF网络之间交流路由信息，而不仅仅在一个网络的区之间。下面将讨论OSPF的这一用法。

3. 网络之间路由

OSPF可以用于互联不同的网络。这个网络可以是另一个完整的OSPF网络或是一个实现了完全不同路由协议的网络。把OSPF网络与其他不同的路由协议相互联是一项复杂的工作，并要使用一种称为路由再分配的技术。这个术语描述了从一个网络到另一个网络路由信息的汇总和重新分布。从非OSPF网络来的路由信息汇总和重新分布到OSPF网络内。

OSPF网络把所有以这种方式学习来的路由标记为外部的。互联两个不同的 OSPF网络更简单一些，因为无需把一种路由信息耗费转变为另一协议能理解的形式。而且，OSPF使建造自治系统成为可能，一个自治系统(AS)是一个内含网络。字面上讲，AS是一个网络管理员或一组网络管理员使用一个路由协议的系统。

AS的实际定义有些不固定。这其实没有关系。真正重要的是 OSPF允许为一个网络分配自治系统号。一个非常大的 OSPF网络能分成两个或多个自治系统。这些系统可以通过第四种类型的OSPF路由器——自治系统边界路由器(autonomous system border router ASBR)来进行互联。ASBR汇总所有自身AS的路由信息并把汇总情况转发给对应相邻的 ASBR。ASBR的作用非常类似于一个区边界路由器。显然，二者的区别是它们组成了自治系统之间的边界而不是一个自治系统或网络内的区边界。

图13-5显示了利用ASBR进行自治系统互联的情形。

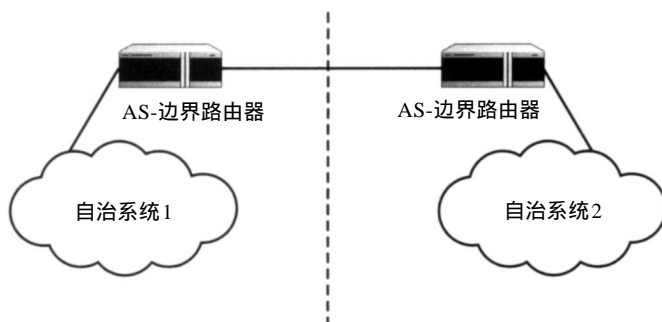


图13-5 互联OSPF自治系统

13.2.2 路由更新

OSPF具有如此可扩展性的一个原因是它的路由更新机制。OSPF使用LSA在OSPF节点之间共享路由信息。这些广播信息会在整个区中进行传播但不会超越一个区。

因此，区中的每一个路由器都知道本区的拓扑。然而，一个区的拓扑对区外是不可知的。考虑到实际上有四种不同类型的 OSPF路由器——区内路由器、区边界路由器、自治系统边界路由器、骨干路由器——很明显每种路由器类型有不同的对等实体集，路由器与这些对等实体交换LSA。

1. 内部区路由器

内部的区路由器必须直接和区中的其他路由器交换 LSA，其中包括每一个区内部路由器，也包括作为区成员的区边界路由器。图 13-6显示了本章前面提及的 OSPF例子网络中，在整个区1中转发或洪泛LSA的情形。需要重点注意的是相同区中的 OSPF路由器无需彼此直接相连就能共享LSA信息。OSPF路由器直接把LSA报文发送到区中每一个知道的路由器，并且使用任何可用的链路来转发那些报文。

蕴含在图 13-5中不太被注意的一点是收敛能够相当快地发生。其中有两个原因，第一个原因是 OSPF路由器能同时直接寻址并发送 LSA至区中所有的路由器(洪泛)，这和RIP使用的“邻居至邻居”的收敛方法完全不同。这样的结果是区内的路由器几乎同时收敛到新拓扑结构。

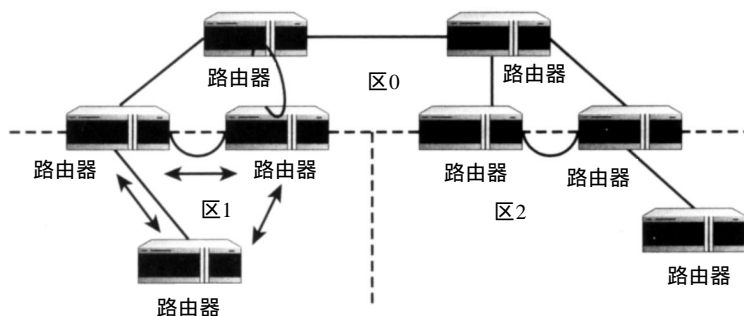


图13-6 在区1内的洪泛

收敛通过区的定义和使用而得到加速。拓扑数据不被传输到区边界之外。因此，收敛不必在自治系统中的所有路由器上发生，而只发生在受影响的区中。这个特点既加速了收敛又增加了网络的稳定性，因为只有自治系统中的一个子网经历不稳定性，这种不稳定性是收敛过程自身带来的。

2. 区边界路由器

区边界路由器负责在数据库中为它们接口所连的每个区维护拓扑信息。因此，如果一个区边界路由器互联了两个不同的区，它必须和两个网络中的对等实体交换 LSA。和区内部路由器一样，这些LSA直接寻址并传输到区中的对等实体。图 13-7显示了这一点。

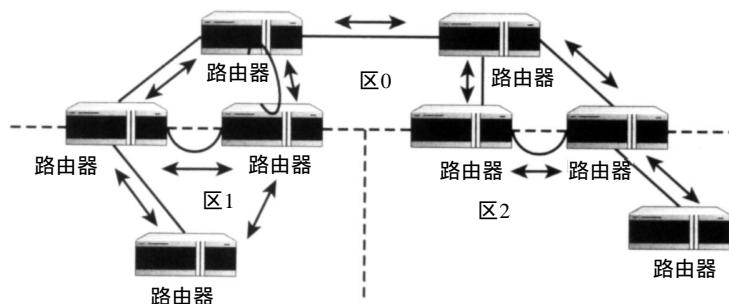


图13-7 在OSPF网络中由区边界路由器引起的区内LSA洪泛

OSPF加强性能的另一个特点是路由汇总。关于一个区的拓扑信息，并不和区外的路由器共享。相反，区边界路由器汇总了所有与其相连的所有区中的地址。这个汇总的路由数据通过LSA报文与其相互联的每个区中的对等路由器实现共享。OSPF使用几种不同类型的LSA：每种有不同的功能。

用于共享汇总路由数据的LSA为类型3 LSA。所有OSPF LSA类型会在本章的剩余部分中描述。

在图13-7中，区边界路由器直接把汇总的数据广播给区0中的所有路由器。OSPF不允许大于或等于1的区之间相互连接。所有这样的互联必须通过区0。因此，其含义是区边界路由器把一个非0编号的区来与区0互联。

3. 骨干路由器

骨干路由器负责维护骨干拓扑信息，并且为自治系统中的每个其他区传播汇总的拓扑信息。

图13-8显示了由骨干路由器交换LSA的情形。

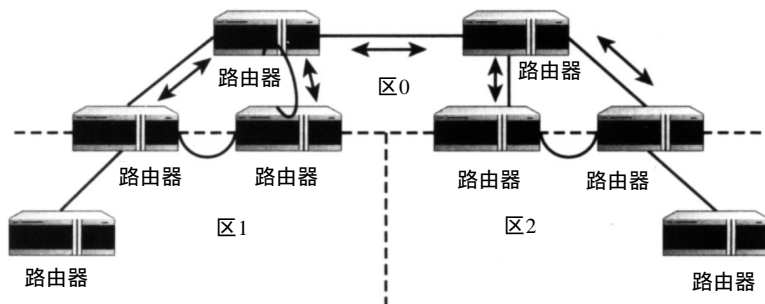


图13-8 在OSPF网络中由骨干路由器引起的区内LSA洪泛

虽然骨干路由器、区边界路由器和区内部路由器之间的差别看起来是清楚的，但由于路由器能支持到其他路由器的多 I/O 端口连接，三者还是会引起混淆。理论上讲，每个端口可以连至一个不同的区。所以，路由器可以在其连接的不同区之间形成边界。

13.3 研究OSPF数据结构

OSPF是相当复杂的路由协议，有许多性能和稳定性方面增强的特点。因此，OSPF使用大量的数据结构就不是奇怪的现象。每个数据结构或信息类型，用于执行一个特定的任务。所有数据结构共享一个通用头，称为OSPF头。OSPF头长度为24字节，包括以下各域：

- 版本号——分配OSPF头的第一个字节用于标识版本号。当前的版本是 2，但是可能会遇到更老的路由器还在运行 RFC 1131 版本1。RFC 1247、1583、2178和2328都对OSPF版本2的向后兼容作了规范。因此无需进一步的标识。
- 类型——第二个字节指出 5种OSPF报文类型中哪一种附加在头结构后面。5种类型(HELLO、数据库描述、链路-状态请求、链路-状态更新和链路-状态应答)用数字标识。
- 报文长度——OSPF头中下面两个字节用于通知接收节点报文的总长度。报文总长度包括数据和头。
- 路由器ID——区中的每个路由器被分配一个惟一的、4字节的标识号。OSPF路由器在发送任何OSPF消息给其他路由器之前，用自己的ID号填充该域。
- 区ID——头中用4字节标识区号。
- 校验和——每个OSPF头包括一个2字节的校验和域，用于检查在传输过程中对报文造成的破坏。发送方对每个消息运行数学计算，然后把结果存储在这个域中。接收方对接收到的报文运行相同的算法并把结果与存储在校验和域中的结果进行比较。如果报文无损到达，两个结果应一样；不相同，说明OSPF报文在传输过程中被破坏。接收方会简单地把受损报文丢弃。
- 认证类型——OSPF能通过认证OSPF信息的发送者来防止会导致假路由信息这样的攻击。两字节的认证类型域标识信息中使用的各种认证形式。
- 认证——头中剩下的9个字节携带的是认证数据，接收方利用此信息来确定信息的发送者。OSPF允许网络管理员使用各种级别的认证：从无认证，到简单认证，到最强大的MD认证，基本结构中包含OSPF节点所需的用于决定报文是否应接收并作进一步处理，

还是应丢弃的所有信息。在传输过程中受损的(校验和指出这一点)及没有通过认证的报文会被丢弃。

OSPF使用5种不同的报文类型。每种类型用于支持不同的,专门的网络功能。这5种类型是:

- HELLO报文(类型1)。
- 数据库描述报文(类型2)。
- 链路-状态请求报文(类型3)。
- 链路-状态更新报文(类型4)。
- 链路-状态应答报文(类型5)。

这5种报文类型有时用编号指明,而不是用名字。所以,OSPF类型5报文实际上是指链路-状态应答报文。所有这些报文类型使用OSPF头。

注意 5种基本的OSPF数据结构用5个纯粹的数表示,对这些结构和大小的详细讨论超出了本章的范围。相反,这一章仅限于讨论这些数据类型的目的和使用。

13.3.1 HELLO报文

OSPF包含一个用于建立和维护相邻站点之间关系的协议(HELLO协议)。这些关系称为连接性。连接性是OSPF交换路由数据的基础。

通过这个协议和报文类型,OSPF节点能发现区中的其他OSPF节点。它的名字表明了其含义,HELLO协议在可能的相邻路由器之间建立通信。HELLO协议使用特别的子报文结构,这个结构附加到标准24字节的OSPF头后面。这些结构共同构成HELLO报文。

OSPF网络中的所有路由器必须遵守一定的规则,这个规则在整个网络中要一致。这些规则包括:

- 网络掩码。
- HELLO报文广播的间隔。
- 网络中的其他路由器认为一个没有反应的路由器为死节点的时间(路由器死时间间隔)。

OSPF中的所有路由器对这些参数必须使用相同值,否则网络可能不会正常工作。这些参数通过HELLO报文进行交换。它们一起构成相邻节点之间通信的基础。它们要确保在不同网络的路由器之间不形成相邻关系(连接性),并且网络中的所有成员要对多久彼此联系一次达成共识。

HELLO报文也包括最近已与其联系过的其他路由器列表(使用它们自己惟一的路由器ID)。这个Neighbor(相邻者)域使邻居发现过程成为可能。HELLO报文还包括几个其他的域,如Designated Router(指定路由器)、Backup Designated Router(备份指定路由器)和其他一些域。这些域对于维护连接性,支持OSPF网络的稳定周期和收敛都是有用的。Designated Router和Backup Designated Router的用处将在本章后面的节中描述。

13.3.2 数据库描述报文

当OSPF中的两个路由器初始化连接时要交换数据库描述(DD)报文。这个报文类型用于描述,而非实际地传送OSPF路由器的链路-状态数据库内容。由于数据库的内容可能相当长,所以可能需要多个数据库描述报文来描述整个数据库。实际上,保留了一个域用于标识数据

库描述报文序列。接收方对报文的重新排序使其能够真实地复制数据库描述报文。

DD交换过程按询问/应答方式进行,在这个过程中,一个路由器作为主路由器。另一个路由器作为从路由器,主路由器向从路由器发送它的路由表内容。显然,主从之间的关系会因每个DD交换的不同而不同。网络中的所有路由器会在不同时刻作用,在这个过程中既可能是主又可能是从。

13.3.3 链路-状态请求报文

OSPF报文的第三种类型为链路-状态请求报文。这个报文用于请求相邻路由器链路-状态数据库中的一部分数据。表面上讲,在收到一个DD更新报文之后,OSPF路由器可以发现相邻信息不是比自己的更新就是比自己的更完全。如果是这样,路由器会发送一个或几个链路-状态请求报文给它的邻居(具有更新信息的路由器)以得到更多的链路状态信息。

请求的信息必须是非常具体的。它必须使用下面的标准规范指明所要求的数据:

- 链路-状态(LS)类型号(1到5)。
- LS标识。
- 通告路由器。

这些规范一起指明了一个具体的OSPF数据库子集,而不是它的一个事例。一个事例是与信息相同的子集,这个子集带有暂时边界(也就是时戳)。记住,OSPF是一个动态路由协议,它能对网络中链路状态的变化自动作出反应。因此,LS请求的接收者把对这些特定路由信息解释为最新数据。

13.3.4 链路-状态更新报文

链路-状态更新报文用于把LSA发送给它的相邻节点。这些更新报文是用于对LSA请求的应答。有5种不同的LSA报文类型。这些报文类型用从1到5的类型号标识。

注意 由于OSPF通常把链路-状态广播看作LSA,因此会存在潜在的混淆。然而,实际上用于更新路由表的机制为链路-状态更新报文——简记为LSU。还有另一个报文结构,链路-状态应答报文,简称为LSA;由于一些不可知的原因,这种报文称为链路-状态应答,而LSA通常是指更新报文。

这些报文类型及其LSA号,如下所述:

- Router LSA (路由器LSA)(类型1)——路由器LSA描述了路由器链路到区的状态和耗费。所有这样的链路必须在一个LSA报文中进行描述。同时,路由器必须为它属于的每个区产生一个路由器LSA。所以,区边界路由器将产生多个路由器LSA,而区内的路由器只需产生一个这样的更新。
- Network LSA (网络LSA)(类型2)——网络LSA与路由器LSA相似,它描述的是连接进网络的所有路由器的链路状态和耗费信息。二者的区别是网络LSA是网络中所有链路-状态和耗费信息的总和。只有网络的指定路由器记录这个信息,并由它来产生网络LSA。
- Summary LSA-IP Network (汇总LSA-IP网络)(类型3)——使用汇总LSA-IP这个名字有些不灵活,因此OSPF的设计者采用了编号策略来记LSA!只有OSPF网络中的区边界路由器能产生这种LSA类型。使用这种LSA类型把一个区的汇总路由信息和OSPF网络中相

邻区路由器信息进行交换。它经常汇总缺省的路由而不是传播汇总的 OSPF 信息至其他网络。

- Summary LSA-Autonomous System Boundary Router(汇总LSA-自治系统边界路由器)(类型4)——类型4与类型3 LSA的关系密切。二者的区别是类型3描述区内路由,而类型4描述的是OSPF网络之外的路由。
- AS-外部LSA(类型5)——第5个LSA是自治系统外部LSA。正如其名,这种LSA用于描述OSPF网络之外的目的地。这些目的地可以是特定主机或是外部网络地址。作为和外部自治系统相联系的ASBR OSPF节点负责把外部路由信息在它属于的整个区中传播。

这些LSA类型用于描述OSPF路由域的不同方面,它们直接寻址到OSPF区中的每一个路由器并同时传输。这样的洪泛确保OSPF区中的所有路由器关于网络的5个不同方面(LSA类型)有一样的信息。路由器完整的LSA数据存储在链路-状态数据库中。当Dijkstra算法应用于这些数据库的内容时会得到OSPF路由表。表和数据库的区别是数据库含有原始数据的完整集合,而路由表包含通过特定路由器接口到已知目的地的最短路径列表。

不必研究每种LSA类型的结构,只需研究它们的头就足够了。

1. LSA头

所有的LSA使用一个通用的头格式。这个头20字节长并附加于标准的24字节OSPF头后面。LSA头唯一地标识了每种LSA。所以,它包括关于LSA类型、链路-状态ID及通告路由器ID的信息。下面是LSA头域:

- LS年龄——LSA头中的前两个字节包含LSA的年龄。这个年龄是自从LSA产生时已消逝的时间秒数。
- OSPF选项——下面的字节由一系列标志组成,这些标志标识了OSPF网络能提供的各种可选的服务。
- LS类型——1字节LS类型指出5种LSA类型中的一种。每种LSA类型的格式是不同的。因此,指出何种类型的数据附加在头后面必不可少。
- 链路-状态ID——链路-状态ID域4字节长用于指明LSA描述的特定网络环境区域。这个域与前面提及的LS类型域关系紧密。实际上,这个域的内容直接依赖于LS类型。比如,在路由器LSA中,链路-状态ID包含产生了这个报文的OSPF路由器ID——通告路由器ID。
- LS顺序号——OSPF路由器会递增每个LSA报文的序列号。所以,接收到两个相同LSA事例的路由器有两种选择来决定哪一个是最新的报文,LS顺序号域4字节长。检查这个域可以确定LSA在网络中已传输了多久。从理论上讲,一个新的LSA年龄比一个老的LSA年龄大是有可能的,特别是在大型复杂的OSPF网络中。所以,接收路由器比较LS顺序号。大号的LSA是最新生成的,这种机制不会因动态路由的变迁而受到损坏,而应认为其是一种更可靠的确定LSA时间的方法。
- LS校验和——3字节的LS校验和用于检查LSA在传输到目的地的过程中是否受到破坏。校验和采用简单的数学算法。它的输出结果依赖于其输入,并且有高度的一致性。给定相同的输入,校验和算法总是给出相同的输出。LS校验和域使用部分LSA报文内容(包括头,不包括LS年龄和校验和域)来生成校验和值。源节点运行Fletcher算法并把结果存于LS校验和域中。目的节点执行相同的算法并把结果与存储在校验和域中的结果比较,如果两个值不相同,就可以认为报文在传输过程中被破坏。之后,产生一个传输请求。

• LS 长度——LS 长度域用于通知接收方 LSA 的长度(以字节为单位)，这个域 1 个字节长。

LSA 报文体的剩余部分包含一个 LSA 的列表。每个 LSA 描述 OSPF 网络 5 个不同方面中的一个。所以，路由器 LSA 报文会广播区内已知存在的路由器信息。

2. 处理 LSA 更新

OSPF 路由表与其他路由表的本质区别是它的更新并不直接被接收站点所使用。从其他路由器接收到的更新包含“从发送路由器角度看”网络得到的信息。所以，在使用和解释接收到的 LSA 数据之前必须由 Dijkstra 算法，把它转化为自己本身的信息。

表面上讲，LSA 的传输是因为一个路由器检测到了链路状态变化。所以，在接收到任何类型的 LSA 之后，OSPF 路由器必须把 LSA 的内容和自身路由表的对应部分进行比较。只有通过 SPF 算法，使用新数据形成新的网络视图之后才能进行比较，SPF 算法输出的结果是得到网络的新视图。这些结果与已存在的 OSPF 路由表相比较，看是否它的路由受到了网络状态变化的影响。

如果由于状态变化必须改变一条或多条路由，就要使用新的信息建造一个新的路由表。

3. 复制 LSA

考虑到 LSA 在整个 OSPF 区内洪泛，就有可能同时存在多个相同 LSA 类型的事例。因此，OSPF 网络的稳定性要求路由器能够识别多个 LSA 中的最新者。收到两个或多个相同 LSA 类型的路由器会检查 LSA 头中的 LS 年龄、LS 顺序号以及 LS 校验和域。只有包含在最新 LSA 中的信息才被接受，并且要经过前面一节中描述的处理过程。

13.3.5 链路-状态应答报文

第 5 种 OSPF 报文是链路-状态应答报文。OSPF 的特点是可靠地分布 LSA 报文(LSA 表示链路-状态通告(advertisement)，通告而不是链路-状态应答)，可靠性意味着通告的接收方必须应答。否则，源节点将没有办法知道是否 LSA 已到达目的地。因此，需要一些应答 LSA 接收的机制。这个机制是链路-状态应答报文。

链路-状态应答报文惟一地标识其要应答的 LSA 报文。标识以包含在 LSA 头中的信息为基础，包括 LS 顺序号和通告路由器。LSA 与应答报文之间无需 1 对 1 的对应关系。多个 LSA 可以用一个报文来应答。

13.4 计算路由

OSPF 虽然很复杂，却使用下面两种相当简单的方法之一计算路由耗费：

- 非带宽敏感的缺省值可以用于每一个 OSPF 接口。
- OSPF 能自动计算使用每个路由接口的耗费。

不管使用哪种方法，任何一条路由的耗费可以通过把路由上遇到的每个路由器接口耗费加起来得到。在 OSPF 的最短路径树中记录了每一个已知目的地的和耗费。

13.4.1 使用自动计算

OSPF 能自动计算一个接口的耗费，这个算法以每个接口类型支持的带宽为基础。一条路由上所有接口计算值的和(sum)形成 OSPF 路由决定的基础。基于冗余链路上可获得的带宽，这些值能使 OSPF 计算出最小耗费的路由。图 13-9 中的网络显示了这一点。

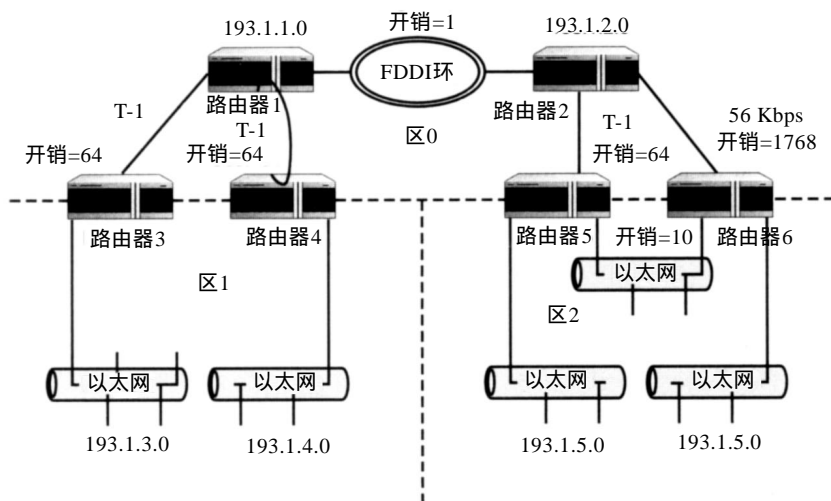


图13-9 自动计算的路由耗费

在图13-9中，位于网络193.1.3.0中的主机与位于网络193.1.4.0中端系统之间的广域网路由耗费为138。这个耗费是这两个网络之间两条T1链路的耗费和（每条64），再加上以太网接口至网络193.1.4.0的耗费。在起点和终点的以太网接口耗费不包含在OSPF耗费计算中，这是因为OSPF只计算向外的路由器接口耗费。

表13-1汇总了图13-9中每个被使用接口自动计算得出的耗费值。

表13-1 每个接口类型的计算耗费

接口类型	耗 费
100Mbps FDDI	1
10Mbps Ethernet	10
1.544Mbps T1 串行链路	64
56Kbps 串行链路	1 768

13.4.2 使用缺省路由耗费

读者最感兴趣的或许是让OSPF自动计算路由耗费，虽然这有点不可能。比如，旧的路由器可能不支持自动计算特性。在这种情况下，所有的接口具有相同的OSPF耗费。因此，T3会和56Kbps的租用线路具有相同的耗费。显然，这两种线路提供了非常不同的性能级别，这点不同应该成为路由决定的基础。

然而，有一些环境会接受使用缺省路由耗费。比如，用户的网络由类似的传输线路组成，那么缺省值将是可接受的。另外，管理员能手动地为某个特定接口修改耗费度量。这样会使网络管理员在主要仍使用缺省路由耗费的前提下，对网络的流量模式进行合理地规划。

1. 同构网络

在同构网络中，所有的传输线路是一样的。比如，所有的LAN接口都是10Mbps的以太网，所有WAN接口都是T1。在这种情形下，使用缺省值不大会引起路由问题。这一点在几乎没有或有很少路由冗余的情况下是非常正确的。

为了显示这一点，考虑图 13-10 中的网络图。

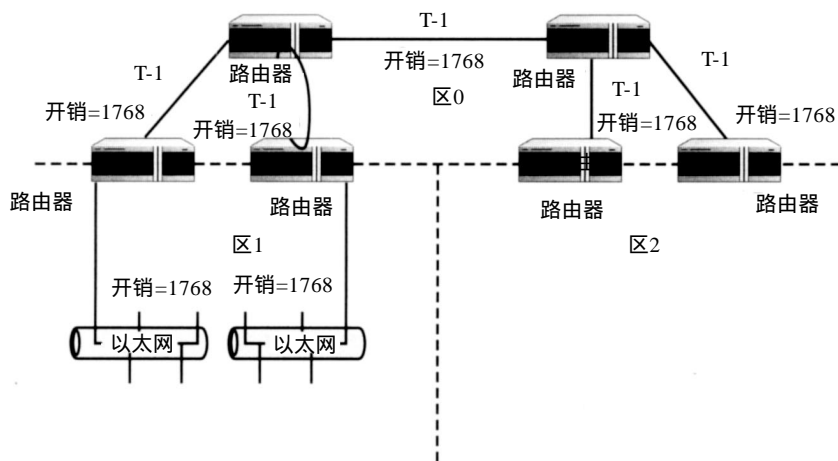


图13-10 可以接受地使用OSPF缺省接口值

在图13-10中，缺省值1 768分配给每一个接口。然而所有的 WAN链路是T1。考虑到所有的值都一样，那么分配值为1 128、1 768还是1 000 000就没太大关系！同构网络中的路由决定变成简单计算和比较跳数，不管网络中有多少路由冗余，这一点都是正确的。

显然，在有相当多路由冗余并且使用不同的传输技术的网络中，缺省值将不会选出到任何目的地的最优路由。

2. 手动设置值

在一些网络中，希望接受 OSPF缺省路由，之后手动地设置那些不同于缺省链路的特殊链路值。比如，用户网络的缺省耗费值可能是 1 768——56Kbps串行链路的计算值。

如果网络中只有一条或两条链路不提供相同的带宽，就能接受缺省值之后为那些特殊链路设置其他值。

是使用自动计算的路由耗费，还是缺省耗费，或是手动配置的耗费对 OSPF节点而言都是不重要的。它们会接受所有这样的耗费值并计算得到网络的最短路径树。

13.4.3 最短路径树

各种LSA复制的目的是使路由器能构造网络拓扑视图。这个拓扑以树的方式安排。OSPF路由器形成树的根。这个树给出到所知目的地地址的完整路径，虽然只有下一跳用于转发报文。其中的原因是简单的，记录到目的地的完整路径使冗余路径的比较和选择最好路径成为可能。如果有多条相同耗费的路径，它们会被 OSPF发现并使用，流量在这些可用链路中大致取得均衡。

1. 路由器3的视图

为了更好地理解最短路径树的概念，考虑图 13-11中的网络。图中的简单网络是一个小型 OSPF网。网络管理员已经启动了路由耗费的自动计算。需要重点注意的是路由器 5和6之间的以太网构成了网络 193.1.5.0和193.1.6.0通过路由器2的另一条路径，所以，OSPF自动计算的耗费为10，而相似的耗费没有分配到其他以太网上。

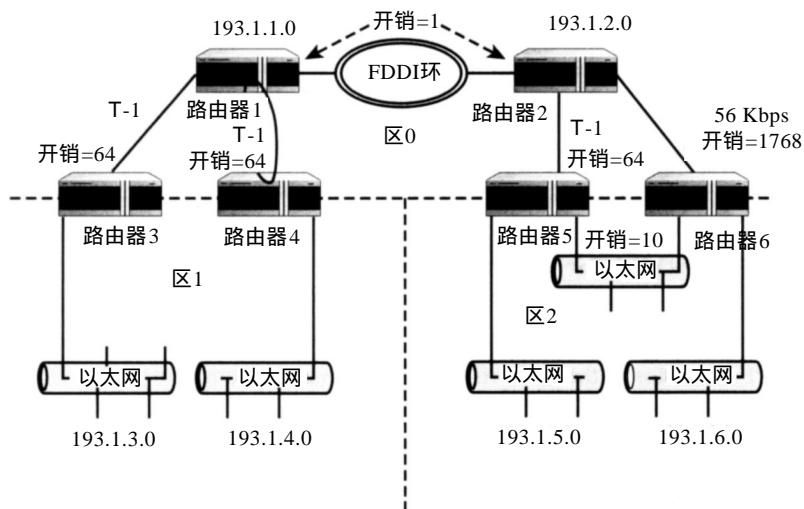


图13-11 具有路由耗费的OSPF网络

这个网络(图13-10)的最短路径树会随路由器的不同而变化。图13-12是从路由器3的角度看到的树。

从图13-12可以明显地看出，树结构使到给定目的地路由耗费的计算简化。根路由器(路由器3——193.1.3.0)能很快地把到任何目的地的路由上所遇到的路由器接口耗费加起来。从路由器3的角度，到任何一个网络的路由耗费汇总在表13-2中。对于多于一跳的目的地，接口耗费相加在括号内。这样可以使读者沿着图13-12中的网络进行路径跟踪。

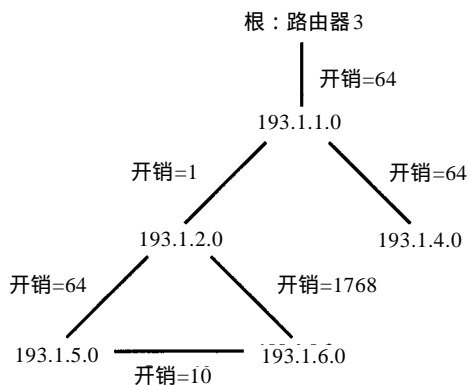


图13-12 路由器3的最短路径树

表13-2 从路由器3到已知目的地的耗费

目的地	跳数	累积耗费
193.1.3.0	—	0
193.1.1.0	1	64
193.1.2.0	2	65(64+1)
193.1.4.0	2	128(64+64)
193.1.5.0	3	129(64+1+64)
193.1.6.0	3	1 833(64+1+1768)
193.1.6.0	4	139(64+1+64+10)

在这个例子中，有两条到网络193.1.6.0的路由。一条路径含更少的跳数，但却有高得多的耗费，这是因为路由器2和6之间的低速串行链路的存在。另一条路由有更多的跳数，但却有少得多的总耗费。在这种情况下，OSPF会抛弃高耗费的路由而使用低耗费的路由。如果这

两条冗余的路由具有相同的总耗费，OSPF会在路由表中维护两条独立的表项并尽可能平均地在二者之间均衡负载。

2. 路由器2的视图

网络中每个路由器的视图是不一样的。虽然考查每个路由器的视图有点枯燥，但是举一个例子来看一下不同路由器对最短路径树的影响还是有必要的。图 13-13显示的是路由器 2 的最短路径树。

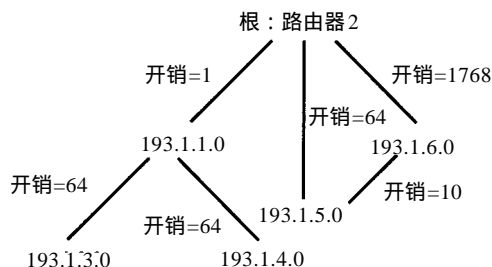


图13-13 路由器2的最短路径树

表13-3汇总了从路由器2到所有已知目的地的路由耗费。

表13-3 从路由器2到已知目的地的耗费

目 的 地	跳 数	积 累 耗 费
193.1.3.0	—	0
193.1.5.0	1	64
193.1.6.0	1	1 768
193.1.6.0	2	74(64+10)
193.1.1.0	1	1
193.1.3.0	2	65(1+64)
193.1.4.0	2	1(1+64)

比较表 13-2和表 13-3，显示出在一个网络中源和目的之间的积累距离随起点不同而不同。视图好像可以说明任何问题。这就是为什么 OSPF路由器使用从其他路由器处通过 LSA更新得到的数据来构造自己的网络视图，而不直接使用那些信息来更新路由表的原因。

13.5 小结

OSPF是功能最强大、特点最丰富的开放式路由协议之一。它的复杂性也是其弱点来源，因为设计、建造和操作一个 OSPF互联网络需要比使用几乎每一种其他路由协议更多的专业知识和精力。采用路由耗费的缺省值可以极大地简化 OSPF网络设计。随着关于 OSPF及网络操作特点知识的增加，用户能够慢慢地通过控制 OSPF变量来优化网络性能。

必须小心地设计区和网络拓扑。做得好，OSPF会使网络用户得到优异的性能和快速的收敛速度。在第 14章中会讨论另一种特殊的路由协议。BGP用于特大型网络如Internet的核心。