

探知；识别；构建；标识；标注；单击、间隔计时器；通断（失效）；废止（删除） （浓缩）
Network Working Group J. Moy
Request for Comments: 2328 Ascend Communications, Inc.
STD: 54 April 1998
Obsoletes: 2178
Category: Standards Track
翻译：马儿快跑 2002 年 12 月

OSPF 版本 2

本备忘录的状态：

本文档讲述了一种 Internet 团体的 Internet 标准跟踪协议，它需要进一步进行讨论和建议以得到改进。请参考最新版的“Internet 正式协议标准”（STD1）来获得本协议的标准化程度和状态。本备忘录的发布不受任何限制。

版权提示：

Copyright (C) The Internet Society (1998). All Rights Reserved.

摘要：

本备忘录说明了 OSPF 协议版本 2。OSPF 是一种连接状态/link-state 路由协议，被设计用于单一的自制系统/Autonomous System 中。每个 OSPF 路由器都维持着同样的数据库以描述 AS 的拓扑结构，并以此数据库来创建最短路径树并计算路由表。

OSPF 在发现拓扑改变后，仅利用很少的路由流量就可以快速的重新计算出路径。OSPF 提供等值多路径。通过提供区域/area 路径，来提供额外的路径保护并可以减少协议所需要的流量。此外，所有的 OSPF 路由信息交换都经过验证。

本备忘录与 RFC 2178 的不同，在附录 G 中说明。所有的不同点都实现向后兼容。

按本备忘录的实现，与按照 RFC2178、1583、1247 的实现能够协同运作。

请将建议发往 ospf@gated.cornell.edu。

请将有关本文翻译的建议发往 raymon@itpub.net

目录

1	绪论
1.1	协议概述
1.2	常用术语的定义
1.3	连接状态路由技术的简要历史
1.4	本文档的结构
1.5	感谢
2	连接状态数据库：组织和计算
2.1	路由器和网络的表示方法
2.1.1	非广播网络的表示方法
2.1.2	一个连接状态数据库的示例
2.2	最短路径树
2.3	使用外部路由信息
2.4	等值多路径
3	将自制系统划分为区域
3.1	自制系统的骨干区域
3.2	区域间路由
3.3	路由器的分类
3.4	一个简单区域配置
3.5	IP 子网化支持
3.6	支持存根区域
3.7	区域的划分
4	功能摘要
4.1	区域间路由
4.2	自制系统外部路由
4.3	路由协议包
4.4	基本实现的需求
4.5	OSPF 可选项
5	协议数据结构
6	区域数据结构
7	形成邻接
7.1	Hello 协议
7.2	数据库同步
7.3	指定路由器
7.4	备份指定路由器
7.5	邻接图
8	协议包处理
8.1	发送协议包
8.2	接收协议包
9	接口数据结构
9.1	接口状态
9.2	引起接口状态改变的事件
9.3	接口状态机
9.4	选举指定路由器
9.5	发送 Hello 包
9.5.1	在 NBMA 网络上发送 Hello 包
10	邻居数据结构
10.1	邻居状态
10.2	引起邻居状态改变的事件

10.3	邻居状态机
10.4	是否形成邻接
10.5	接收到 Hello 包
10.6	接收到数据库描述包
10.7	接收到连接状态请求包
10.8	发送数据库描述包
10.9	发送连接状态请求包
10.10	示例
11	路由表结构
11.1	查找路由表
11.2	路由表示例，无区域
11.3	路由表示例，有区域
12	连接状态宣告 (LSA)
12.1	LSA 头部
12.1.1	连接状态时限
12.1.2	选项
12.1.3	连接状态类型
12.1.4	连接状态标识
12.1.5	宣告路由器
12.1.6	连接状态序号
12.1.7	连接状态校验和
12.2	连接状态数据库
12.3	TOS 表现
12.4	生成 LSA
12.4.1	Router-LSA
12.4.1.1	描述点对点接口
12.4.1.2	描述广播和 NBMA 接口
12.4.1.3	描述虚拟通道
12.4.1.4	描述点对多点接口
12.4.1.5	Router-LSA 示例
12.4.2	Network-LSA
12.4.2.1	Network-LSA 示例
12.4.3	Summary-LSA
12.4.3.1	向存根区域生成 Summary-LSA
12.4.3.2	Summary-LSA 示例
12.4.4	AS-external-LSA
12.4.4.1	AS-external-LSA 示例
13	洪泛过程
13.1	判定较新的 LSA
13.2	将 LSA 加入数据库
13.3	洪泛过程的下一步操作
13.4	接收自生成的 LSA
13.5	发送连接状态确认包 (LSAck 包)
13.6	重传 LSA
13.7	接收连接状态确认包 (LSAck 包)
14	老化连接状态数据库
14.1	提前老化 LSA
15	虚拟通道
16	计算路由表
16.1	计算一个区域的最短路径树

16.1.1	计算下一跳
16.2	计算区域间路径
16.3	查看传输区域的 Summary-LSA
16.4	计算 AS 外部路径
16.4.1	外部路径参数
16.5	增量更新——Summary-LSA
16.6	增量更新——AS-external-LSA
16.7	路由表改变引起的事件
16.8	等值多路径
	脚注
	引用
A	OSPF 数据格式
A.1	OSPF 包的封装
A.2	选项域
A.3	OSPF 包格式
A.3.1	OSPF 包头
A.3.2	Hello 包
A.3.3	数据库描述包 (DD 包)
A.3.4	连接状态请求包 (LSR 包)
A.3.5	连接状态更新包 (LSU 包)
A.3.6	连接状态确认包 (LSAck 包)
A.4	LSA 格式
A.4.1	LSA 头部
A.4.2	Router-LSA
A.4.3	Network-LSA
A.4.4	Summary-LSA
A.4.5	AS-external-LSA
B	结构常量
C	可配置变量
C.1	全局参数
C.2	区域参数
C.3	路由器接口参数
C.4	虚拟通道参数
C.5	NBMA 网络参数
C.6	点对多点网络参数
C.7	主机路径参数
D	验证
D.1	空验证
D.2	简单口令验证
D.3	密码验证
D.4	信息生成
D.4.1	生成空验证
D.4.2	生成简单口令验证
D.4.3	生成密码验证
D.5	信息校验
D.5.1	校验空验证
D.5.2	校验简单口令验证
D.5.3	校验密码验证
E	设定 LS 标识的一种算法
F	多接口接入同一网络/子网

G	与 RFC 2178 的不同
G.1	洪泛过程的修改
G.2	外部路径优先级的改变
G.3	解决不完整的虚拟下一跳
G.4	路由表查找
	安全性考虑
	作者的地址
	完整的版权声明

1. 绪论

本文档描述了开放最短路径优先/Open Shortest Path First (OSPF) TCP/IP 网际路由协议。OSPF 是一种典型的内部网关协议/Interior Gateway Protocol (IGP)。这意味着其路由信息是描述属于同一个自制系统/Autonomous System (AS) 中的路由器。OSPF 协议是基于连接状态或被称为 SPF 的技术，这与传统 TCP/IP 网际路由协议所使用的 Bellman-Ford 技术不同。

OSPF 协议是由 Internet Engineering Task Force 的 OSPF 工作组所开发的，特别为 TCP/IP 网络而设计，包括明确的支持 CIDR 和标记来源于外部的路由信息。OSPF 也提供了对路由更新的验证，并在发送/接收更新时使用 IP 多播。此外，还作了很多的工作使得协议仅用很少的路由流量就可以快速地响应拓扑改变。

1.1. 协议概述

OSPF 仅通过在 IP 包头中的目标地址来转发 IP 包。IP 包在 AS 中被转发，而没有被其他协议再次封装。OSPF 是一种动态路由协议，它可以快速地探知 AS 中拓扑的改变（例如路由器接口的失效），并在一段时间的收敛后计算出无环路的新路径。收敛的时间很短且只使用很小的路由流量。

在连接状态路由协议中，每台路由器都维持着一个数据库以描述 AS 的拓扑结构。这个数据库被称为连接状态数据库，所有参与的路由器都有着同样的数据库。数据库中的各项说明了特定路由器自身的状态（如该路由器的可用接口和可以到达的邻居）。该路由器通过洪泛/flooding 将其自身的状态传送到整个 AS 中。

所有的路由器同步地运行完全相同的算法。根据连接状态数据库，每台路由器构建出一棵以其自身为树根的最短路径树。最短路径树给出了到达 AS 中各个目标的路径，路由信息的起源在树中表现为树叶。

当有多条等值的路径到达同一目标时，数据流量将在这些路径上平均分摊。路径的距离值表现为一个无量纲数。

OSPF 允许将一些网络组合到一起。这样的组被称为区域/area。区域对 AS 中的其他部分隐藏其内部的拓扑结构，信息的隐藏极大地减少了路由流量。同时，区域内的路由仅由区域自身的拓扑来决定，这可使区域抵御错误的路由信息。区域通常是一个子网化了的 IP 网络。

OSPF 允许灵活的配置 IP 子网。由 OSPF 发布的每条路径都包含目标和掩码。同一个 IP 网络的两个子网可以有不同的大小（即不同的掩码），这常被称为变长子网/variable length subnetting。数据包按照最佳匹配（最长匹配）来转发。主机路径被看作掩码为“全 1”（0xffffffff）的子网来处理。

OSPF 协议中所有的信息交换都经过验证。这意味着，在 AS 中只有被信任的路由器才能参与路由。有多种验证方法可以被选择。事实上，可以为每个 IP 子网选用不同的验证方法。

来源于外部的路由信息（如路由器从诸如 BGP [引用 23] 的外部网关协议中得到的路径）向整个 AS 内部宣告。外部数据与 OSPF 协议的连接状态数据相对独立。每条外部路径可以由所宣告的路由器作出标记，在自制系统边界路由器 (ASBR) 之间传递额外的信息。

1.2. 常用术语的定义

本节定义了贯穿本文的，在 OSPF 协议中有特定含义的术语。对 IP 协议不熟悉的读者可使用 [引用 13] 作为 IP 协议的绪论。

路由器/Router:

一种三层 IP 包的交换设备。在早期的 IP 文献中被称为网关/gateway。

自制系统/Autonomous System:

一组使用相同路由协议交换路由信息的路由器，缩写为 AS。

内部网关协议/Interior Gateway Protocol:

被一个 AS 内的路由器所使用的路由协议，缩写为 IGP。每个 AS 使用单一的 IGP，不同的 AS 会使用不同的 IGP。

路由器标识/Router ID:

一个 32 位的数字，用以识别每台运行 OSPF 协议的路由器。在一个 AS 中，这个数字可以唯一地表示出一台路由器。

网络/Network:

在本备忘录中，会表示 IP 网络/子网/超网。一个物理网络上可能设置有多个网络/子网号，我们把它按照独立的网络来对待。物理点对点/point-to-point 网络是个例外——无论在上面设置了多少网络/子网号（如果有的话），都将其看作是一个网络。

网络掩码/Network mask:

一个 32 位的数字，表示 IP 地址的范围来说明这是一个 IP 网络/子网/超网。本文以 16 进制来表示网络掩码。如将 C 类 IP 地址的网络掩码显示为 0xfffff00，这一掩码在其他文献中经常被表示为 255.255.255.0。

点对点网络/Point-to-point networks:

仅仅连接一对路由器的网络。56k 的串行线路是一个点对点网络的例子。

广播网络/Broadcast networks:

支持多台（大于两台）路由器接入的网络，同时有能力发送一条信息就能到所有接入的路由器（广播）。网络上邻居路由器可以通过 OSPF 的 Hello 协议来动态发现。如果可能，OSPF 协议将进一步使用多播。广播网络上的每一对路由器都被认为可以直接通讯。以太网/ethernet 是一个广播网络的例子。

非广播网络/Non-broadcast networks:

支持多台（大于两台）路由器接入的网络，但没有广播能力。网络上的邻居路由器通过 OSPF 的 Hello 协议来维持。但由于缺乏广播能力，需要一些配置信息的帮助来发现邻居。在非广播网络上，OSPF 协议的数据通常需要被轮流发送到每一台邻居路由器上。X.25 公用数据网/Public Data Network (PDN) 是一个非广播网络的例子。

在非广播网络上运行的 OSPF 有两种模式。第一种被称为非广播多路接入/non-broadcast

multi-access (NBMA)，模拟 OSPF 在广播网络上的操作；第二种被称为点对多点/Point-to-MultiPoint，将非广播网络看作是一系列点对点的连接。非广播网络被作为 NBMA 网络还是点对多点网络，取决于 OSPF 在该网络上所配置的运行模式。

接口/Interface:

是指路由器与所接入的网络之间的一个连接。接口通过下层协议和路由协议获取与其相关的状态信息。指向网络的接口只和单一的 IP 地址及掩码相关（除非是无编号的点对点网络）。接口有时也被称为连接/link。

邻居路由器/Neighboring routers:

在同一网络中都有接口的两台路由器。邻居关系是由 OSPF 的 Hello 协议来维持，并通常依靠 Hello 协议来动态发现。

邻接/Adjacency:

用以在所选择的邻居路由器之间交换路由信息的关系。不是每对邻居路由器都会成为邻接。

连接状态宣告/Link state advertisement:

描述路由器或网络自身状态的数据单元。对路由器来说，这包含它的接口和邻接状态。每一项连接状态宣告都被洪泛到整个路由域中。所有路由器和网络连接状态宣告的集合形成了协议的连接状态数据库。在本备忘录中，连接状态宣告被缩写为 LSA。

Hello 协议/Hello Protocol:

在 OSPF 协议中，用于建立和维持邻居关系的部分。在广播网络中还被用于动态发现邻居路由器。

洪泛/Flooding:

在 OSPF 协议中，用于 OSPF 路由器之间发送及同步连接状态数据库的部分。

指定路由器/Designated Router:

在每个接入了至少两台路由器的广播和 NBMA 网络中都有一台作为指定路由器。指定路由器生成 Network-LSA 并在运行协议时完成其他特定职责。指定路由器通过 Hello 协议选举。（译注：本文后边将缩写为 DR）

指定路由器的概念减少了广播和 NBMA 网络上所需要的邻接数量。同时也减少了路由协议所需要的流量及连接数据库的大小。

下层协议/Lower-level protocols:

为 IP 及 OSPF 协议提供服务的下层网络接入协议。如为 X.25 PDN 服务的 X.25 packet and frame level；以及为以太网服务的以太网数据链路层。

1.3. 连接状态路由技术的简要历史

OSPF 是一种连接状态路由协议。此类协议在其他文献中也被称为基于 SPF 或基于分布数据库/distributed-database 的协议。本节简要地介绍连接状态技术发展过程中影响 OSPF 协议的部分。

第一种连接状态路由协议是为 ARPANET 网络开发的。该协议在 [引用 3] 中被描述。它成为其他连接状态协议的起点。所谓 ARPANET 网络环境，就是指通过同步串行线路连接的单一厂商的包交换网络，设计简单且只实现基本的协议。

对此协议的修改在 [引用 4] 中被提出。修改包括增加路由协议的容错性，其中之一的手段就是在 LSA 中添加了校验和（用于发现数据库错误）。该文还提及在连接状态协议中要减少路由流量。这通过将产生 LSA 的间隔增加相当长时间来实现。

连接状态算法同时也被 ISO 的 IS-IS 路由协议所使用。该协议在 [引用 2] 中被描述。该协议工作在广播网络上时，通过为每个网络选举 DR，来减少数据及路由流量。由 DR 产生表示该网络的 LSA。

IETF 的 OSPF 工作组长期以来从事 OSPF 协议的开发工作。增强了指定路由器的概念以减少对路由流量的需求；还进一步使用多播能力来减少路由带宽；开发出区域路由而允许信息的隐藏/保护/汇聚；最后，修改算法以专为 TCP/IP 服务。

1.4. 本文档的结构

本文档开始的三章大致介绍了协议的功能，第 4-16 章介绍了协议内容的细节，在附录中列出了包格式、协议常量和可配置变量的清单。

在文档中出现的诸如 HelloInterval 的标注是协议所需要的参数，包括可配置的和不可配置的。结构化常量在附录 B 中汇总，可配置变量在附录 C 中汇总。

为了使描述更为精确，本文档定义了协议所需要的数据结构。对本协议的实现必须支持所描述的功能，但不必使用和本备忘录完全一致的数据结构。

1.5. 感谢

作者感谢 Ran Atkinson、Fred Baker、Jeffrey Burgan、Rob Coltun、Dino Farinacci、Vince Fuller、Phanindra Jujjavarapu、Milo Medin、Tom Pusateri、Kannan Varadhan、Zhaohui Zhang 和其他 OSPF 工作组成员对本项目的建议和支持。

OSPF 点对多点/Point-to-MultiPoint 接口是基于 Fred Baker 的工作。

OSPF 密码验证选项是由 Fred Baker 和 Ran Atkinson 开发的。

另外，本文译者感谢 tigerfish、chenghit、cisco800、4099 和其他 ITPUB 论坛成员的鼓励和支持。

最后还要特别感谢我的女儿，她正是出生在我翻译本文的这段日子里。她让我知道，这世界上除了网络还有很多其他的东西。

2. 连接状态数据库：组织和计算

下面的章节描述 OSPF 连接状态数据库的结构，以及如何从数据库来计算路由器的路由表。

2.1. 路由器和网络的表示方法

在 AS 内的连接状态数据库直接表现为一张图表。图表的行、列用来表示路由器和网络。当两台路由器通过物理点对点网络接入时，图表上的标记就连接这两台路由器。当路由器在某个网络有接口时，图表标记就连接该路由器和这个网络。网络分传输/transit 网络和存根/stub 网络。传输网络可以传送既不是由本网络产生、也不是到达本网络的数据流量。传输网络在图表中有输入、输出两个标记，而存根网络只有输入标记。

网络中各节点在图表上的邻居关系取决于网络的类型（点对点、广播、NBMA 或点对多点），以及连接到该网络的路由器数量。图 1a 描述了三种情况：方框表示路由器，纵横线表示网络。路由器名由字母 RT 开头、网络名由字母 N 开头、路由器的接口名由字母 I 开头。两台路由器之间的直接连线表示点对点网络。左边的图表示网络和路由器的连接，右边的表显示结果。

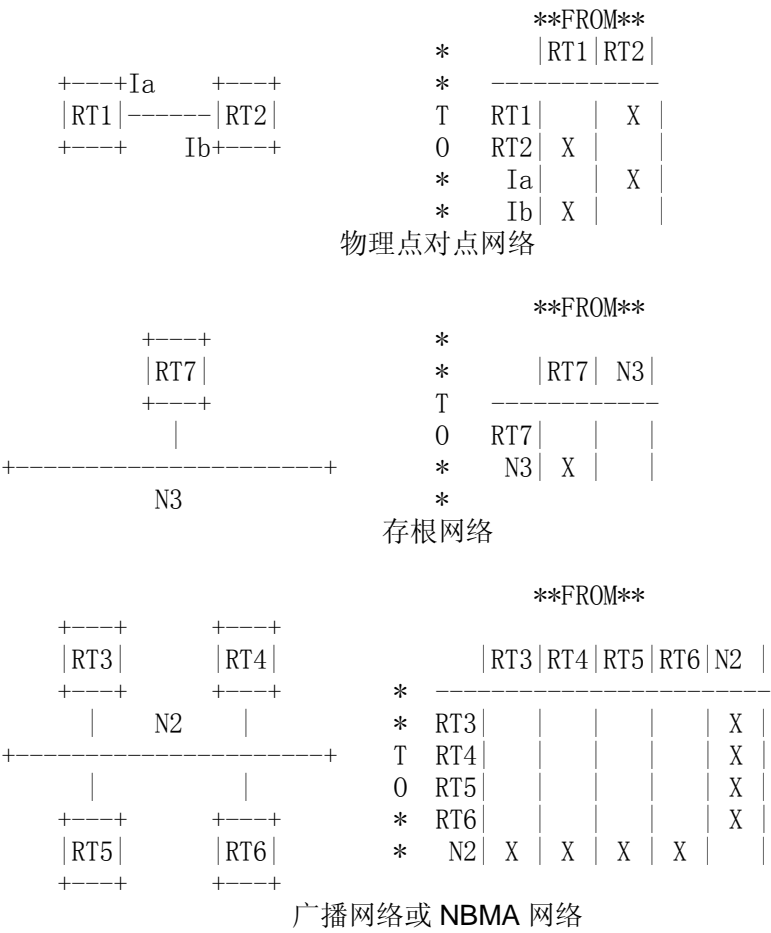


图 1a：网络表的构成

图表的行、列用来表示路由器和网络。当且仅当 A 列 B 行上有“X”记号时，表示 A、B 两项之间有连接。

图 1a 的最上部分显示了两台通过点对点线路连接的路由器。在连接状态数据库表的结果中，两个路由器项通过一对标记相互直接连接，每个标记表示一个方向。点对点接口不必设定 IP 地址。如果设定了 IP 地址，每台路由器就宣告对端路由器的接口地址为一个存根连接。如果在点对点网络上设定了 IP 子网，每台路由器就宣告这个子网为存根连接，而不是宣告对端路由器的接口地址。

图 1a 的中间显示了只接入一台路由器的网络（即存根网络）。这时，连接状态数据库中表现为存根连接后的网络。

当广播网络上接入了多台路由器后，连接状态数据库表现为这些路由器与网络的双向连接。如图 1a 的底部表示。

每个在表中的网络（传输和存根）都有 IP 地址以及相关联的网络掩码，掩码说明了在网络上的节点数量。直接连接到路由器的主机（称为主机路径），表现为存根网络。主机路径的网络掩码始终为 0xffffffff，表示只有一个节点。

2.1.1. 非广播网络的表示方法

正如前面所述,OSPF 在非广播网络上有两种运行方式:NBMA 和点对多点。运行的方式决定了 Hello 协议及洪泛在非广播网络上的工作方法，以及该网络在连接状态数据库中的表示方法。

在 NBMA 模式下，OSPF 模拟在广播网络上的操作，并选举 DR 为该网络来生成 Network-LSA。正如图 1a 中显示的那样，NBMA 网络和广播网络的图表表示是一致的。

不论从连接状态数据库的大小，还是从路由协议所需的流量，NBMA 模式都是在非广播网络上运行 OSPF 最有效的方式。但这需要一个很重要的限制：所有接入 NBMA 网络的路由器都能够直接通讯。这一限制适合某些非广播网络，如使用 SVC 的 ATM 子网；但在另一些非广播网络上就不适合，如仅使用 PVC 的帧中继网络。可以将这种网络分为几个逻辑上的子网，使每个子网中的路由器都能够直接通讯，并在每个子网上按 NBMA 模式运行（见 [引用 15]）。但是这需要更多些的管理，且容易配置错误。在此类网络上使用点对多点模式更好。

在点对多点模式中，OSPF 将非广播网络里，每个路由器之间的连接都看作是点对点的连接。不在网络中选举 DR，也不为该网络产生 Network-LSA。事实上，连接状态数据库的图表中并没有表示点对多点网络的项。

		FROM				
		RT3 RT4 RT5 RT6				
		* -----				
		* RT3				
		T RT4				
		O RT5				
		* RT6				
		* I3				
		I4				
		I5				
		I6				

图 1b: 点对多点网络的网络表构成

除了 RT4 与 RT5 外，其他的路由器之间都可以通过 N2 直接通讯。I3 到 I6 表示路由器的 IP 接口地址。

图 1b 说明了点对多点网络在连接状态数据库中的表现。图的左边一半给出了一个点对多点网络。并假设除了 RT4 与 RT5 外，其他的路由器之间都可以直接通讯。I3 到 I6 表示路由器在该网络上的 IP 接口地址。在图表中，能够通过点对多点网络而直接通讯的路由器使用双向标记连接；此外每台路由器都有一个表示自身 IP 接口地址的存根连接（对照图 1a 中真正点对点连接中的表示）。

在某些非广播网络上，使用点对多点模式和诸如逆向 ARP（见 [引用 14]）的数据链路层协议，可以在没有广播支持的情况下自动发现 OSPF 邻居。

2.1.2. 一个连接状态数据库的示例

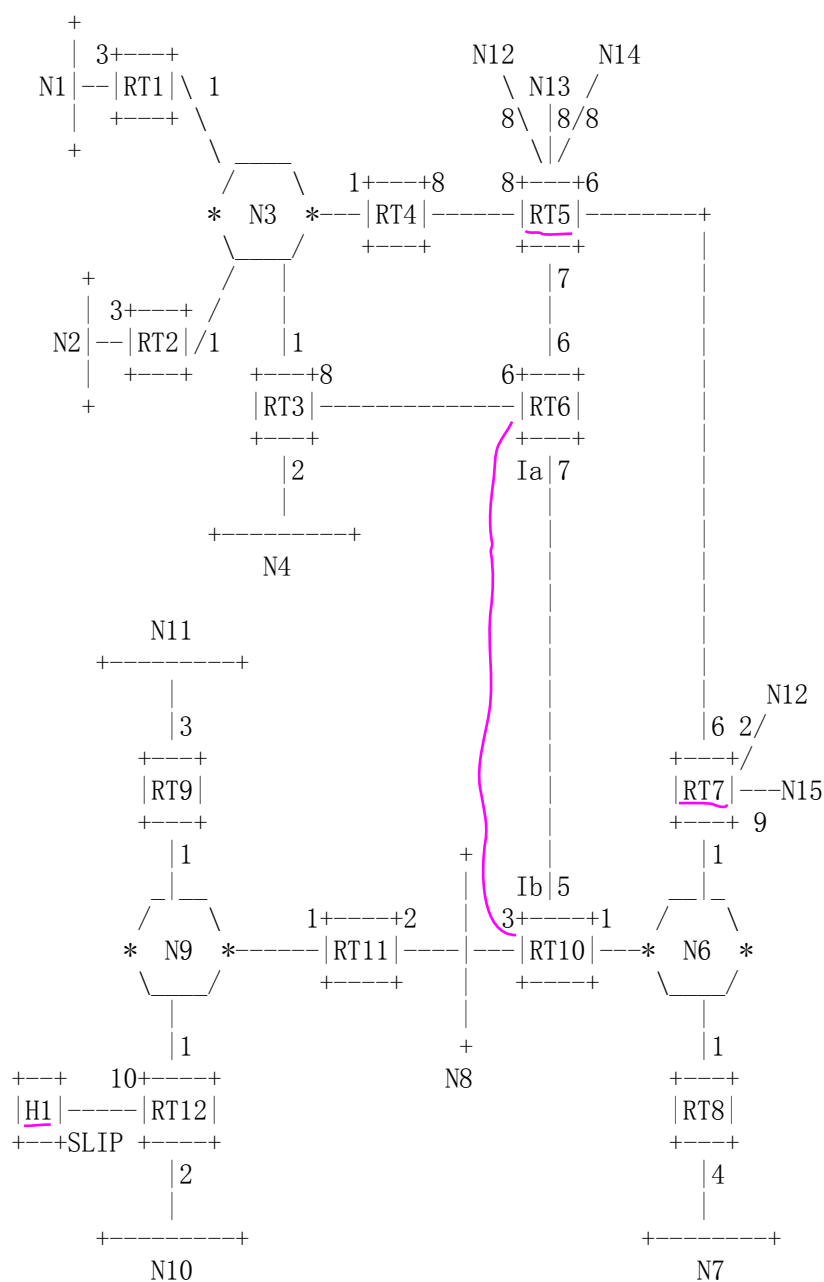


图 2：一个简单的自制系统

图 2 显示了一个简单的 AS 示意图。标有 H1 的方框表示一台主机通过 SLIP 与路由器 RT12 连接，路由器 RT12 因此宣告主机路径。路由器之间的连线表示物理点对点网络，唯一设定了接口地址的点对点网络用以连接路由器 RT6 和 RT10。路由器 RT5 和 RT7 通过 BGP 与其他 AS 连接，这些路由器各自从 BGP 得到一系列的路径。

每个路由器输出接口都有数值与其关联，这些值由系统管理员设定。值越小，该接口就越会被用来转发数据流量。来源于外部的路由数据也有值与其关联（如从 BGP 得到的路径）。

图 2 表示的示意图所产生的结果在图 3 中描述。对应路由器输出接口的数值被填写在连接中。没有标明数值的连接填入 0。注意，从网络到路由器的连接，其数值虽然总是 0，但却有重要意义。还要注意，来源于外部的路由数据，在表中表现为存根。

连接状态数据库是由产生于路由器的 LSA 集合而成。结合图表，每一个单独的 LSA 表示一个相邻的路由器或传输网络。图 4 显示了这些 LSA。路由器 RT12 与两个广播网络各有一个接口连接，还与另一主机通过 SLIP 连接。网络 N9（译注：本小节中的 N9，原文均为 N6，但相信为笔误）是一个广播网络并有 3 台路由器接入，从网络 N9 到接入路由器的数值都是 0。网络 N9 的 LSA 是由接入到该网络上的一台路由器来生成的，该路由器就是在网络上选举出的 DR。

FROM

	RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	N3	N6	N8	N9
RT1													0			
RT2													0			
RT3						6							0			
RT4					8								0			
RT5				8		6	6									
RT6			8		7				5							
RT7					6									0		
* RT8														0		
* RT9																0
T RT10						7							0	0		
O RT11														0	0	
* RT12															0	
* N1	3															
N2		3														
N3	1	1	1	1												
N4			2													
N6							1	1		1						
N7								4								
N8										3	2					
N9									1		1	1				
N10												2				
N11									3							
N12					8		2									
N13					8											
N14					8											
N15							9									
H1												10				

图 3：连接图结果

FROM						**FROM**					
RT12 N9 N10 H1						RT9 RT11 RT12 N9					
*	-----					*	-----				
*	RT12					*	RT9				0
T	N9	1				T	RT11				0
O	N10	2				O	RT12				0
*	H1	10				*	N9				
*						*					

图 4: 单独的连接状态构成

2.2. 最短路径树

当不在 OSPF 中配置区域的时候，AS 中每一台路由器都有着相同的连接状态数据库，并使用相同的图表表示。每台路由器根据图表，构建出一棵以其自身为树根的最短路径树，并生成路由表。显然，最短路径树是由路由器来计算的。例子中路由器 RT6 的最短路径树在图 5 中描述。

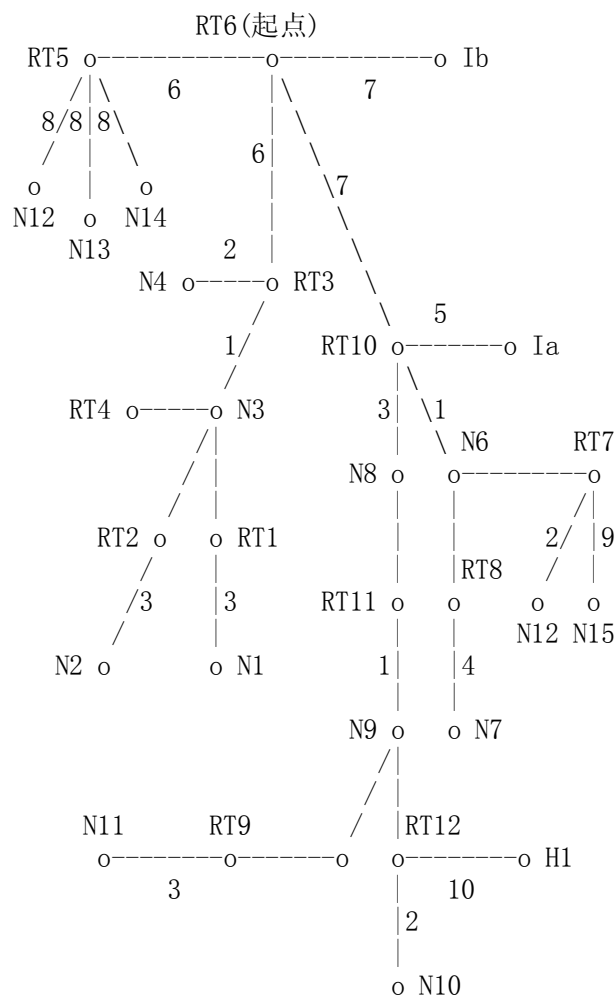


图 5: 路由器 RT6 的最短路径树

没有列出数值的连接标记,其数值为 0 (网络到路由器的连接)。路由器使用外部信息到达网络 N12-N15,这在第 2.3 节中描述。

最短路径树给出了到达任何目标网络或主机的完整路径。虽然在转发过程中，只使用通向目标的下一跳。注意，也同时计算到达任何路由器的最佳路径。为了处理外部数据，我们标注了到达这些宣告外部路由的路由器的下一跳和距离值。路由器 RT6 计算后的路由表在表 2 中描述。注意，各有一条路径到达有编号点对点网络的两端（本例中是路由器 RT6 与 RT10 之间的串行线）。

在图 5 中，从路由器到属于其他 AS 的网络（如 N12）的连接使用虚线（译注：实际图中没有分别）。在下一节中考虑，如何使用外部信息。

目标	下一跳	距离
N1	RT3	10
N2	RT3	10
N3	RT3	7
N4	RT3	8
Ib	*	7
Ia	RT10	12
N6	RT10	8
N7	RT10	12
N8	RT10	10
N9	RT10	11
N10	RT10	13
N11	RT10	14
H1	RT10	21
RT5	RT5	6
RT7	RT10	8

表 2：路由器 RT6 路由表，本地路由部分

2.3. 使用外部路由信息

当产生最短路径树后，开始解释外部路由信息。外部路由信息可以是由其他诸如 BGP 的路由协议所产生的，或是被静态配置的（静态路由）。默认路径也可以被看作是 AS 外部路由信息的一部分。

外部路由信息将不作修改地在整个 AS 中被洪泛。本例中，AS 中的所有路由器都知道 RT7 有两条外部路径，其距离分别为 2 和 9。

OSPF 支持两种类型的外部距离。类型 1 被解释为与 OSPF 接口值（也就是连接状态中的距离值）使用同样的计量单位。类型 2 使用在更大层次上，类型 2 的距离被认为大于任何 AS 内部的路径距离。在使用外部距离类型 2 时假设：在路由一个数据包时，其距离主要用于 AS 之间的路径，并消除了将外部距离转换为连接状态数值的必要。

作为一个处理外部距离类型 1 的例子，假设图 2 中的 RT5 和 RT7 都宣告类型 1 的距离。对于路由器 RT6 来说，到达每一条被宣告的外部路径的距离值，等于所宣告的外部路径距离加上 RT6 到达宣告路由器的距离值。当有两台路由器宣告同一条外部路径时，RT6 选择总值最小的宣告路由器。并将到达该外部路径的下一跳设定为到达所选择宣告路由器的下一跳。

在图 2 中，RT5 和 RT7 都宣告了外部路径：网络 N12。对 RT6 来说，会首选 RT7，因为从 RT7 到达 N12 的距离是 10（8+2），要比 RT5 的 14（6+8）小。表 3 显示了加上外部路由的路由表。

目标	下一跳	距离
N12	RT10	10
N13	RT5	14
N14	RT5	14
N15	RT10	17

表 3: 路由器 RT6 路由表，外部路由部分

处理外部距离类型 2 要简单些。将选择 ASBR 所宣告的较小距离，而不管到达 ASBR 的内部距离。假设例子中的 RT5 和 RT7 宣告类型 2 的距离。所有通往网络 N12 的流量都将由 RT7 转发，因为 $2 < 8$ 。当有多条等值的类型 2 路径存在时，将比较 AS 内部到达宣告路由器的距离来作出选择。

在一个 AS 中，类型 1 和类型 2 的外部距离可以同时存在。这时，类型 1 将始终被优先选择。

本节中假设，通往外部目标的数据包始终通过 ASBR 来转发，而事实不尽如此。例如，假设图 2 中的 N6 上还接入了一台路由器 RTX。RTX 不参与 OSPF 路由，但与 ASBR RT7 交换 BGP 信息。这样 RT7 将无法宣告需通过 RTX 转发的目标为 OSPF 外部路径。这些数据包将首先被转发到 RT7(宣告路由器)，从而增加了额外的一跳。

为了解决这个问题，OSPF 协议允许 ASBR 在 AS-external-LSA 中增加“转发地址/forward address”。在上面的例子中，为使需要的数据包被直接转发到 RTX，RT7 将 RTX 的 IP 地址填入那些路径的“转发地址”中。

“转发地址”还有另一个应用。能够使 AS 内部的路由器作为一台“路由服务器/route server”。例如，图 2 中的 RT6 可以成为路由服务器，通过静态配置和外部路由协议收集外部路由信息。RT6 通过宣告自己为 ASBR，生成一系列 AS-external-LSA。RT6 使用 LSA 中的“转发地址”为目标设定正确的 AS 出口点。

2.4. 等值多路径

上面的描述都简化为到达任何目标时，都只考虑一条路径。事实上，当有多条等值路径到达一个目标时，这些路径都会被发现并使用。这不需要修改算法的概念，在下面讨论树生成过程时，我们会再仔细研究。

当存在等值多路径时，路由器会有到达指定目标的多个可用下一跳。

3. 将自制系统划分为区域

在 OSPF 中允许将一系列连续的网络和主机组合在一起。这样的组合，以及至少有一个接口接入这些网络的路由器，称为区域/area。每个区域独立地运行一套连接状态路由算法。正如在前面所解释的，这意味着每个区域有其自己的连接状态数据库和相应的图表。

区域内的拓扑结构对于区域外来说是不可见的。反过来，在给定区域内的路由器也不知道区域外的拓扑细节。这种隔绝与将整个 AS 作为单一的连接状态域相比，可以使协议极大地减少路由流量。

随着区域的引入，AS 中的路由器不再有完全相同的连接状态数据库。事实上路由器为它所连接的每个区域建立单独的连接状态数据库。（连接多个区域的路由器被称为区域边界路由器（ABR））。两台属于同一区域的路由器，为该区域建立完全相同的区域连接状态数据库。

在 AS 内部的转发有两个层次，取决于数据包的源及目标地址是在同一个区域内（使用区域内路由/intra-area routing），或者不同的区域间（使用区域间路由/inter-area routing）。在使用区域内路由时，数据包仅根据从该区域内得到的信息来转发，而不使用从区域外得到的信息。这可以保护区域内路由，避免错误路由信息。在第 3.2 节中讨论区域间路由。

3.1. 自制系统的骨干区域

OSPF 的骨干区域是 OSPF 的特殊区域 0（由于 OSPF 的区域号是按 IP 地址的格式，所以经常被写为区域 0.0.0.0）。OSPF 骨干区域始终包含所有的 ABR。骨干区域负责发布其他区域之间的路由信息。骨干区域必须是连续的，虽然不一定是物理上连续。骨干区域的连续性可以通过配置虚拟通道/virtual link 来建立和维持。

虚拟通道可以在任何两台拥有同一非骨干区域接口的骨干路由器之间建立。虚拟通道属于骨干区域。协议将通过虚拟通道所连接的两台路由器，视为通过骨干区域内的无编号点对点网络连接。在骨干区域的图表中，这样两台路由器通过数值上等于路由器之间区域内距离的标记连接。在虚拟通道内的路由协议流量只包含区域内路由。

3.2. 区域间路由

当数据包在两个非骨干区域之间转发时，使用骨干区域。数据包所经过的路径可以被分为三段连续的过程：从源到 ABR 的区域内路径；从源区域到目标区域的骨干路径；到达目标的另一个区域内路径。算法用来发现拥有最小距离值的路径。

换一个角度看，区域间路由可以被理解为一个星形配置的 AS，骨干区域是中心，每个非骨干区域是分支。

骨干区域的拓扑结构决定了区域间的骨干路径。骨干区域的拓扑结构可以通过增加虚拟通道来增强。这可以使系统管理员控制区域间流量。

选择恰当的 ABR 作为离开源区域出口的方法，与选择宣告外部路径的路由器的方法完全相同。区域内的每个 ABR 计算到达区域外所有网络的距离。在计算了本区域的最短路径树以后，路由器到达外部目标的距离值等于该数值加上到达 ABR 所需的距离。

3.3. 路由器的分类

在引入区域之前，路由器的唯一特殊功能是宣告外部路由信息，如图 2 中的路由器 RT5。当把 AS 划分为 OSPF 区域以后，根据功能，可以将路由器进一步分为以下功能重叠的四类：

内部路由器/Internal routers (IR)：

路由器所直接连接的网络都属于同一个区域。这些路由器只运行路由算法的一个副本。

区域边界路由器/Area border routers (ABR)：

接入多个区域的路由器。ABR 运行路由算法的多个副本，每个接入的区域一个。ABR 将所接入区域的拓扑信息汇聚后发布到骨干区域。骨干区域再将这些信息发布到其他区域。

骨干路由器/Backbone routers：

有至少一个接口在骨干区域的路由器。这包括所有接入多个区域的路由器（即 ABR）。但骨干路由器不一定是 ABR，所有接口都在骨干区域的路由器也是骨干路由器。

自制系统边界路由器/AS boundary routers (ASBR)：

与属于其他 AS 的路由器交换路由信息的路由器。这样的路由器在 AS 内宣告 AS 外部路由信息。到达各台 ASBR 的路径要被 AS 中的每台路由器所知晓。该分类与前面的分类完全独立：ASBR 可能是内部路由器或 ABR，也可能在或可能不在骨干区域中。

3.4. 一个简单区域配置

图 6 显示了一个简单的区域配置。第一个区域包含网络 N1-N4，以及接入的路由器 RT1-RT4。第二个区域包括网络 N6-N8，以及接入的路由器 RT7、RT8、RT10 和 RT11。第三个区域包括网络 N9-N11 和主机 H1，以及接入的路由器 RT9、RT11 和 RT12。第三个区域中的配置把 N9-N11 和 H1 对外宣告为一条单一路径（其细节见第 3.5 节）。

在图 6 中，路由器 RT1、RT2、RT5、RT6、RT8、RT9 和 RT12 是 IR。路由器 RT3、RT4、RT7、RT10 和 RT11 是 ABR。如前所述路由器 RT5 和 RT7 是 ASBR。

图 7 显示了区域 1 连接状态数据库的结果。此图完整地描述了区域内路由。也显示了内部路由器 RT1 和 RT2 对整个网络的理解。ABR RT3 和 RT4 的工作，就是将所有外部目标到本区域的路径距离，向区域 1 内宣告。这些在图 7 中表示为虚线存根路径（译注：实际图中没有分别）。同时，RT3 和 RT4 必须向区域 1 内宣告 ASBR RT5 和 RT7 的位置。最后，由 RT5 和 RT7 产生的 AS-external-LSA 被洪泛到整个 AS 中，包括区域 1。这些描述了到达网络 N12-N15 路径的 LSA，被包含在区域 1 的数据库中。

路由器 RT3 和 RT4 必须将区域 1 的拓扑结构汇总，并发布到骨干区域中。这些骨干 LSA 在表 4 中显示，这些信息概要的显示了区域 1 中所包含的网络（即网络 N1-N4），以及分别从 RT3 和 RT4 到达这些网络的距离。

骨干区域的连接状态数据库在图 8 中显示。其中显示的路由器是骨干路由器。路由器 RT11 因为属于两个区域而成为骨干路由器。为了使骨干区域连续，在 RT10 与 RT11 之间配置了一条虚拟通道。

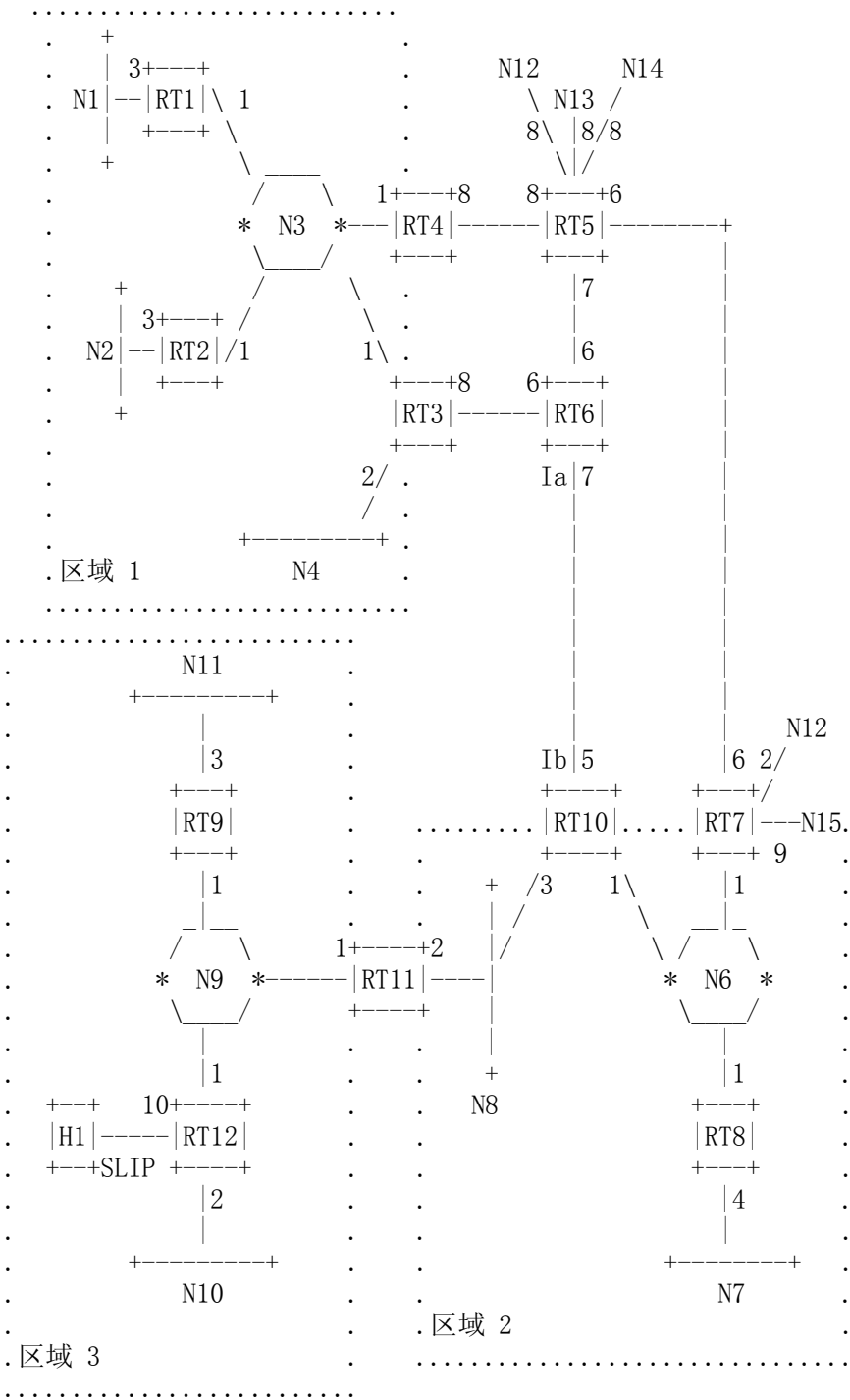


图 6: 一个简单的 OSPF 区域配置

网络	RT3 宣告	RT4 宣告
N1	4	4
N2	4	4
N3	1	1
N4	2	3

表 4: 路由器 RT3 和 RT4 向骨干区域所宣告的网络

FROM

	RT 1	RT 2	RT 3	RT 4	RT 5	RT 7	N3
RT1							0
RT2							0
RT3							0
* RT4							0
* RT5			14	8			
T RT7			20	14			
O N1	3						
* N2		3					
* N3	1	1	1	1			
N4			2				
Ia, Ib			20	27			
N6			16	15			
N7			20	19			
N8			18	18			
N9-N11, H1			29	36			
N12					8	2	
N13					8		
N14					8		
N15						9	

图 7: 区域 1 的数据库

FROM

	RT 3	RT 4	RT 5	RT 6	RT 7	RT 10	RT 11
RT3				6			
RT4			8				
RT5		8		6	6		
RT6	8		7			5	
RT7			6				
* RT10				7			2
* RT11						3	
T N1	4	4					
O N2	4	4					
* N3	1	1					
* N4	2	3					
Ia						5	
Ib				7			
N6					1	1	3
N7					5	5	7
N8					4	3	2
N9-N11, H1							11
N12			8		2		
N13			8				
N14			8				
N15					9		

图 8: 骨干数据库

ABR RT3、RT4、RT7、RT10 和 RT11 将其所接入的非骨干区域的信息汇聚后，通过骨干区域发布；这些在图 8 中表示为虚线存根路径（译注：实际图中没有分别）。其中第三个区域的配置将网络

第20页/总132页

N9-N11 和主机 H1 精简为一条路径。所以网络 N9-N11 和主机 H1 在图 8 中表示为一行。路由器 RT5 和 RT7 是 ASBR，来源于外部的信息在图 8 中表示为存根。

ABR 通过骨干区域交换汇总信息。每个 ABR 从其他 ABR 获得区域汇总。并通过检查所收集到的 LSA 来获取区域外网络距离的描述，并加上到达各自宣告路由器的骨干距离。

再次以路由器 RT3 和 RT4 为例子，处理过程如下：首先计算骨干区域的最短距离优先树。这给出了到达其他所有 ABR 的距离。同时注意到达网络（Ia 和 Ib）和 ASBR（RT5 和 RT7）的距离也属于骨干区域。计算的结果在表 5 中。

然后，通过查看从 ABR 获取的区域汇总，RT3 和 RT4 可以决定到达所有区域外网络的距离。这些距离可以由 RT3 和 RT4 向区域内宣告。RT3 和 RT4 向区域 1 内宣告的信息在表 6 中显示。注意，表 6 假设在骨干区域的配置中，Ia 和 Ib 被组合进一个 LSA 中。

这些由 RT3 和 RT4 输入到区域 1 中的信息，可以使 IR，如 RT1，灵活的选择 ABR。路由器 RT1 将使用 RT4 到达网络 N6、使用 RT3 到达网络 N10、并使用两台路由器作负载均衡来到达网络 N8。

		从 RT3 的距离	从 RT4 的距离
to	RT3	*	21
to	RT4	22	*
to	RT7	20	14
to	RT10	15	22
to	RT11	18	25
to	Ia	20	27
to	Ib	15	22
to	RT5	14	8
to	RT7	20	14

表 5: 路由器 RT3 和 RT4 计算的骨干区域距离

目标	RT3 的宣告	RT4 的宣告
Ia, Ib	20	27
N6	16	15
N7	20	19
N8	18	18
N9-N11, H1	29	36
RT5	14	8
RT7	20	14

表 6: 路由器 RT3 和 RT4 向区域 1 宣告的距离

路由器 RT1 可以同样获取到达 ASBR RT5 和 RT7 的最短路径，并查看 RT5 和 RT7 的 AS-external-LSA。路由器 RT1 可以决定使用 RT5 或 RT7 以到达另一个 AS 中的目标（网络 N12-N15）。

注意，当路由器 RT6 和 RT10 之间的链路失效时，将使骨干区域变为不连续。在路由器 RT7 和 RT10

之间建立一条虚拟通道，可以提高骨干区域的连通性，以进一步抵御此类错误。

3.5. IP 子网化支持

在每条宣告的路径上，OSPF 都有相关的 IP 地址掩码。掩码说明了特定路径所描述的地址范围。如一条指向目标 128.185.0.0 的 Summary-LSA，使用掩码 0xffff0000，就描述了一条到达目标 128.185.0.0 到 128.185.255.255 的路径。同样，在宣告主机路径时，始终使用 0xffffffff 为掩码，表示只存在单一的目标。

在宣告每条路径时包含掩码，可以实现变长子网。这意味着一个 IP 的 A、B 或 C 类网络可以被分为多个不同大小的子网。例如，网络 128.185.0.0 可以被分为 62 个不同大小的子网：15 个大小为 4k 的子网、15 个大小为 256 的子网、和 32 个大小为 8 的子网。表 7 显示了这些地址和子网划分的结果。

网络地址	IP 地址掩码	子网大小
128.185.16.0	0xfffff000	4K
128.185.1.0	0xfffff00	256
128.185.0.8	0xfffffff8	8

表 7：一个子网大小的例子

有很多方法，可以将 A、B 或 C 类网络划分为变长子网。对这些做法的精确描述超出了本规范的范围。本规范所需要的是建立如下的方法：当转发 IP 数据包时，始终转发到与数据包目标地址最佳匹配的网络。这里的最佳匹配与最长匹配同义。例如，默认路径使用目标地址 0.0.0.0 和掩码 0x00000000，始终匹配任何 IP 目标，但也始终比其他匹配要差。子网掩码在匹配任何 IP 目标时不能有任何歧义。

在每条路径中加入地址掩码同样可以支持 IP 超网。例如可以设定地址、掩码对[192.9.4.0, 0xffffc00]到一个物理网络。这样一个 IP 网络可以包含从 192.9.4.0 到 192.9.7.0 四个连续的 C 类网络地址。这样的地址，随着 CIDR（见 [引用 10]）的使用而变得普遍。

为了在区域边界上更好得汇聚，可以使用区域地址范围（细节见附录 C.2）。每个地址范围被定义为 [地址、掩码] 对。就象子网化了的网络包含了很多子网一样，一个地址范围中可以包含很多独立的网络。ABR 通过为每个地址范围宣告一条路径，来汇总区域中的内容（并向骨干区域发布）。汇总路径的距离值为该范围所包含网络距离值中的最大值。

例如：一个子网化的 IP 网络被配置为一个 OSPF 区域。这时，就可以单独配置一个地址范围加上其原始的 IP 掩码。而在区域内，可以定义任何可变大小的子网。通过向区域外发布整个网络的单一路径，而彻底隐藏了划分子网的情况。这条路径的距离值等于这些子网中，一系列路径距离的最大值。

3.6. 支持存根区域

在某些 AS 中，连接状态数据库中主要包含的是 AS-external-LSA。一般情况下，OSPF 将 AS-external-LSA 洪泛到整个 AS 中。但 OSPF 允许将一些区域配置为“存根区域/stub area”，AS-external-LSA 将不被洪泛进这些区域。默认时，将以区域为单位，向 AS 外的目标转发数据。这减少了存根区域内部路由器中连接状态数据库的大小，以及所需要的内存。

为了利用 OSPF 存根区域的优点，在存根区域内必须使用默认路径，并按如下说明来完成。存根区域中的一台或多台 ABR 通过 Summary-LSA 向存根区域内宣告一条默认路径。这条汇总的默认路径仅被洪泛进存根区域，而不是更远（这样作的理由是，这条默认路径只属于特定的存根区域）。当所到达的目标既不在区域内路径，也不在区域间路径的时候（即 AS 外部目标），使用这条汇总的默认路径。

当一个区域内只有单一出口，或者不需要按每条外部路径来选择离开区域的出口时，可以将其配置为存根区域。例如，图 6 中的区域 3 可以被配置为存根区域，因为所有的外部流量都必须通过唯一的 ABR RT11。这时，RT11 将向区域 3 内部宣告一条默认路径（通过一个 Summary-LSA），而不是将网络 N12-N15 的 AS-external-LSA 洪泛进区域。

OSPF 协议确信同一区域内的路由器，对于是否把该区域配置为存根区域达成一致。这将保证在洪泛 AS-external-LSA 时，不出现混淆。

使用存根区域也有些限制。在存根区域中不能配置有虚拟通道；ASBR 也不能存在于存根区域中。

3.7. 区域的划分

OSPF 并不试图修复分隔的区域。当一个区域被分隔后，每个部分就成为一个独立的区域，这两部分间的数据转发将通过骨干区域。原来的一些区域内路径将会变为区域间路径。

为了保证区域被分隔后，还能保持完整的路由，同一个地址范围就不能出现在区域的多个部分中；同样骨干区域不能被分隔。不然，AS 中的一部分网络就会变为不可到达。骨干区域的分隔可以通过配置虚拟通道来修复（见第 15 章）。

通过查看第 2 章的 AS 图表，可以换一个角度来理解区域分隔。区域号被视作图表中连接标记 [脚注 1] 的颜色。图表中的每个标记用来连接网络，或代表点对点网络。标记的颜色表示网络的区域号。

一组同样颜色的标记，以及其相连接的节点表示为一个区域。如果 AS 的拓扑结构完整，图表中就会有多种颜色的部分，每个不同的区域号用一种颜色表示。

当 AS 的拓扑改变后，一个区域被分隔。AS 图表中就会有多个部分拥有同样的颜色（同样的区域号）。只要这些部分连接到同一个骨干区域上，AS 仍将正常工作。

4. 功能摘要

OSPF 算法为每个区域运行独立的副本。在多个区域拥有接口的路由器运行算法的多个副本。下面描述路由算法的简明摘要。

路由器启动时，首先初始化路由协议的数据结构。然后等待下层协议提示接口可以开始工作。

然后路由器使用 OSPF 的 Hello 协议来探知邻居。路由器向其邻居发送 Hello 包，并等待接收邻居发送的 Hello 包。在广播和点对点网络上，路由器通过向多播地址 `AllSPFRouters` 发送自己的 Hello 包来动态的探知邻居路由器。在非广播网络上，需要一些其他的配置信息来发现邻居。在广播和 NBMA 网络上，Hello 协议还被用来选择网络中的 DR。

路由器试图与新探知的邻居建立邻接/adjacencies 关系。在邻接路由器之间，连接状态数据库是同步的。在广播和 NBMA 网络上，哪些路由器需要形成邻接是由 DR 来决定的。

邻接控制路由信息地发布，只有在邻接路由器之间才会发送、接收路由更新。

路由器周期性的宣告其状态：即连接状态。在路由器状态改变时也会宣告连接状态。在其 LSA 中会包含相邻接的路由器。邻接和连接状态的关系，使协议能够通过时间周期来探知失效的路由器。

LSA 在整个区域中被洪泛。洪泛算法是可靠的，以确保区域中所有的路由器拥有完全相同的连接状态数据库。数据库中包含了区域中每个由路由器生成的 LSA。从这个数据库，每台路由器构建出以自己为树根的最短路径树。再由最短路径树计算出协议的路由表。

4.1. 区域间路由

前一节描述了协议在单一区域中的操作。对于区域内路由，不再需要其他信息。为了能够到达区域外的目标，ABR 需要把额外的路由信息向区域内发布。这些额外的信息是 AS 中其他部分拓扑结构的汇总。

汇聚过程按如下步骤完成：按定义每台 ABR 都是连接到骨干区域的。每台 ABR 将其非骨干区域内的拓扑结构汇总后，发往骨干区域，从而到达其他所有的 ABR。这样 ABR 就有了骨干区域的完整拓扑信息和从其他 ABR 获得的区域汇总，并从这些信息计算出到达所有区域间目标的路径，再将这些路径向区域内宣告。这使区域内部的路由器可以选择最佳的出口路由器来转发区域间的流量。

4.2. 自制系统外部路由

了解其他 AS 信息的路由器，可以将这些信息洪泛进整个 AS 中。这些外部路由信息被逐个发布到每台参与的路由器。一个例外是：外部路由信息不被洪泛进存根区域（见第 3.6 节）。

为了使用外部路由信息，整个 AS（除了存根区域内）中的路由器，都需要知道到达宣告外部信息路由器的路径。因此，ASBR 的位置要由非存根的 ABR 来汇总。

4.3. 路由协议包

OSPF 协议直接基于 IP，使用 IP 协议号 89。OSPF 没有明确的提供分片/组装功能。当需要分片时，直接使用 IP 分片/组装。在 OSPF 协议中，按照设计较大的数据包可以被分割为较小的数据包，并推荐使用这种方法；应当尽可能地避免 IP 分片。

路由协议包应当始终将 IP TOS 域中设定为 0。如果可能，在发送和接收的时候，路由协议包的优先级应当高于正常的 IP 数据。为实现这一点，OSPF 协议包的 IP 优先级域应当被设置为 Internet Control（见 [引用 5]）。

所有的 OSPF 协议包使用相同的协议头部，协议头部在附录 A 中描述。OSPF 包的类型在表 8 中列出，其格式同样在附录 A 中描述。

类型	数据包名称	协议功能
1	Hello	发现/维持邻居关系
2	数据库描述/Database Description (DD)	数据库内容汇总
3	连接状态请求/Link State Request (LSR)	数据库下载
4	连接状态更新/Link State Update (LSU)	数据库更新
5	连接状态确认/Link State Ack (LSAck)	洪泛确认

表 8: OSPF 数据包类型

OSPF 的 Hello 协议使用 Hello 包来发现和维持邻居关系。DD 包和 LSR 包用来形成邻接。OSPF 的可靠更新机制是由 LSU 包和 LSAck 包来实现的。

每个 LSU 包中包括一系列新的 LSA，并将其发送到距离其起源更远的一跳。一个 LSU 包中可以包括有多个路由器产生的 LSA。每个 LSA 由其生成路由器的标识来识别，并对其内容生成校验和。每个 LSA 同样有一个类型域；不同类型的 LSA 在下面的表 9 中列出。

LSA 类型	LSA 名称	LSA 描述
1	Router-LSA	由所有路由器产生。该 LSA 表述了在路由器在一个区域内所有接口的状态。仅在一个区域内洪泛。
2	Network-LSA	由广播或 NBMA 网络中的指定路由器产生。该 LSA 包含了接入该网络的路由器列表。仅在一个区域内洪泛。
3, 4	Summary-LSA	由 ABR 产生，并在与该 LSA 相关联的区域内洪泛。每条 Summary-LSA 描述了一条在区域外，但在 AS 内的路径（区域间路径）。类型 3 描述到达网络的路径。类型 4 描述到达 ASBR 的路径。
5	AS-external-LSA	由 ASBR 产生，并在整个 AS 内洪泛。每条 AS-external-LSA 描述了另一 AS 中的一条路径。AS 的默认路径也可被描述为 AS-external-LSA。

表 9: OSPF 连接状态宣告（LSA）

OSPF 路由协议包（除了 Hello 包外）都只在邻接路由器之间传送。这意味着除了虚拟邻接外，所有的 OSPF 协议包都只在 IP 网络中传播一跳。OSPF 协议包的源地址是邻接一端的路由器 IP 地址，目标地址是邻接另一端的 IP 地址或 IP 多播地址。

4.4. 基本实现的需求

要实现 OSPF，需要系统支持以下一些内容。

记时器/Timers:

需要两种不同的记时器。第一种称为“单击记时器/single shot timers”，激活一次，并导致执行协议事件。第二种称为“间隔记时器/interval timers”，按持续的时间间隔激活。这常用于周期性的发送数据包，例如周期性的广播 Hello 包。这两种记时器的记时精度都为 1 秒。

单击记时器应当避免偏离/drift。在一些路由器上，处理数据包会影响记时器工作。当有多个使用广播的路由器接入同一网络时，会导致路由包的同步（这应当被避免）。如果不能保证避免偏离，每次激活记时器时，应当加上或减去一小段随机时间。

IP 多播/IP multicast:

一些 OSPF 包以 IP 多播的形式发送。所以需要通过适当的下层协议，来支持收发 IP 多播。OSPF 传送的 IP 多播包决不超过一跳。因此，不需要转发 IP 多播的能力。关于 IP 多播的信息，见 [引用 7]。

变长子网支持/Variable-length subnet support:

路由器的 IP 协议必须有能力将一个 A、B 或 C 类的 IP 网络划分为多个不同大小子网。这常被称为变长子网，细节见第 3.5 节。

IP 超网支持/IP supernetting support:

路由器的 IP 协议必须有能力将一系列连续的 A、B 或 C 类的 IP 网络汇聚为一个超网。超网被提出用来改善 Internet 上的 IP 路由。关于 IP 超网的更多细节见 [引用 10]。

下层协议支持/Lower-level protocol support:

这里说的下层协议是指网络接入协议，如以太网数据链路层。这些协议必须向 OSPF 发出信号，表示网络接口的通断。例如，知道以太网接口被拔下是个很重要的信号。

非广播下层协议支持/Non-broadcast lower-level protocol support:

在非广播网络中，当 OSPF 向一个失效或不存在的路由器发送包时，可以由 Hello 协议来提供标识。例如在 X.25 PDN 中，由于适当的原因或诊断会收到 X.25 clear，以表示邻居失效，这样的信息应当被传送到 OSPF。

列表处理简化/List manipulation primitives:

很多 OSPF 功能，是通过对 LSA 列表的处理来实现的。例如一系列 LSA 将被重传到邻接路由器，直至接收到确认。任一特定的 LSA 可能出现在多个列表中。实现 OSPF 需要维持这些列表，按照需要增删特定的 LSA。

任务支持/Tasking support

在本规范中的特定过程要调用其他的过程时，其他的过程要被及时执行，即在特定过程结束前被执行，这通过编写调用过程的指令来实现。而有的时候，其他过程要等特定过程结束后才能执行，这通过编写任务调度的指令来实现。

4.5. OSPF 可选项

OSPF 协议定义了一些可选项。路由器在 OSPF Hello 包、DD 包和自生成的 LSA 中标识出所支持的可选项。这使支持不同可选项的路由器能够在同一个 AS 中共存。

有些选项必须被特定区域内所有的路由器所支持。这时，如果邻居的 Hello 包中没有报告这一项，路由器将不接受（即不匹配的选项将不能形成邻居关系）。如下面所说的 ExternalRoutingCapability。

其他选项可以在数据库交换过程中协商。这通过在 DD 包中指定可选项来实现。如果邻居间的选项不匹配，将导致在这两个邻居间，只交换一部分的连接状态数据库。

是否存在可选项同样影响路由表的建立。由于可以在 LSA 中报告可选项，所以在生成最短路径树时，就不会选择不支持特定功能的路由器。

本备忘录中定义的 OSPF 可选项在后面被列出。细节见附录 A.2。

ExternalRoutingCapability

整个 OSPF 区域可以被配置成为“存根”（见第 3.6 节）。不会将 AS-external-LSA 洪泛进存根区域。这在 OSPF 可选项中表现为 E 位（见附录 A.2）。为确保存根区域的配置一致，区域中所有的路由器必须在 Hello 包中清除 E 位（见第 9.5 和 10.5 节）。

5. 协议数据结构

在此描述 OSPF 协议操作时所需要的多种协议数据结构。下面的列表包含了顶层的 OSPF 数据结构。并说明所有必须的初始化。与 OSPF 的区域、接口和邻居相关的数据结构在本规范的后面描述。

路由器标识/Router ID:

一个 32 位数，在 AS 中唯一的识别出一台路由器。一个可行的策略是使用该路由器接口上设定的最小 IP 地址。当路由器的 OSPF 路由器标识改变时，需要重新启动 OSPF 软件使新的路由器标识生效。在重新启动前，路由器应当将其自身产生的 LSA 从路由域中废止（见第 14.1 节），不然它们会一直存在至多 MaxAge 分钟。

区域结构/Area structures:

路由器为其接入的每个区域建立自己的数据结构。该数据结构在讲述基本 OSPF 算法时描述。每个区域各自运行基本 OSPF 算法的独立副本。

骨干区域结构/Backbone (area) structure:

OSPF 骨干区域负责发布区域间路由信息。

虚拟通道配置/Virtual links configured:

在路由器上配置虚拟通道，使其成为一个端点。为了配置虚拟通道，路由器自身必须成为 ABR。虚拟通道由另一端点（另一 ABR）的路由器标识来识别。这两台路由器必须接入同一个区域，被称为虚拟通道传输区域。虚拟通道是骨干区域的一部分，就象在两台路由器之间有一个无编号点对点网络。虚拟通道使用传输区域的区域内路径来转发数据包。通过构建传输区域的最短路径树控制虚拟通道的通断。

外部路由器列表/List of external routes:

可以到达 AS 外部目标的路由器。可能是通过与其他路由协议（如 BGP）的直接交互，或通过配置信息，或是两者的组合（例如 OSPF 使用配置了的距离来宣告外部动态信息）。任何拥有外部信息的路由器都被称为 ASBR。外部路径由这些路由器通过 AS-external-LSA 向 OSPF 路由域内部宣告。

AS-external-LSA 列表/List of AS-external-LSAs:

连接状态数据库的一部分，是由 ASBR 所生成。包含了到达 AS 外部目标的路径。注意，如果路由器本身为 ASBR 的话，其中一部分的 AS-external-LSA 是由其自身生成的。

路由表/The routing table:

来源于连接状态数据库。路由表中的每一项以目标为索引，包含了到达目标的距离值和用于转发到达该目标数据包的一系列路径。路径是按类型和下一跳来描述的。具体的细节见第 11 章。

图 9 显示了一台典型路由器所包含的数据结构。图中显示的路由器是图 6 中的 RT10。注意，路由器 RT10 和 RT11 之间配置了一条虚拟通道，使用区域 2 作为传输区域，这在图 9 中表现为星号。当建立了区域 2 的最短路径树后，虚拟通道被开通，从而成为了骨干区域中的一个接口（在图 9 中，有两个接入骨干区域的接口）。

+----+

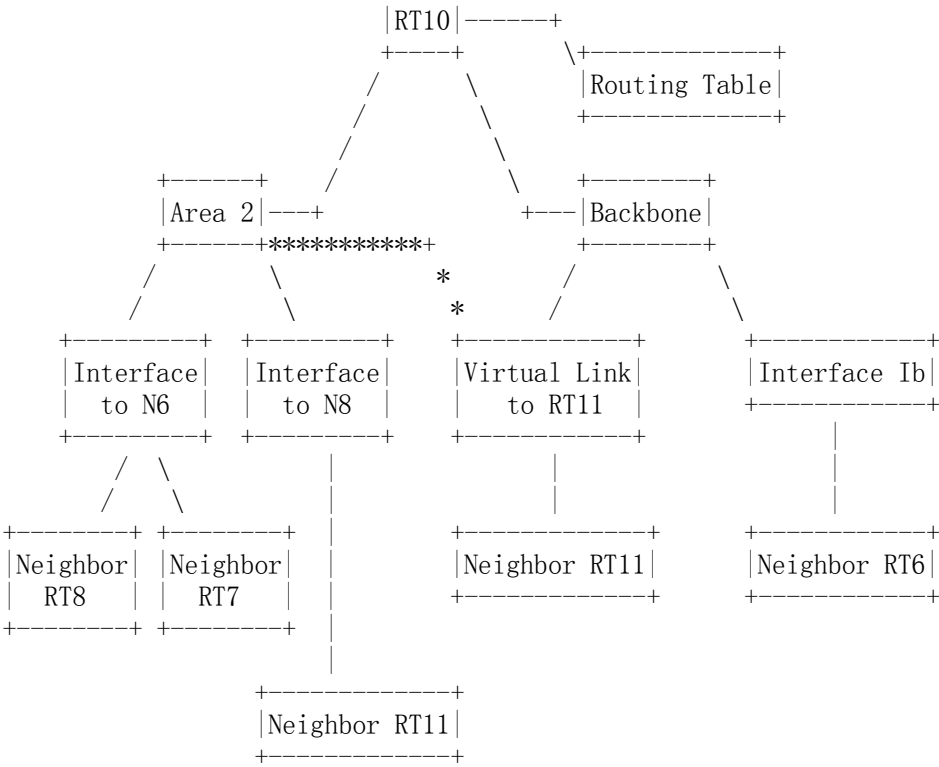


图 9：路由器 RT10 的数据结构

6. 区域数据结构

所有用于基本 OSPF 路由算法的信息都包含在区域数据结构中。每个区域维持着自己的连接状态数据库。每个网络只属于一个区域，每个路由器接口只连接一个区域，每对邻接路由器也属于同一区域。

OSPF 骨干区域是一个特殊的 OSPF 区域，负责发布区域间路由信息。

区域的连接状态数据库包含了由本区域内路由器生成的一系列 Router-LSA、Network-LSA 和 Summary-LSA。这些信息只在单一的区域洪泛。AS-external-LSA 列表（见第 5 章）也被认为是每个区域连接状态数据库的一部分。

区域标识/Area ID:

一个 32 位数以识别区域。区域标识 0.0.0.0 被保留用来表示骨干区域。

区域地址范围列表/List of area address ranges:

为了在区域边界汇总路由信息，需要使用区域地址范围。每个地址范围由一个 [地址、掩码] 对和一个表示是否宣告的状态组成。

相关路由器接口/Associated router interfaces:

路由器连接到区域的接口。每个路由器接口仅属于一个区域（包括骨干区域）。骨干区域的相关接口列表包含所有的虚拟通道。虚拟通道是由另一端的路由器标识来识别的，其距离值是这两台路由器在传输区域内的最短区域内路径。

Router-LSA 列表/List of Router-LSAs:

Router-LSA 是由区域中的每台路由器生成。它描述了该路由器在本区域内各接口的状态。

Network-LSA 列表/List of Network-LSAs:

Network-LSA 是为区域内每个广播和 NBMA 网络生成的。它描述了当前连接在该网络上的路由器集合。

Summary-LSA 列表/List of Summary-LSAs:

Summary-LSA 是由该区域的 ABR 生成。它描述了一条在本区域外，但在 AS 内的路径（即区域间路径）。

最短路径树/Shortest-path tree:

以路由器自身为树根的区域最短路径树。以收集到的 Router-LSA 和 Network-LSA，通过 Dijkstra 算法（见第 16.1 节）而构建得出。

TransitCapability:

此参数表示该区域是否能够传输既不是由本区域产生、也不是到达本区域的数据流量。该参数在生成区域最短路径树时使用（见第 16.1 节，当且仅当有一条或多条完全邻接的虚拟通道使用本区域作为传输区域时，TransitCapability 被设为 TRUE），还在生成路由表时作为输入参数（见第 16.3 节）。当一个区域的 TransitCapability 被设为 TRUE 时，该区域也被称为“传输区域/transit area”。

ExternalRoutingCapability:

表示是否将 **AS-external-LSA** 洪泛进本区域。这是一个可配置的变量。如果 **AS-external-LSA** 被拒绝在区域外，该区域就是“存根”。在存根区域中，仅根据默认的汇总路径来判断到达 **AS** 外部的目标。骨干区域不能被配置为存根区域，同时在存根区域中不能配置虚拟通道。更多信息见第 3.6 节。

StubDefaultCost:

如果一台路由器被配置为存根区域的 **ABR**，那么 **StubDefaultCost** 就说明了该路由器向区域内宣告的默认 **Summary-LSA** 的距离值。更多细节见第 12.4.3 节。

除非特别说明，本文的其他部分都是说明 **OSPF** 协议在一个区域内的操作。

7. 形成邻接

OSPF 在邻居路由器之间形成邻接以交换路由信息。不是所有的邻居路由器之间都形成邻接。本章包含了产生邻接的一般性规则。更多细节参见第 10 章。

7.1. Hello 协议

Hello 协议负责建立和维持邻居关系，它同时确保邻居间的通讯是双向的。Hello 包从路由器的所有接口周期性地发出。双向通讯是指路由器发现其邻居的 Hello 包中包含了自己。在广播和 NBMA 网络上，Hello 协议被用来为网络选举 DR。

在广播网络、NBMA 网络和点对多点网络上，Hello 协议的工作不同。在广播网络上，每台路由器通过周期性地发送多播 Hello 包来宣告自身，这使得邻居能被动态探知。在这些 Hello 包中包含了自己所认为的 DR 和最近收到过 Hello 包的路由器列表。

在 NBMA 网络上，需要一些配置信息来使 Hello 协议运行。每台可能成为 DR 的路由器，都有接入该网络其他路由器的完整列表。每台可能成为 DR 的路由器，在第一次可以操作 NBMA 网络接口时，向其他可能成为 DR 的路由器发送 Hello 包，以试图在网络上找到 DR。如果该路由器被选举为 DR，它开始向接入该网络的其他所有路由器发送 Hello 包。

在点对多点网络上，路由器向所有能与它直接通讯的邻居发送 Hello 包。这些邻居可能使用诸如逆向 ARP（见 [引用 14]）的协议来动态探知，或通过静态配置而知。

当通过双向通讯发现邻居，且（在广播或 NBMA 网络上）选举了 DR 后，就要判断邻居间是否需要形成邻接（见第 10.4 节）。如果形成邻接，那么第一步就是同步邻居间的连接状态数据库。这在下一节阐述。

7.2. 数据库同步

对于连接状态路由算法，十分重要的一点是：在所有路由器之间保持连接状态数据库的同步。OSPF 将其简化为保持邻接路由器之间的同步。同步过程在路由器之间试图形成邻接时开始。每台路由器通过向其邻居发送一系列 DD 包，来描述自身的数据库。每个 DD 包中包含了路由器数据库中的一组 LSA。当邻居发现其中的 LSA 比自己数据库中的要新时，就记录下需要请求的 LSA。

这种发送和接收 DD 包的过程被称为“数据库交换过程/Database Exchange Process”。在这个过程中，两台路由器形成主从关系。每个 DD 包都有一个序列号。主机发送描述包后，从机通过回应描述包的序列号来确认。在发送和回应包中都包含了连接状态数据的摘要。只有主机能够按照固定的时间间隔重传 DD 包，其长度是按接口配置的常量 RxmtInterval。

每个 DD 包都通过一个标识——M（更多）位，来表示后面是否还有更多的包。当路由器收到并发出清除了 M 位的 DD 包后，表示数据库交换过程结束。

在数据库交换过程期间及完成后，每台路由器都有一个需要从邻居请求更新的 LSA 列表。在 LSR 包中发出对这些 LSA 的请求。没有被确认的 LSR 包，按固定间隔 RxmtInterval 重发。当数据库描述过程结束，并且所有的连接状态请求都得到确认时，数据库即被认为已经同步，邻居路由器也被

标记为完全邻接。这时，可以实现邻接的所有功能，并在两台路由器的 Router-LSA 中宣告邻接。

从数据库交换过程开始，邻接就被用于洪泛过程。这简化了数据库同步，并保证可以在预计的时间内结束。

7.3. 指定路由器

在每个广播和 NBMA 网络上都有一台指定路由器 (DR)。DR 在路由协议中实现两项主要功能。

DR 为该网络生成 **Network-LSA**。这个 LSA 中列出了当前接入该网络的路由器列表 (包括 DR 自身)。这个 LSA 的 LS 标识 (见第 12.1.4 节) 是 DR 的接口 IP 地址，可以通过网络的掩码获得网络的 IP 地址。

DR 与网络上的其他所有路由器邻接。由于连接状态数据库是通过邻接来同步的 (先通过邻接形成，再通过洪泛过程)，DR 在同步过程中起着重要作用。

DR 是通过 Hello 协议来选举的。每个路由器接口都可以配置路由器优先级/Router Priority，并被包含在路由器的 Hello 包中。通常，当路由器的网络接口开始工作时，它首先检查网络上是否已经存在 DR。如果有，它便接受这一 DR，而忽略路由器优先级 (这使得难以预知 DR 的身份，但确保 DR 很少变化。见下文)。否则，网络上拥有最高优先级的路由器成为 DR。更多更精确关于 DR 选举的细节在第 9.4 节介绍。

DR 是很多邻接的端点。为了优化广播网络上的洪泛过程，DR 使用多播地址 AllSPFRouters 发送 LSU 包，而不是向每个邻接分别发送。

在本文的第 2 章中讨论了区域的图表表示。路由器节点以其路由器标识来识别，传输网络节点事实上是由 DR 的 IP 接口地址来识别。当 DR 改变时，图表上的网络节点就会变为一个全新的节点。这会导致该网络 and 所有接入的路由器生成新的 LSA。在连接状态数据库再次收敛前，会导致暂时失去网络的连通，会使数据流量产生 ICMP 不可到达响应。为此，DR 应当较少更改，应当通过配置路由器优先级，使较可靠的路由器最终成为 DR。

7.4. 备份指定路由器

为了能够平滑地转换到新的 DR，在每个广播和 NBMA 网络上都有一台备份指定路由器/Backup Designated Router (BDR)。BDR 同样与网络上所有的路由器邻接，并在上一台 DR 失效时成为 DR。如果不存在 BDR，那么当新的 DR 出现后，就要在新的 DR 和其他路由器之间建立新的邻接。形成邻接中的一步是同步连接状态数据库，而这可能会持续一段时间。在这段时间里，网络不能传输数据流量。BDR 与其他路由器之间已经存在了邻接，而不需要再次形成。这使传输中断的时间缩短为洪泛新的 LSA (宣告新的 DR) 所需要的时间。

BDR 不为网络生成 Network-LSA。(如果生成的话，可以更快地转换为 DR。这实际上是在数据库大小与 DR 失效后的收敛速度之间的一种折中)。

BDR 同样通过 Hello 协议中选举。每个 Hello 包中都列出了网络的 BDR。

在某些洪泛过程中，BDR 处于被动地位，而让 DR 完成主要工作。这减少了本地的路由流量。细节见第 13.3 节。

7.5. 邻接图

邻接存在于两台接入同样网络的路由器之间。如果两台路由器有多个同样的网络，那在这两台路由器之间就有多对邻接。

可以通过形成一个图表来描述网络上的一系列邻接。行、列为路由器，在邻接的路由器之间标上标记。邻接图描述了在 AS 内路由协议包（具体而言是 LSU 包）的流向。

可能出现两种图表，取决于是否选举出 DR。在物理点对点网络、点对多点网络和虚拟通道上，当相互可以通讯就会形成邻接。相对应的，在广播和 NBMA 网络上，只有 DR 和 BDR 与网络上其他所有的路由器形成邻接。

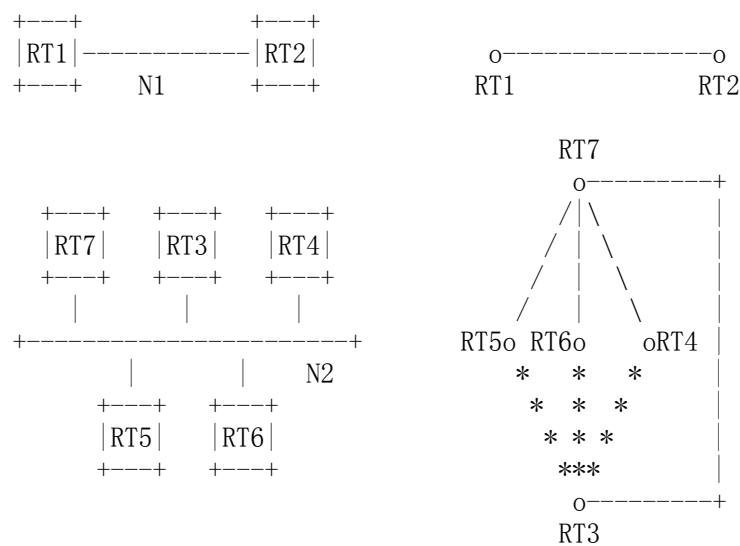


图 10：邻接图

图 10 显示了这些图表。这里假设网络 N2 中，路由器 RT7 是 DR、而路由器 RT3 是 BDR。BDR 与 DR 相比，在洪泛过程中完成更少的工作（见第 13.3 节）。所以在图中其他路由器与 RT3 的连接表现为星号。

8. 协议包处理

本章讨论处理 OSPF 路由协议包的一般过程。因为必须持续地保持路由器之间连接状态数据库的同步，所以不论收发，路由协议数据包都应当比普通的数据包优先处理。

路由协议数据包仅在邻接间发送（除了用来探知邻居的 Hello 包），这意味着除了虚拟通道外，所有的路由协议包都只传送一跳。

所有的路由协议包开始于相同的头部。本章的后面提供了如何填充和校验标准头部的细节。此外本章给出了特定包类型的处理细节。

8.1. 发送协议包

当路由器发送路由协议包时，按如下要求填充 OSPF 包的标准头部。头部的格式细节见附录 A.3.1。

版本号/Version #:

设为 2，为本规范所说明的协议版本。

包类型/Packet type:

OSPF 包的类型，如 LSU 包或 Hello 包等。

包长度/Packet length:

整个 OSPF 包的字节长度，包括标准的 OSPF 头部。

路由器标识/Router ID:

表示路由器自身（包的起源）。

区域标识/Area ID:

包将被发送到的区域。

校验和/Checksum:

除了 64 位的验证域外，整个 OSPF 包的标准 16 位 1 补足校验和。校验和也被作为验证正确性的一部分，在某些 OSPF 验证类型下，将忽略校验和的计算。细节见附录 D.4。

验证类型和验证域/AuType and Authentication:

每个 OSPF 包的交换都被验证。协议所设定的验证类型在附录 D 中描述。每个 IP 网络/子网可以使用不同的验证类型。AuType 说明了验证所选用的类型。64 位的验证域被所选择的验证类型所使用。在生成包的最后调用验证过程。细节见附录 D.4。

数据包的 IP 目标地址按如下选择：在物理点对点网络上，IP 目标地址始终被设为 AllSPFRouters。在其他网络上（包括虚拟通道），主要的 OSPF 包使用单播/unicast 发送，也就是直接发往邻接的对端。这时 IP 目标地址就被设置为相关联的对端邻接的 IP 地址（见第 10 章）。唯一不接单播发送的是在广播网络上：这时 Hello 包被发往多播地址 AllSPFRouters；DR 和 BDR 向多播地址 AllSPFRouters 发送 LSU 包和 LSAck 包；其他的路由器向多播地址 AllDRouters 发送 LSU 包和 LSAck 包。

重传的 LSU 包“始终”被直接发往邻居。在多路接入/multi-access 网络上，这意味着重传应当发送到邻居的 IP 地址。

数据包的 IP 源地址应当被设定为发送接口的 IP 地址。对于没有关联 IP 地址的无编号点对点网络，应当将 IP 源地址设定为属于该路由器的任意其他 IP 地址。因此，在路由器上至少需要设定一个 IP 地址 [脚注 2]。注意，在绝大多数情况下，虚拟通道表现为无编号点对点网络。而事实上，每个虚拟通道确实有一个 IP 接口地址（在路由表建立过程中取得），并被用于虚拟通道中发送包的 IP 源地址。

更多特定 OSPF 包的格式细节，见表 10 中列出的章节。

类型	协议数据包名称	发送细节所在章节
1	Hello	第 9.5 节
2	数据库描述（DD）	第 10.8 节
3	连接状态请求（LSR）	第 10.9 节
4	连接状态更新（LSU）	第 13.3 节
5	连接状态确认（LSAck）	第 13.5 节

表 10：描述 OSPF 协议数据包发送的章节

8.2. 接收协议包

当接收到协议包后，路由器标记出是从哪个接口上被接收。当在路由器上配置有虚拟通道时，可能不能立即判断出协议包所关联的接口。例如图 6 中的路由器 RT11，从连接到网络 N8 的接口收到 OSPF 协议包，可能是与区域 2 关联的接口，也可能是与 RT10 关联的虚拟通道（是骨干区域的一部分）。下面，我们假设数据包不是与虚拟通道相关联。[脚注 3]

为了使数据包能够被 IP 层所接受，在将包传送给 OSPF 处理前，必须经过一些检测。

IP 校验和必须正确。

数据包的目标地址必须是接收接口的 IP 地址，或是多播地址 AllSPFRouters 与 AllDRouters。

IP 协议域必须标明为 OSPF（89）。

自身生成的数据包不应当被传送到 OSPF。也就是说，必须检查 IP 源地址，以确保该多播包不是由自身生成的。

其次，校验 OSPF 包头。包头中的各域必须与接收接口上的配置相匹配。不然，丢弃该数据包。

版本号域必须标明为版本 2。

必须验证 OSPF 包头中的区域标识。如果不满足下列两个条件之一，就丢弃该数据包。

- （1）与接收接口的区域标识相匹配。在这种情况下，数据包仅被传送过一跳。即数据

包的源地址与接收接口在同一网络上。这通过将源地址和接收接口的地址，同地址掩码进行与操作的结果比较而得出。这一比较不应当在点对点网络上进行，在点对点网络上，两端的接口地址被独立设置，甚至根本没有设置。

(2)标识为骨干区域。这时，数据包是从虚拟通道传送而来的。接收路由器必须是 ABR，并且包中的路由器标识（源路由器）必须是所配置虚拟通道的另一端。接收接口也必须接入该虚拟通道所配置的传输区域。如果这个数据包通过了所有这些检查就被接受，并同虚拟通道相关联（同时与骨干区域相关联）。

IP 目标地址为 AllDRouters 的数据包，仅能被接收接口为 DR 或 Backup 状态的路由器接收。

数据包中说明的 AuType 必须与关联区域说明的 AuType 匹配。

数据包必须被验证。由 AuType 决定验证类型（见附录 D）。每个接口可以配置一个或多个验证密码，以用于验证过程。验证过程也可能包括 OSPF 包头中的校验和（使用标准 16 位 1 补足校验和，验证除了 64 位的验证域外的整个 OSPF 包）。如果验证失败，该数据包被丢弃。

如果是 Hello 包，需要由 Hello 协议（见第 10.5 节）作进一步处理。其他的包仅在邻接间收发，这意味着数据包必定是由路由器的一个活跃邻居所发出的。当接收接口在广播网络、点对多点网络或 NBMA 网络时，可以在数据包的 IP 头部找到发送者的 IP 源地址。当接收接口在点对点网络或虚拟通道时，可以在数据包的 OSPF 头部找到发送者的路由器标识。接收接口的数据结构中有活跃邻居的列表，不能匹配该列表的包被丢弃。

因此，所有接收到的协议数据包都和一个活跃的邻居相关联。特定协议包的输入细节，见表 11 所列的章节。

类型	协议数据包名称	接收细节所在章节
1	Hello	第 10.5 节
2	数据库描述 (DD)	第 10.6 节
3	连接状态请求 (LSR)	第 10.7 节
4	连接状态更新 (LSU)	第 13 章
5	连接状态确认 (LSAck)	第 13.7 节

表 10: 描述 OSPF 协议数据包接收的章节

9. 接口数据结构

OSPF 接口用以连接路由器和网络。我们假设每个 OSPF 接口接入各自的网络/子网。多个接口接入同一网络的情况在附录 F 中考虑。每个接口数据结构中有至多一个 IP 接口地址。

OSPF 接口可以被认为属于包含其所接入网络的区域。所有由这台路由器生成并从该接口发出的协议数据包，都被标记上接口所在的区域标识。在路由器的接口上可能产生一个或多个邻接。由路由器发出的 LSA 反应出其接口的状态及其相关联的邻接。

下列的数据项是与一个接口相关联的。注意，其中一些项实际上配置的是所接入的网络，所有接入该网络的路由器必须配置一致。

类型/Type:

OSPF 接口的类型可能是点对点、广播、NBMA、点对多点或虚拟通道。

状态/State:

接口的功能层次。状态决定了该接口上能否建立完全邻接。状态也反应在路由器的 LSA 中。

接口 IP 地址/IP interface address:

与该接口关联的 IP 地址，出现在所有由该接口发出的路由协议数据包的 IP 源地址中。无编号点对点网络没有相关联的 IP 地址。

接口 IP 掩码/IP interface mask:

也被称为子网掩码，表示出接口 IP 地址中的哪些部分说明了所接入的网络。掩码和地址一起，说明了所接入网络的网络号。在点对点网络和虚拟通道上，没有定义接口 IP 掩码。在这些网络上，可能没有设定 IP 网络号；即使被设定，两端的设定也是相互独立的。

区域标识/Area ID:

接入网络所属于的区域。所有从该接口发出的协议数据包，都被标记上该区域标识。

HelloInterval:

路由器从该接口发送 Hello 包的时间间隔，以秒计。该接口发出的 Hello 包中宣告该数值。

RouterDeadInterval:

当不再收到路由器的 Hello 包后，过多少秒以后宣告邻居断开。该接口发出的 Hello 包中宣告该数值。

InfTransDelay:

在该接口上发送一个 LSU 包所需要的大致时间，按秒数评估。包含 LSA 的 LSU 包，在发送前要在其时限/age 上增加这一数值。这一数值应当被计算进发送和传播延迟，并必须大于 0。

路由器优先级/Router Priority:

一个 8 位无符号整数。当接入同一网络的两台路由器都试图成为 DR 的时候，优先选择具有较高优先级的路由器。路由器优先级为 0 的路由器不能成为 DR。该接口发出的 Hello 包中宣告该数值。

Hello 记时器/Hello Timer:

引起接口发送 Hello 包的间隔记时器。该记时器每过 HelloInterval 秒被激活。注意，在非广播网络上，需要向每个邻居发送单独的 Hello 包。

Wait 记时器/Wait Timer

使接口退出等待/Waiting 状态的单击记时器，并作为网络上选择 DR 的结果。其长度为 RouterDeadInterval 秒。

邻居路由器列表/List of neighboring routers:

接入该网络的其他路由器。这一列表由 Hello 协议形成。会与其中的一些邻居形成邻接。通过检查所有邻居的状态可以决定邻接路由器的集合。

指定路由器/Designated Router:

为其接入的网络所选举的 DR。在所有的广播和 NBMA 网络上，通过 Hello 协议选举出 DR。通过其路由器标识和接口 IP 地址两项，在网络上识别。DR 宣告该网络的连接状态，在这个 Network-LSA 中，标识出 DR 的 IP 地址。初始的 DR 值为 0.0.0.0，表示不存在。

备份指定路由器/Backup Designated Router:

在广播和 NBMA 网络上，同样通过 Hello 协议选举 BDR。接入该网络的其他路由器与 DR 和 BDR 相邻接。当前 DR 失效时，BDR 就成为 DR。初始的 BDR 值为 0.0.0.0，表示不存在。

接口输出值/Interface output cost(s):

从接口发送数据的输出值，在连接状态距离中表示。这一数值在 Router-LSA 中宣告为该接口的连接状态距离值。该数值必须大于 0。

RxmtInterval:

向该接口的邻接重传 LSA 所间隔的秒数。也在重传 DD 包和 LSR 包时使用。

AuType:

接入网络/子网所使用的验证类型。验证类型在附录 D 中定义。每个 OSPF 包的交换都被验证。每个 IP 网络/子网可以使用不同的验证方法。

验证密码/Authentication key:

这一数据被验证过程用于生成和检验 OSPF 协议包。验证密码可以按接口配置。例如，当 AuType 表示简单口令验证时，验证密码就是被插入 OSPF 包头中的 64 位明文。而当 AuType 表示密码验证时，验证密码就成为消息摘要，并追加到 OSPF 包后。当使用密码验证时，可支持多个密码，以使密码平滑过度（见附录 D.3 节）。

9.1. 接口状态

在本节中说明路由器接口可能达到的多种状态。状态按处理过程排序。例如，不能工作的状态排在最前面，经过一系列中间状态达到完全工作。在本文中会按此顺序，使用诸如“该接口高于某状态”的说法。图 11 显示了接口状态的转变。图中标记的事件引起状态的改变，这些事件在第 9.2 节中描述。接口状态机的细节描述见第 9.3 节。

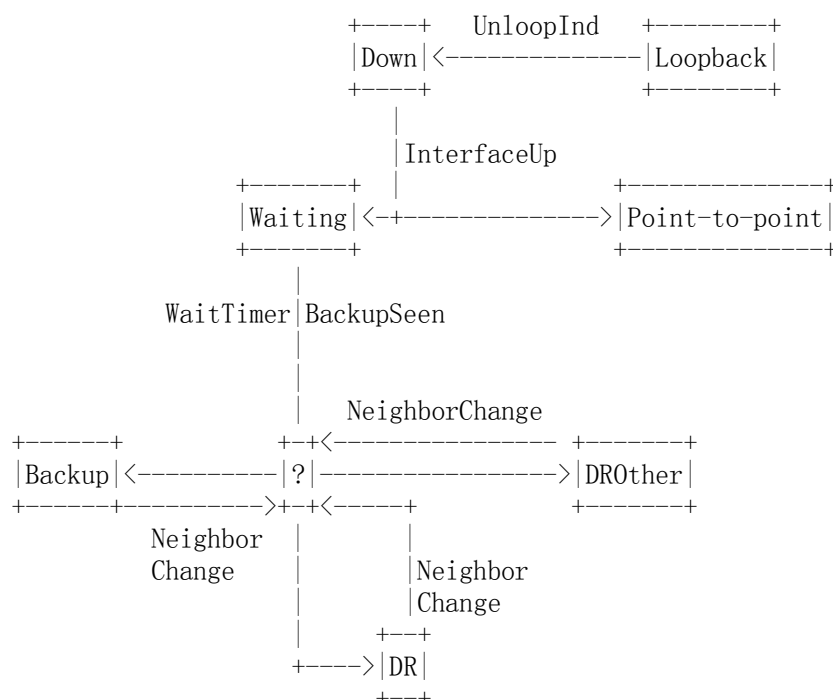


图 11：接口状态改变

除了图上标记的事件，事件 **InterfaceDown** 始终导致 **Down** 状态，事件 **LoopInd** 始终导致 **Loopback** 状态。

Down:

这是接口的初始状态，这时下层协议指出接口为断开，接口上没有协议流量的收发。这时，接口上的所有参数都被设为初始值，关闭所有的接口计时器，该接口上也没有相关联的邻接。

Loopback:

这时，路由器到网络的接口处于回环/Loopback，回环可能以硬件或软件的方式实现。回环接口不能用于正常的数据传输，但仍能通过 **ICMP ping** 或位错误检测来收集接口信息。**IP** 包仍需要被发往回环接口。为此，要在 **Router-LSA** 中宣告此接口为接口 **IP** 地址的主机路径 [脚注 4]。

Waiting:

在此状态时，路由器试图判定网络上 **DR** 和 **BDR**。为此，路由器对其接收到的 **Hello** 包进行监听。在结束等待前，路由器不能被选举为 **DR** 或 **BDR**。这可以避免不必要地改变 **DR** 和 **BDR**。

Point-to-point:

这时，连接到物理点对点网络或虚拟通道的接口开始工作。进入此状态之后，路由器试图与邻居路由器形成邻接。并按 **HelloInterval** 的间隔发送 **Hello** 包。

DR Other:

广播或 **NBMA** 网络上的其他路由器被选举为 **DR**，其自身也没有被选为 **BDR**。路由器开始与 **DR** 和 **BDR**（如果存在的话）形成邻接。

Backup:

在此状态时，路由器是所接入网络的 **BDR**。并将在当前的 **DR** 失效时成为 **DR**。该路由器与接入该网络的所有其他路由器形成邻接。在洪泛过程中，**BDR** 的工作与 **DR** 稍有不同（见第 13.3 节）。关于 **BDR** 的工作细节见第 7.4 节。

DR:

在此状态时，路由器是所接入网络的 **DR**。该路由器与接入该网络的所有其他路由器形成邻接。路由器必须为网络节点生成一个 **Network-LSA**。该 **Network-LSA** 包含了所有接入该网络的路由器（包括 **DR** 自身）。关于 **DR** 的工作细节见第 7.3 节。

9.2. 引起接口状态改变的事件

一系列事件可能引起接口状态的改变。这些事件在图 11 中被标注，并在下面定义。这些事件所引起的 **OSPF** 协议操作参见第 9.3 节。

InterfaceUp:

下层协议指出网络接口开通。这使接口转出 **Down** 状态。对虚拟通道而言，接口开通是最短路径计算的结果。

WaitTimer:

等待计时器到期，表示在选举 **DR** 和 **BDR** 前的等待状态已经结束。

BackupSeen:

路由器已经探知在网络上是否存在 **BDR**。有两个方法可以得到。第一种：收到邻居的 **Hello** 包，并标明自己是 **BDR**；第二种：收到邻居的 **Hello** 包，标明自己是 **DR**，但没有 **BDR**。在这两种情况下，都必须与邻居实现双向通讯，也就是，路由器必须出现在邻居的 **Hello** 包中。这一事件结束等待状态。

NeighborChange:

该接口上双向通讯的邻居发生了改变。**DR** 和 **BDR** 需要重新计算。邻居的以下改变产生 **NeighborChange** 事件。邻居状态的解释见第 10.1 节。

与一个邻居建立了双向通讯。换句话说，该邻居状态达到 **2-Way** 或更高。

与一个邻居不再存在双向通讯。换句话说，该邻居状态成为 **Init** 或更低。

一个双向通讯的邻居最近宣告自己为 **DR** 或 **BDR**。这通过查看邻居 **Hello** 包而探知。

一个双向通讯的邻居不再宣告自己为 **DR** 或 **BDR**。同样通过查看邻居 **Hello** 包而探知。

一个双向通讯的邻居宣告路由器优先级改变。同样通过查看邻居 **Hello** 包而探知。

LoopInd:

收到指令，表示接口现在为 **Loopback**。指令可能是从网管，或下层协议而来。

UnloopInd:

收到指令，表示接口现在不再为 **Loopback**。同 **LoopInd** 一样，指令可能是从网管，或下层协议而来。

InterfaceDown:

下层协议指出接口断开。不管当前是什么接口状态，新的接口状态都将为关闭。

9.3. 接口状态机

下面描述接口状态改变的细节。每次变化都是由事件所调用（第 9.2 节）。根据当前的接口状态，同样的事件可能产生不同的影响。因此，状态机是根据当前接口状态和事件来组织的。每一项都描述了新的接口状态和所需要的一系列附加操作。

当接口状态改变后，需要生成一个新的 **Router-LSA**。更多细节见第 12.4 节。

下面的一些操作会引起改变邻居状态机的事件。例如，当接口变为不可操作时，所有与该接口关联的邻居都要被清除。邻居状态机的更多细节见第 10.3 节。

状态: **Down**

事件: **InterfaceUp**

新状态: 取决于操作过程

操作: 启动 **Hello** 计时器，开始从接口上周期性地发送 **Hello** 包。如果接入的是物理点对点网络、点对多点网络或虚拟通道，接口状态转换为 **Point-to-Point**。否则，如果路由器不能成为 **DR**，接口状态转换为 **DR Other**。

如果可能成为 **DR** 的路由器接入了广播或 **NBMA** 网络，其接口状态成为 **Waiting**，并开始 **Wait** 计时器，以试图发现所接入网络中的 **DR**。此外，在 **NBMA** 网络上，对于接口上邻居列表中可能成为 **DR** 的路由器发出 **Start** 邻居事件。

状态: **Waiting**

事件: **BackupSeen**

新状态: 取决于操作过程

操作: 按第 9.4 节的介绍，选举出所接入网络的 **BDR** 和 **DR**。按照选举的结果，接口的状态可能成为 **DR Other**、**Backup** 或 **DR**。

状态: **Waiting**

事件: **WaitTimer**

新状态: 取决于操作过程

操作: 按第 9.4 节的介绍，选举出所接入网络的 **BDR** 和 **DR**。按照选举的结果，接口的状态可能成为 **DR Other**、**Backup** 或 **DR**。

状态: **DR Other**、**Backup** 或 **DR**

事件: **NeighborChange**

新状态: 取决于操作过程

操作: 按第 9.4 节的介绍，重新选举出所接入网络的 **BDR** 和 **DR**。按照选举的结果，接口的状态可

能成为 DR Other、Backup 或 DR。

状态：任意状态

事件：InterfaceDown

新状态：Down

操作：重置接口上的所有变量，关闭接口计时器。通过生成 KillNbr 邻居事件，来清除所有关联的邻居连接（见第 10.2 节）。

状态：任意状态

事件：LoopInd

新状态：Loopback

操作：由于接口不再接入网络，执行与上面 InterfaceDown 事件同样的操作。

状态：Loopback

事件：UnloopInd

新状态：Down

操作：不需要执行任何操作。诸如重置接口变量等操作都在进入 Loopback 时完成了。注意，在接口再次可以工作前，需要接收到 InterfaceUp 事件。

9.4. 选举指定路由器

本节描述用于选举网络上 DR 和 BDR 的算法。该算法被接口状态机所调用。网络上的路由器第一次运行选举算法的时候，把网络上的 DR 和 BDR 都初始为 0.0.0.0，这说明 DR 和 BDR 都不存在。

DR 的选举算法按如下进行：将执行计算的路由器称为 X。并检查该网络上已经建立了双向通讯的路由器列表。这个列表，与路由器 X 至少达到 2-Way 状态（见第 10.1 节）的邻居（在该网络上）正好相同，路由器 X 自己也被认为在列表上。从列表上排除那些不能成为 DR 的路由器（路由器优先级为 0 的路由器，不能成为 DR）。根据剩余的列表，执行下面的步骤：

（1）记录网络上当前的 DR 和 BDR 数值，以用于以后的比较。

（2）按如下计算网络上的 BDR：列表上的路由器中，只有那些不宣告自己为 DR 的才可能成为 BDR。如果有一台或多台这样的路由器宣告自己为 BDR（也就是说，在其 Hello 包中将自己列为 BDR，而不是 DR），选择其中拥有最高路由器优先级的成为 BDR；如果相同，选择拥有最大路由器标识的。如果没有路由器宣告自己为 BDR，选择列表中路由器拥有最高优先级的成为 BDR，（同样排除宣告自己为 DR 的路由器），如果相同，再根据路由器标识。

（3）按如下计算网络上的 DR。如果有一台或多台路由器宣告自己为 DR（也就是说，在其 Hello 包中将自己列为 DR），选择其中拥有最高路由器优先级的成为 DR；如果相同，选择拥有最大路由器标识的。如果没有路由器宣告自己为 DR，将新选举出的 BDR 设定为 DR。

（4）如果路由器 X 新近成为 DR 或 BDR，或者不再成为 DR 或 BDR，重复步骤 2 和 3，然后执行 5。假设路由器 X 新近成为 DR，在重复第 2 步时就不再参与 BDR 选举。这样做是为了确保路由器不会同时宣告自己为 DR 和 BDR。[脚注 5]

(5) 作为选举结果，路由器可能成为 DR 或 BDR。在第 7.3 和 7.4 节中描述了更多的职责。如果路由器成为 DR，那么新的接口状态就是 DR；如果路由器成 BDR，那么新的接口状态就是 Backup；否则新的接口状态就是 DR Other。

(6) 如果路由器接入 NBMA 网络，并成为 DR 或 BDR，它必须向不能成为 DR 的路由器发送 Hello 包（见第 9.5.1 节）。这通过向路由器优先级为 0 的邻居调用 Start 邻居事件来完成。

(7) 如果上面的选举导致了 DR 或 BDR 的改变，就会改变关联接口上的邻接。有些邻接需要建立，而有些需要拆除。为此，对所有达到至少 2-Way 状态的邻居调用事件 AdjOK?。这会导致重新检查邻接（见第 10.3 和 10.4 节）。

选举算法如此复杂的原因是：为了在当前 DR 失效时，让 BDR 顺利转变为 DR。通过一定的滞后来确保顺利转换：在老的 BDR 接受了新的 DR 职责之前，不会选择新的 BDR。

上面的过程可能选举一台路由器同时为 DR 和 BDR，虽然这不会是进行计算的路由器（路由器 X）自身。被选择的 DR 不一定是拥有最高优先级的路由器；被选择的 BDR 也不一定是第二高优先级的路由器。如果路由器 X 自身不能成为 DR，按上面的过程可能选举不出 BDR 和 DR。注意：如果路由器 X 是网络上唯一可能成为 DR 的路由器，它将选择自己为 DR，而没有 BDR。

9.5. 发送 Hello 包

在每个工作的路由器接口上都发送 Hello 包。它们用于发现和维持邻居关系 [脚注 6]。在广播和 NBMA 网络上，Hello 包同样被用于 DR 和 BDR 的选举。

Hello 包的格式细节在附录 A.3.2 中说明。Hello 包中包含有路由器优先级（用于选择 DR）；接口上发送 Hello 包的时间间隔（HelloInterval）；多长时间内必须收到邻居消息（RouterDeadInterval）。接入同一网段的所有路由器，其 HelloInterval 和 RouterDeadInterval 必须相同。Hello 包中还包含了所接入网络的 IP 地址掩码。在无编号点对点网络和虚拟通道上，该域应当被设为 0.0.0.0。

在 Hello 包的 Options 域中描述了路由器的 OSPF 可选项。本规范定义了其中的一个选项（见第 4.5 节和附录 A.2）。当且仅当所接入的区域能够处理 AS-external-LSA 时，才在可选项中设置 E 位（不是存根区域）。如果邻居路由器的 Hello 包中包含了不匹配的 E 位，就会被拒绝接收（见第 10.5 节）。Hello 包中 Options 域里不能识别的位，应当被设定为 0。

为了确保邻接路由器之间的双向通讯，发出的 Hello 包中还包含最近接收到 Hello 包的路由器列表。此外还包括路由器当前所选举的 DR 和 BDR。在没有选举前，该域被设为 0.0.0.0。

在广播网络和物理点对点网络上，Hello 包以 HelloInterval 秒的间隔向 IP 多播地址 AllSPFRouters 发送。在虚拟通道上，Hello 包以 HelloInterval 秒的间隔以单播发送（其地址直接指向虚拟通道的另一端）。在点对多点网络上，在 HelloInterval 秒内，向各个邻居发送独立的 Hello 包。在 NBMA 网络上的 Hello 包发送见下一节。

9.5.1. 在 NBMA 网络上发送 Hello 包

在非广播网络上实现 Hello 协议需要静态配置信息（见附录 C.5 和附录 C.6）。在 NBMA 网络上，每

台能够成为 DR 的路由器，都需要知道网络上的所有其他路由器（不论是通过配置，或其他未说明的机制）。还要标识出每台邻居路由器是否有能力成为 DR。

接口状态必须达到 **Waiting** 才能从 NBMA 接口上发出 **Hello** 包。**Hello** 包直接以单播的方式发往路由器的邻居。**Hello** 包可能按计时器周期性发送，或作为接收到 **Hello** 包的回应。路由器发送 **Hello** 包的行为，取决于它是否有能力成为 DR。

如果路由器有能力成为 DR，它必须周期性的向其他有能力成为 DR 的邻居发送 **Hello** 包。此外，如果路由器自身成为了 DR 或 BDR，它必须周期性的向所有其他邻居发送 **Hello** 包。这意味着，所有有能力成为 DR 的路由器之间始终交换 **Hello** 包，以正确执行 DR 选择算法。为了减少 **Hello** 包的发送，应当控制 NBMA 网络上有能力成为 DR 的路由器数量。

如果路由器不能成为 DR，它必须周期性的向 DR 和 BDR 发送 **Hello** 包（如果存在的话）。也必须在接收到其他有能力成为 DR（不是当前的 DR 和 BDR）的 **Hello** 包后，回应 **Hello** 包。这用于与其他潜在的 DR 建立起双向关系。

向邻居发送 **Hello** 包的时间间隔取决于邻居的状态。如果邻居状态是 **Down**，每隔 **PollInterval** 秒发送 **Hello** 包；其他状态时，每隔 **HelloInterval** 秒发送 **Hello** 包。

10. 邻居数据结构

每台 OSPF 路由器都与其邻居产生会话，会话由“邻居数据结构/neighbor data structure”所描述。每个会话被限定在特定的路由器接口上，并由邻居的 OSPF 路由器标识、或邻居 IP 地址（见下）所识别。因此，如果两台路由器共同接入多个相同的网络，就会引起多个会话，并由各自的邻居数据结构所描述。每个独立的会话，在本文档中被通称为“邻居”。

邻居数据结构中，包含了两台路由器之间形成邻接所需要的所有相关信息（虽然，并不是所有的邻居都会成为邻接）。邻接可以被看作是两台路由器之间更高层次的会话。

状态/State:

这标识出邻居间会话的层次。更多的细节见第 10.1 节。

非活跃计时器/Inactivity Timer:

一个单击计时器，在最近没有接收到 Hello 包后被激活。其长度为 RouterDeadInterval 秒。

主从/Master、Slave:

当两个邻居交换数据库时，形成主从关系。主机发送第一个 DD 包，并且只能由其重发。从机仅能响应主机的 DD 包，主从关系在 ExStart 状态下协商。

数据库描述序号/DD Sequence Number (DD 序号):

当前被发往邻居的 DD 包序号。

最近接收到的数据库描述包/Last received Database Description packet:

从邻居最近接收到的 DD 包中的 DD 序号，以及初始 (I)、更多 (M) 和主从 (MS) 位。用于判断从邻居接收到的下一个 DD 包是否重复。

邻居标识/Neighbor ID:

邻居的 OSPF 路由器标识。邻居标识是通过所接收到的邻居 Hello 包而取得，在虚拟邻接中是通过配置而取得（见附录 C.4）。

邻居优先级/Neighbor Priority:

邻居的路由器优先级。包含在邻居的 Hello 包中，为所接入的网络选举 DR 时使用。

邻居 IP 地址/Neighbor IP address:

邻居路由器接入该网络的接口 IP 地址。在向邻接发送单播的路由协议包时，将其作为目标 IP 地址。如果该邻居路由器被选举为 DR，也被 Router-LSA 用做网络的连接标识/Link ID（见第 12.4.1 节）。邻居 IP 地址通过所接收到的邻居 Hello 包而取得，在虚拟通道中是通过路由表计算而取得（见第 15 章）。

邻居选项/Neighbor Options:

邻居的 OSPF 可选项，在数据库交换过程中取得（见第 10.6 节）。邻居的 OSPF 可选项也在 Hello 包中被列出。当某个重要的 OSPF 选项不匹配时（见第 10.5 节），可以拒绝接收 Hello 包（也就是说开始形成邻居关系）。OSPF 可选项的说明在第 4.5 节。

邻居的指定路由器/Neighbor's Designated Router:

邻居所认为的 DR。如果 DR 为邻居自身时,就是计算网络中 DR 的重要因素。仅在广播和 NBMA 网络上被定义。

邻居的备份指定路由器/Neighbor's Backup Designated Router:

邻居所认为的 BDR。如果 BDR 为邻居自身时,就是计算网络中 BDR 的重要因素。仅在广播和 NBMA 网络上被定义。

下面的变量是 LSA 列表。这些列表描述了部分区域连接状态数据库。本备忘录定义了 5 种不同类型的 LSA,都出现在区域连接数据库中: Router-LSA、Network-LSA 和类型 3、4 的 Summary-LSA (都存储在区域数据结构中)以及 AS-external-LSA (存储在全局数据结构中)。

连接状态重传列表/Link state retransmission list:

已经被洪泛,但还没有从邻接得到确认的 LSA 列表。将按间隔重发直至确认,或邻接消失。

数据库汇总列表/Database summary list:

区域连接状态数据库中 LSA 的完整列表。在邻居进入数据库交换状态时,以 DD 包的形式向邻居发送此列表。

连接状态请求列表/Link state request list:

需要从邻居接收,以同步两者之间连接状态数据库的 LSA 列表。该列表在接收到 DD 包时创建,并向邻居发送 LSR 包,在收到适当的 LSU 包后被清除。

10.1. 邻居状态

本节描述邻居的状态(实质上是与邻居路由器之间对话的状态),状态按处理过程排序。例如,不能工作的状态排在最前面,经过一系列中间状态达到完全工作。在本文中会按此顺序,使用诸如“该邻居/邻接之间达到某状态”的说法。图 12 和图 13 显示了邻居状态的转变。图中标记的事件引起状态的改变,这些邻居事件在第 10.2 节中描述。

图 12 显示了由于 Hello 协议而引起的状态改变。Hello 协议用于邻居的建立和维持,并保证邻居间的双向通讯。

图 13 显示了邻接的形成。并不是所有的两台邻居之间都形成邻接(见第 10.4 节)。邻接状态从 ExStart 状态开始;当两台路由器确定了主从关系后,转变为 Exchange 关系;这时在洪泛过程中使用邻居状态,并开始同步邻居间的数据库;当同步完成,邻居状态转变为 Full,就达到完全邻接,并将该邻接在 LSA 中列出。

关于邻居状态转变的更多细节,以及改变过程中调用的过程,见第 10.3 节。

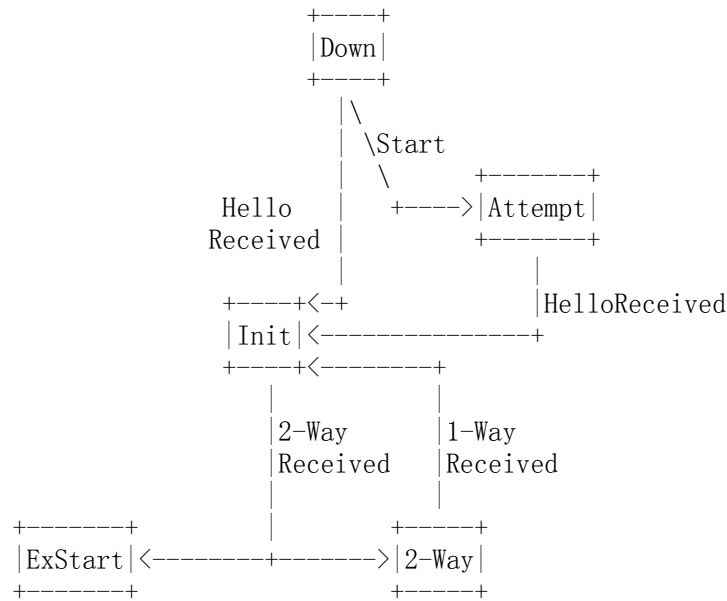


图 12: 邻居状态转变 (Hello 协议)

除了图中列出的转变, 事件 **KillNbr** 始终导致 **Down** 状态; 事件 **InactivityTimer** 始终导致 **Down** 状态; 事件 **LLDown** 始终导致 **Down** 状态。

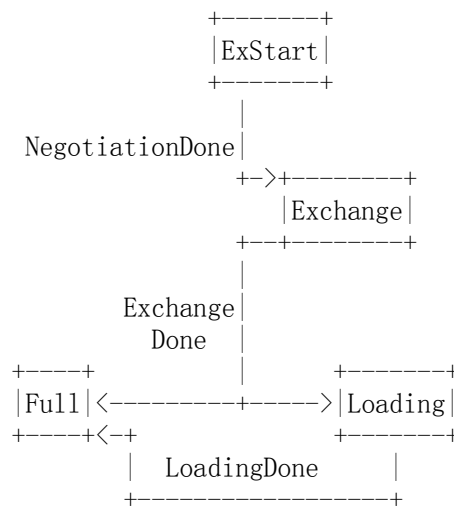


图 13: 邻居状态转变 (数据库交换)

除了图中列出的转变, 事件 **SeqNumberMismatch** 导致 **ExStart** 状态; 事件 **BadLSReq** 导致 **ExStart** 状态; 事件 **1-Way** 导致 **Init** 状态; 事件 **KillNbr** 始终导致 **Down** 状态; 事件 **InactivityTimer** 始终导致 **Down** 状态; 事件 **LLDown** 始终导致 **Down** 状态; 事件 **AdjOK?** 导致邻接的形成和拆除。

Down:

这是邻居状态转换的初始状态, 表示最近没有从邻居收到信息。在 **NBMA** 网络上, 仍然使用较低的频率向 “**Down**” 状态的邻居发送 **Hello** 包 (见第 9.5.1 节)。

Attempt:

该状态仅对 NBMA 网络上的邻居有效，表示最近没有从邻居收到信息，但仍需作出进一步的尝试，用以与邻居联系。这时按 HelloInterval 的间隔向邻居发送 Hello 包（见第 9.5.1 节）。

Init:

在此状态时，表示收到了由邻居发送的 Hello 包，但还没有建立与邻居的双向通讯（也就是说，路由器自身没有出现在该邻居的 Hello 包中）。从关联接口发出的 Hello 包中列出所有达到此状态（或更高状态）的邻居。

2-Way:

在此状态时，两台路由器的通讯达到双向，这是通过 Hello 协议而实现的。并且是建立邻接前的最高状态。DR（BDR）是从达到 2-Way 或更高状态的邻居中选举出来的。

ExStart:

这是两台邻居路由器之间建立邻接的第一步。此步的目标是确定路由器的主从，并确定初始的 DD 序号。达到此状态或更高状态的邻居会话被称为邻接。

Exchange:

在此状态的路由器通过向邻居发送 DD 包，来描述其完整的连接状态数据库。每个 DD 包都有一个 DD 序号，并需要被明确地确认。在任何时候，一次只能发送一个 DD 包。在此状态下，允许向邻居发送 LSR 包来取得较新的 LSA。所有达到 Exchange 或更高状态的邻接参与洪泛过程。事实上，这些状态的邻接完全有能力传送和接收所有类型的 OSPF 路由协议包。

Loading:

在此状态下，向邻居发送 LSR 包来取得较新的 LSA。就是在 Exchange 状态下发现（但还没有接收到）的那些 LSA。

Full:

在此状态下，邻居路由器达到完全邻接。现在开始这些邻接将出现在 Router-LSA 和 Network-LSA 中。

10.2. 引起邻居状态改变的事件

状态的改变受一系列事件的影响。这些事件标注在图 12 和图 13 中，并在下面定义：

HelloReceived:

从邻居接收到一个 Hello 包。

Start:

表示将以 HelloInterval 秒的间隔向邻居发送 Hello 包。这一事件仅与 NBMA 网络上的邻居相关。

2-WayReceived:

两台邻居路由器之间达到双向通讯。这表明在邻居的 Hello 包中包含了路由器自身。

NegotiationDone:

已经协商好主从关系，并交换了 DD 序号。这一信号表示开始收发 DD 包。生成这一事件的细节，参见第 10.8 节。

ExchangeDone:

两台路由器都已成功交换了完整的 DD 包。每台路由器也知道其连接状态数据库中过期的部分。生成这一事件的细节，参见第 10.8 节。

BadLSReq:

接收到的连接状态请求中，包含有并不存在于数据库中的 LSA。这说明在数据库交换过程中出现了错误。

LoadingDone:

连接状态更新已经接收了数据库中所有需要更新的部分。这是由数据库交换过程完成后，连接状态请求列表为空而表明的。

AdjOK?:

决定是否需要与邻居建立/维持邻接关系。这将导致一些邻接的形成和拆除。

下面的事件会导致邻居状态的降低。与上面的事件不同，这些事件会在任何邻居会话的状态时发生。

SeqNumberMismatch:

接收到的 DD 包出现下列情况：a) 含有意外的 DD 序号；b) 意外地设定了 Init 位；c) 与上一个接收到的 DD 包有着不同的选项域。这些情况都说明，在建立邻接的过程中出现了错误。

1-Way:

从邻居接收到 Hello 包，但并不包含路由器自身。这说明与该邻居的通讯不再是双向。

KillNbr:

这说明现在不可能与该邻居有任何通讯，强制转换邻居状态到 Down。

InactivityTimer:

非活跃记时器被激活。这说明最近没有从邻居接收到 Hello 包。强制转换邻居状态到 Down。

LLDown:

由下层协议说明，邻居不可到达。例如在 X.25 PDN 中，由于适当的原因或诊断会收到 X.25 clear，以表示邻居关闭。强制转换邻居状态到 Down。

10.3. 邻居状态机

下面描述邻居状态改变的细节。每次状态的改变都是由事件（见第 10.2 节）所调用的。根据当前的邻居状态，同样的事件可能有不同的影响。因此，状态机是根据当前的邻居状态和接收到的事件来组织的。每一项都描述了新的邻居状态和所需要的一系列额外操作。

当邻居状态改变后，也许需要重新运行 DR 选举算法。这通过是否生成 NeighborChange 接口事件（见第 9.2 节）来决定。同时，当接口为 DR 状态时（该路由器为 DR），邻居状态的改变需要生成

一个新的 Router-LSA（见第 12.4 节）。

当邻居状态机需要调用接口状态机时，应当被作为任务调度（见第 4.4 节）。这样做是为了简化过程，确保两个状态机之间不会出现递归调用。

状态：Down

事件：Start

新状态：Attempt

操作：向邻居发送 Hello 包（这一邻居始终是与 NBMA 网络相关联），并启动该邻居的非活跃计时器。计时器以后的激活说明没有取得与该邻居的通讯。

状态：Attempt

事件：HelloReceived

新状态：Init

操作：因为现在侦听到邻居，而重置该邻居的非激活计时器。

状态：Down

事件：HelloReceived

新状态：Init

操作：启动该邻居的非活跃计时器，计时器以后的激活说明该邻居关闭。

状态：Init 或更高

事件：HelloReceived

新状态：不变

操作：因为再次侦听到邻居，而重置该邻居的非激活计时器。

状态：Init

事件：2-WayReceived

新状态：取决于操作过程

操作：决定是否需要与邻居建立邻接（见第 10.4 节）。如果不需要，新的邻居状态为 2-Way。

否则（需要建立邻接），邻居状态转换为 ExStart。进入该状态后，路由器增加邻居数据结构中的 DD 序号。如果这是第一次试图建立邻接，DD 序号应当被设定为某些唯一值（如时钟上的时间）。声明自己为主机（设定主从位为主），并开始发送设定了初始（I）、更多（M）和主从（MS）位的 DD 包。DD 包的其他部分应当为空。DD 包以 RxmtInterval 的间隔重发，直到进入下一个状态（见第 10.8 节）。

状态：ExStart

事件：NegotiationDone

新状态：Exchange

操作：路由器必须列出在邻居数据库汇总列表中，所包含的的全部区域连接状态数据库。区域连接状态数据库包含有区域数据结构中的 Router-LSA、Network-LSA 和 Summary-LSA，以及全局数据结构中的 AS-external-LSA。在虚拟邻居数据库汇总列表中忽略 AS-external-LSA，对于存根区域的邻居数据库汇总列表也忽略 AS-external-LSA（见第 3.6 节）。时限等于 MaxAge 的 LSA 被改为加入邻居连接状态重传列表。在 DD 包中发送数据库汇总列表的摘要，每个 DD

包都有一个 DD 序号，并需要被明确确认。在任何时候，一次只允许发出一个 DD 包。关于发送和接收 DD 包的细节，见第 10.8 和 10.6 节。

状态: Exchange

事件: ExchangeDone

新状态: 取决于操作过程

操作: 如果邻居连接状态请求列表为空，则新的邻居状态为 Full。无须其他操作，已经达到完全邻接状态。

否则，新的邻居状态为 Loading。并开始（或继续）向邻居发送 LSR 包（见第 10.9 节），以获取较新的邻居 LSA（这些 LSA 在 Exchange 状态下被发现，但还没有接收到）。在相关邻居的连接状态请求列表中列出这些 LSA。

状态: Loading

事件: LoadingDone

新状态: Full

操作: 无须操作。已达到完全邻接状态。

状态: 2-Way

事件: AdjOK?

新状态: 取决于操作过程

操作: 决定是否需要与邻居路由器形成邻接（见第 10.4 节）。如果不需要，邻居状态就保持在 2-Way。

否则，邻居状态变为 ExStart，并执行上面 Init 状态下，收到 2-WayReceived 事件时所执行的操作。

状态: ExStart 或更高

事件: AdjOK?

新状态: 取决于操作过程

操作: 决定是否需要与邻居路由器继续保持邻接。如果是，就无须改变状态，且无须更多操作。

否则，邻接（可能仅部分建立）必须被拆除。邻居状态变为 2-Way。清除连接状态重传列表、数据库汇总列表和连接状态请求列表中的 LSA。

状态: Exchange 或更高

事件: SeqNumberMismatch

新状态: ExStart

操作: 邻接（可能仅部分建立）被拆除，并尝试重新建立。邻居状态首先变为 ExStart。清除连接状态重传列表、数据库汇总列表和连接状态请求列表中的 LSA。路由器增加邻居数据结构中的 DD 序号，声明自己为主机（设定主从位为主），并开始发送设定了初始（I）、更多（M）和主从（MS）位的 DD 包。DD 包中的其他部分应当为空（见第 10.8 节）。

状态: Exchange 或更高

事件: BadLSReq

新状态: ExStart

操作: 对 BadLSReq 事件的操作，与对 SeqNumberMismatch 事件的操作完全相同。邻接（可能仅

部分建立) 被拆除, 并尝试重新建立。更多细节见 **Exchange** 或更高状态下, **SeqNumberMismatch** 事件所执行的操作。

状态: 任意状态

事件: **KillNbr**

新状态: **Down**

操作: 清除连接状态重传列表、数据库汇总列表和连接状态请求列表中的 **LSA**。同时取消非活跃计时器。

状态: 任意状态

事件: **LLDown**

新状态: **Down**

操作: 清除连接状态重传列表、数据库汇总列表和连接状态请求列表中的 **LSA**。同时取消非活跃计时器。

状态: 任意状态

事件: **InactivityTimer**

新状态: **Down**

操作: 清除连接状态重传列表、数据库汇总列表和连接状态请求列表中的 **LSA**。

状态: **2-Way** 或更高

事件: **1-WayReceived**

新状态: **Init**

操作: 清除连接状态重传列表、数据库汇总列表和连接状态请求列表中的 **LSA**。

状态: **2-Way** 或更高

事件: **2-WayReceived**

新状态: 无须改变

操作: 无须操作。

状态: **Init**

事件: **1-WayReceived**

新状态: 无须改变

操作: 无须操作。

10.4. 是否形成邻接

仅在部分邻居路由器之间形成邻接。由点对点网络、点对多点网络和虚拟通道连接的路由器始终形成邻接。在广播网络和 **NBMA** 网络上, 所有的路由器与 **DR** 和 **BDR** 形成邻接。

在邻居状态机中有两处需要作出是否形成邻接的判断。第一, 当与邻居的双向通讯初次建立; 第二, 当所接入网络上的 **DR** (**BDR**) 发生改变。如果决定不形成邻接, 邻居状态就停留在 **2-Way**。

当满足下列至少一个条件时, 在双向邻居间建立邻接:

下层网络的类型为点对点；

下层网络的类型为点对多点；

下层网络的类型为虚拟通道；

路由器自身为指定路由器；

路由器自身为备份指定路由器；

邻居路由器为指定路由器；

邻居路由器为备份指定路由器。

10.5. 接收到 Hello 包

本节介绍接收到 Hello 包后的处理过程（Hello 包的格式见附录 A.3.2）。通常，都要检查所输入的 OSPF 包中 IP 包头和 OSPF 包头的有效性。其次，对于所接收到的网络掩码、HelloInterval 和 RouterDeadInterval 域都必须与接收接口上的配置进行比较。任何的不匹配都将导致终止处理过程并丢弃该包。换句话说，上面这些域描述的都是所接入的网络。有一个例外：在点对点网络和虚拟通道上，忽略所接收 Hello 包中的网络掩码。

接收的接口接入单一的 OSPF 区域（可能是骨干区域）。Hello 包选项域中 E 位的设定，必须与该区域的 ExternalRoutingCapability 相匹配。如果 AS-external-LSA 不被洪泛进该区域（即存根区域），必须清除 Hello 包中的 E 位，不然必须设定 E 位。任何的不匹配都将导致终止处理过程并丢弃该包。应当忽略 Hello 包中选项域内其他位的设定。

此时，尝试将 Hello 包中的源地址与接收接口的某个邻居相关联。如果接收接口连接的是广播、点对多点或 NBMA 网络，源地址就是 Hello 包中 IP 头部里的 IP 源地址。如果接收接口连接的是点对点网络或虚拟通道，源地址就是 Hello 包中 OSPF 头部里的路由器标识。该接口的数据结构中包含有当前的邻居列表。如果没有找到相关联的邻居结构（即这是第一次发现该邻居），则将其创建。将新创建的邻居初始状态设定为 Down。

当从广播、点对多点或 NBMA 网络上的邻居接收到 Hello 包时，将邻居结构中的邻居标识设定为 OSPF 包头中的路由器标识。在这些网络类型上，邻居结构中的邻居优先级、邻居的指定路由器、邻居的备份指定路由器，都要按接收到的 Hello 包中的内容设定。这些域的改动需要注意到下面提到的使用情况。在点对点网络上（但不包括虚拟通道），将邻居结构中的邻居标识设定为 IP 包的源地址。

现在检查 Hello 包中的其余部分，并向邻居及接口状态机生成事件。这些状态机被指定为直接执行或被调度执行（见第 4.4 节）。例如当接收到的一个 Hello 包引起了多个邻居状态的转换，就要指定哪个邻居状态机被在线执行。

每个 Hello 包引起邻居状态机执行事件 HelloReceived。

然后检查 Hello 包中的邻居列表。如果路由器自身出现在列表中，邻居状态机执行事件 **2-WayReceived**。否则，邻居状态机执行事件 **1-WayReceived**，并终止包处理过程。

下面，如果发现邻居优先级有改变，接收接口状态机调度执行事件 **NeighborChange**。

如果邻居宣告自己为 **DR**（Hello 包的指定路由器域 = 邻居 IP 地址），且包中的备份指定路由器域为 0.0.0.0，而且接收接口状态机的状态为 **Waiting**，接收接口状态机调度执行事件 **BackupSeen**。否则，如果以前不宣告的邻居宣告自己为 **DR**，或以前宣告的邻居现在不宣告自己为 **DR**，接收接口状态机调度执行事件 **NeighborChange**。

如果邻居宣告自己为 **BDR**（Hello 包的备份指定路由器域 = 邻居 IP 地址），且接收接口状态机的状态为 **Waiting**，接收接口状态机调度执行事件 **BackupSeen**。否则，如果以前不宣告的邻居宣告自己为 **BDR**，或以前宣告的邻居现在不宣告自己为 **BDR**，接收接口状态机调度执行事件 **NeighborChange**。

在 NBMA 网络上，收到 Hello 包还可能引起 Hello 包的回送。更多细节见第 9.5.1 节。

10.6. 接收到数据库描述包

本节介绍接收到 DD 包的处理细节。通过调用通用包输入过程（第 8.2 节），所输入的 DD 包已经有了相关联的邻居和接收接口。是否应当接收 DD 包，以及如何进一步处理，取决于邻居状态。

如果接收了一个 DD 包，包中的 DD 序号，以及初始（I）、更多（M）和主从（MS）选项位都应当被保存在对应邻居数据结构的“最近接收到的 DD 包”中。如果从一个邻居连续收到这些域都相同的两个包，第二个 DD 包会被下面的处理过程认为是“重复”。

如果 DD 包中表示 IP 包大小的接口 MTU 域，大于该路由器接口所能接收的不分片大小，拒绝该 DD 包。否则，如果邻居路由器的状态是：

Down:

包应当被拒绝。

Attempt:

包应当被拒绝。

Init:

邻居状态机应当执行事件 **2-WayReceived**。这导致立即变为状态 **2-Way** 或状态 **ExStart**。如果新状态是 **ExStart**，按照下面 **ExStart** 情况的介绍，继续当前包的处理。

2-Way:

包应当被忽略。DD 包仅为建立邻接而使用。[脚注 7]

ExStart:

如果接收到的包满足下列一个条件，则邻居状态机需要执行 **NegotiationDone** 事件（这将使状态变为 **Exchange**）。包的选项域应当被记录在邻居结构的选项域中，包的序号应当是下一个

要处理的号码（见下）。否则，忽略该包。

设定了初始（I）、更多（M）和主从（MS）选项位，包的其他部分为空，且邻居路由器标识比自身路由器标识要大。这时，路由器作为从机/Slave。设定主从位为从，并将邻居数据结构中的邻居 DD 序号设定为主机所提出的号码。

清除了初始（I）和主从（MS）选项位，且包中的 DD 序号等于邻居数据结构中的 DD 序号（标明为确认），而且邻居路由器标识比自身路由器标识要小。这时，路由器作为主机/Master。

Exchange:

主机丢弃所收到的重复 DD 包；从机收到重复的 DD 包时，则应当重发前一个 DD 包。否则（不是重复包）：

如果主从（MS）位的状态与当前的主从关系不匹配，生成 SeqNumberMismatch 邻居事件，并终止包的处理。

如果意外设定了初始（I）位，生成 SeqNumberMismatch 邻居事件，并终止包的处理。

如果包的选项域与以前接收到的 OSPF 可选项不同（记录在邻居结构的邻居选项中），生成 SeqNumberMismatch 邻居事件，并终止包的处理。

DD 包必须按包的 DD 序号顺序处理。对于主机，下一个包中的 DD 序号应当等于邻居数据结构中的 DD 序号；对于从机，下一个包中的 DD 序号应当等于邻居数据结构中的 DD 序号再加一。不论哪种情况，如果下一个包中的序号正确，都应当被接收并按下面的说明处理。

否则，生成 SeqNumberMismatch 邻居事件，并终止包的处理。

Loading 或 Full:

在此状态时，路由器已经收发了全部 DD 包。只可能接收重复的 DD 包（见上）。这时，包的选项域应当与以前接收到的 OSPF 可选项相同（记录在邻居结构的邻居选项中）。如果收到其他的包，包括设定了初始（I）位的包，都应当生成 SeqNumberMismatch 邻居事件 [脚注 8]。主机丢弃所收到的重复 DD 包；从机收到重复的 DD 包时，则应当重发前一个 DD 包。

当路由器接收到序列号匹配的 DD 包后，按如下处理。对于每一个 LSA 列表，检查其 LS 类型的有效性。如果 LS 类型为未知（即不是本规范所定义的 LS 类型 1-5），或者是一个 AS-external-LSA（LS 类型 = 5）而邻居关联到一个存根区域，就生成 SeqNumberMismatch 邻居事件，并终止包的处理。否则，路由器在数据库中是否有该 LSA 的实例，如果没有，或者数据库中的数据较旧（见第 13.1 节），就将该 LSA 放入连接状态请求列表中，并在 LSR 包中（立即或一段时间后）发送。

当路由器接收到序列号匹配的 DD 包后，根据其主从方式，有如下操作：

主机:

将邻居数据结构中的 DD 序号加一。如果路由器已经发送了全部的 DD 包，并收到了清除了更

多（M）位的包，就生成 **ExchangeDone** 邻居事件。否则，向从机发送新的 DD 包。

从机：

将接收包中的 DD 序号设定为邻居数据结构中的 DD 序列号。从机必须发送一个 DD 包作为回应。如果接收到清除了更多（M）位的包，而且从机发送的包也将更多（M）位清 0，就生成 **ExchangeDone** 邻居事件。注意，从机始终比主机早生成这一事件。

10.7. 接收到连接状态请求包

本节介绍接收到 LSR 包的处理细节。接收到 LSR 包，意味着邻居希望接收一系列 LSA。当邻居状态为 **Exchange**、**Loading** 或 **Full** 时，应当接收 LSR 包；而在其他状态下应当忽略。

应当在路由器的数据库中检索 LSR 包中的每个 LSA，并将其复制到 LSU 包中发往邻居。这些 LSA “不” 应当被放入邻居的连接状态重传列表中。如果某个 LSA 没有在数据库中被找到，说明在数据库交换过程中出错，应当生成 **BadLSReq** 邻居事件。

10.8. 发送数据库描述包

本节描述如何向邻居发送 DD 包。DD 包中的接口 MTU 域，应当被设定为该接口所能发送的最大不分片 IP 包大小。Internet 中常用的 MTU 可以在 [引用 22] 的表 7-1 中找到。在虚拟通道上发送 DD 包时，应当将接口 MTU 设定为 0。

路由器的 OSPF 可选项（见第 4.5 节）在 DD 包的选项域中发往邻居。在数据库交换过程和洪泛过程中，各路由器需要维持一组相同的选项。如果因为某些原因改变了路由器的可选项，必须将邻居状态回到 **ExStart**，以重新开始数据库交换过程。本规范定义了一个可选项（见第 4.5 节和附录 A.2）。当且仅当所接入的网络不是存根网络时，才设定 E 位。选项域中未知的位应当被设为 0。

DD 包的发送取决于邻居的状态。在 **ExStart** 状态时，路由器发送设定了初始（I）、更多（M）和主从（MS）位的空包。这些包按每 **RxmtInterval** 秒重传。

在 **Exchange** 状态中的 DD 包，实际上包含了路由器数据库中连接状态信息的摘要。在邻居数据库汇总列表中，列出了区域连接状态数据库中的每个 LSA（这时，邻居进入 **Exchange** 状态）。每个新的 DD 包从邻居数据结构中复制出 DD 序号，并描述数据库汇总列表顶端的数据。当收到前一个包的确认后，将该项从数据库汇总列表中删除。

在 **Exchange** 状态下，何时发送 DD 包，取决于路由器的主从状态。

主机：

在下列两个条件之一时，发送 DD 包：a) 从机通过回显 DD 序号，确认了前一个 DD 包；b) 经过了 **RxmtInterval** 秒也没有收到确认，这时重传前一个 DD 包。

从机：

只能在接收到由主机发送的 DD 包时，才能发送 DD 包作为回应。如果从主机收到了一个新的 DD 包，则发送新包；否则，重新发送前一个 DD 包。

在 **Loading** 和 **Full** 状态下，当收到主机重复的 **DD** 包时，从机必须重新发送最后一个 **DD** 包来作出回应。为此，从机必须等待 **RouterDeadInterval** 秒，才能释放最后一个 **DD** 包。在此之后收到的 **DD** 包，会生成 **SeqNumberMismatch** 邻居事件。

10.9. 发送连接状态请求包

当在 **Exchange** 或 **Loading** 邻居状态时，连接状态请求列表中包含了需要从邻居更新的 **LSA** 列表。路由器将连接状态请求列表的开始部分，包含在 **LSR** 包中发给邻居，来取得这些 **LSA**。

当邻居以适当的 **LSU** 包，回应了这些请求后，删减连接状态请求列表，并发出新的 **LSR** 包。继续这一过程，直到连接状态请求列表为空。那些在连接状态请求列表中，已经发出，但还没有收到的 **LSA**，被包含在 **LSR** 包中，以 **RxmtInterval** 秒的周期重传。在任何时候，都至多发送一个 **LSR** 包。

当邻居状态为 **Loading** 而连接状态请求列表为空时（也就是已经与邻居完整地收发了全部的 **DD** 包），生成 **LoadingDone** 邻居事件。

10.10. 示例

图 14 显示了一个邻接形成的例子。路由器 **RT1** 和 **RT2** 都连接到一个广播网络。并假设 **RT2** 是网络上的 **DR**，**RT2** 的路由器标识比 **RT1** 的大。

在图旁边列出了每台路由器所认识到的邻居状态改变。

在图 14 的开始，路由器 **RT1** 的网络接口接通，并开始发送 **Hello** 包。虽然它并不知道 **DR** 的身份，也不知道其他任何邻居路由器。路由器 **RT2** 收到 **Hello** 包（转变邻居状态为 **Init**），并在下一个 **Hello** 包中标明自己为 **DR**，并且已经收到了 **RT1** 的 **Hello** 包。这使 **RT1** 的状态转为 **ExStart**，并开始形成邻接。

RT1 一开始假设自己为主机，但当发现 **RT2** 为主机后（因为 **RT2** 的路由器标识较大），**RT1** 变为从机状态，并采用邻居的 **DD** 序号。开始交换 **DD** 包，由主机（**RT2**）开始发出，并由从机（**RT1**）响应。当发出和响应包的 **M** 位都清除后，**DD** 包的序列结束。

在本例中，假设 **RT2** 的数据库已经完全更新。这样，**RT2** 就立即进入 **Full** 状态。**RT1** 的数据库在经过必须的更新后，也进入 **Full** 状态。这是通过发送 **LSR** 包，接收 **LSU** 包的回应来实现的。注意，例子中，**RT1** 是在接收了（从 **RT2**）完整的 **DD** 包以后，才发送 **LSR** 包的，而实际上不需要如此。**RT1** 可以在接收 **DD** 包的同时，发送 **LSR** 包。

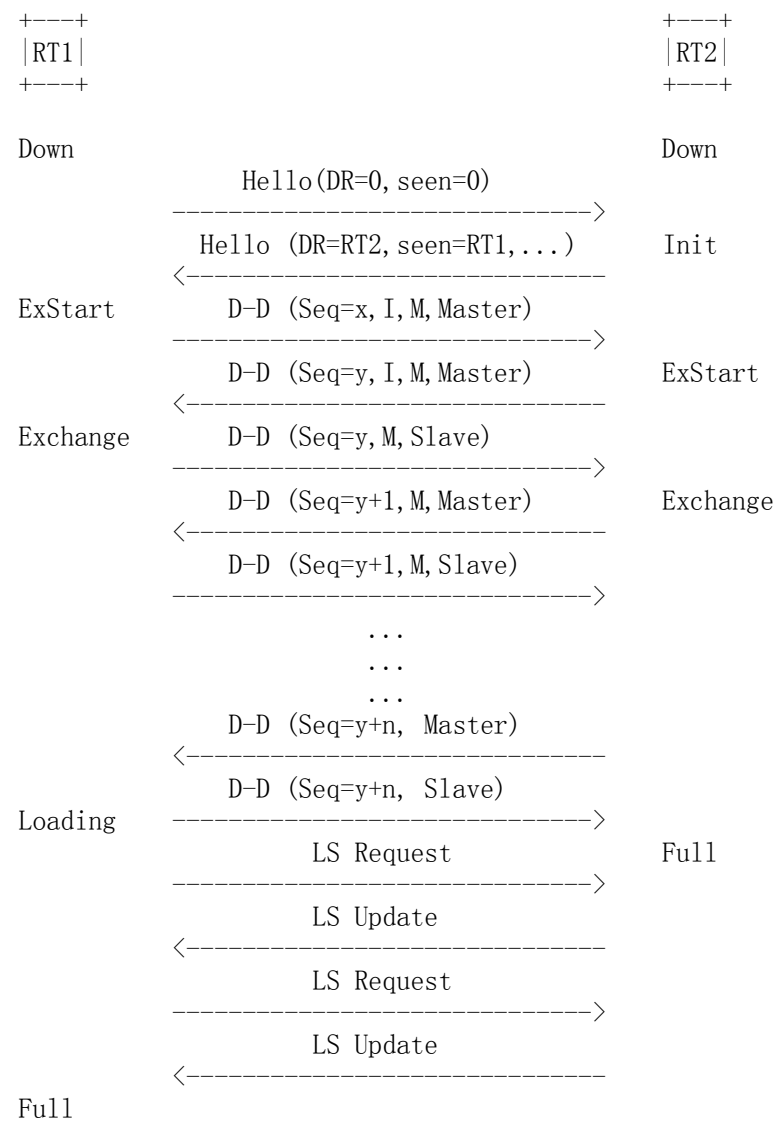


图 14：形成邻接的例子

11. 路由表结构

向目标转发 IP 数据包所需要的所有信息，都包含在路由表数据结构中。路由表中的每一项都描述了到达特定目标的一组最佳路径。当转发 IP 数据包时，路由表提供到达 IP 目标的最佳匹配项。所匹配的路由表项，提供了到达目标的下一跳地址。OSPF 同样提供默认路径（目标标识 = `DefaultDestination`，地址掩码 = `0x00000000`）。当存在默认路径时，它可以匹配所有的 IP 目标（尽管其他项提供更佳匹配）。下面的第 11.1 节描述，如何在路由表中查找最佳匹配 IP 目标的项。

各路由器有独立的路由表。第 11.2 和 11.3 节描述两个路由表的示例，第 16 章描述路由表的建立。

本节的剩余部分定义了路由表项中的各域。首先说明描述路由表目标的各域。

目标类型/Destination Type:

目标类型为“网络”或“路由器”。在转发 IP 数据时，仅使用网络项。路由器项仅用于建立路由表的中间步骤。

网络定义了要转发数据的 IP 地址范围。这包括 IP 网络（A、B 或 C 类）、IP 子网、IP 超网和单一 IP 主机。默认路径也被归为这一类。

路由器项记录了 ABR 和 ASBR。ABR 的路由表项用于计算区域间路径（见第 16.2 节），并维持虚拟通道的配置（见第 15 章）。ASBR 的路由表项用于计算 AS 外部路径（见第 16.4 节）。

目标标识/Destination ID:

目标的标识或名称，取决于目标类型。对网络项，标识是所关联的 IP 地址；对路由器项，标识是 OSPF 的路由器标识。[脚注 9]

地址掩码/Address Mask:

仅为网络项定义。网络的 IP 地址与地址掩码一起，定义了 IP 地址的范围。对于 IP 子网，地址掩码就是子网掩码。对主机路径，掩码为“全 1”（`0xffffffff`）。

可选项/Optional Capabilities:

当目标为路由器时，该项表示目标路由器所支持的 OSPF 可选项。本规范唯一定义的可选项是处理 AS-external-LSA 的能力。更多关于 OSPF 可选项的描述，见第 4.5 节。

到达一个目标所使用的路径，可能根据所属的 OSPF 区域而不同。这意味着，依据下面各项的值，可能有多个到达同一目标的路由表项。

区域/Area:

该项表示，路由表是从哪个区域的连接状态信息中取得该路由表项。这也被称为该项的关联区域。对于 AS 外部路径，该域没有定义。对于“路由器”类型的目标，可能存在多个相关联的区域（也就有多个独立的路由表项）。例如，当两台 ABR 同时接入多个相同区域时。对于“网络”类型的目标，只保留最佳路径所在的区域（只提供一条路径）。

路由表项的其他部分描述了到达目标的一系列路径。下面的各域，作为一个整体描述了一条路径。也就是说，在路由表项中的每一条路径都有相同的路径类型和距离值（见下）。

路径类型/Path-type:

有四种到达目标的路径类型，按优先选择的次序分别是：区域内路径、区域间路径、类型 1 外部路径和类型 2 外部路径。区域内路径表示目标属于路由器所接入的某一区域；区域间路径是到达 OSPF 其他区域目标的路径，通过检查所收到的 Summary-LSA 而发现，外部路径是到达 AS 外部目标的路径。通过检查所收到的 AS-external-LSA 而发现。

距离值/Cost:

到达目标的连接状态距离值。对于类型 2 外部路径，该值表示了 AS 内部网络部分的距离。在其他路径类型中表示了整条路径的距离。这一数值是由一系列连接距离值求和计算而得来。

类型 2 距离值/Type 2 cost:

仅对类型 2 外部路径有效。在这些路径中，该域表示路径中 AS 外部网络部分的距离。该数值由 ASBR 所宣告，并且是整条路径中最首要的部分。例如，类型 2 距离值为 5 的路径，始终优于类型 2 距离值为 10 的路径，而不考虑两条路径的内部距离。

连接状态起源/Link State Origin:

仅对区域内路径有效，该域标明了直接涉及目标的 LSA（Router-LSA 或 Network-LSA）。例如，如果目标为传输网络，那就是传输网络的 Network-LSA；如果目标为存根网络，那就是所接入路由器的 Router-LSA。在计算最短路径树时发现该 LSA（见第 16.1 节）。也许有多个 LSA 对应同一个目标，但通过一个选择方案始终只选择一个 LSA。OSPF 协议并不使用连接状态起源域，而是在 OSPF 多播路由扩展（MOSPF）的路由表计算中使用。

当有相同路径类型和距离值的多条路径到达同一目标时（或被称为“等值”路径），被存储在一个路由表项中。每条“等值”路径由下面的域区分。

下一跳/Next hop:

当向目标转发流量时，所使用的路由器输出接口。在广播、点对多点和 NBMA 网络上，下一跳还包括到达目标路径中下一跳路由器的 IP 地址。

宣告路由器/Advertising router:

仅对区域间路径和 AS 外部路径有效。该域包含了宣告 Summary-LSA 或 AS-external-LSA 的路由器标识，该 LSA 产生了这条路径。

11.1. 查找路由表

当接收到 IP 数据包时，OSPF 路由器在路由表中查找最匹配目标的项。该路由表项提供了转发数据包所需要的输出接口和下一跳路由器。本节描述查找最匹配项的过程。

在开始查找前，应当为路由器的活跃区域地址范围（见第 3.5 节），向路由表添加“丢弃/discard”路由表项。（当通过区域内路径可以到达区域地址范围所包含的网络时，该区域范围就被认为是“活跃”的。）该“丢弃”项的目标被设为活跃区域地址范围所描述的地址，其路径类型被设为“区域间”。[脚注 10]

也许会有多个路由表项与目标地址匹配。这时，最精确（最长）的匹配提供了“最佳”匹配的路由

表项。换句话说，就是选择具有最小 IP 地址范围的项 [脚注 11]。例如，地址/掩码对（128.185.1.0，0xfffff00）就要比（128.185.0.0，0xffff0000）来得精确。默认路径能够匹配任何目标，但却是最不精确的。（注意在一个路由表项中，可能会有多条路径。这时，在第 16.1 节、第 16.2 节和第 16.4 节中的计算始终按第 11 章所述，选择最优先的路径类型。）

如果没有相匹配的路由表项，或者最匹配的是上面所说的“丢弃”项，那么该包的 IP 目标就被认为不可到达。该包不被转发，而是被丢弃并应当向包的源地址发出 ICMP 目标不可达信息。

11.2. 路由表示例，无区域

参见图 2 中的 AS，没有配置 OSPF 区域，在每个输出接口上都配置了简单尺度。在第 2.2 节中描述了路由器 RT6 的路由表计算过程。其路由表结果显示在表 12 中。在类型中“网络”类型缩写为“N”、“路由器”类型缩写为“R”。

类型	目标	区域	路径类型	距离值	下一跳	宣告路由器
N	N1	0	区域内	10	RT3	*
N	N2	0	区域内	10	RT3	*
N	N3	0	区域内	7	RT3	*
N	N4	0	区域内	8	RT3	*
N	Ib	0	区域内	7	*	*
N	Ia	0	区域内	12	RT10	*
N	N6	0	区域内	8	RT10	*
N	N7	0	区域内	12	RT10	*
N	N8	0	区域内	10	RT10	*
N	N9	0	区域内	11	RT10	*
N	N10	0	区域内	13	RT10	*
N	N11	0	区域内	14	RT10	*
N	H1	0	区域内	21	RT10	*
R	RT5	0	区域内	6	RT5	*
R	RT7	0	区域内	8	RT10	*
N	N12	*	类型 1 外部	10	RT10	RT7
N	N13	*	类型 1 外部	14	RT5	RT5
N	N14	*	类型 1 外部	14	RT5	RT5
N	N15	*	类型 1 外部	17	RT10	RT7

表 12: 路由器 RT6 的路由表（未配置区域）

在本例中没有出现等值多路径。同时因为没有区域，也就没有区域间路径。

路由器 RT5 和 RT7 是 ASBR。需要计算到达 RT5 和 RT7 的区域内路径，并以此计算由这两台路由器所宣告的外部路径（即网络 N12、N13、N14 和 N15）。在这里假设 RT5 和 RT7 所生成的 AS-external-LSA 都是类型 1 外部距离。并计算到达目标 N12-N15 的类型 1 外部路径。

11.3. 路由表示例，有区域

考虑前面的例子，这次划分了 OSPF 区域，区域的配置按图 6。本例描述路由器 RT4 的路由表，RT4 接入了区域 1 和骨干区域。这时 RT4 将 AS 视为图 7 和图 8 的串联。其路由表计算的结果显示在表 13 中。

类型	目标	区域	路径类型	距离值	下一跳	宣告路由器
N	N1	1	区域内	4	RT1	*
N	N2	1	区域内	4	RT2	*
N	N3	1	区域内	1	*	*
N	N4	1	区域内	3	RT3	*
R	RT3	1	区域内	1	*	*
N	Ib	0	区域内	22	RT5	*
N	Ia	0	区域内	27	RT5	*
R	RT3	0	区域内	21	RT5	*
R	RT5	0	区域内	8	*	*
R	RT7	0	区域内	14	RT5	*
R	RT10	0	区域内	22	RT5	*
R	RT11	0	区域内	25	RT5	*
N	N6	0	区域间	15	RT5	RT7
N	N7	0	区域间	19	RT5	RT7
N	N8	0	区域间	18	RT5	RT7
N	N9-N11, H1	0	区域间	36	RT5	RT11
N	N12	*	类型 1 外部	16	RT5	RT5, RT7
N	N13	*	类型 1 外部	16	RT5	RT5
N	N14	*	类型 1 外部	16	RT5	RT5
N	N15	*	类型 1 外部	23	RT5	RT7

表 13: 存在区域时路由器 RT4 的路由表

同样，路由器 RT5 和 RT7 是 ASBR，而 RT3、RT4、RT7、RT10 和 RT11 是 ABR。注意，因为与 RT3 有两个相同的区域（区域 1 和骨干区域），所以有两条到达 ABR RT3 的路由项。

计算所有 ABR 在骨干区域中的路径，用以决定区域间路径。注意，所有的区域间路径都与骨干区域相关，当路由器自身为 ABR 时也始终进行这样的计算。路由信息在区域边界汇聚，本例中我们假设区域 3 中定义，将网络 N9-N11 以及主机 H1 汇总到一条路径中，向骨干区域宣告（通过路由器 RT11）。注意，其距离值是一系列独立部分距离值中的最大值。

在路由器 RT10 和 RT11 之间配置了一条虚拟通道。如果不配置的话，RT11 无法将网络 N9-N11 和主机 H1 的路径宣告进骨干区域，这一项也就不会出现在路由器 RT4 的路由表中。

本例中，有两条到达网络 N12 的等值路径，但它们使用同样的下一跳（路由器 RT5）。

如果在路由器 RT4 和 RT3 之间增加配置虚拟通道，路由器 RT4 的路由表将会得到改善（即某些路由表中的路径将变短）。新的虚拟通道所关联的第一项是表 13 中的 ABR RT3（通过区域 1 的区域内路径），其距离值为 1。加入了虚拟通道后，路由表的改变部分显示在表 14 中。

类型	目标	区域	路径类型	距离值	下一跳	宣告路由器
N	Ib	0	区域内	16	RT3	*
N	Ia	0	区域内	21	RT3	*
R	RT3	0	区域内	1	*	*
R	RT10	0	区域内	16	RT3	*
R	RT11	0	区域内	19	RT3	*
N	N9-N11, H1	0	区域间	30	RT3	RT11

表 14：增加了虚拟通道后的改变

12. 连接状态宣告 (LSA)

AS 中的每台路由器都生成一个或多个连接状态宣告 (LSA)，本备忘录按第 4.3 节的描述定义了五种不同类型的 LSA。LSA 的集合形成了连接状态数据库。每种不同类型的 LSA 有不同的功能：Router-LSA 和 Network-LSA 描述了区域内路由器和网络的互联；Summary-LSA 提供了区域路由器信息的汇总；AS-external-LSA 提供了将外部路由信息传入 AS 内部的方法。

每个 LSA 开始于 20 个字节的标准头部。LSA 头部在下面描述。

12.1. LSA 头部

LSA 头部包含有连接状态类型/LS 类型、连接状态标识/LS 标识以及宣告路由器。这三个域的组合唯一的标识出 LSA。

在同一时刻，可能会有 LSA 的多个实例存在于 AS 中，必须判定哪一个实例较新。这通过检查 LS 序号、LS 校验和以及 LS 时限来实现。这些域都包含在 20 个字节的 LSA 头部中。

一些 OSPF 包中需要列出 LSA。当不需要实例版本时，按 LS 类型、LS 标识以及宣告路由器来引用 LSA（见 LSR 包）。否则必须加上 LS 序号、LS 时限以及 LS 校验和域。

LSA 头部的细节在下面描述。

12.1.1. 连接状态时限

该域以秒数表示 LSA 存在的时间，必须按 16 位无符号整数处理。在洪泛过程中，每经过一跳必须增加 InfTransDelay。当保存在各路由器中的数据库时，LSA 同样需要被老化。

LSA 的时限增加决不超过 MaxAge。时限达到 MaxAge 的 LSA 不再用于路由表的计算。当 LSA 的时限第一次到达 MaxAge 时，将被重新洪泛。当不再需要保持数据库同步时，达到 MaxAge 的 LSA 被从数据库中删除。关于 LSA 老化的更多信息，参见第 14 章。

当路由器接收到两个具有相同 LS 序号和 LS 校验和的 LSA 实例时，检查 LS 时限域。拥有 MaxAge 的实例始终被认为是较新的，这使得老的 LSA 可以被快速的从路由域中删除。否则，如果其时限差异大于 MaxAgeDiff 的话，具有较小时限的实例被认为较新 [脚注 12]。更多细节见第 13.1 节。

12.1.2. 选项

LSA 头部中的选项域表示所关联 LSA 的可选项。OSPF 可选项在第 4.5 节中描述。本规范所定义的一个可选项，在选项中表示为 E 位。选项中未知的位应当被设为 0。

E 位表示 OSPF 的 ExternalRoutingCapability。该位应当被设置在所有与骨干区域关联的 LSA，以及所有与非存根区域关联的 LSA 上；也应当被设置在所有的 AS-external-LSA 上。在存根区域的所有 Router-LSA、Network-LSA 和 Summary-LSA 上，应当被清除。所有 LSA 上的 E 位设置都只是为了通告，并不影响路由表的计算。

12.1.3. 连接状态类型

LS 类型域描述 LSA 的格式和功能。各类型的 LSA 有各自的名称(即 Router-LSA 或 Network-LSA)。本备忘录所定义的各种 LSA，除了 AS-external-LSA (LS 类型 5) 外，都只在一个区域内洪泛。而 AS-external-LSA 在除存根区域的整个 AS 内被洪泛 (见第 3.6 节)。不同类型的 LSA 介绍见表 15。

LS 类型	LSA 描述
1	即 Router-LSA。描述了路由器接口的状态。更多信息见第 12.4.1 节。
2	即 Network-LSA。描述了网络上所接入的路由器。更多信息见第 12.4.2 节。
3 或 4	即 Summary-LSA。描述了区域间路径，并允许在区域边界上汇总路由信息。由 ABR 生成，类型 3 的 Summary-LSA 描述了到达网络的路径，而类型 4 的 Summary-LSA 描述了到达 ASBR 的路径。
5	即 AS-external-LSA。由 ASBR 生成，描述 AS 外部路径。AS 的默认路径也被描述为 AS-external-LSA。

表 15: OSPF 连接状态宣告 (LSA)

12.1.4. 连接状态标识

该域用于在路由域中描述一个 LSA。根据 LSA 的 LS 类型，LS 标识的取值见表 16。

LS 类型	LS 标识
1	生成路由器的路由器标识。
2	该网络上 DR 的 IP 接口地址。
3	目标网络的 IP 地址。
4	所描述的 ASBR 的路由器标识。
5	目标网络的 IP 地址。

表 16: LSA 的 LS 标识

事实上，类型 3 的 Summary-LSA (LS 类型 = 3) 和 AS-external-LSA (LS 类型 = 5) 的 LS 标识可能会加上一位或多位目标网络的“主机”位。例如，当为使用掩码 255.0.0.0 的网络 10.0.0.0 生成 AS-external-LSA 时。LS 标识可以设定为 10.0.0.0 到 10.255.255.255 的任意值 (虽然应当竟可能使用 10.0.0.0)。这种随意性可以使路由器为两个使用同样网络地址，但不同掩码的网络生成 LSA。更多细节见附录 E。

当 LSA 描述一个网络 (LS 类型=2、3 或 5) 时，可以将 LS 标识与 LSA 中所包含的网络/子网掩码进行操作，而得到网络的 IP 地址。当 LSA 描述一个路由器 (LS 类型=1 或 4)，LS 标识始终描述了路由器的 OSPF 路由器标识。

当 AS-external-LSA (LS 类型=5) 描述默认路径时，其 LS 标识被设为 DefaultDestination (0.0.0.0)。

12.1.5. 宣告路由器

该域说明了 LSA 起源的 OSPF 路由器标识。对于 Router-LSA，该域与 LS 标识域相同；Network-LSA

由网络上的 DR 生成；Summary-LSA 由 ABR 生成；AS-external-LSA 由 ASBR 生成。

12.1.6. 连接状态序号

序号是一个 32 位符号整数。用于判定旧的或相同的 LSA。序号空间是线形的，较大的序号（按 32 位符号整数比较）表示较新的 LSA。为了更精确的描述序号，下面定义 N 为常数 2^{31} 。

序号 $-N$ (0x80000000) 被保留未用。而 $-N+1$ (0x80000001) 表示最小（也就是最旧）的序号，这一序号被定义为常数 InitialSequenceNumber。路由器在第一次生成任何 LSA 的时候使用 InitialSequenceNumber。然后，当每次路由器生成新的 LSA 实例时，将 LSA 的序号加一。当试图增加最大序号 $N-1$ (0x7fffffff, 也被定义为 MaxSequenceNumber) 时，必须先将当前 LSA 从路由域中废止。这通过将 LSA 提早老化（见第 14.1 节）并重新洪泛而实现。当从所有邻接的邻居收到确认后，生成以 InitialSequenceNumber 为序号的新实例。

当在洪泛过程中收到意外的 LSA 新实例时，路由器可能会强制提高其 LSA 序号。这应当绝少发生，这可能是路由器在上一次重启前发出的旧 LSA，仍存在于 AS 中。更多信息见第 13.4 节。

12.1.7. 连接状态校验和

该域是除了 LS 时限域外，LSA 各域的校验和。将 LS 时限域排除在外，是为了在不更新校验和的情况下增加 LS 时限。该校验和的使用与在 ISO 无连接数据报中的使用相同，通常被称为 Fletcher 校验和，在 [引用 6] 的附录 B 中说明。LSA 中还包括了 LSA 的字节长度，但要减去 LS 时限域的大小（2 个字节），与校验和的校验范围相同。

校验和用于判断 LSA 的数据错误。错误可能发生在 LSA 洪泛过程或保存在路由器内存中时。LS 校验和不能被设为 0；这样的值被视为校验错误。换句话说，校验和的计算不是可选的。

在两种情况下检查 LSA 的校验和：a) 当接收到 LSU 包；b) 老化连接状态数据库时。不同情况下的校验错误将导致不同的操作。更多细节见第 13 章和第 14 章。

当两个 LSA 实例具有相同的 LS 序号时，检查 LS 校验和。如果两者不同，较大 LS 校验和的实例被认为较新 [脚注 13]。更多细节见第 13.1 节。

12.2. 连接状态数据库

路由器为其所接入的每个区域都建有独立的连接状态数据库。所有属于同一区域的路由器，都有着相同的区域连接状态数据库。

各区域的数据库始终被分开处理，最短路径树的构建也按区域分开（见第 16 章）。区域连接数据库中的内容仅在区域内洪泛。最后，当一对邻接形成（属于区域 A）时，两台路由器之间也仅同步区域 A 的数据库。

区域数据库的组成部分有 Router-LSA、Network-LSA 和 Summary-LSA（都在区域数据结构中列出）。此外，外部路径（AS-external-LSA）也被包含在所有非存根区域中（见第 3.6 节）。

OSPF 的实现必须能够访问区域数据库中的各个部分。要有基于 LS 类型、LS 标识以及宣告路由器的 LSA 查找函数 [脚注 14]。在数据库中的 LSA 都只有一个实例 (最新的)。在 LSA 洪泛过程 (见第 13 章) 以及路由表计算过程 (见第 16 章) 中调用数据库查找函数。此外, 路由器通过查找函数判断某个特定的 LSA 是否是自生成的, 以及它的 LS 序号。

添加一个 LSA 到路由器数据库: a) 在洪泛过程中接收 (见第 13 章); b) 路由器自己生成 (见第 12.4 节)。从路由器数据库删除一个 LSA: a) 在洪泛过程中被较新的实例所覆盖 (见第 13 章); b) 路由器自己生成了较新的实例 (见第 12.4 节); c) LSA 超时, 并从路由域中被废止 (见第 14 章)。当从数据库中删除 LSA 时, 必须将其同时从所有的邻居连接状态重传列表中删除 (见第 10 章)。

12.3. TOS 表现

为了与以前版本的 OSPF [引用 9] 向后兼容, 在 Router-LSA、Summary-LSA 和 AS-external-LSA 中包含了 TOS 附加信息。OSPF LSA 的 TOS 编码见表 17。该表建立了 OSPF 编码与 IP 包头中 TOS 域 (定义于 [引用 12]) 之间的关系。OSPF 编码为十进制, IP 包头中的 TOS 域按 [引用 12] 的二进制表现。

OSPF 编码	RFC 1349 TOS 值
0	0000 normal service
2	0001 minimize monetary cost
4	0010 maximize reliability
6	0011
8	0100 maximize throughput
10	0101
12	0110
14	0111
16	1000 minimize delay
18	1001
20	1010
22	1011
24	1100
26	1101
28	1110
30	1111

表 17: OSPF 中的 TOS 表现

12.4. 生成 LSA

在给定的 OSPF 区域中, 一台路由器将生成多个 LSA。每台路由器都生成一个 Router-LSA; 如果路由器是区域网络中的 DR, 就生成该网络的 Network-LSA。

ABR 为每个已知的区域间目标生成一个 Summary-LSA; ASBR 为每个已知的 AS 外部目标生成一个 AS-external-LSA。一次只宣告一个目标, 所以一条路径的改变不必洪泛全部的路径。在洪泛过程中, 在一个 LSU 包中可以包含多个 LSA。

例如, 考虑图 6 中的路由器 RT4。这是个连接区域 1 和骨干区域的 ABR。路由器 RT4 生成 5 个不同的 LSA 进入骨干区域 (一个 Router-LSA, 以及为网络 N1-N4 各生成的一个 Summary-LSA)。路由器同时向区域 1 生成 8 个不同的 LSA (一个 Router-LSA, 以及图 7 中显示的七个 Summary-LSA)。

如果 RT4 被选举为网络 N3 的 DR，它还将向区域 1 生成 N3 的 Network-LSA。

同样在该图中，路由器 RT5 将生成 3 个不同的 AS-external-LSA（分别为网络 N12-N14）。假设没有区域被配置为存根区域，这些 LSA 将被洪泛到整个 AS 中。但是，如果区域 3 被配置为存根区域，网络 N12-N14 的 AS-external-LSA 就不会被洪泛进区域 3（见第 3.6 节）。相反，路由器 RT11 将生成默认 Summary-LSA，并洪泛进区域 3（见第 12.4.3 节）。这使区域 3 中的路由器，将其 AS 外部流量发往 RT11。

当生成 LSA 的新实例时，增加其 LS 序号，其 LS 时限设为 0，计算 LS 校验和，将 LSA 加入连接状态数据库并从适当的接口开始洪泛。将 LSA 加入连接状态数据库的细节见第 13.2 节；将新生成的 LSA 洪泛的细节见第 13.3 节。

有十种事件可以导致生成 LSA 的新实例：

（1）路由器自己生成的 LSA，其 LS 时限达到 LSRefreshTime。这时，生成 LSA 的新实例，即使其 LSA 体的内容（除了 LSA 头）完全相同。这将保证周期性的生成所有的 LSA，这种周期性的 LSA 刷新增强了连接状态算法的强壮性。仅仅描述不可到达目标的 LSA 不会被重新刷新，而将被从路由域中废止（见第 14.1 节）。

当 LSA 所描述的内容改变时，生成新的 LSA。然而，不能在时间 MinLSInterval 内连续生成同一 LSA 的两个实例。这意味着可能会延迟至多 MinLSInterval 秒以生成新的实例。下面的事件会导致 LSA 内容的改变。当且仅当内容有所不同时，才会生成 LSA 的新实例。

（2）当接口状态改变（见第 9.1 节），这可能需要生成一个 Router-LSA 的新实例。

（3）当所接入网络的 DR 改变时，生成一个新的 Router-LSA。此外，如果路由器新成为 DR 的话，生成一个新的 Network-LSA；如果路由器不再成为 DR，其原来为该网络生成的任何 Network-LSA 都应当被从路由域中废止（见第 14.1 节）。

（4）其邻居路由器的状态达到 FULL 状态、或不再是 FULL 状态，都将导致生成 Router-LSA 的新实例。此外，如果路由器是该网络的 DR 的话，还需要生成一个新的 Network-LSA。

下面的四种事件仅关联 ABR。

（5）当路由表中的区域内路径改变（增加/删除/修改）时，可能需要向其所接入的区域（可能包含骨干区域）生成新的 Summary-LSA（表示此条路径）。

（6）当路由表中的区域间路径改变（增加/删除/修改），可能需要向其所接入的区域（但“决不”包含骨干区域）生成新的 Summary-LSA（表示此条路径）。

（7）当路由器新接入一个区域。路由器必须为新的区域，生成所有路由表中相关的区域内和区域间路径的 Summary-LSA。具体细节见第 12.4.3 节。

（8）当路由器上配置的虚拟通道状态改变，可能需要向传输区域生成新的 Router-LSA（见第 12.4.1 节中所描述的 Router-LSA 中的 V 位），同时需要向骨干区域生成新的 Network-LSA。

最后两种事件仅关联 ASBR（或形成 ASBR）。

（9）当直接从其他路由协议（如 BGP）获取的外部路径改变时，将导致 ASBR 生成新的 AS-external-LSA。

（10）可能因为重新启动，而使路由器不再成为 ASBR。这将使路由器删除所有其原先生成的 AS-external-LSA。这通过在第 14.1 节中描述的提前老化过程来实现。

下面说明各种类型 LSA 的构成细节，也就是描述其 LSA 体（20 个字节的 LSA 头部后面的内容）。关于 LSA 头部的构造见第 12.1 节。

12.4.1. Router-LSA

路由器为其所接入的每个区域生成 Router-LSA。每个 LSA 描述路由器接入该区域的接口状态。该 LSA 仅在特定的区域内被洪泛，而不是更远。

Router-LSA 的格式见附录 A.4.2。LSA 的前 20 个字节为一般的 LSA 头部，如第 12.1 节所述，Router-LSA 的 LS 类型 = 1。

通过 Router-LSA 中的特定位，路由器说明自己是否为 ABR 或 ASBR（分别通过 B 位和 E 位）。这使得在路由表中保留到达这些类型路由器的路径，并为后面处理 Summary-LSA 和 AS-external-LSA。当路由器接入两个或更多区域时设定 B 位，即使路由器当前没有接入 OSPF 骨干区域。在存根区域中决不能在 Router-LSA 中设置 E 位（ASBR 不能存在于存根区域中）。

此外，当且仅当路由器配置的一条或多条以区域 A 为传输区域的虚拟通道达到完全邻接时，要在区域 A 的 Router-LSA 中设置 V 位。V 位的设定使得区域中的其他路由器能够发现，区域是否支持传输流量（见第 6 章中的 TransitCapability）。

Router-LSA 描述了路由器在区域中运作的连接（及接口或通道）。依据所接入的网络类型，而决定连接类型。每个连接都使用连接标识来识别。连接标识向连接的另一端给出了名称。表 18 汇总了类型值和连接标识。

连接类型	描述	连接标识
1	点对点连接	邻居路由器标识
2	连接到传输网络	DR 的接口地址
3	连接到存根网络	网络 IP 地址
4	虚拟通道	邻居路由器标识

表 18: Router-LSA 中的连接描述

此外，每个连接有连接数据域，给出了此连接的 32 位附加信息。对于传输网络、编号点对点连接和虚拟通道，该域给出了关联接口的 IP 地址（在路由表计算中需要，见第 16.1.1 节）；对于存根网络，该域给出了存根网络的 IP 地址掩码；对于无编号点对点连接，该域被设为无编号接口的 MIB-II [引用 8] 接口索引值。

最后，是该连接的输出值，连接的输出数值可配置。除了到存根网络的连接，输出数值始终不为 0。

为了进一步描述建立连接列表的过程，假设路由器需要为区域 A 建立 Router-LSA。路由器开始检查接口数据结构，对于每个接口，执行下面的步骤：

如果所接入的网络不属于区域 A，不在 LSA 中加入连接，开始检查下一个接口。

如果接口状态为 Down，不增加连接。

如果接口状态为 Loopback，只要不是无编号点对点接口，就增加类型 3 连接（存根网络）。连接标识为接口 IP 地址，连接数据设为掩码 0xffffffff（表示主机路径），其距离值设为 0。

否则，将连接描述按 OSPF 接口类型加入 Router-LSA。点对点接口的连接描述在第 12.4.1.1 节说明；虚拟通道的在第 12.4.1.2 节；广播和 NBMA 接口的在第 12.4.1.3 节；点对多点接口的在第 12.4.1.4 节。

处理完路由器接口后，通过检查属于区域 A 的主机接入列表，将主机连接加入到 Router-LSA 中。主机路径表现为类型 3 连接（存根区域）其连接标识为主机 IP 地址，连接数据为全 1（0xffffffff），输出值为所配置的数值（见附录 C.7）。

12.4.1.1. 描述点对点接口

对点对点网络，按如下在 Router-LSA 中加入一个或多个连接描述：

如果邻居路由器为完全邻接，加入类型 1 连接（点对点）。连接标识应当被设为邻居路由器标识。对于编号点对点网络，连接数据应当被设为接口 IP 地址；对于无编号点对点网络，连接数据应当被设为接口的 MIB-II [引用 8] 索引值。其输出值应当被设为点对点接口的距离。

此外，只要接口状态为“点对点”（而不论邻居路由器的状态），加入类型 3 连接（存根区域）。这一存根连接可能有下面两种格式：

选择 1：假设已知邻居路由器的 IP 地址，设定该类型 3 连接的连接标识为邻居 IP 地址；连接数据为掩码 0xffffffff（表示主机路径），输出值为接口上的设定值 [脚注 15]。

选择 2：如果在点对点连接上设定了子网，设定该类型 3 连接的连接标识为子网的 IP 地址；连接数据为子网掩码，输出值为接口上的设定值 [脚注 16]。

12.4.1.2. 描述广播和 NBMA 接口

对于运作的广播和 NBMA 接口，按如下在 Router-LSA 中加入连接描述：

如果接口状态为 Waiting，加入类型 3 连接（存根网络）。设定连接标识为所接入网络的 IP 地址；连接数据为所接入网络的网络掩码；输出值为接口上的设定值。

不然，所接入网络上会有被选举出的 DR。如果路由器与 DR 完全邻接，或路由器自身为 DR

且与至少一台其他路由器邻接，加入类型 2 连接（传输网络）。设定连接标识为所接入网络中 DR 的 IP 接口地址（可能为路由器自身）；连接数据为路由器自己的 IP 接口地址；输出值为接口上的设定值。否则，按照接口状态为 **Waiting** 加入连接（见上）。

12.4.1.3. 描述虚拟通道

对于虚拟通道，仅当虚拟邻居为完全邻接时，才在 **Router-LSA** 中加入连接描述。这时，加入类型 4 连接（虚拟通道）。设定连接标识为虚拟邻居的路由器标识；连接数据为关联接口的 IP 接口地址；输出值为路由表计算时得出的虚拟通道值（见第 15 章）。

12.4.1.4. 描述点对多点接口

对于运作的点对多点接口，按如下在 **Router-LSA** 中加入一个或多个连接描述：

加入类型 3 连接（存根网络）。设定连接标识为路由器自己的 IP 接口地址；连接数据为掩码 **0xffffffff**（表示主机路径）；输出值为 0。

对于关联接口上每个完全邻接的邻居，另加上类型 1 连接（点对点）。设定连接标识为邻居路由器标识；连接数据为接口 IP 地址；输出值为接口上的设定值。

12.4.1.5. Router-LSA 示例

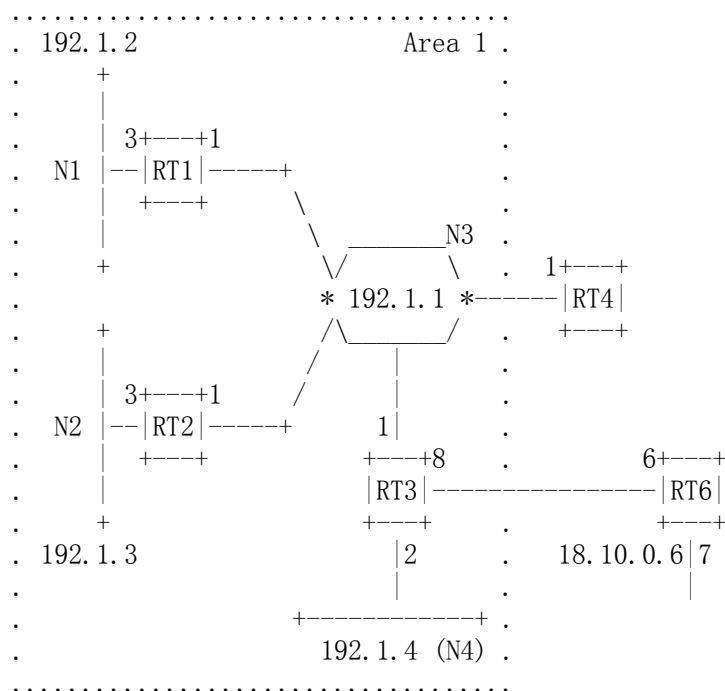


图 15：带 IP 地址显示的区域 1

考虑图 6 中路由器 RT3 所生成的 **Router-LSA**。在图 15 中显示带网络地址的 RT3 所在区域（区域 1）。假设 RT3 的接口地址最后字节均为 3，即其接口地址为 192.1.1.3 和 192.1.4.3，并且其他路由器的地址也这样安排。此外，假设所有的接口都正常工作，路由器标识被设为最小 IP 接口地址。

RT3 分别为区域 1 和骨干区域生成两个 Router-LSA。假设路由器 RT4 被网络 192.1.1.0 选举为 DR，那么 RT3 在区域 1 中的 Router-LSA 如下：显示出 RT3 在区域 1 中有两个连接，第一个是传输网络 192.1.1.0，第二个是存根网络 192.1.4.0。注意，传输网络是由 DR 的 IP 接口地址来标识的（即连接标识 = 192.1.1.4，是 192.1.1.0 网络中的 DR RT4 的 IP 地址）。同时，RT3 标记出自己为 ABR。

路由器 RT3 在区域 1 中的 Router-LSA:

LS 时限 = 0	; 在生成时总是如此
选项 = E 位	
LS 类型 = 1	; 说明为 Router-LSA
LS 标识 = 192.1.1.3	; RT3 的路由器标识
宣告路由器 = 192.1.1.3	; RT3 的路由器标识
E 位 = 0	; 不是 ASBR
B 位 = 1	; 是 ABR
连接数量 = 2	
连接标识 = 192.1.1.4	; DR 的 IP 地址
连接数据 = 192.1.1.3	; RT3 的 IP 接口地址
类型 = 2	; 连接到传输网络
TOS 值 = 0	
输出值 = 1	
连接标识 = 192.1.4.0	; IP 网络地址
连接数据 = 0xfffff00	; 网络掩码
类型 = 3	; 连接到存根网络
TOS 值 = 0	
输出值 = 2	

下面是 RT3 在骨干区域中的 Router-LSA。显示出 RT3 通过与 RT6 的无编号点对点连接和骨干区域相连。RT3 再次标记出自己为 ABR。

路由器 RT3 在骨干区域中的 Router-LSA:

LS 时限 = 0	; 在生成时总是如此
选项 = E 位	
LS 类型 = 1	; 说明为 Router-LSA
LS 标识 = 192.1.1.3	; RT3 的路由器标识
宣告路由器 = 192.1.1.3	; RT3 的路由器标识
E 位 = 0	; 不是 ASBR
B 位 = 1	; 是 ABR
连接数量 = 1	
连接标识 = 18.10.0.6	; 邻居的 IP 地址
连接数据 = 0.0.0.3	; 点对点接口在 MIB-II 中的索引值
类型 = 1	; 连接到路由器
TOS 值 = 0	
输出值 = 8	

12.4.2. Network-LSA

为每个传输的广播或 NBMA 网络生成 Network-LSA（传输网络是指接入了两台或更多路由器的网络）。Network-LSA 描述了所有接入网络的路由器。

网络上的 DR 生成这个 LSA。仅当 DR 与网络上至少一台路由器完全邻接后，才生成该 LSA。该 Network-LSA 仅在包含该传输网络的区域中洪泛，而不是更远。Network-LSA 中包含了与 DR 完全

邻接的邻居列表，各台路由器由其 OSPF 路由器标识来识别。DR 自己也包含在列表中。

Network-LSA 的 LS 标识是 DR 的接口 IP 地址。将该值与网络地址掩码（也包含在该 Network-LSA 中）进行运算可得出网络 IP 地址。

当路由器不再是网络上的 DR，就要删除以前生成的 Network-LSA。该 LSA 不再用于路由表计算。这通过将 LSA 提前老化到 MaxAge 并重新洪泛而实现（见第 14.1 节）。此外还有一种罕见的情况，当路由器标识改变时，必须删除以旧路由器标识所生成的 Network-LSA。该 Network-LSA 的 LS 标识等于某一接口 IP 地址，但宣告路由器不为路由器标识，而路由器不可能知道以前的路由器标识（更多细节见第 13.4 节）。

12.4.2.1. Network-LSA 示例

再次考虑图 6 中的区域配置。区域 1 中的网络 N3、区域 2 中的 N6 和 N8 以及区域 3 中的 N9 需要生成 Network-LSA。假设 RT4 是网络 N3 上选举出的 DR，下面的 Network-LSA 就是 RT4 为网络 N3 所生成的（地址设置见图 15）。

网络 3 的 Network-LSA:

LS 时限 = 0	; 在生成时总是如此
选项 = E 位	
LS 类型 = 2	; 说明为 Network-LSA
LS 标识 = 192.1.1.4	; DR 的接口 IP 地址
宣告路由器 = 192.1.1.4	; RT4 的路由器标识
网络掩码 = 0xfffff00	
接入路由器 = 192.1.1.4	; 路由器标识
接入路由器 = 192.1.1.1	; 路由器标识
接入路由器 = 192.1.1.2	; 路由器标识
接入路由器 = 192.1.1.3	; 路由器标识

12.4.3. Summary-LSA

Summary-LSA 所描述的目标可能是一个 IP 网络、一台 ASBR 或一个 IP 地址范围。Summary-LSA 仅在一个区域内洪泛，它所描述的目标是在区域之外，但仍属于这个 AS。

Summary-LSA 由 ABR 生成的。按下面描述的算法检查路由表结构（见第 11 章），以精确地向区域宣告汇总路径。注意，向骨干区域只宣告区域内路径；而向其他区域要宣告区域内和区域间路径。

为了决定向所接入的区域 A 宣告哪些路径，按下面的过程处理每条路由表项。切记，每条路由表项描述了到达特定目标的一系列等值路径：

只有目标类型为网络或 ASBR 时，才在 Summary-LSA 中被宣告。如果路由表项的目标类型为 ABR 的话，检查下一项。

决不在 Summary-LSA 中宣告 AS 外部路径，如果路由表项中有类型 1 或 2 的外部路径，检查下一项。

否则，如果该条路径与区域 A 自身有关联的话，不为该路径生成 Summary-LSA [脚注 17]。

否则，如果该条路径的下一跳地址与区域 A 有关联，不为该路径生成 Summary-LSA [脚注 18]。这在逻辑上与 DV 算法的水平分割相同。

否则，如果路由表中的距离值等于或超过 LSInfinity，不为该路径生成 Summary-LSA。

否则，如果目标类型为 ASBR，当且仅当路由表项描述了到达 ASBR 的最佳路径时（见第 16.4 节第 3 步），生成类型 4 的 Summary-LSA。该 LSA 的 LS 标识等于 ASBR 的路由器标识，输出值等于路由表项中的距离值。注意，如果区域 A 为存根区域的话，不生成该 LSA。

否则，目标类型为网络。如果是区域间路径，为目标生成类型 3 的 Summary-LSA。该 LSA 的 LS 标识等于网络地址（如果需要，LS 标识中可能有一位或多位主机位，细节见附录 E），输出值等于路由表项中的距离值。

剩下的一种情况是网络的区域内路径。这意味着该网络包含在路由器直接接入的区域内。通常，这些信息出现在 Summary-LSA 之前会被汇总。记住，区域配置了地址范围的列表，每个列表都由 [地址、掩码] 对加上 Advertise 或 DoNotAdvertise 的状态，通常为每个范围生成一个类型 3 的 Summary-LSA。当地址范围的状态为 Advertise 时，所生成的 Summary-LSA 其 LS 标识被设置为范围地址（如果需要，LS 标识可能有一位或多位主机位，细节见附录 E），其距离值等于其组成部分中最大的距离值。当范围的状态为 DoNotAdvertise 是，Summary-LSA 被抑制，该部分的网络对其他区域隐藏。

默认情况下，如果没有明确的配置地址范围，所生成的类型 3 Summary-LSA 的 LS 标识等于网络地址（如果需要，LS 标识可能有一位或多位主机位，细节见附录 E），其输出值等于路由表中该网络的距离值。

如果一个区域能够传送传输流量（即 TransitCapability 被设为 TRUE），关于骨干网络的路由信息在汇总进区域前，不应当被汇聚。同样也不应当抑制向传输区域宣告骨干网络。换句话说，在向传输区域生成 Summary-LSA 时，应当忽略骨干区域上所配置的范围。

如果路由器宣告的 Summary-LSA 中有网络变为不可达，路由器必须将 LSA 的时限设为 MaxAge 并重新洪泛（见第 14.1 节），以废止该 LSA。同样，如果目标仍可达，但不再按照以前的方式，（也就是说，以前是一条区域间路径，现在使用非骨干的区域内路径，就应当不再向骨干区域宣告），该 LSA 应当从路由域中废止。

12.4.3.1. 向存根区域生成 Summary-LSA

当区域 A 是一个存根区域时，对于第 12.4.3 节所描述的算法有另一个选项。连接到存根区域的 ABR 可以按照第 12.4.3 节的算法向区域生成 Summary-LSA，或者在一定控制下选择仅生成一部分 Summary-LSA。生成的 LSA 越少，存根区域的连接状态数据库就越小，可以进一步减少路由器所需要的资源。当然，忽略的 LSA 可能导致区域间路径虽可以使用，但不是最佳。

如第 12.4.3 节所述，决不向存根区域生成类型 4 Summary-LSA（ASBR-Summary-LSA）。

在存根区域，各台 ABR 不再输入外部路径，而是生成“default Summary-LSA”，其输出值是可配

置（按区域）的参数 `StubDefaultCost`。注意，存根区域中的各个 ABR，其 `StubDefaultCost` 不一定配置相同。

12.4.3.2. Summary-LSA 示例

再次考虑图 6 中的区域配置。路由器 RT3、RT4、RT7、RT10 和 RT11 都是 ABR，都能够生成 Summary-LSA。考虑特定路由器 RT4，其路由表作为例子在第 11.3 节中计算。RT4 分别为骨干区域和区域 1 生成 Summary-LSA。对于骨干区域，路由器 RT4 分别为网络 N1-N4 生成 Summary-LSA；对于区域 1，RT4 分别为网络 N6-N8 和 ASBR RT5、RT7 生成 Summary-LSA；还将 la 和 lb 的主机路径汇总进一个 Summary-LSA 中；最后将网络 N9、N10、N11 和主机 H1 的路径在一个 Summary-LSA 中宣告。汇总过程是由路由器 RT11 来生成的。

这些 LSA 在图 7 和图 8 中说明。下面说明由路由器 RT4 生成的两个 Summary-LSA。带网络和路由器的 IP 地址配置见图 15。

网络 N1 的 Summary-LSA，由路由器 RT4 为骨干区域生成：

LS 时限 = 0	； 在生成时总是如此
选项 = E 位	
LS 类型 = 3	； 说明为类型 3 Summary-LSA
LS 标识 = 192.1.2.0	； 网络 N1 的 IP 地址
宣告路由器 = 192.1.1.4	； RT4 的路由器标识
输出值 = 4	

路由器 ASBR RT7 的 Summary-LSA，由路由器 RT4 为区域 1 生成：

LS 时限 = 0	； 在生成时总是如此
选项 = E 位	
LS 类型 = 4	； 说明为类型 4 Summary-LSA
LS 标识 = RT7 的路由器标识	
宣告路由器 = 192.1.1.4	； RT4 的路由器标识
输出值 = 14	

12.4.4. AS-external-LSA

AS-external-LSA 描述了到达 AS 外部目标的路径。大多数 AS-external-LSA 描述了到达特定外部目标的路径，这时 LSA 的 LS 标识被设为目标网络的 IP 地址（如果需要，LS 标识可能加上一位或多位主机位，细节见附录 E）。然而，AS 的默认路径可以被描述为 LS 标识设为 `DefaultDestination`（0.0.0.0）的 AS-external-LSA。AS-external-LSA 是由 ASBR 所生成。不论是从其他路由协议（如 BGP），或通过配置信息得到，ASBR 为每条外部路径生成单独的一个 AS-external-LSA。

AS-external-LSA 是唯一在整个 AS 中洪泛的 LSA，其他类型的 LSA 都只在特定区域内洪泛。但是，AS-external-LSA 不被洪泛进存根区域（见第 3.6 节）。这将减小存根区域内部路由器的连接状态数据库。

所宣告的外部路径有两种类型的距离值。类型 1 的距离值可以与连接状态距离值相比较；类型 2 的

距离值被假设大于任何 AS 内部路径的距离值。

如果由 AS-external-LSA 宣告的目标变为不可达，路由器必须将该 LSA 从路由域中废止，这通过将 LS 时限设为 MaxAge 并重新洪泛而实现（见第 14.1 节）。

12.4.4.1. AS-external-LSA 示例

再次考虑图 6 中的 AS。有两台 ASBR：RT5 和 RT7。路由器 RT5 分别为网络 N12-N14 生成三条 AS-external-LSA，；路由器 RT7 分别为网络 N12 和 N15 生成两条 AS-external-LSA，。假设 RT7 是从 BGP 得到 N12 的路径，并希望按类型 2 距离值向 AS 内宣告。RT7 将为 N12 生成如下的 LSA。

网络 12 的 AS-external-LSA，由路由器 RT7 生成

LS 时限 = 0； 在生成时总是如此

选项 = E 位

LS 类型 = 5； 说明为 AS-external-LSA

LS 标识 = N12 的网络 IP 地址

宣告路由器 = RT7 的路由器标识

E 位 = 1； 类型 2 距离值

输出值 = 2

转发地址 = 0.0.0.0

在上例中，转发地址域被设为 0.0.0.0，表示到达该外部目标的包，应当由宣告的 OSPF 路由器（RT7）来转发。但并不总是这样。考虑图 16 中的例子。有三台 OSPF 路由器（RTA、RTB 和 RTC）连接到同一网络上，只有一台路由器 RTA 与非 OSPF 路由器 RTX 交换 BGP 信息。RTA 必须为从 RTX 得到的目标生成 AS-external-LSA。通过使用 AS-external-LSA 中的转发地址域，RTA 使到达那些目标的包直接转发到 RTX，如果没有这一功能，从路由器 RTB 和 RTC 到达目标将经过额外一跳。

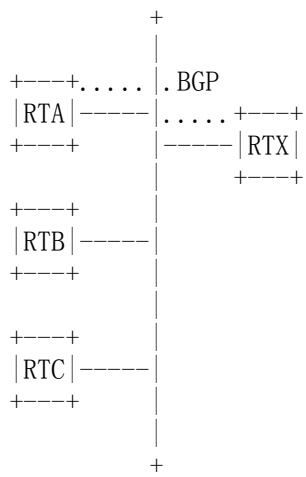


图 16：转发地址示例

注意，当转发地址为非零时，应当指向属于另一个 AS 的路由器。

转发地址同样可以用于默认路径。例如图 16 中，RTA 可能希望所有到达外部目标的包都通过其 BGP 伙伴 RTX。这样的 AS-external-LSA 在下面列出，注意其 LS 标识被设为 DefaultDestination。

通过 RTX 转发包的默认路径，由路由器 RTA 生成

LS 时限 = 0	； 在生成时总是如此
选项 = E 位	
LS 类型 = 5	； 说明为 AS-external-LSA
LS 标识 = DefaultDestination	； 默认路径
宣告路由器 = RTA 的路由器标识	
E 位 = 1	； 类型 2 距离值
输出值 = 1	
转发地址 = RTX 的 IP 地址	

在图 16 中，假设 RTA 和 RTB 都与 RTX 交换 BGP 信息。这时，RTA 和 RTB 将生成一系列相同的 AS-external-LSA。如果使用了同样的距离值，那么这些说明了同样目标和同样转发地址（RTX）的 LSA 将完全等价，这是明显的重复。如果 RTA 或 RTB 中只有一台生成这一系列 AS-external-LSA，路径还是相同，但减少了连接状态数据库的大小。但是，这必须明确地定义由哪台路由器来生成这些 LSA（如果没有的话，生成 LSA 的路由器将会来回颠簸）。因此建立下面的规则：如果两台路由器，相互可以到达，生成了功能上相同的 AS-external-LSA（即相同目标、距离值和非零的转发地址），使用具有最大 OSPF 路由器标识的路由器所产生的 LSA。较小路由器标识的路由器按照第 14.1 节的描述删除其 LSA。

13. 洪泛过程

LSU 包提供了洪泛 LSA 的机制。在一个 LSU 包中可能包含多个不同的 LSA，并将 LSA 洪泛到距离其起源更远的一跳。为了使洪泛过程可靠，每个 LSA 必须被分别确认。确认在 LSAck 包中传送，一个包中也可以包含多个独立的确认。

当接收到 LSU 包时，开始洪泛过程。在处理洪泛之前，要对所接收到的包进行很多一致性检查（见第 8.2 节）。特别的，LSU 包要与特定的邻居、及特定的区域相关联。如果所关联的邻居状态没有达到 Exchange，该包将被丢弃而不作进一步操作。

除了 AS-external-LSA，其他所有类型的 LSA 都与特定的区域相关联。但是 LSA 中并没有包含区域。必须从 LSU 包的头部推算出 LSA 的区域。

对于每个包含在 LSU 包中的 LSA，进行下面的步骤：

（1）确认 LSA 的 LS 校验和。如果校验和无效，丢弃该 LSA，并从 LSU 包中取得下一个。

（2）检查 LSA 的 LS 类型。如果 LS 类型为未知，丢弃该 LSA，并从 LSU 包中取得下一个。本规范定义的 LS 类型为 1-5（见第 4.3 节）。

（3）否则，如果是一个 AS-external-LSA（LS 类型= 5），而该区域被配置为存根区域，丢弃该 LSA，并从 LSU 包中取得下一个。AS-external-LSA 不被洪泛进存根区域（见第 3.6 节）。

（4）否则，如果 LSA 的 LS 时限等于 MaxAge，而且路由器的连接状态数据库中没有该 LSA 的实例，而且路由器的邻居都不处于 Exchange 或 Loading 状态。就执行下面操作：a）通过发送一个 LSAck 包到发送的邻居（见第 13.5 节）来确认收到该 LSA，并且 b）丢弃该 LSA，并从 LSU 包中取得下一个（如果存在）。

（5）否则，在路由器当前的连接状态数据库中查找该 LSA 的实例。如果没有找到数据库中的副本，或所接收的 LSA 比数据库副本新（判断哪个 LSA 较新的方法见第 13.1 节），执行下面的步骤：

（a）如果已经有一个数据库副本，而且是在 MinLSArrival 秒内通过洪泛而加入数据库的，丢弃新的 LSA（没有确认）并检查 LSU 包中的下一个 LSA（如果存在）。

（b）否则，立即将新 LSA 洪泛出路由器的某些接口（见第 13.3 节）。在某些情况下（如，接收接口的状态为 DR，而且 LSA 不是从 BDR 发出），该 LSA 将被从接收接口洪泛出去。这一事件需要在后面的确认过程中被注意。

（c）将当前数据库中的副本，从所有的邻居连接状态重传列表中删除。

（d）将新的 LSA 加入连接状态数据库（取代当前数据库的副本），这可能导致按调度计算路由表。此外，将新的 LSA 加上当前时间的时间戳（也就是接收时间）。在经过 MinLSArrival 秒之前，新加入的 LSA 不能被洪泛过程所覆盖。加入 LSA 的过程在后面第 13.2 节中描述。

(e) 也许需要从接收接口发送 LSAck 包以确认所收到的 LSA。这在第 13.5 节说明。

(f) 如果这个新的 LSA 是由路由器自身所生成的（即被作为自生成 LSA），路由器执行特殊的操作，或许更新该 LSA，或将其从路由域中删除。自生成 LSA 的删除以及随后操作的描述，见第 13.4 节。

(6) 否则，如果该 LSA 的实例正在邻居的连接状态请求列表上，产生数据库交换过程的错误。生成该邻居的 BadLSReq 事件，重新启动数据库交换过程，并停止处理 LSU 包。

(7) 否则，如果接收的 LSA 与数据库副本为同一实例（没有哪个较新），应当完成下面两步：

(a) 如果 LSA 在所接收邻居的连接状态重传列表上，表示路由器自身正期待着这一 LSA 的确认。路由器可以将这一 LSA 作为确认，并将其从连接状态重传列表中去除。这被称为“隐含确认”，这需要在后面的确认过程中注意（见第 13.5 节）。

(b) 也许需要从接收接口发送 LSAck 包以确认所收到的 LSA。这在第 13.5 节说明。

(8) 否则，数据库中的副本较近。如果数据库副本的 LS 时限等于 MaxAge，且 LS 序号等于 MaxSequenceNumber，则丢弃接收到的 LSA 而不作确认（这时，LS 序号正在回绕，在加入 LSA 新实例前必须删除 MaxSequenceNumber 的 LSA）。否则，只要数据库中的副本没有在最近 MinLSArrival 秒中被 LSU 包发送，就用一个 LSU 包发回给邻居。这个 LSU 包应当被直接发往邻居，而不是将数据库中 LSA 的副本放到邻居的邻接状态重传列表中，对于所接收到的 LSA 实例（较旧的）不作确认。

13.1. 判定较新的 LSA

当路由器遇到一个 LSA 的两个实例时，它必须判定哪一个较新。这在收到一个在数据库中有副本的 LSA 时发生。在形成邻接的数据库交换过程中也必须比较。

LSA 是通过 LS 类型、LS 标识和宣告路由器来识别的。同一 LSA 的两个实例是通过 LS 序号、LS 时限和 LS 校验和来判断哪个较新。

较大 LS 序号的 LSA 较新。见第 12.1.6 节对 LS 序号空间的解释。如果两个实例的序号相同：

则，如果两个实例的 LS 校验和不同，具有较大校验和（按 16 位无符号整数）的实例较新。

否则，如果其中一个实例的 LS 时限为 MaxAge，则这个实例为较新。

否则，如果两个实例 LS 时限的差异大于 MaxAgeDiff，较小时限（较近生成）的实例为较新。

否则，两个实例被认为相同。

13.2. 将 LSA 加入数据库

不论是因为洪泛的结果或是自生成了新的 LSA，将新的 LSA 加入数据库将导致重新计算 OSPF 路由表。新 LSA 所包含的内容应当与旧的实例（如果存在的话）不同，如果没有不同，就不需要重新计算路由表。在比较新旧 LSA 时，要考虑所有下面的组成部分：

LSA 选项域的改变。

一个 LSA 实例的 LS 时限为 MaxAge 而另一个不是。

LSA 头部中的长度域改变。

LSA 体改变（除了 20 个字节 LSA 头部的任何部分），但不包括 LS 序号和 LS 校验和的改变。

如果包含内容有改变，根据新 LSA 的 LSA 类型域，必须重新计算下面的路由表：

Router-LSA 和 Network-LSA:

必须从每个区域的最短路径树开始，重新计算整个路由表（而不仅仅是改变了连接状态数据库的区域）。最短路径计算不局限于一个区域的原因是，ASBR 可能接入多个区域。在当前提供最短路径的区域中改变路径，可能会导致使用另一个区域中的区域内路径。[脚注 19]

Summary-LSA:

必须重新计算由 Summary-LSA 所说明的路径（见第 16.5 节）。如果目标为 ASBR，还必须检查所有的 AS-external-LSA。

AS-external-LSA:

必须重新计算由 AS-external-LSA 所说明的路径（见第 16.6 节）。

同时，在加入新的 LSA 时，必须从数据库中删除旧的 LSA。旧的实例必须同时从所有邻居的连接状态重传列表中删除（见第 10 章）。

13.3. 洪泛过程的下一步操作

当接收到较新的 LSA 后，必须将其向路由器的一些接口洪泛出去。本节描述洪泛过程的第二部分（第一部分在第 13 章描述），也就是说，选择输出接口，并将 LSA 加到邻居的连接状态重传列表上。同时，这部分的洪泛过程还包括维护邻居的连接状态请求列表。

本节的描述同样适用于由路由器自己生成的 LSA（见第 12.4 节）。这些 LSA 需要按本节的描述进行洪泛（这些过程在第 13 章没有涉及，因为这些 LSA 不是从邻居所接收，而且不需要确认）。

根据 LSA 的 LS 类型，LSA 可以被洪泛到特定的接口上。这些接口，按照下面的定义，叫做合格接口。

AS-external-LSA（LS 类型 = 5）:

除了存根区域，AS-external-LSA 将被洪泛到整个 AS 中（见第 3.6 节）。合格接口就是除了虚拟通道和接入存根区域的所有其他路由器接口。

所有其他 LS 类型:

所有其他的类型都和一个区域（区域 A）相关。合格接口就是接入区域 A 的所有接口。如果区域 A 为骨干区域，也包括所有的虚拟通道。

连接状态数据库必须在所有合格接口上的邻接之间保持同步。这通过在每个合格接口上执行下面的步骤而完成。如果所接入的邻居很可能已经接收了这个 LSA，该过程可能会不在特定接口将 LSA 洪泛出去。但是，为了洪泛过程必须保证邻居确实接收到了该 LSA，所以这个 LSA 仍旧被加到每个邻接的连接状态重传列表上。对于每个合格接口：

（1）检查接口上的各个邻居，判断是否必须接收新的 LSA，对每个邻居执行下面的步骤：

（a）如果邻居状态小于 **Exchange**，它不参与洪泛，检查下一个邻居。

（b）如果邻接还没有完全（邻居状态为 **Exchange** 或 **Loading**），检查邻接所关联的连接状态请求列表。如果存在有该 LSA 的实例，表示邻居已经有了该 LSA。将新的 LSA 与邻居的 LSA 作如下比较：

如果新的 LSA 较老，检查下一个邻居。

如果两个副本为相同实例，删除连接状态请求列表中的 LSA，检查下一个邻居。

[脚注 20]

否则，如果新的 LSA 较新，删除连接状态请求列表中的 LSA。

（c）如果新的 LSA 是从该邻居所接收，检查下一个邻居。

（d）这时，如果不能肯定邻居有 LSA 的最新实例，将新的 LSA 加到邻接的连接状态重传列表中。为了确保洪泛过程的可靠；LSA 应当按间隔重传，直到从邻居收到确认。

（2）路由器必须决定是否将新的 LSA 洪泛出接口。如果在上一步中，“没有”向连接状态重传列表加入任何 LSA，就不需要将 LSA 洪泛出接口。检查下一个接口。

（3）如果 LSA 是由该接口接收，且是从 DR 或 BDR 接收到的，说明其他邻居都已经接收到该 LSA。检查下一个接口。

（4）如果 LSA 是由该接口接收，且接口状态为 **Backup**（路由器是 BDR），检查下一个接口。DR 必须在该接口上进行洪泛。但是，如果 DR 失效，该路由器（即 BDR）将最终重传更新。

（5）如果到达这步，接口必须洪泛该 LSA。发送一个 LSU 包（包含新 LSA）出接口。当复制该 LSA 时，其 LS 时限必须增加 **InfTransDelay**（必须 > 0）（直到 LS 时限域达到 **MaxAge**）。

在广播网络上，LSU 包是多播。LSU 包的目标地址取决于接口状态。如果接口状态是 DR 或 BDR，使用地址 **AllSPFRouters**；否则，使用地址 **AllDRouters**。

在非广播网络上，各个 LSU 包按单播分别发送到各个邻接的邻居（即状态达到 **Exchange** 或更高）。

包的目标地址为邻居的 IP 地址。

13.4. 接收自生成的 LSA

在洪泛过程中接收到自生成的 LSA 是很正常的。判断自生成 LSA 是根据 1) LSA 的宣告路由器等于自己的路由器标识；或 2) 对于 Network-LSA，其 LS 标识等于路由器的某一接口 IP 地址。

但是，如果所接收到的自生成 LSA 比当前生成的实例还要新，路由器必须作出特殊动作。接收到这种情况，是因为路由器上次重新启动前所生成的 LSA 仍旧存在路由域中。在大多数情况下，路由器必须生成一个新的 LSA 实例，并将其 LS 序号超过所接收到的 LS 序号。

有可能路由器不再希望生成所收到的 LSA。可能的例子有：1) LSA 为 Summary-LSA 或 AS-external-LSA，而路由器不再有一条（宣告的）路径到达该目标；2) LSA 为 Network-LSA，而路由器不再是该网络的 DR；或 3) LSA 为 Network-LSA，其 LS 标识是路由器自己的接口 IP 地址，而宣告路由器不等于自己的路由器标识（最后这种情况很少发生，这表示路由器在生成上个 LSA 后改变了路由器标识）。所有这些情况时，不应当更新 LSA，而应当将接收到的 LSA 中 LS 时限域增加到 MaxAge，并重新洪泛，而从路由域中废止。

13.5. 发送连接状态确认包（LSAck 包）

每一个接收到的 LSA 必须被确认。这通过发送 LSAck 包来实现。虽然可以通过发送 LSU 包来隐含地实现确认（见第 13 章的 7a 步骤）。

多个确认可以被组合在一个 LSAck 包中。这样的包被从接收到 LSA 的接口上发出。有两种发送包的方法：延迟并按间隔时间、或直接发送到特定邻居。具体的确认策略取决于接收到 LSA 的情况。

可以用延迟发送的确认包完成以下功能：1) 可以将多个确认装入一个 LSAck 包中；2) 可以使用一个 LSAck 包来一次向多个邻居表明确认（使用多播）；以及 3) 当多个路由器接入同一网络时，使 LSAck 包的发送时间随机化。路由器发送确认的固定延迟时间必须很短（小于 RxmtInterval），不然会引起不必要的重传。

收到重复的 LSA 时，向特定邻居发送直接确认。当收到重复时，立即发送直接确认。在多路复用网络上，确认被直接发送到邻居的 IP 地址。

发送 LSAck 包的精确过程在表 19 中描述。收到 LSA 的情况在左列说明。确认操作的说明在右边两行中。具体的操作取决于接口的状态，在 Backup 状态的接口，其操作与其他状态下的接口不同。延迟确认必须向关联接口上的所有邻接路由器发送。在广播网络上，这通过使用多播发送 LSAck 包来实现。所使用的目标 IP 地址取决于接口状态。如果接口状态为 DR 或 Backup，使用地址 AllSPFRouters；在其他状态时，使用 AllDRouters。在非广播网络上，延迟 LSAck 包必须以单播分别发送到每个邻接（即邻居状态 \geq Exchange）。

可以用一个例子来说明使用多播来发送上述包的原因。考虑图 15 描述的网络配置。假设 RT4 被选举为网络 N3 的 DR，而 RT3 为 BDR。当路由器 RT4 洪泛一个新的 LSA 到网络 N3 时，会被路由器 RT1、RT2 和 RT3 收到。这些路由器不会将 LSA 洪泛回网络 N3，但必须确保他们的连接状态数据库与邻接的邻居同步。所以 RT1、RT2 和 RT4 希望见到 RT3 的确认，同样 RT4 和 RT3 都希望

见到 RT1 和 RT2 的确认。所以最好的方法就是以多播发送确认。

BDR 发送确认的逻辑与其他状态不同，是因为在洪泛 LSA 时功能的不同（见第 13.3 节，步骤 4）。

情况	Backup 状态下操作	其他状态下操作
LSA 被从所接收的接口洪泛（见第 13 章，步骤 5b）	不发送确认	不发送确认
LSA 比数据库中的副本要近，但不从所接收的接口洪泛	如果是从 DR 接收到的宣告，发送延迟确认。否则无操作	发送延迟确认
LSA 重复，并被作为隐含确认（见第 13 章，步骤 7a）	如果是从 DR 接收到的宣告，发送延迟确认。否则无操作	不发送确认
LSA 重复，但不被作为隐含确认	发送直接确认	发送直接确认
LSA 的 LS 时限等于 MaxAge，而且该 LSA 没有存在于数据库中的实例，并且路由器的邻居都不处于 Exchange 或 Loading 状态（见第 13 章，步骤 4）	发送直接确认	发送直接确认

表 19：发送连接状态确认（LSAck）

13.6. 重传 LSA

通过将 LSA 加入连接状态重传列表上，来向邻接洪泛 LSA。为了确保洪泛的可靠性，这些 LSA 将被重传直至收到确认。重传的间隔为 RxmtInterval，可按接口配置。如果在一个接口上设置的太短，会造成不必要的重传；如果在一个接口上设置的太长，在丢包时就会影响洪泛的速度。

在一个 LSU 包中可以包含多个重传的 LSA。当 LSA 重传时，只能发送一个 LSU 包可以包含的 LSA。当收到部分 LSA 确认，或下一重传时间到达时，可以重传另一个包。

重传的 LSU 包始终被直接发送到邻居。在多路复用网络上，这意味着重传直接发送到邻居的 IP 地址。当将 LSA 复制到输出的 LSU 包时，其 LS 时限必须增加 InfTransDelay（必须 > 0）（直到 LS 时限到达最大值 MaxAge）。

如果邻接路由器断开，仍将重传到 Hello 包取消邻接。邻接取消后，连接状态重传列表将被清除。

13.7. 接收连接状态确认包（LSAck 包）

在处理洪泛之前，要对所接收到的 LSAck 包进行很多一致性检查。特别的，需要与特定的邻居相关联。如果所关联的邻居状态小于 Exchange，则丢弃该 LSAck 包。

否则，对于 LSAck 包中的每个确认，执行下面的步骤：

该 LSA 的确认是否有一个实例在邻居的连接状态重传列表中？如果没有，检查下一个确认。

否则，如果确认所对应的实例包含在列表中，将其从列表中删除，并检查下一个确认。

记录有问题的确认，并检查下一个确认。

14. 老化连接状态数据库

每个 LSA 都有 LS 时限域。LS 时限表现为秒数。当 LSA 被保存在路由器的数据库中时，LS 时限被不断增加；同样，当 LSU 包被洪泛出特定接口时，LSA 的 LS 时限也被增加 `InfTransDelay`。

LSA 的 LS 时限决不超过 `MaxAge`。具有 `MaxAge` 的 LSA 不被用于路由表计算。当路由器老化连接状态数据库时，LSA 的 LS 时限可能会达到 `MaxAge` [脚注 21]。这时，路由器需要将其从路由域中废止。这通过将达到 `MaxAge` 的 LSA，作为新生成的 LSA 洪泛出特定的接口而实现。（见第 13.3 节）

当在形成邻接的过程中生成数据库汇总列表时，连接状态数据库中达到 `MaxAge` 的 LSA 将被加到邻居的连接状态重传列表，而不是邻居的数据库汇总列表。更多细节见第 10.3 节。

当同时满足下两个条件时，达到 `MaxAge` 的 LSA 必须立即从路由器的连接状态数据库中删除：a) 不存在任何邻居连接状态重传列表；b) 没有邻居处于 `Exchange` 或 `Loading` 状态。

在老化连接状态数据库的过程中，当 LSA 的 LS 时限到达某些 `CheckAge` 时，应当校验 LS 校验和。如果 LS 校验和不正确，就可以发现程序或内存的错误，在最坏的情况下，路由器应当重新启动。

14.1. 提前老化 LSA

不改变 LS 序号而将 LS 时限设为 `MaxAge`，并重新洪泛，可以将 LSA 从路由域中废止。这一过程与 LS 时限正常到达 `MaxAge` 而废止是一样的（见第 14 章）。达到 `MaxAge` 的 LSA 必须立即从路由器的连接状态数据库中删除，当 a) 不存在任何邻居连接状态重传列表；以及 b) 没有邻居处于 `Exchange` 或 `Loading` 状态。我们将设置 LS 时限为 `MaxAge` 称为提前老化。

在自生成 LSA 进行回绕时使用提前老化。这时，必须提前老化当前的 LSA 实例（其 LS 序号为 `MaxSequenceNumber`），并在生成新的序号为 `InitialSequenceNumber` 的实例前将其删除。更多细节见第 12.1.6 节。

在下面情况也使用提前老化，例如某条路由器宣告的外部路径不再可以到达。在这时，路由器可以通过提前老化来将 `AS-external-LSA` 从路由域中废止。这比另一种方法：生成距离为 `LSInfinity` 的新 LSA 要好。提前老化也被用于在洪泛过程中收到意外的自生成 LSA（见第 13.4 节）。

路由器只能提前老化自生成的 LSA，而不能提前老化其他路由器生成的 LSA。当 1) LSA 的宣告路由器等于自己的路由器标识，或 2) `Network-LSA` 的 LS 标识等于路由器的某个接口 IP 地址时，被认为是路由器自生成的 LSA。

15. 虚拟通道

骨干区域（区域标识=0.0.0.0）不能被分割，不然 AS 中的某些区域将变为不可达。为了建立/维持骨干区域的连通性，可以在非骨干区域配置虚拟通道。虚拟通道可以连接物理上分割的骨干区域，虚拟通道的两个端点都是 ABR，虚拟通道必须配置在两台路由器上。在一个端点上的配置信息包括另一个端点（另一个 ABR），以及两台路由器共同接入的非骨干区域（被称为传输区域）。虚拟通道不能配置在存根区域内（见第 3.6 节）。

虚拟通道被视为属于骨干区域内，连接两台 ABR 的无编号点对点网络。在虚拟通道上会试图建立邻接。当邻接建立后，在骨干区域的 Router-LSA 会包含虚拟通道，属于骨干区域的 OSPF 包也会被传送到邻接。这种邻接在本文档中被称为“虚拟邻接”。

每台端点路由器中，虚拟通道的距离值和存在性，是通过查找路由表中另一端点路由器而得到的（该项所关联的区域必须被配置为传输区域）。该项被称为虚拟通道的相关路由表项。当对应路由表项可以到达时，生成虚拟通道的 InterfaceUp 事件；相反的，当路由表项不可达时，生成 InterfaceDown 事件。换句话说，虚拟通道是否存在，取决于两个端点间是否存在通过传输区域的区域内路径。注意，如果虚拟通道的距离值大于十六进制 0xffff（Router-LSA 中最大的接口值）应当被认为不可操作（即与路径不存在时一样对待）。

关于虚拟通道的其他细节如下：

AS-external-LSA “决不”在虚拟邻接间洪泛。这会产生重复，因为同样的 AS-external-LSA 已经在传输区域内洪泛。同样，在数据库交换过程中，AS-external-LSA 不能被汇总。

虚拟通道的距离值“不能”配置。该值是由两台 ABR 之间的区域内路径值来定义的。该值出现在虚拟通道的相关路由表项中，当虚拟通道的距离值改变时，应当为骨干区域生成一个新的 Router-LSA。

正象虚拟通道的距离值和存在性一样，虚拟通道接口的 IP 地址以及虚拟邻居的 IP 地址也是由路由表生成过程决定的（通过建立另一端点的路由表项）。它们用于在虚拟通道上发送 OSPF 协议包。注意当虚拟通道一方（或双方）的端点是由无编号点对点连接而接入传输区域时，可能会无法计算虚拟接口的 IP 地址和/或虚拟邻居的 IP 地址，这将导致虚拟通道失效。

在各端点为骨干区域生成的 Router-LSA 中，虚拟通道表现为类型 4 连接，其连接标识被设为虚拟邻居的 OSPF 路由器标识，连接数据为虚拟接口的 IP 地址。更多信息见第 12.4.1 节。

当且仅当传输区域中有一条或多条完全邻接的虚拟通道时（见第 6 章和第 16.1 节中的 TransitCapability），非骨干区域可以通过传输流量（即被作为“传输区域”）。在汇总骨干区域（见第 12.4.3 节）和路由计算时（见第 16.3 节），这样的区域需要特殊对待。

连接状态重传的时间间隔 RxmtInterval，可以在虚拟通道上配置。其值应当超过两台路由器之间所预期的来回时间。在虚拟通道上可能难以评估，将其配置的稍大会更好。

16. 计算路由表

本章描述计算 OSPF 路由表的细节。输入的是所接入区域的连接状态数据库，路由器运行如下的算法，一步步地建立路由表。在每一步中，路由器必须访问连接状态数据库的不同部分（例如，由特定路由器生成的 Router-LSA）。由第 12.2 节所描述的查找函数来完成访问。查找函数可能会返回 LS 时限等于 MaxAge 的 LSA。这样的 LSA 不应当用于路由表计算，并被视为查找失败。

OSPF 路由表的结构在第 11 章解释。第 11.2 和 11.3 节说明了两个路由表生成的例子。这一过程可以被分为下列步骤：

- （1）当前的路由表为无效，重新建立路由表。保存旧的路由表以发现路由表项的改变。
- （2）通过为所接入的每个区域建立最短路径树来计算区域内路径。特别的所有目标类型为 ABR 的路由表项也在此步计算。这一步分为两个部分。首先构建的树，仅考虑路由器和传输网络之间的连接，随后将存根网络加入树中。在构建区域最短路径树时，同时计算用于步骤 4 中区域的 TransitCapability。
- （3）通过查看 Summary-LSA 计算区域间路径。如果路由器接入多个区域（即为 ABR），仅查看骨干区域的 Summary-LSA。
- （4）如果 ABR 接入一个或多个传输区域（即为非骨干区域而 TransitCapability 为 TRUE），查看传输区域的 Summary-LSA，是否有使用传输区域的路径比上面步骤 2-3 找到的路径好。
- （5）通过查看 AS-external-LSA 计算到达外部目标的路径。ASBR（生成 AS-external-LSA）的位置在上面步骤 2-4 中计算。

步骤 2-5 的说明细节。

路由表的改变作为计算的结果，可能会导致 OSPF 协议执行更多的操作。例如区域间路径的改变会导致 ABR 生成新的 Summary-LSA（见第 12.4 节）。路由表改变而引起 OSPF 协议操作的完整列表见第 16.7 节。

16.1. 计算一个区域的最短路径树

下面的计算得出相关区域（下面称为区域 A）的区域内路径。路由器以自己为树根构建最短路径树 [脚注 22]。这里的最短路径树按两步形成，第一步，仅考虑路由器和传输网络之间的连接。通过 Dijkstra 算法，根据连接状态数据库的子集形成树。第二步，考虑存根网络连接，作为叶子加入树。

按照第 2 章中的拓扑结构作出说明。区域的连接状态数据库表现为直接图表，图表的行列节点为路由器、传输网络和存根网络。第一步仅处理传输节点（路由器和传输网络）以及它们的连接。通过计算最短路径，为每个传输节点建立下面的关联数据。

节点标识/Vertex (node) ID:

32 位数并带有节点类型（路由器或网络）唯一的识别出节点。对于路由器节点，节点标识就是 OSPF 路由器标识；对于网络节点，就是网络上 DR 的 IP 地址。

一个 LSA/An LSA:

每一个传输节点都有相关联的 LSA。对于路由器节点就是 Router-LSA；对于传输网络，就是 Network-LSA（由网络上的 DR 生成）。这时，LSA 的 LS 标识始终等于上面的节点标识。

下一跳列表/List of next hops:

从树根到该节点最短路径的下一跳列表。由于等值多路径，可能会有多条最短路径。每个下一跳都标记出转发流量的路由器输出接口。在广播、点对多点和 NBMA 网络上，下一跳还包括路径中，下一个路由器的 IP 地址。

从树根的距离/Distance from root:

从树根到节点最短路径的连接状态距离。该路径的距离值是路径中各组成部分距离值之和（在 Router-LSA 和 Network-LSA 中宣告）。距离值越小，该路径就越短。

计算过程的第一部分（即 Dijkstra 算法）摘要如下。每次递归运行算法的时，都有一个候选节点列表。找到从树根到达这些节点的路径，但不必要是最短的。最靠近树根的节点被认为是最短的，这个节点被加入最短路径树，并从候选节点列表删除。再检查其邻接节点，并增加/修改候选列表。算法被再次递归，直至候选列表为空。

下面描述算法的细节。这里所计算的是区域 A 的最短路径树。所有对连接状态数据库的查找都是在区域 A 的数据库中。

（1）初始化算法数据结构。清除候选节点列表，初始化最短路径树使其只包含树根（进行计算的路由器）。设定区域 A 的 TransitCapability 为 FALSE。

（2）将新加入树的节点称为节点 V。查看与节点 V 关联的 LSA。这是在区域 A 的连接状态数据库中基于节点标识的查找。如果是一个 Router-LSA，且 Router-LSA 中设置了 V 位（见附录 A.4.2），将区域 A 的 TransitCapability 设位 TRUE。在任何情况下，LSA 描述的每个连接都给出了到达邻接节点的距离值。对于所描述的每个连接 W（称为将节点 W 加入节点 V）：

（a）如果连接为存根网络，检查节点 V 的 LSA 中下一个连接。到存根网络的连接在最短路径计算的第二步中考虑。

（b）否则，节点 W 为传输节点（路由器或传输网络），在区域 A 的连接状态数据库中查找节点 W 的 LSA（Router-LSA 或 Network-LSA）。如果不存在该 LSA，或 LS 时限等于 MaxAge，或不存在指回节点 V 的连接，检查 LSA 中下一个连接 [脚注 23]。

（c）如果节点 W 已经在最短路径树上，检查 LSA 中下一个连接。

（d）计算从树根到节点 W 路径的连接状态距离值 D。D 是到达节点 V 最短路径的连接状态距离值（已经计算过），加上节点 V、W 之间的连接所宣告距离值。如果 D：

大于候选列表中节点 W 已经有的值，检查下一个连接。

等于候选列表中节点 W 已经有的值，使用所宣告的连接计算下一跳。计算的输入

为目标 (W) 及其父节点 (V)，计算在第 16.1.1 节描述。计算的结果应当加入候选列表中节点 W 的下一跳值。

小于候选列表中节点 W 已经有的值，或者 W 不在候选列表中，将 W 加入候选列表，并说明到达树根的距离为 D。同样使用所宣告的连接计算下一跳，并以此设定 W 的下一跳值。计算在第 16.1.1 节描述，其输入为目标 (W)，及父节点 (V)。

(3) 如果候选列表为空，最短路径树（传输节点）就被构建完成，这一部分过程结束。否则，在候选列表中选择最靠近树根的节点，将其加入最短路径树（同时在候选列表中删除该节点）。注意，如果存在多个节点最靠近树根时，网络节点必须比路由器节点先选择，以用于找到所有的等值路径。在 OSPF 多播路由扩展 (MOSPF) 所使用的 Dijkstra 修改算法中，使用了一致的选择方法。

(4) 可能修改路由表。这些修改的路由表项，其关联区域必须是区域 A，路径类型必须是区域内路径，其距离值是由最短路径计算新找到的距离。

如果新加入的节点是 ABR 或 ASBR，在路由表中加入目标类型为“路由器”的项。将所关联的 Router-LSA 中选项域的内容复制到路由表的选项域中。将新加入的节点称为路由器 X。如果路由器 X 是执行计算路由器虚拟通道的另一端，该虚拟通道使用区域 A 为传输区域，就宣告该虚拟通道开通，虚拟接口的 IP 地址就被设为计算得出的输出接口的 IP 地址，虚拟邻居的 IP 地址就被设为路由器 X 指回树根的接口 IP 地址（包含在路由器 X 的 Router-LSA 中）；同样的，该接口也在最短路径树中指回路器 X 的父节点（类似与第 16.1.1 节中的计算）。

如果新加入的节点是传输网络，就在路由表项中定位了该网络。该项的目标标识为 IP 网络地址，可以通过节点标识 (LS 标识) 与所关联的子网掩码（可以在 Network-LSA 中找到）运算而得到。如果路由表项已经存在（即在路由表中，已经存在了到达目标的区域内路径），就有多个节点映射到相同的 IP 网络。当有新的 DR 建立时，可能出现这种情况。当且仅当新找到的路径距离与原来一样，并且当前路由表项连接状态起源的 LS 标识，比新加入节点 LSA 要小时，应当覆盖当前的路由表。

如果没有该网络的路由表项（这是通常状态），应当加入该 IP 网络的路由表项。该路由表项的连接状态起源应当被设定为新加入的节点 LSA。

(5) 回到步骤 2，递归调用本算法。

存根网络在本过程的第二步中被加入树。在这步中，再次检查所有的路由器节点。并丢弃在上一步中，已经被判定为不可到达的。对于每一个可以到达的路由器节点（称其为 V），找到其在连接状态数据库中所关联的 Router-LSA。检查其中出现的每个存根网络，并执行下面的步骤：

(1) 计算从存根网络到树根的距离 D。D 等于从树根到路由器节点的距离（在第一步中计算），加上存根网络的宣告距离。通过在当前路由表中查找该存根网络，将当前到达存根网络的距离值与计算出的距离相比较。如果计算出的距离 D 较大，检查 LSA 中下一个存根网络。

(2) 如果到达本步骤，必须更新该存根区域的路由表项。计算使用该存根连接的下一跳。计算在第 16.1.1 节中描述；计算的输入是目标（该存根网络）及父节点（路由器节点）。如果距

离 D 与路由表中的距离值相同，仅仅将该下一跳加入到路由表的下一跳列表中。这时，路由表中已经有连接状态起源。如果该连接状态起源的 Router-LSA 中 LS 标识比 V 的路由器标识要小，重设连接状态起源为 V 的 Router-LSA。

否则， D 比路由表中的距离值小。覆盖当前路由表，设定路由表项的距离值为 D ，并设定下一跳列表为新的计算值。设定路由表项中连接状态起源为 V 的 Router-LSA。并检查下一个存根网络连接。

对于所有在第二步中增加/修改的路由表，必须是相关于区域 A ，并且路径类型为区域内。当可到达的 Router-LSA 检查完毕，第二步就告完成。此时，就决定了所有与区域 A 相关联的区域内路径。

本规范并不要求完全按照上面说的两步来计算最短路径树。但是，如果使用其他的算法，必须构建出与之相同的最短路径树。因此，树中所包含的传输节点间的连接必须是双向的。需要提及的是，存在有更高效率的算法构建树，如 [引用 1] 中所描述的增强 SPF 算法。

16.1.1. 计算下一跳

本节描述如何为一个目标计算当前的一系列下一跳。每个下一跳都包含用于转发包的输出接口以及下一跳路由器的 IP 地址（如果有）。每次发现到达目标的最短路径时，调用下一跳的计算。这发生在构建最短路径树的任何步骤中（见第 16.1 节）。在最短路径树计算的第一步中，当目标被加入候选列表，或候选列表中的目标项被改变时（第一步的步骤 2d）发现最短路径；或在第二步中，目标路由表项被修改时（第二步的步骤 2）发现最短路径。

在构建最短路径树时，随着发现越来越短的路径，可能会多次计算目标所使用的下一跳。最终，目标的路由表项将始终反应最短路径的下一跳。

下一跳计算所需要的输入有：**a)** 目标；以及 **b)** 在树根（进行计算的路由器）与目标之间，当前最短路径上的父节点。该父节点始终为一个传输节点（即始终是路由器或传输网络）。

如果在目标与树根之间的当前最短路径上有不止一个中间路由器，目标就简单的从父节点继承下一跳。否则，有两种情况。第一种情况，父节点为树根（计算路由器自身）。这意味着目标是一个直接连接的网络或直接连接的路由器。输出接口就是连接目标网络/路由器的 OSPF 接口。如果目标通过点对多点接口与计算路由器相联，就可以从目标的 Router-LSA 中得到下一跳 IP 地址：任何指回计算路由器，并且连接数据域属于该点对多点网络的连接，就提供了下一跳地址。如果目标是一个直接连接的网络，或者路由器通过点对点接口与计算路由器连接，就不需要提供下一跳地址。如果目标是与计算路由器通过虚拟通道连接的路由器，下一跳地址的定义在第 16.3 节的计算中。

第二种情况，父节点是一个连接计算路由器与目标的网络。下一跳列表通过检查目标的 Router-LSA 来决定。在 Router-LSA 中任何指回父网络的连接，其连接数据域就提供了下一跳路由器的 IP 地址。其输出接口可以从下一跳 IP 地址而获得（或者从父网络继承）。

16.2. 计算区域间路径

通过检查 Summary-LSA 可以计算区域间路径。如果路由器接入了多个区域，则仅检查骨干区域的 Summary-LSA。接入一个区域的路由器检查该区域的 Summary-LSA。不论哪种情况，被检查的

Summary-LSA 都是区域连接状态数据库的一部分（称为区域 A）。

Summary-LSA 由 ABR 生成。依次考虑区域 A 中的 Summary-LSA，Summary-LSA 描述的目标可能是网络（类型 3 Summary-LSA）或 ASBR（类型 4 Summary-LSA）。对于每个 Summary-LSA：

- （1）如果该 LSA 的距离为 LSInfinity，或其 LS 时限等于 MaxAge，检查下一个 LSA。
- （2）如果该 LSA 是由计算路由器自己生成的，检查下一个 LSA。
- （3）如果是类型 3 的 Summary-LSA，且该 LSA 描述的目标等于路由器上所配置的区域地址范围（见第 3.5 节），并且该区域地址范围为活跃，就应当忽略该 Summary-LSA。“活跃”意味着有一个或多个可以到达的网络（通过区域内路径），包含在该区域范围内。
- （4）否则，称该 LSA 为 N（对于类型 3 的 Summary-LSA，N 的地址可以通过 LSA 中的 LS 标识域 LSA 体中的网络/子网掩码运算而得到）描述了该目标，生成 LSA 的路由器称为 BR。在区域 A 的路由表中查找 BR。如果不存在路由器 BR 的项（即区域 A 内不能到达 BR），不对该 LSA 操作，并考虑下一个 LSA。否则，该 LSA 就描述了到达目标 N 的区域间路径，其距离值为到达 DR 的距离，加上 LSA 中说明的距离之和。将这个区域间路径的距离值称为 IAC。
- （5）下来，查找目标 N 的路由表项（如果 N 是 ASBR，查找该“路由器”相关于区域 A 的路由表项）。如果没有针对 N 的项，或该项的路径类型为“类型 1 外部”或“类型 2 外部”，就将 N 加入区域间路径。并设定关联区域为区域 A，距离值 IAC，下一跳等于 BR 的下一跳列表，宣告路由器为 BR。
- （6）否则，如果在路由表中的路径是区域内路径，不对该 LSA 操作（区域内路径已经完成）。
- （7）否则，路由表中的路径也是区域间路径。如果通过 BR 的路径较短则覆盖原有路径。否则，如果新的路径距离值一样，也将其加入路由表项中。

16.3. 查看传输区域的 Summary-LSA

这一步仅在接入了一个或多个运送传输数据的非骨干区域（即“传输区域”，或者说区域的 TransitCapability 参数在 Dijkstra 算法的步骤 2 中被设为 TRUE，见第 16.1 节）的 ABR 上执行。

这一步的计算是为了检查传输区域是否能提供，比第 16.1 节和第 16.2 节计算出的结果更好（更短）的路径。任何找到的更好的或与前面相同的路径，都被加入路由表中。

这一计算也决定了在第 16.1 节和第 16.2 节中计算出虚拟通道的下一跳地址。在计算完成后，第 16.1 节和第 16.2 节计算出的路径，如果仍不能解决下一跳，将被丢弃。

计算过程如下。依次检查所有传输区域中的 Summary-LSA。每个 Summary-LSA 描述了通过传输区域 A 达到网络 N 的路径（N 的地址通过 LSA 中 LS 标识与 LSA 体中的网络/子网掩码运算而得到），或者对于类型 4 Summary-LSA 而说，是 ASBR N。假设该 Summary-LSA 是由被称为 BR 的 ABR 所生成。

(1) 如果该 LSA 的距离为 LSInfinity，或其 LS 时限等于 MaxAge，检查下一个 LSA。

(2) 如果该 LSA 是由计算路由器自己生成的，检查下一个 LSA。

(3) 查找 N 的路由表项。(如果 N 是 ASBR，查找该“路由器”相关于骨干区域的路由表项)。如果不存在，或路径类型不是区域内或区域间，或者关联区域不是骨干区域，检查下一个 LSA。换句话说，计算仅仅更新在第 16.1 节中找到的骨干区域中区域内路径，或第 16.2 节找到的区域间路径。

(4) 查找宣告路由器 BR 相关于区域 A 的路由表项。如果不可达，检查下一个 LSA。否则，到达目标 N 的距离就是 BR 在区域 A 路由表中的距离加上 LSA 中宣告的距离，称之为 IAC。

(5) 如果该值比当前 N 路由表中的距离要小，使用 BR 的下一跳列表来覆盖 N 的，并设定 N 的路由表距离为 IAC。否则，如果 IAC 和 N 的当前值相同，将 BR 的下一跳列表加入 N 的。在任一种情况下，N 的路由表所关联的区域必须是骨干区域，其路径类型（区域内或区域间）必须和原来的相同。

要注意，上面的计算决不会把不可达的目标变为可达，而仅是可能找到更好的路径。计算中找到的更好的路径被加入路由表项后，会在 Summary-LSA 中向其他区域重新宣告。

这种计算的例子，参见图 17 中的 AS。一个非骨干区域（区域 1）将骨干区域物理地分割为两个部分。为了维持骨干区域的连通性，在路由器 RT1 和 RT4 之间配置了虚拟通道。图的右边，网络 N1 属于骨干区域。可以看出从路由器 RT5 到网络 N1 的骨干区域内路径（距离 20）要比从路由器 RT4 到网络 N1（距离 100）的小。路由器 RT4 和 RT5 都在 Summary-LSA 中向区域 1 宣告网络 N1。

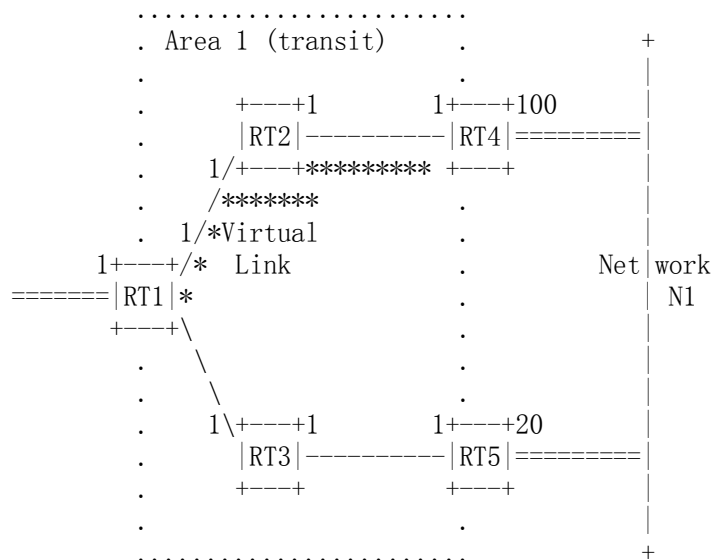


图 17：传输区域的路由

在按照第 16.1 节计算了骨干区域的最短路径树后，路由器 RT1（虚拟通道的左端）会选择 RT4 来传送所有到达网络 N1 的数据流量。而实际上，由于路由器 RT5 更靠近网络 N1，区域 1 中的所有路由器（即路由器 RT2 和 RT3）将使用 RT5，而不是 RT4 来转发到达网络 N1 的流量。并且最后，在按照上面的计算，重新检查了区域 1 中的 Summary-LSA 后，RT1 也使用 RT5 来转发网络 N1 的

流量。注意，在本例中，虚拟通道允许传输流量通过区域 1，但是所通过的实际路径并不按照虚拟通道的定义。换句话说，虚拟通道允许传输流量通过区域，但并不精确的定义所通过的路径。

16.4. 计算 AS 外部路径

通过检查 AS-external-LSA 可以计算 AS 外部路径。每个 AS-external-LSA 被依次考虑。大多数 AS-external-LSA 描述了到达特定 IP 目标的路径。AS-external-LSA 也可以描述 AS 的默认路径（目标标识 = DefaultDestination，网络/子网掩码 = 0x00000000）。对于每个 AS-external-LSA：

（1）如果该 LSA 的距离为 LSInfinity，或其 LS 时限等于 MaxAge，检查下一个 LSA。

（2）如果该 LSA 是由计算路由器自己生成的，检查下一个 LSA。

（3）称 LSA 描述目标的为 N。N 的地址通过 LSA 中 LS 标识和 LSA 体中的网络/子网掩码计算而来。在路由表中查找（可能每个接入区域都需要一次）生成该 LSA 的 ASBR。如果没有到达该 ASBR 的路径（即 ASBR 不可达），不作关于该 LSA 的操作，并考虑列表中的下一项。

否则，该 LSA 就描述了到达 AS 外部目标 N 的路径。检查 AS-external-LSA 中的转发地址，这说明了到达目标的包应当被转发的地址。

如果转发地址被设为 0.0.0.0，包应当被转发到 ASBR 自身。如果 ASBR 有多个路由表项，按下面的描述选择。如果 RFC1583Compatibility 被设为 “disable”，去除按照第 16.4.1 节描述的 ASBR 路由表项。不论哪种情况，在剩余的路由表项中，选择最小距离的项；当有多个最小距离项时，选择关联区域中 OSPF 区域标识（按 32 位无符号整数考虑）最大的项。

如果转发地址非零，在路由表中查找转发地址 [脚注 24]。所匹配的路由表项必须是区域内或区域间路径，如果没有如此的路径存在，不作关于该 LSA 的操作，并考虑列表中的下一项。

（4）设定 X 为到达 ASBR/转发地址首选路径的距离值，而 Y 为 LSA 中说明的距离值。X 被称为连接状态距离，而 Y 为类型 1 或 2 的外部距离。

（5）查找到达目标 N 的路由表项。如果没有，就加入到达 N 的 AS 外部路径，并设定下一跳为转发地址的下一跳列表，宣告路由器为 ASBR。如果外部路径的类型为 1，则路径类型为类型 1 外部路径，距离等于 X+Y。如果外部路径的类型为 2，则路径类型为类型 2 外部路径，连接状态中的路由器距离为 X，类型 2 距离为 Y。

（6）将该 LSA 描述的外部路径与路由表中已经存在的到达 N 的路径按如下比较。如果新的路径更好，将当前路径替换到 N 的路由表项中。如果新旧路径一样好，将新路径加到 N 路由表项的路径列表中。

（a）区域内路径与区域间路径，始终比 AS 外部路径好。

（b）类型 1 外部路径始终比类型 2 外部路径好。当所有的路径都为类型 2 外部路径时，宣告最小类型 2 距离的路径始终最好。

(c) 如果新旧路径还是不能区别出好坏，而且 `RFC1583Compatibility` 被设定为“`disable`”，按第 16.4.1 节的描述，选择使用 AS 内部路径到达 ASBR/转发地址的路径为最好。

(d) 如果新旧路径还是不能区别出好坏，选择具有最小距离的路径为最好。类型 1 外部路径之间通过到达转发地址的距离和宣告的类型 1 距离之和 ($X+Y$) 比较。类型 2 外部路径之间如果宣告了同样的类型 2 距离，就比较到达转发地址的距离。

16.4.1. 外部路径参数

当提供有多条到达 ASBR/转发地址的 AS 内部路径时，由下面的规则来说明哪一条更好。当可以通过多个区域到达同一 ASBR，或决定应当使用哪个 AS-external-LSA 时，这些规则起作用。在前面的例子中，所有的路径在同一个 ASBR 上终止，而在下面，路径会在不同的 ASBR 上终止。不论那种情况，每条路径都按照第 11 章的定义，表现为不同的路由表项。

本节仅在 `RFC1583Compatibility` 被设定为“`disable`”时被考虑。

从高到低的路径优先级按下面定义。注意，使用这些规则后，仍可能有多条最高优先级的路径。这些路径必须按第 16.4 节的描述，按距离值决定。

非骨干区域的区域内路径始终是最高优先。

其他的路径，骨干区域中的区域内路径和区域间路径，优先级相同。

16.5. 增量更新——Summary-LSA

当接收到新的 Summary-LSA 时，不必重新计算整个路由表。把该 Summary-LSA 所描述的目标称为 N (N 的地址可以由 LSA 中 LS 标识和 LSA 体中的网络/子网掩码计算而得出)，并假设该 LSA 属于区域 A。有下面的两种情况：

情况 1：区域 A 为骨干区域或者路由器不为 ABR。

这时执行下面的计算。首先，如果当前存在到达目标 N 的区域间路径，记录下该项内容以便后面的比较，并使该项路由表无效。为目标 N 单独执行第 16.2 节中描述的计算。在计算中，检查所有在区域 A 中描述到达 N 路径的 Summary-LSA。此外，如果路由器是接入一个或多个传输区域的 ABR，为这个目标单独执行第 16.3 节中的计算。如果计算后，到达一台 ASBR，或转发地址的路径或距离发生了改变（这时为类型 4 的 Summary-LSA），必须重新执行第 16.4 节的计算，以再次检查所有的 AS-external-LSA。否则，如果 N 变为不可达，必须单独为目标 N 重新运行第 16.4 节的计算，以发现是否有到达 N 的其他外部路径。

情况 2：区域 A 为传输区域并且路由器为 ABR。

这时执行下面的计算。记录下该项内容以便后面的比较，删除 N 的路由表项中使用传输区域 A 的路径，如果这删除了路由表项中的所有路径，必须使该项无效。必须按第 16.3 节的描述单独为目标 N 重新计算。如果计算的结果导致了到达 N 距离的增加，必须按第 16.1 节的描述重新执行 Dijkstra 算法以计算完整的路由表。如果计算后，到达一台 ASBR，或转发地址的路径或距离发生了改变（这时为类型 4 的 Summary-LSA），必须重新执行第 16.4 节的计算，以再

次检查所有的 **AS-external-LSA**。否则，如果 **N** 变为不可达，必须单独为目标 **N** 重新运行第 16.4 节的计算，以发现是否有到达 **N** 的其他外部路径。

16.6. 增量更新——AS-external-LSA

当收到新的 **AS-external-LSA** 时，不需要重新计算整个路由表。称该 **AS-external-LSA** 描述的目标为 **N**。**N** 的地址可以由 **LSA** 中 **LS** 标识和 **LSA** 体中的网络/子网掩码计算而得出。如果已经有一条区域内或区域间的路径到达目标，不需要重新计算（内部路径优先考虑）。

否则，执行第 16.4 节的过程，但只针对目标为 **N** 的 **AS-external-LSA**。在执行过程前，应当使当前到达 **N** 的路由表项无效。

16.7. 路由表改变引起的事件

路由表的改变，有时会导致 **OSPF** 的 **ABR** 执行额外操作。在如下路由表改变时，路由器需要执行：

路由表项的距离或类型有了改变。如果该项所描述的目标是网络或 **ASBR**，而且该项所改变的不是简单的 **AS** 外部路径，也许需要生成新的 **Summary-LSA**（可能要为每个所接入的区域，包括骨干区域）。更多信息见第 12.4.3 节。如果删除了以前所宣告的项，或不再向特定的区域宣告，必须通过将 **LS** 时限域设定为 **MaxAge** 并重新洪泛，将该 **LSA** 从路由域中废止（见第 14.1 节）。

关联虚拟通道的路由表项有了改变。该项的目标为 **ABR**，这样的改变会影响虚拟通道的距离值或存在性。

如果该项表示 **ABR** 新近可以到达，所对应的虚拟通道现在可以操作。生成该虚拟通道的 **InterfaceUp** 事件，引起虚拟邻接的形成（见第 10.3 节）。并计算虚拟通道的 **IP** 地址以及虚拟邻居的邻居 **IP** 地址。

如果该项表示 **ABR** 新近不可达，消除所关联的虚拟通道，这意味着为所关联的虚拟通道生成 **InterfaceDown** 事件。

如果该项的距离改变，而该项为一完全建立了的虚拟邻接，为骨干区域生成新的 **Router-LSA**。这可能引起进一步的路由表改变。

16.8. 等值多路径

OSPF 为任何目标维持等值多路径，这可以在上面的路由表计算中看出，并在路由表结构中定义。

这些多条的路径都有着同样的类型（区域内、区域外、类型 1 外部或类型 2 外部），距离值，并和相同的区域相关联。但每条路径可能有不同的下一跳和宣告路由器。

没有必要在运行 **OSPF** 的路由器上记录下所有的等值路径。每个实现可以保持固定数量的路径以到达任何给定目标。这对本规范介绍的算法没有任何影响。

脚注：

[脚注 1]：图表标记可以表示路由器、传输网络或存根网络。由于路由器可能属于多个区域，这就不可能在图表标记上标出颜色。

[脚注 2]：一台路由器的所有接口有可能都是无编号点对点连接。这时，必须设定一个 IP 地址给路由器。这一地址在路由器的 **Router-LSA** 中，作为主机路径宣告。

[脚注 3]：注意，这两种情况下，不论是否为虚拟通道接口，可能会有相同的 IP 地址。

[脚注 4]：注意，不论接口处于什么状态，在无编号点对点网络上不会生成主机路径，也不会向其发送 IP 包。

[脚注 5]：看一下网络上的 DR 失效后的情形，会很有帮助。将网络上的 DR 称为 RT1，BDR 称为 RT2。当 RT1 断开时（或其连接到网络的接口失效），其他的路由器会在 **RouterDeadInterval** 秒内发觉。但所有的路由器不可能恰好同时发觉。比 RT2 早发觉的路由器，会在一段时间内，选择 RT2 同时为 DR 和 BDR。当 RT2 发觉 RT1 失效后，将自己转换为 DR。同时，剩余路由器中最高优先级的路由器被选为 BDR。

[脚注 6]：在点对点网络上，由下层协议来说明邻居是否运行；同样，虚拟通道上的邻居由路由表计算来说明。但在这两种情况下，仍旧使用 **Hello** 协议。用以确保邻居间的双向通讯，并保证邻居能够在网络层正常工作。

[脚注 7]：当指定路由器改变的时候，恰巧有邻居向路由器发送 DD 包，这会导致 DR 身份瞬间的混淆。

[脚注 8]：通过将邻接状态退回到 **ExStart** 状态，可以使路由器重新同步其已经建立的完全邻接。这会使邻接的对端处理 **SeqNumberMismatch** 事件，并且也回到 **ExStart** 状态。

[脚注 9]：IP 网络的地址范围与 OSPF 路由器标识的范围可能有重叠。也就是，网络所包含的 IP 地址，可能正好是某各路路由器的标识（被作为 32 位数考虑时）。

[脚注 10]：“丢弃”项是使区域边界的汇总不产生环路而必须的。

[脚注 11]：在这里假设，有两个地址范围匹配一个目标时，其中的一个比另一个更精确。非连续子网掩码的配置会违背这一假设。OSPF 协议不能处理这种类型的子网掩码配置。

[脚注 12]：**MaxAgeDiff** 是一个常量。它以秒数表示一个 LSA 洪泛到整个路由域，所可能的最大传播时限。如果两个 LSA 的差异大于此，就可以假设为同一 LSA 的不同实例。当路由器重启动，或丢失了 LSA 以前的 LS 序号时会发生。更多细节见第 13.4 节。

[脚注 13]：当两个 LSA 有着不同的 LS 校验和时，被假设为不同的实例。这可能在路由器重启动，或丢失了 LSA 以前的 LS 序号时会发生。这时，两个 LSA 有相同的 LS 序号，不可能从中判断哪个 LSA 更新。可是，如果将错误的 LSA 认为较新，生成 LSA 的路由器会简单的再生成一个实例。更多细节见第 13.4 节。

[脚注 14]: 还要实现一个按部分信息的查找函数。在路由表计算过程中, 需要仅根据 LS 标识找到一个 Network-LSA。这一查找很好定义, 因为没有两个 Network-LSA 会有相同的 LS 标识。

[脚注 15]: 这是在 RFC1583 中所说明的点对点情况。这样有三个优点: a) 不需要为点对点连接分配一个子网; b) 这导致路由的偏离, 发往点对点接口的包将被接口所接收 (在诊断上有用); c) 这允许对邻居网络引导, 而不需要在引导程序中包含 OSPF 的实现。

(? ? ?) b) it tends to bias the routing so that packets estined for the point-to-point interface will actually be received over the interface (which is useful for diagnostic purposes)

[脚注 16]: 这是更典型的点对点情况, 常用于诸如 RIP 等协议。

[脚注 17]: 这一句包含这种情况: 区域间路径不被汇总到骨干区域。因为区域间路径始终与骨干区域相关联。

[脚注 18]: 这仅发生在非骨干区域 A 支持传输数据流量时 (即 TransitCapability 被设为 TRUE)。例如图 6 中的区域配置, 区域 2 因为 RT10 与 RT11 之间的虚拟通道而支持传输流量。其结果是 RT11 仅需要为区域 2 生成一个 Summary-LSA (描述目标 N9-N11、H1), RT11 其他符合条件的路径, 其下一跳都在区域 2 中 (仅需要其他 ABR 来宣告, 如本例中的 RT10 和 RT7)。

[脚注 19]: 如果在路由表中记录更多的内容, 可以实现仅计算一个区域的最短路径树。事实上另一个增强算法的实现可以仅构建一个特定区域的最短路径树 [引用 1]。对这个算法的描述超过了本规范的范围。

[脚注 20]: 这会使连接状态请求列表为空, 并导致邻居状态转为 Full。更多细节见第 10.9 节。

[脚注 21]: LSA 的 LS 时限很少会达到 MaxAge。事实上, 在老化之前, LSA 将被一个较新的 LSA 所替代。

[脚注 22]: 严格地说, 由于存在等值多路径, 该算法生成的并不是树。但我们仍旧使用术语“树”, 是因为在已存在的文献中, 它是最多出现的。

[脚注 23]: 注意, 指回节点 V 的连接是必须的; 这不需要考虑从 V 到 W 的单向连接。这样可以保证, 在邻居路由器之间交换数据流量之前, 连接状态数据库将被同步。

[脚注 24]: 当转发地址为非零时, 应当指向属于另一个 AS 的路由器。细节见第 12.4.4 节。

引用:

- [引用 1] McQuillan, J., I. Richer and E. Rosen, "ARPANET Routing Algorithm Improvements", BBN Technical Report 3803, April 1978.
- [引用 2] Digital Equipment Corporation, "Information processing systems -- Datacommunications -- Intermediate System to Intermediate System Intra-Domain Routing Protocol", October 1987.
- [引用 3] McQuillan, J., et.al., "The New Routing Algorithm for the ARPANET", IEEE Transactions on Communications, May 1980.
- [引用 4] Perlman, R., "Fault-Tolerant Broadcast of Routing Information", Computer Networks, December 1983.
- [引用 5] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [引用 6] McKenzie, A., "ISO Transport Protocol specification ISO DP 8073", RFC 905, April 1984.
- [引用 7] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, May 1988.
- [引用 8] McCloghrie, K., and M. Rose, "Management Information Base for network management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.
- [引用 9] Moy, J., "OSPF Version 2", RFC 1583, March 1994.
- [引用 10] Fuller, V., T. Li, J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC1519, September 1993.
- [引用 11] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.
- [引用 12] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.
- [引用 13] Leiner, B., et.al., "The DARPA Internet Protocol Suite", DDN Protocol Handbook, April 1985.
- [引用 14] Bradley, T., and C. Brown, "Inverse Address Resolution Protocol", RFC 1293, January 1992.
- [引用 15] deSouza, O., and M. Rodrigues, "Guidelines for Running OSPF Over Frame Relay Networks", RFC 1586, March 1994.
- [引用 16] Bellovin, S., "Security Problems in the TCP/IP Protocol Suite", ACM Computer Communications Review, Volume 19, Number 2, pp. 32-38, April 1989.

- [引用 17] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [引用 18] Moy, J., "Multicast Extensions to OSPF", RFC 1584, March 1994.
- [引用 19] Coltun, R., and V. Fuller, "The OSPF NSSA Option", RFC 1587, March 1994.
- [引用 20] Ferguson, D., "The OSPF External Attributes LSA", work in progress.
- [引用 21] Moy, J., "Extending OSPF to Support Demand Circuits", RFC 1793, April 1995.
- [引用 22] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.
- [引用 23] Rekhter, Y., and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.
- [引用 24] Hinden, R., "Internet Routing Protocol Standardization Criteria", BBN, October 1991.
- [引用 25] Moy, J., "OSPF Version 2", RFC 2178, July 1997.
- [引用 26] Rosen, E., "Vulnerabilities of Network Control Protocols: An Example", Computer Communication Review, July 1981.

A. OSPF 数据格式

本附录描述 OSPF 协议包以及 OSPF LSA 的格式。OSPF 协议直接运行在 IP 网络层之上。在描述数据格式之前，先解释 OSPF 封装的细节。

下面描述 OSPF 可选项，该域描述了 OSPF 路由域中可能支持或不支持的选项。OSPF 选项出现在 OSPF Hello 包，DD 包和 OSPF LSA 中。

OSPF 包格式的 details 在附录 A.3 中描述。OSPF LSA 的 details 在附录 A.4 中描述。

A.1 OSPF 包的封装

OSPF 直接运行于 Internet Protocol (IP) 网络层之上。OSPF 包被封装时仅加上 IP 和本地数据链路层头部。

OSPF 并没有定义协议包的分片，当所传输的包大于网络 MTU 时，就依赖于 IP 分片。如果需要 OSPF 包可以达到 65,535 字节（包含 IP 包头）。较大的 OSPF 包（DD 包、LSR 包、LSU 包和 LSAck 包）可以被分割为几个协议包，而不影响功能。但是推荐尽可能的避免 IP 分片。因此，在虚拟通道上的 OSPF 包应当被限定在 576 字节，除非执行了发现路径 MTU（见 [引用 22]）。

OSPF 包的 IP 封装中的其他重要功能有：

使用 IP 多播。在广播网络上，一些 OSPF 消息使用多播。使用两个独特的 IP 多播地址。这些包只会传输一跳，而决不会被转发。为了确保不会传送多跳，IP 包的 TTL 必须设定为 1。

AllSPFRouters:

这一多播地址被设定为 224.0.0.5。所有运行 OSPF 的路由器必须准备接收发送到该地址的包。Hello 包始终发往该目标。此外在洪泛过程中，一些 OSPF 包被发往该地址。

AllDRouters:

这一多播地址被设定为 224.0.0.6。DR 和 BDR 必须准备接收发送到该地址的包。在洪泛过程中，一些 OSPF 包被发往该地址。

OSPF 的 IP 协议号为 89。该数字在 Network Information Center 中注册。IP 协议号的设定见文档 [附录 11]。

所有的 OSPF 路由协议包使用数值为二进制 0000 的普通 TOS 服务，其定义见 [附录 12]。

路由协议包中的 IP 优先级被设定为 Internetwork Control。不论发送或接收，OSPF 协议包都应当比普通 IP 数据包的优先级高。通过将 IP 包头中的 IP 优先级设定为 Internetwork Control [引用 5] 而实现。

A.2 选项域

OSPF 选项域出现在 OSPF Hello 包、DD 包和所有的 LSA 中。选项域使 OSPF 路由器支持（或不

支持）可选择的功能，并向其他 OSPF 路由器宣告其能力。这一机制使得不同能力的路由器可以混合在一个 OSPF 路由域中。

在使用 Hello 包时，如果选项域中的能力不匹配，可以使路由器拒绝一个邻居。此外，在交换 DD 包时，路由器可以因为功能的不同，而选择不将特定的 LSA 转发到邻居。最后，路由器可以根据 LSA 中列出的选项将其排除出路由表的计算，而将流量转发到特定的路由器。

OSPF 包的选项域中定义了五个位，而其中只有一个位（E 位）在本备忘录中定义。每个位都在下面摘要描述。在发送 Hello 包、DD 包和生成 LSA 时，路由器应当清除选项域中未定义的位。相对应的，当接收到 Hello 包、DD 包或 LSA 中未知的位时应当忽略，而按正常的包/LSA 来处理。



选项域

E 位：
该位描述是否洪泛 AS-external-LSA，在本备忘录的第 3.6、9.5、10.8 和 12.1.2 节中描述。

MC 位：
该位描述是否按照 [引用 18] 的说明转发 IP 多播包。

N/P 位：
该位描述了处理类型 7 LSA，见 [引用 19] 的说明。

EA 位：
该位描述了是否按 [引用 20] 的说明忽略还是接收并转发 External-Attributes-LSA。

DC 位：
该位描述了按 [引用 21] 的说明处理按需链路。

A.3 OSPF 包格式

有五种不同的 OSPF 包类型。所有的 OSPF 包都以 24 个字节的头部开始。首先描述包头，而后描述各种类型。在本节中列出包中的各域，并列举出各域的定义。

所有的 OSPF 包（除了 OSPF Hello 包）都处理 LSA 列表。例如，LSU 包实现在 OSPF 路由域中洪泛 LSA。因此，必须理解 LSA 的格式才能分析 OSPF 协议包。LSA 的格式在附录 A.4 中描述。

接收及处理 OSPF 包的细节见第 8.2 节。发送 OSPF 包的解释见第 8.1 节。

A.3.1 OSPF 包头

每个 OSPF 包都以 24 字节的头部开始。头部所包含的所有信息用于决定包如何进行下一步操作。这在本规范的第 8.2 节中定义。



版本号/Version #:

OSPF 的版本号，本规范所说明的协议版本号为 2。

类型/Type

OSPF 包的类型按如下定义。具体细节见附录 A.3.2 到附录 A.3.6。

类型	描述
1	Hello 包
2	数据库描述包/Database Description
3	连接状态请求包/Link State Request
4	连接状态更新包/Link State Update
5	连接状态确认包/Link State Ack

包长度/Packet length:

整个 OSPF 包的字节长度，包括标准的 OSPF 头部。

路由器标识/Router ID:

生成包的路由器标识。

区域标识/Area ID:

一个 32 位数表示包所属于的区域。所有的 OSPF 包都关联到一个区域。大多数都只传输一跳。在虚拟通道中的传输关联到骨干区域 0.0.0.0。

校验和/Checksum:

从 OSPF 包头开始，除了 64 位的验证域外，整个包的标准 IP 校验和。该校验和作为 16 位 1 补足校验和而计算包中除验证域外的所有 16 位字。如果包的长度不满足 16 位字，就在校验和前加上一个字节的 0 来补足。校验和也被作为正确性验证的一部分，在某些 OSPF 验证类型下，校验和计算被忽略。

验证类型/AuType:

说明所使用验证过程的类型。附录 D 描述验证。参见附录 D 中当前定义的验证类型列表。

验证域/ Authentication:
64 位的验证域被所选择的验证类型所使用。细节见附录 D。

A.3.2 Hello 包

Hello 包的 OSPF 包类型为 1。这些包被周期性的从各个接口（包含虚拟通道）发送，以建立和维持邻居关系。此外，在支持多播或广播的物理网络上，Hello 包使用多播来动态的发现邻居路由器。

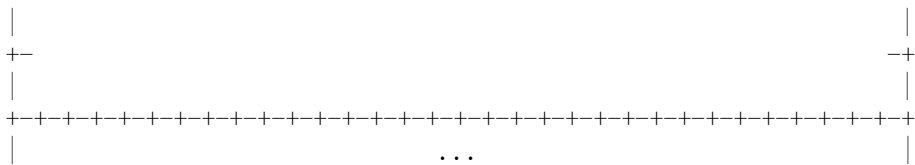
所有接入同一网络的路由器必须使用相同的参数（网络掩码、HelloInterval 和 RouterDeadInterval）。这些参数都包含在 Hello 包中，参数的不同将阻止形成邻居关系。接收 Hello 的细节描述出现在第 10.5 节；发送 Hello 包出现在第 9.5 节。



网络掩码/Network mask:
该接口所关联的网络掩码。例如，接口上设定的是 B 类网络并使用第三个字节作为子网，则网络掩码为 0xfffff00。

选项/Options:
在附录 A.2 中说明的路由器所支持的选项。

HelloInterval:
路由器发送 Hello 包的间隔秒数。



接口 MTU/Interface MTU:
从所关联的接口不分片而能发送的最大 IP 包字节大小。常见接口类型的 MTU 可以在 [附录 22] 的表 7-1 中找到。在虚拟通道上的 DD 包里，接口 MTU 被设为 0。

选项/Options:
按附录 A.2 的描述，路由器所支持的选项。

I 位/I-bit:
初始位，在第一个 DD 包中设定为 1。

M 位/M-bit:
更多位。当后面还有更多的 DD 包时设定为 1。

MS 位/MS-bit:
主从位。在数据库交换过程中的主机设定为 1，否则该路由器为从机。

DD 序号/DD sequence number:
用于描述 DD 包的序号。其初始值（设定了初始位）应当唯一。在整个数据库描述过程中，所发送的 DD 需要应当线形增加。

包中的其他部分是包含了连接状态数据库内容的列表（可能是局部列表）。每个 LSA 由其 LSA 头部来描述。LSA 头部在附录 A.4.1 中说明。这些信息可以唯一的识别出该 LSA 及其当前实例。

A.3.4 连接状态请求包/LSR 包

LSU 包的 OSPF 包类型为 3。在与邻居交换了 DD 包后，路由器后发现它的一部分连接状态数据库已经过期。这时就使用 LSR 包来取得邻居数据库中较新的部分。也许需要使用多个 LSR 包。

发送 LSR 包的路由器确切的知道所请求的实例。每个实例由 LS 序号、LS 校验和以及 LS 时限来定义，虽然这些域不是在 LSR 包中说明。在响应中，路由器会收到最新的实例。

发送 LSR 包的文档在第 10.9 节，接收 LSR 包的文档在第 10.7 节。



校验和/Checksum	验证类型/AuType
验证/Authentication	
验证/Authentication	
LS 类型/LS type	
LS 标识/Link State ID	
宣告路由器/Advertising Router	
...	

所请求的每个 LSA 由 LS 类型、LS 标识和宣告路由器来说明。这唯一的识别出一个 LSA，但没有指明实例。LSR 包被理解为请求该 LSA 最新的实例（而且也确实如此）。

A.3.5 连接状态更新包/LSU 包

LSU 包的 OSPF 包类型为 4。此包实现了 LSA 的洪泛。每个 LSU 包将其包含的 LSA 传送到距其起源更远的一跳。多个 LSA 可能被包含在一个包中。

LSU 包在支持多播/广播的物理网络上使用多播。为了使洪泛过程可靠，洪泛的 LSA 由 LSAck 包所确认。如果需要重传当前的 LSA，重传的 LSA 始终被直接发送到邻居。可靠性洪泛 LSA 的更多内容见第 13 章。

0	1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																	
版本号/Version#										4										包长度/Packet length																												
路由器标识/Router ID																																																
区域标识/Area ID																																																
校验和/Checksum																验证类型/AuType																																
验证/Authentication																																																
验证/Authentication																																																
LSA 数量/# LSA																																																
LSA																																																
...																																																

LSA 数量/# LSA:

该更新包中包含的 LSA 数量。

LSU 包内包含有一系列 LSA。每个 LSA 的开始是 20 个字节的通用头部，在附录 A.4.1 中描述。不

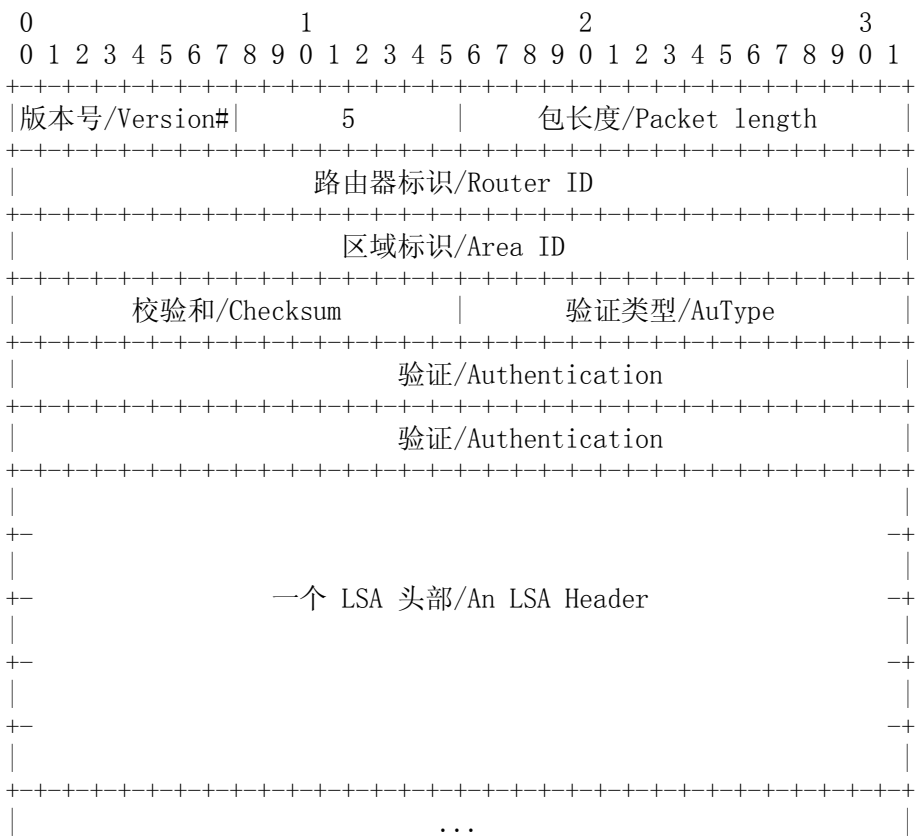
同类型 LSA 的描述细节见附录 A.4。

A.3.6 连接状态确认包/LSAck 包

LSAck 包的 OSPF 包类型为 5。通过明确的确认，该包使洪泛 LSA 变为可靠。确认是通过发送和接收 LSAck 包而实现。多个 LSA 的确认可以包含在一个 LSAck 包中。

根据当前发送接口的状态以及 LSU 包的发送路由器，LSAck 包可能发送到多播地址 AllSPFRouters 或 AllDRouters 或者使用单播。发送 LSAck 包的文档在第 13.5 节，接收 LSAck 包的文档在第 13.7 节。

包的格式与 DD 包类似。包中包含了 LSA 头部的列表。



通过 LSA 头部的描述确认每个 LSA。LSA 头部在附录 A.4.1 中描述。这些信息唯一的标明了 LSA 及其实例。

A.4 LSA 格式

本规范定义了五种不同类型的 LSA。每种 LSA 开始于 20 个字节的 LSA 头部。该头部在附录 A.4.1 中描述。接下来的章节描述不同的 LSA 类型。

每个 LSA 描述了 OSPF 路由域中的一部分内容。每台路由器生成一个 Router-LSA；此外如果路由器被选举为 DR，它生成一个 Network-LSA；还有其他类型的 LSA 也会被生成（见第 12.4 节）。所有的 LSA 都在 OSPF 路由域中洪泛。洪泛算法是可靠的，以确保所有的路由器有着一样的 LSA 集

合（洪泛算法的更多细节见第 13 章）。LSA 的集合被称为连接状态数据库。

从连接状态数据库，每台路由器构建以自己为树根的最短路径树。并以此生成路由表（见第 11 章）。建立路由表的更多细节见第 16 章。

A.4.1 LSA 头部

所有的 LSA 开始于 20 个字节的头部。头部包含的信息可以唯一的识别出 LSA (LS 类型、LS 标识和宣告路由器)。在同一时间里,可能存在 LSA 的多个实例。必须判定哪个实例较新。这通过检查 LSA 头部中的 LS 时限、LS 序号和 LS 校验和来判定。

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
LS 时限/LS age										选项/Options										LS 类型/LS type																			
LS 标识/Link State ID																																							
宣告路由器/Advertising Router																																							
LS 序号/LS sequence number																																							
LS 校验和/LS checksum															长度/length																								

LS 时限/LS age:

从 LSA 生成开始的时间秒数。

选项/Options:

所描述的路由域中支持的可选项。OSPF 可选项见附录 A.2。

LS 类型/LS type:

LSA 的类型。各种类型的 LSA 使用不同的格式。本备忘录定义的 LSA 类型如下（更多细节见第 12.1.3 节）。

LS 类型	描述
1	Router-LSA
2	Network-LSA
3	Summary-LSA (IP 网络)
4	Summary-LSA (ASBR)
5	AS-external-LSA

LS 标识/Link State ID:

该域描述由 LSA 所描述的网络部件。具体的内容取决于 LSA 中的 LS 类型。例如, Network-LSA 中的 LS 标识为网络上 DR 的接口 IP 地址 (从中可以计算出网络的 IP 地址)。LS 标识的更多描述见第 12.1.4 节。

宣告路由器/Advertising Router:

生成该 LSA 的路由器标识。例如，Network-LSA 中该域等于网络上 DR 的路由器标识。

- LS 序号/LS sequence number:
用于判定旧的或重复的 LSA。连续的 LSA 实例使用连续的 LS 序号。更多细节见第 12.1.6 节。
- LS 校验和/LS checksum:
整个 LSA 的 Fletcher 校验和，包括除 LSA 时限域外的 LSA 头部。更多细节见第 12.1.7 节。
- 长度/length:
LSA 的字节长度。包含 20 字节的 LSA 头部。

A.4.2 Router-LSA

Router-LSA 是类型 1 LSA。每台路由器生成一个 Router-LSA 。该 LSA 描述了路由器连接（接口）到区域的状态和距离值。连接到一个区域的所有接口必须在一个 Router-LSA 中描述。更多关于建立 Router-LSA 的内容，见第 12.4.1 节。



在 Router-LSA 中，LS 标识被设为路由器的 OSPF 路由器标识。Router-LSA 仅在一个区域内洪泛。

- V 位/bit V:
设置时，该路由器是一条或多条完全邻接的虚拟通道的端点，并说明区域为传输区域（V 表示虚拟通道）。

E 位/bit E:

设置时，该路由器为一个 **ASBR**（**E** 表示外部）。

B 位/bit B:

设置时，该路由器为一个 **ABR**（**B** 表示边界）。

连接数/# links:

该 **LSA** 所描述的路由器连接数量。必须是该区域路由器连接（接口）的总和。

下面的各域用于描述各个路由器连接（接口）。每个接口都有类型（见下面的类型域），类型域说明了所描述连接的种类。连接可能为传输网络、到达另一路由器或存根网络。其他描述连接的域都取决于连接类型。例如，每个连接有相关的 **32** 为连接数据域。对存根区域的连接来说表示网络的 **IP** 地址掩码，而对其他类型来说，表示的是路由器接口 **IP** 地址。

类型/Type:

路由器连接的基本描述，可为以下值。注意，主机路径被看作网络掩码为 **0xffffffff** 的存根网络，

类型	描述
1	点对点连接到另一路由器
2	连接到传输网络
3	连接到存根网络
4	虚拟通道

连接标识/Link ID:

表示路由器连接所接入的目标。其值取决于连接的类型。当所连接的目标也生成 **LSA**（即另一路由器或传输区域），连接标识就等于邻居 **LSA** 的 **LS** 标识。在路由表计算中，这提供了在连接状态数据库中查找邻居 **LSA** 的关键。更多细节见第 12.2 节。

类型	LS 标识
1	邻居的路由器标识
2	DR 的 IP 接口地址
3	IP 网络/子网号
4	邻居的路由器标识

连接数据/Link Data:

其值同样取决于连接的类型。对于存根网络连接，连接数据说明的是网络的 **IP** 地址。对于无编号点对点网络；说明的是接口的 **MIB-II** [引用 8] **ifIndex** 接口索引值；对于其他类型的连接，说明的是路由器接口的 **IP** 地址。在计算路由表过程中，计算下一跳的 **IP** 地址时需要使用这些信息。更多细节见第 16.1.1 节。

TOS 数/# TOS:

该连接不同 **TOS** 的数量，不包括所需要的连接距离（在 [引用 9] 中被称为 **TOS 0** 距离）。例如，如果没有给出额外的 **TOS** 距离，该域被设定为 0。

距离值/metric:

路由器连接的距离。

可能包括附加的 TOS 信息，用于和以前版本的 OSPF 兼容 ([引用 9])。对于每条连接，对应的 TOS 附加信息按下面定义：

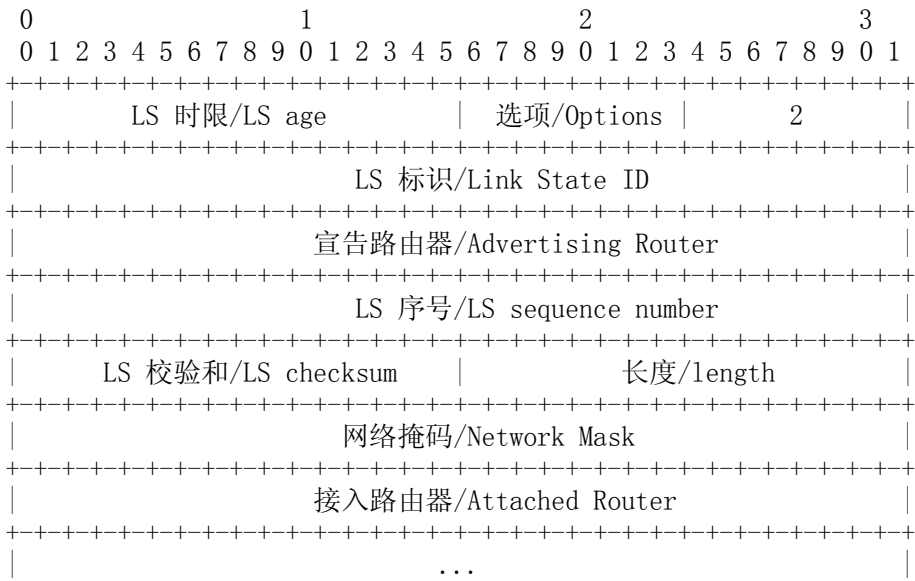
TOS 是服务类型（Type of Service）的意思。其在 LSA 中编码的描述见第 12.3 节。

TOS 距离值/TOS metric:
TOS 附加距离信息。

A.4.3 Network-LSA

Network-LSA 是类型 2 LSA。Network-LSA 是为区域中接入了两个或多个路由器的广播和 NBMA 网络而生成。Network-LSA 由网络中的 DR 生成。每个 LSA 描述了接入网络的所有路由器，包括 DR 自身。在该 LSA 中，LS 标识域表示 DR 的接口 IP 地址。

从网络到所接入路由器的距离为零。所以在 Network-LSA 中不需要距离域。关于建立 Network-LSA 的内容见第 12.4.2 节。



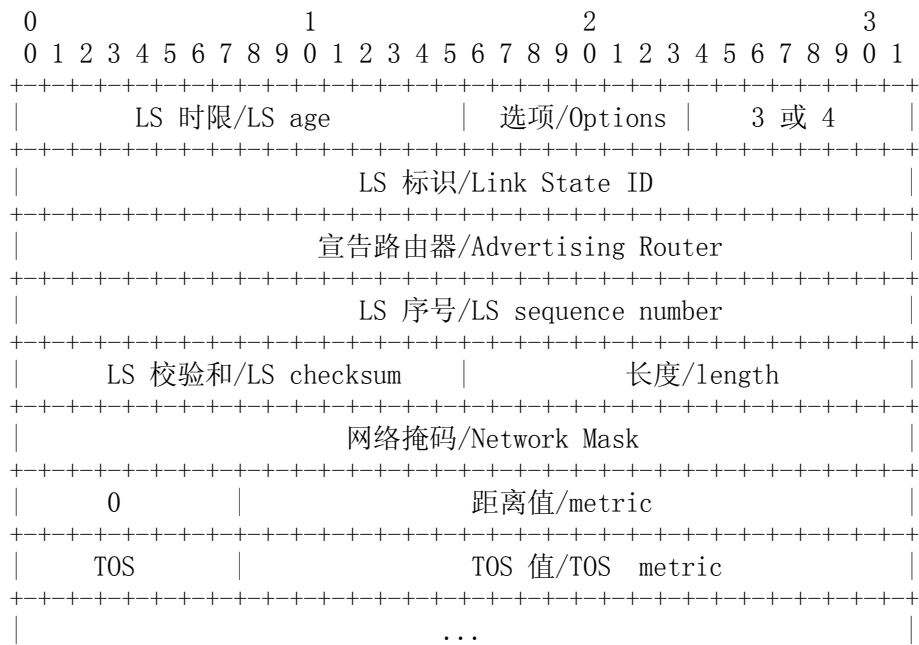
网络掩码/Network Mask:
该网络的 IP 地址掩码。例如 A 类网络的掩码为 0xff000000。

接入路由器/Attached Router:
接入该网络的各个路由器的路由器标识。注意，只有与 DR 达到完全邻接的路由器才被列出。
DR 自身也被包含在列表中。列表中的路由器数量可以从 LSA 头部中的长度域中计算而得到。

A.4.4 Summary-LSA

Summary-LSA 是类型 3 和 4 的 LSA。这些 LSA 由 ABR 生成，描述区域间目标。关于建立 Summary-LSA 的细节见第 12.4.3 节。

当目标为 IP 网络时使用类型 3 的 Summary-LSA。这时 LSA 的 LS 标识为 IP 网络号（如果需要，LS 标识会包含一位或多位主机位，细节见附录 E）。当目标为 ASBR 时，使用类型 4 的 Summary-LSA，这时 LS 标识是 ASBR 的 OSPF 路由器标识（必须宣告每台 ASBR 位置的原因见第 16.4 节）。除了 LS 标识域外，类型 3 和 4 的 Summary-LSA 格式相同。



对于存根区域，类型 3 的 Summary-LSA 同样被用于描述默认路径（每区域）。在存根区域中，不使用一系列的外部路径，而使用默认路径。当描述默认路径时，Summary-LSA 的 LS 标识始终为 DefaultDestination（0.0.0.0）并且其网络掩码被设为 0.0.0.0。

网络掩码/Network Mask:

对于类型 3 的 Summary-LSA。表示网络的 IP 地址掩码。例如，A 类网络使用 0xff000000。
对于类型 4 的 Summary-LSA，该域无意义，必须设定为 0。

距离值/metric:

路径的距离。与在 Router-LSA 中的接口距离意义相同。

可能包括附加的 TOS 信息，用于和以前版本的 OSPF 兼容（[引用 9]）。对于每条连接，对应的 TOS 附加信息按下面定义：

TOS 是服务类型（Type of Service）的意思。其在 LSA 中编码的描述见第 12.3 节。

TOS 距离值/TOS metric:

TOS 附加距离信息。

A.4.5 AS-external-LSA

AS-external-LSA 是类型 5 的 LSA。是由 ASBR 所生成，并描述 AS 外部的路径。关于建立 AS-external-LSA 的细节，见第 12.4.3 节。

AS-external-LSA 通常描述外部目标。这些 LSA 的 LS 标识域为网络的 IP 地址（如果需要，LS 标识会包含一位或多位主机位，细节见附录 E）。AS-external-LSA 同样被用于描述默认路径。当一个目标没有特定路径时使用默认路径。描述默认路径时，LS 标识始终被设定为 DefaultDestination (0.0.0.0)，网络) 并且其网络掩码被设为 0.0.0.0。



网络掩码/Network Mask:
表示所宣告目标的 IP 地址掩码。例如，A 类网络使用 0xff000000。

E 位/bit E:
外部路径类型。如果设定了 E 位，该数值就为类型 2 外部路径。这意味着该距离比任何连接状态路径的距离都大。如果 E 位为 0，说明为类型 1 外部路径。这意味着该距离和连接状态距离使用同样的单位（即与接口距离相同）。

距离值/metric:
路径的距离。其含义取决于外部类型（上面的 E 位）。

转发地址/Forwarding address:
到达所宣告目标的流量应该被转发的地址。如果转发地址被设为 0.0.0.0，数据就应当被转发到 LSA 的起源（相对应的 ASBR）。

外部路径标识/External Route Tag:

附加在每条外部路径上的 32 位数。并不被 OSPF 协议自身所使用，而是被用于 ASBR 之间的通讯。精确的描述该内容超出了本规范的范围。

可能包括附加的 TOS 信息，用于和以前版本的 OSPF 兼容 ([引用 9])。对于每条连接，对应的 TOS 附加信息按下面定义：

TOS 是服务类型 (Type of Service) 的意思。其在 LSA 中编码的描述见第 12.3 节。

E 位/bit E:

为与 [附录 9] 向下兼容。

TOS 距离值/TOS metric:

TOS 附加距离信息。

转发地址/Forwarding address:

为与 [附录 9] 向下兼容。

外部路径标识/External Route Tag:

为与 [附录 9] 向下兼容。

B. 结构常量

OSPF 协议中的一些参数使用固定值。在本文中这些参数以类似 **LSRefreshTime** 的名称被使用，可配置的参数也使用类似的命名，并在附录 C 中定义。

下面是各个常量，包括名称和简单的功能描述。

LSRefreshTime:

生成一个特定 LSA 的最大间隔。如果由路由器自己生成的 LSA 的 LS 时限达到了 **LSRefreshTime**，就需要生成 LSA 的新实例，即使 LSA 体中的内容完全一样（除了 LSA 头部）。**LSRefreshTime** 的值被设为 30 分钟。

MinLSInterval:

生成一个特定 LSA 的最小间隔，**MinLSInterval** 的值被设为 5 秒。

MinLSArrival:

对于特定的 LSA，洪泛过程中接收到 LSA 新实例之间必须间隔的时间。以更高频率到达的 LSA 实例被忽略，**MinLSInterval** 的值被设定为 1 秒。

MaxAge:

LSA 可以存在的最长时间，当 LSA 的 LS 时限域达到 **MaxAge**，为废止 LSA 将重新在路由域中洪泛（见第 14 章）。达到 **MaxAge** 的 LSA 不用于路由表计算，**MaxAge** 的值被设为 1 小时。

CheckAge:

当连接状态数据库中 LSA 的时限到达 **CheckAge** 的倍数时，检查 LSA 的有效性。错误的校验和将表示严重的错误。**CheckAge** 的值被设为 5 分钟。

MaxAgeDiff:

当一个 LSA 在 AS 中洪泛时，可以使用的最长时间。时间主要花费在洪泛过程中 LSA 在路由器输出队列中的等待（这时不被老化）。**MaxAgeDiff** 的值被设为 15 分钟。

LSInfinity:

在 LSA 中表示目标不可到达时的距离值。在 Summary-LSA 和 AS-external-LSA 中作为提前老化的一种替代方法（见第 14.1 节）。该值被定义为 24 位二进制 1：0xfffff。

DefaultDestination:

描述默认路径的目标标识。当路径不能匹配路由表项时使用该路径。默认路径仅在 AS-external-LSA 和存根区域的类型 3 Summary-LSA 中使用。该值被设为 IP 地址 0.0.0.0，关联的网络掩码也为 0.0.0.0。

InitialSequenceNumber:

生成任何第一个 LSA 实例时所使用的 LS 序号值。该值为有符号 32 位整数 0x80000001。

MaxSequenceNumber

LS 序号可以使用的最大值。该值为有符号 32 位整数 0x7fffffff。

C. 可配置变量

OSPF 协议有一些可以配置的参数。这些参数在下面列出，并按功能分组（区域参数、接口参数等）。并给出了一些参数的示例值。

一些参数需要在一组路由器上配置一致。例如，一个区域中的路由器必须使用同样的区域参数，接入同一网络的路由器必须使用同样的 IP 网络掩码。

一些参数可能是由本规范之外的路由器算法所决定（如通过 SLIP 链路连接的主机地址）。从 OSPF 的角度看，这些参数仍旧被视为可以配置。

C.1 全局参数

通常为每个区域建立一个独立的 OSPF 协议的副本。因此，大多数可配置的参数都是按区域定义的。但还是有下面一些全局参数。

路由器标识/Router ID:

一个 32 位数，在 AS 中唯一的识别出路由器。选择路由器标识的算法可以为选择在路由器上设定的最大或最小 IP 地址。如果 OSPF 路由器标识改变，在新的路由器标识起作用前，必须重新启动路由器上的 OSPF 软件。在为了改变路由器标识而重新启动前，必须将自生成的 LSA 从路由域中废止（见第 14.1 节），否则这些 LSA 会存在至多 MaxAge 分钟。

RFC1583Compatibility:

当有多条 AS-external-LSA 宣告同一目标时，按第 16.4 节的描述控制优先级规则。当设定为“enable”时，优先级规则按 RFC1583 ([引用 9]) 的规定；当设定为“disable”时，优先级规则按第 16.4.1 节的规定，以避免不同区域生成同一目标的 AS-external-LSA 时可能的路由环路。默认设定为“enable”。

为了减少路由环路的可能，路由域中所有 OSPF 路由器的 RFC1583Compatibility 设置必须相同。如果还有路由器没有更新以实现第 16.4.1 的功能，所有路由器应当设定为“enable”；否则，所有路由器应当设定为“disable”，以彻底避免路由环路。

C.2 区域参数

一个区域中的所有路由器必须使用同样的区域配置。两台不同配置的路由器将不能形成邻接，以避免路由协议流量和数据流量的通过。每个区域必须配置下面的参数：

区域标识/Area ID:

表示区域的 32 位数。区域标识 0.0.0.0 为骨干区域保留。如果区域描述了一个 IP 网络，可以使用该网络的 IP 网络地址来作为区域标识。

地址范围列表/List of address ranges:

OSPF 区域可以定义地址范围列表。每个地址范围都包括下面的项：

[IP 地址/IP address, 掩码/mask]:

描述该地址范围内的 IP 地址。网络和主机是否属于一个区域，取决于其地址是否在于区域所定义的某个地址范围之内。如果路由器所接入的网络属于多个区域，则该路由器属于多个区域。

是否宣告的状态/**Status Set to either Advertise or DoNotAdvertise:**

路由信息在区域边界被汇总。对于区域外部，大多数情况时一条路径宣告（通过 **Summary-LSA**）一个地址范围。当且仅当地址范围的状态为“宣告”时，才会宣告这条路径。通过不宣告的范围，可以使当前存在的网络对于区域外故意的隐藏。状态的默认值是“宣告”。

例如，某个 IP 网络有其自己的 OSPF 区域。该区域可以配置一个地址范围，其 IP 地址为网络地址，掩码为该类网络的本来掩码。该区域将向外部宣告一条路径，以描述整个网络。

ExternalRoutingCapability:

是否将 **AS-external-LSA** 洪泛进区域。如果 **AS-external-LSA** 被排除在区域之外，该区域就成为“存根”。在存根区域内，仅仅根据默认汇总路径决定到达外部目标的路径。骨干区域不能配置为存根区域，同样虚拟通道不能在存根区域内配置。更多信息见第 3.6 节。

StubDefaultCost:

如果区域被配置为存根区域，并且该路由器为 ABR，则 **StubDefaultCost** 表示向存根区域内部宣告的默认 **Summary-LSA** 的距离值。

C.3 路由器接口参数

某些在路由器接口上的配置参数（如 IP 接口地址和子网掩码）事实上反应了所接入网络的属性，这些属性必须在接入网络的所有路由器上保持一直。必须在路由器接口上配置的参数有：

IP 接口地址/IP interface address:

该接口的 IP 协议地址。必须在整个网络上保持唯一。在点对点网络上不一定需要 IP 地址，这种点对点网络称为“无编号”。

IP 接口掩码/IP interface mask:

也被称为网络/子网掩码，说明了 IP 接口地址中表示网络的部分。将 IP 接口地址和 IP 接口掩码与操作可以得到网络的 IP 地址。在点对点网络和虚拟通道上，没有定义 IP 接口掩码。在这些网络上，没有设定 IP 网络，即使都设定了 IP 地址，两端也可以是相互独立的。

区域标识/Area ID:

接入网络所属的 OSPF 区域。

接口输出距离值/Interface output cost:

从接口发送包的花费，表现为连接状态距离值。在路由器的 **Router-LSA** 中宣告为连接距离。接口输出距离值必须始终大于 0。

RxmtInterval:

为该接口上邻接重传 LSA 的间隔秒数。也用于重传 DD 包和 LSR 包。必须大于网络上两台路

由器之间的来回延迟。该值应当设定的较为保守以避免不必要的重传。如本地网络上的数值可以设为 5 秒。

InfTransDelay:

评估接口传送一个 LSU 包所需要的秒数。在传送前，包内各 LSA 中的时限都要增加这一值，该值应当被计入接口的传播时延。其值必须大于 0，本地网络上的值可以设为 1 秒。

路由器优先级/Router Priority:

8 位无符号整数。当接入网络的两台路由器都试图成为 DR 时，较高优先级的路由器优先。如果相同，较大路由器标识的优先。路由器标识为 0 的路由器不能成为 DR。路由器优先级仅在广播和 NBMA 网络接口上配置。

HelloInterval:

路由器接口上发送 Hello 包的间隔秒数。该值在 Hello 包中宣告，同一网络上的所有路由器必须使用相同的值。较小的 HelloInterval 可以快速的探知拓扑的改变，但使用更多的 OSPF 网络协议流量。例如 X.25 PDN 网络上被设为 30 秒；本地网络上被设为 10 秒。

RouterDeadInterval:

当不再接收到 Hello 包后，过多少秒宣告路由器关闭。该值也再 Hello 包中宣告，应当是 HelloInterval 的倍数（4 倍）。同一网络上的所有路由器也必须使用相同的值。

验证类型/AuType:

表示所接入网络上使用的验证类型。同一网络上的所有路由器必须使用相同的值。在附录 D 中定义了验证类型。

验证密码/Authentication key:

为了使接收接口验证 OSPF 协议包而配置的数据。例如，当验证类型为简单口令时，验证密码就是 64 位的明码口令。与其他 OSPF 验证类型相关联的验证密码见附录 D 的描述。

C.4 虚拟通道参数

虚拟通道是为了恢复/增强骨干区域的连通性。虚拟通道可以在任何一对接入同一区域（非骨干区域）的 ABR 之间配置。在骨干区域的连接图上，虚拟通道表现为无编号点对点网络。虚拟通道必须在两台 ABR 上都配置。

在 Router-LSA（为骨干区域）中，虚拟通道表现为接入骨干区域的独立接口（见附录 C.3）。虽然虚拟通道更象无编号点对点网络，但实际却有相关联的 IP 接口地址。该地址在建立路由表的过程中动态设定，并用于通过虚拟通道的 OSPF 协议包的 IP 源地址。接口的输出距离值等于两台路由器之间的区域内路径的距离。必须设定 RxmtInterval 参数，并应当适当的大于两台路由器之间的来回延迟，但在虚拟通道上很难估计，可以将其设定得较大。在虚拟通道上不使用路由器优先级。

虚拟通道是由下面两个可配置的参数而定义的：虚拟通道另一端点的路由器标识，以及虚拟通道使用的（非骨干）区域（或被称为虚拟通道的传输区域）。在存根区域中不能配置虚拟通道。

C.5 NBMA 网络参数

OSPF 对待 NBMA 就象是广播网络。由于在网络上接入了多个路由器，需要为网络选举一台 DR。DR 为网络生成 Network-LSA，其中包含了接入 NBMA 网络的所有路由器。

但是由于缺乏广播能力，需要一些配置参数而选举 DR。这些参数仅当不能自动发现邻居时，在能够称为 DR 的路由器（即路由器优先级不为零）上配置。

所有其他接入路由器的列表/List of all other attached routers:

所有接入 NBMA 网络的路由器列表。列表中列出各台路由器的 IP 接口地址，同时在列表中要说明路由器能否成为 DR。当 NBMA 网络接口开通时，路由器仅向能够成为 DR 的路由器发送 Hello 包，直到发现 DR。

PollInterval:

当邻居路由器不接通时（经过了 RouterDeadInterval 秒后还没有收到 Hello 包），仍旧需要向该邻居发送 Hello 包。这时 Hello 包以比 HelloInterval 更慢的速率 PollInterval 发送。例如在 X.25 PDN 上设定为 2 分钟。

C.6 点对多点网络参数

在点对多点网络上，也许需要配置通过点对多点网络能够直接到达的邻居。每台邻居由接入点对多点网络的接口 IP 地址来表示。在点对多点网络上不需要选举 DR，所以不用定义能否成为 DR。

或者，在点对多点网络上的邻居可能可以通过下层协议，诸如逆向 ARP ([引用 14])，而动态发现。

C.7 主机路径参数

主机路径在 Router-LSA 中表现为掩码为 0xfffff 的存根网络。可以通过接入点对点网络的路由器接口地址、或路由器的回环地址、或连接路由器的 IP 主机地址（也就是说，通过 SLIP 链路）来表示。对于直接连接路由器的每台主机，必须配置下面各项：

主机 IP 地址/Host IP address:

主机的 IP 地址。

连接主机的距离值/Cost of link to host:

按连接状态距离来计算的，向主机发送包的距离值。但是，由于主机往往只有这一条接入网络的路径，实际的配置值就变的不重要了（即不影响路由选择）。

区域标识/Area ID:

主机路径所属于的 OSPF 区域。

D. 验证

所有 OSPF 协议包的交换都被验证。OSPF 包头（见附录 A.3.1）中包含了验证类型域，以及 64 位用于验证的数据（取决于验证类型）。

验证类型是按接口配置的（或者说是按网络/子网）。附加的验证数据也是按接口配置的。

本规范定义了验证类型 0、1 和 2。其他所有的类型被 IANM (iana@ISI.EDU) 所保留。当前的验证类型在下面的表 20 中描述。

验证类型	描述
0	空验证
1	简单口令
2	密码验证
其他	被 IANA (iana@ISI.EDU) 所保留

表 20: OSPF 验证类型

D.1 空验证

使用该验证类型意味着在网络/子网上的路由交换没有被验证。OSPF 包头中的 64 位的验证域不包含任何内容；在接收到时也不做任何检查。在使用空验证时，OSPF 包中的所有内容（除了 64 位验证域）被计算校验和，以检查数据的正确性。

D.2 简单口令验证

使用该验证类型时，按网络配置 64 位域。所有从特定网络发送的包必须在 OSPF 包头的 64 位验证域中包含所配置的值。实际上是一个 64 位的“明文”密码。此外，OSPF 包中的所有内容（除了 64 位验证域）被计算校验和，以检查数据的正确性。

简单口令验证可以避免路由器错误的加入路由域，在参与路由前，每台路由器必须先配置该网络的密码。但是简单口令验证在当前的网络环境中很容易受到被动攻击（见 [引用 16]）。任何可以访问到该网络的人可以得到密码，并危及 OSPF 路由域的安全。

D.3 密码验证

使用该验证类型时，所有接入同一网络/子网的路由器配置有一个共享的密码。对于每一个 OSPF 路由协议包，该密码用于生成/检验加在 OSPF 包最后的“信息摘要”。该信息摘要是根据 OSPF 协议包和密码通过单向函数而得到的。由于密码从不在网络上传输，可以抵御被动攻击。

生成/检验信息摘要的算法由密码隐含决定。本规范完整地定义了使用 MD5 算法的 OSPF 密码验证。

此外，在 OSPF 协议包中还包含了一个非减的序号，以避免重复攻击。这可以提供长效保护，但在序号改变前仍有可能收到重复的 OSPF 包。为了实现这一功能，每个邻居数据结构中新加上“密码序号/cryptographic sequence number”域。该域初始设为零。

当邻居状态变为“Down”，当接收到一个可信的 OSPF 包时，密码序号被设定为接收包中的序号。

本规范并没有提供序号回滚的功能。当发送的序号达到最大值时，路由器应当重设序号为 0。这时，邻居路由器会在 RouterDeadInterval 时间内拒绝该路由器 OSPF 包，并试图在接口上重新建立所有的邻接。但是，事实上很多实现使用“自启动的秒数”（或“自 1960 年来的秒数”）作为密码序号。由于密码序号长达 32 位，这一选择事实上避免了回滚。

OSPF 密码验证选项不提供机密性。

当使用密码验证时，标准 OSPF 包头中的验证域按图 18 定义。其中的新域在下面定义：

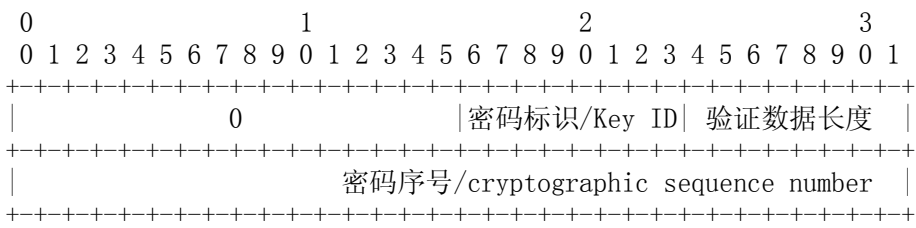


图 18：使用密码验证时 OSPF 包头中的验证域

密码标识/Key ID:

该域表示 OSPF 包头中生成信息摘要的算法和密码。每个接口（或者说每个子网）的密码标识唯一。

验证数据长度/Auth Data Len:

追加到 OSPF 包后面的信息摘要的字节长度。

密码序号/ cryptographic sequence number:

32 位无符号非减数值。用于抵御重复攻击。

追加到 OSPF 包的信息摘要事实上不被看作 OSPF 协议包的一部分：信息摘要的长度不包括在 OSPF 包头的长度中，虽然它被包含在 IP 包头的长度中。

每个密码由接口和密码标识的组合来识别。一个接口上可能同时有多个密码被激活，这是为了平滑地过度到新密码。每个密码有相关联的四个常量，这些时间常量可以用实时钟或路由器本地时钟（即从上次重启来的秒数）来表示：

KeyStartAccept:

从该时间起，路由器将开始接受以该密码生成的包。

KeyStartGenerate:

从该时间起，路由器将开始以该密码生成数据包。

KeyStopGenerate:

从该时间起，路由器将不再以该密码生成数据包。

KeyStopAccept:

从该时间起，路由器将不再接受以该密码生成的包。

为了达到密码的平滑过度，**KeyStartAccept** 应当小于 **KeyStartGenerate**，并且 **KeyStopGenerate** 应当小于 **KeyStopAccept**。如果没有定义 **KeyStopGenerate** 和 **KeyStopAccept**，该密码的生存周期为无限。如果用新密码代替旧的，新密码的 **KeyStartGenerate** 必须小于或等于旧密码的 **KeyStopGenerate**。

不论经过系统的热启动或冷启动，密码都应当被保持，以避免操作中断。如果接口上最后一个密码过期，不应当回到非验证的情况，也不应当中断路由。这时，路由器应当向网管发出“最后的密码过期”信息，并扩展该密码的生存周期为无限，直到网管删除该密码或配置了新的密码。

D.4 信息生成

当生成了 OSPF 包以后，在从接口发送前，应当根据接口的验证类型调用验证过程。验证过程按下面的描述修改 OSPF 包：

D.4.1 生成空验证

当使用空验证时，按如下修改包：

(1) OSPF 包头中的验证类型域被设定为 0。

(2) OSPF 包头中的校验和域被设定为整个包的标准 IP 校验和，计算从 OSPF 包头开始但不包括 64 位验证域。按 16 位 1 补足校验和计算包中除了验证域外的 16 位字，如果包长度不是 16 位字的整数倍，在校验和前加上一个字节的 0 补足。

D.4.2 生成简单口令验证

当使用简单口令验证时，按如下修改包：

(1) OSPF 包头中的验证类型域被设定为 1。

(2) OSPF 包头中的校验和域被设定为整个包的标准 IP 校验和，计算从 OSPF 包头开始但不包括 64 位验证域。按 16 位 1 补足校验和计算包中除了验证域外的 16 位字，如果包长度不是 16 位字的整数倍，在校验和前加上一个字节的 0 补足。

(3) OSPF 包头中的 64 位验证域被设定为按接口配置的 64 位口令（即验证密码）。

D.4.3 生成密码验证

当使用密码验证时，接口上可能配置有多个密码。这时，从有效生成密码中（即 **KeyStartGenerate** \leq 当前时间 $<$ **KeyStopGenerate**）选择具有最近 **KeyStartGenerate** 时间的密码。使用该密码，按如下修改包：

- (1) OSPF 包头中的验证类型域被设定为 2。
- (2) 不计算标准 OSPF 包头中的校验和，而将其设为 0。
- (3) 将密码标识（见图 18）设定为所选择密码的密码标识。
- (4) 将验证数据长度域设定为追加到 OSPF 包的数据摘要长度。当使用 MD5 验证算法时，验证数据长度为 16。
- (5) 将 32 位的密码序号（见图 18）设定为非减数值（不能小于上次从该接口发出的数值）。具体选择的数值根据不同的实现而不同。例如可以基于简单的计数器，或基于系统时钟。
- (6) 计算信息摘要并追加到 OSPF 包。所使用的验证算法由密码自身决定，将 OSPF 包和密码作为输入到验证算法。当使用 MD5 验证算法时，按如下计算信息摘要：
 - (a) 将 16 字节的 MD5 密码追加到 OSPF 包。
 - (b) 按 [引用 17] 的说明，加入尾部填充和长度域。
 - (c) 将 OSPF 包、密码、填充和长度域串联，运行 MD5 验证算法，得到 16 字节信息摘要（见 [引用 17]）。
 - (d) 将 MD5 摘要覆盖 OSPF 密码（即追加到原始的 OSPF 包后）。摘要不被包含在 OSPF 包长度中，但包含在 IP 包长度中。摘要后面的尾部填充和长度域不被计算长度也不被传送。

D.5 信息校验

当从接口接收到 OSPF 包后，必须进行验证。验证类型由标准 OSPF 包头中的验证类型域所说明，并且与 OSPF 接收接口的配置相匹配。

如果 OSPF 包被接收并验证，按第 8.2 节的描述继续处理。验证失败的包被丢弃。

D.5.1 校验空验证

当使用空验证时，必须检查 OSPF 包头中的校验和的有效性。其按 16 位 1 补足校验和计算包中除了验证域外的 16 位字（如果包长度不是 16 位字的整数倍，在校验和前加上一个字节的 0 补足）。

D.5.2 校验简单口令验证

当使用简单口令验证时，按如下检验所收到的 OSPF 包：

- (1) 必须检查 OSPF 包头中的校验和的有效性。其按 16 位 1 补足校验和计算包中除了验证域外的 16 位字（如果包长度不是 16 位字的整数倍，在校验和前加上一个字节的 0 补足）。

(2) OSPF 包头中的 64 位验证域必须等于接口上配置的 64 位口令（即验证密码）。

D.5.3 校验密码验证

当使用密码验证时，按如下检验所收到的 OSPF 包：

(1) 查找接口上所配置密码中，密码标识等于所接收 OSPF 包中密码标识（见图 18）的密码。如果没有找到密码，或密码对于接收无效（即当前时间 < KeyStartAccept，或当前时间 > KeyStopAccept），丢弃该 OSPF 包。

(2) 如果 OSPF 包头中的密码序号（见图 18），小于发送该包的邻居数据结构中的密码序号，丢弃该 OSPF 包。

(3) 按下面的步骤检验信息摘要：

(a) 保存所接收到的摘要。

(b) 按附录 D.4.3 中的第 6 步计算新的摘要。

(c) 将计算出的摘要与所接收到的进行比较。如果不匹配，丢弃该 OSPF 包；如果匹配，该 OSPF 包就通过了验证，并将邻居数据结构中的“密码序号”设定为所接收到的 OSPF 包头中的密码序号。

E. 设定 LS 标识的一种算法

AS-external-LSA 和 Summary-LSA 中的 LS 标识通常被设定为所描述的网络 IP 地址。但是，有时需要在连接标识上加上一位或多位主机位。这使得路由器可以为拥有相同网络地址，但不同掩码的网络生成不同的 LSA。当网络中存在超网和零子网时会出现这种情况（见 [引用 10]）。

本附录给出了在 LS 标识中加上主机位的一种可行算法。算法的选择是独立的，因为仅对路由器自己生成的 LSA 起作用，所以不同的路由器可以选择不同的算法。所必须的是，应当尽可能得将 LS 标识设定为网络 IP 地址，这将与早先根据 RFC 1583 的实现达到最大互通。

下面的算法是以 AS-external-LSA 为例的。这仅仅是为了描述的明确，完全相同的算法可以用于 Summary-LSA。假设路由器要为一个网络生成 AS-external-LSA，其地址为 NA，掩码为 NM1。使用下面的步骤来决定该 LSA 的 LS 标识。

（1）检查路由器是否已经生成了一个 AS-external-LSA，其 LS 标识等于 NA（LSA 中该路由器为 LSA 的宣告路由器）。如果没有，设定 LS 标识为 NA，并终止算法。否则，

（2）取得已经存在 AS-external-LSA 的 LS 标识，称其为 NM2。有两种情况：

NM1 比 NM2 要长（即更精确）。这时，设定表示网络 [NA,NM1] 的新 LSA 中 LS 标识为全部主机位（即等于 NA 加上 NM1 的反码，也就是网络 [NA,NM1] 的广播地址）。

NM2 比 NM1 要长。通过增加序列号，使已存在的 LSA（其 LS 标识为 NA）改为表示新的网络 [NA,NM1]，改变 LSA 体中的掩码为 NM1，并设定新的距离值。然后为旧网络 [NA,NM2] 生成新的 LSA，设定其 LS 标识为 NA 加上 NM2 的反码（即网络 [NA,NM2] 的广播地址）。

上面的算法假设所有的掩码是连续的，当两个网络有相同的网络地址时，可以确保一个掩码比另一个更精确。该算法同时假设一个网络的网络地址不会等于另一个网络的广播地址。在满足上面两个假设时，上述算法始终能生成唯一的 LS 标识。上面的算法也可以表述如下：当生成 AS-external-LSA 时，首先试图将其 LS 标识设定为网络地址。如果有冲突，检查所冲突的两个网络，其中一个网络将是另一个的子网。对于不精确的网络，设定 LS 标识为网络地址；而较精确的网络则使用该网络的广播地址（即主机位全 1）。如果先生成的是较精确的网络，则需要同时生成两个 LSA。

作为该算法的例子，考虑在一台路由器（路由器 A）上发生的一系列事件。

（1）路由器 A 为网络 [10.0.0.0,255.255.255.0] 生成 AS-external-LSA:

（a）使用 10.0.0.0 作为 LS 标识。

（2）路由器 A 又要为网络 [10.0.0.0,255.255.0.0] 生成 AS-external-LSA:

（a）为网络 [10.0.0.0,255.255.255.0] 重新生成新的 LS 标识为 10.0.0.255。

（b）网络 [10.0.0.0,255.255.0.0] 使用 LS 标识 10.0.0.0。

(3) 路由器 A 又要为网络 [10.0.0.0,255.0.0.0] 生成 AS-external-LSA:

(a) 为网络 [10.0.0,0,255.255.0.0] 重新生成新的 LS 标识为 10.0.255.255。

(b) 网络 [10.0.0.0,255.0.0.0] 使用 LS 标识 10.0.0.0。

(c) 网络 [10.0.0.0,255.255.255.0] 仍旧使用 LS 标识 10.0.0.255。

F. 多接口接入同一网络/子网

有至少两种方法支持多物理接口接入同一 IP 子网。所有方法都与 RFC 1583 兼容（当然也和本规范兼容）。这两种方法大略的在下面描述。在这里假设各个接口都设定了不同的 IP 地址（否则，支持多接口就是链路层的功能而非 OSPF 需要实现）。

方法 1:

在各个接口上运行完整的 OSPF 功能，发送接收 Hello 包、洪泛、在各接口上执行邻居有限机等等。这时，因为邻居是通过 IP 地址而识别的（在广播和 NBMA 网络上），网络上的其他路由器将两个接口看作为不同的邻居。

方法 1 有以下缺点：

- （1）增加了邻居和邻接的数量。
- （2）因为双向通讯是基于路由器标识的，在两个接口上都失去了双向测试。
- （3）必须将两个接口合并来参与 DR 选举，不然会产生混乱（DR 选举是基于路由器标识）。

方法 2:

仅在一个接口上运行 OSPF（称之为主接口），但在 Router-LSA 中包含主从接口。

方法 2 有以下缺点：

- （1）在从接口上失去了双向测试。
- （2）当主接口失效时，需要将从接口提升为主工作状态。

G. 与 RFC 2178 的不同

本附录描述本规范与 RFC 2178 的不同。所有的不同都提供向后兼容。按本规范的实现可以与按照 RFC 2178、1583 和 1247 的实现协同操作。

G.1 洪泛过程的修改

第 13 章的洪泛过程中有三处改动。

第一处改变是第 13 章的步骤 4。现在达到 MaxAge 的 LSA 仅在下面情况下被确认并丢弃：a) 数据库中没有该 LSA 的副本，和 b) 没有邻居路由器的状态是 Exchange 或 Loading。在其他情况时，达到 MaxAge 的 LSA 就象其他 LSA 一样处理，当比数据库中的副本较新时，被装入数据库并洪泛处相关的接口（第 13 章的步骤 5）。这些修改同样影响表 19 的内容。

第二处改变是第 13 章的步骤 5a。仅在接受到洪泛时，对 MinLSArrival 进行检查，并不影响那些由路由器自己生成的 LSA。

第三处改变是第 13 章的步骤 8。对新旧 LSA 实例的混淆会导致连接状态协议的崩溃（见[引用 26]）。OSPF 用两种方法来避免这个问题：a) 在洪泛中，象 TTL 一样处理 LS 时限，以避免 LSA 在网络中的循环（见第 13.3 节），和 b) 路由器不接收比每 MinLSArrival 秒一次更高频率的 LSA 更新（见第 13 章）。但在 RFC 2178 中提出了一种情况，认为较新的 LSA 可能产生很多洪泛流量：为了回应旧的 LSA，需要重新洪泛数据库中的副本。因此，第 13 章中的步骤 8 变成了，仅当在最近 MinLSArrival 秒内没有被 LSU 包发送时，才用数据库中的副本回应。

G.2 外部路径优先级的改变

按照 RFC 2178 的实现，在满足下面两个条件时，仍有可能产生路由环路：a) 使用虚拟连接，和 b) 同一条外部路径，被不同区域的多个 ASBR 所引入。为了解决这个问题，提出了第 16.4.1 节。在选择正确的 ASBR/转发地址时，始终优先选择非骨干区域的区域内路径；而区域间路径和骨干区域（区域 0）的区域内路径优先级相同，而必须比较距离值。

这一改变背后的原因是：当使用虚拟通道时，骨干区域的区域内路径可能变为一条更短的区域间路径。所以区域间路径和骨干区域的区域内路径必须有相同的优先级。仅需要比较其距离值，而按照第 16.3 节的计算取得最佳路径。

感谢 UNH InterOperability Lab 的 Michael Briggs 和 Jeremy McCooey 提出了这个问题。

G.3 解决不完整的虚拟下一跳

第 16.3 节计算的其中一项是计算虚拟通道的下一跳，这些目标是在第 16.1 和 16.2 节中计算的。当第 16.3 节的计算完成后，还没有得出下一跳的路径应当被丢弃。

G.4 路由表查找

第 11.1 节中的路由表查找算法被修改以满足当前需要。最精确（最长）匹配的路由表项始终是“最

佳匹配”的。假设有到达 192.9.1.1 的包要转发，192.9.1/24 的路由表项始终优于 192.9/16 的项，而不考虑其路径类型。注意，当给定的路由表项提供了多条路径时，在第 16.1、16.2 和 16.4 节中的计算始终选择最优先的路径类型（区域内路径最优先，其次分别为路径间路径、类型 1 外部路径和类型 2 外部路径；见第 11 章）。

安全性考虑

所有的 OSPF 包交换都经过验证。OSPF 支持多种验证类型，验证的类型可以按网络来配置。其中一种 OSPF 验证类型，称为密码验证/Cryptographic authentication 选项，被认为可以安全地抵御被动攻击并对主动攻击提供保护。在使用密码验证选项时，路由器在所传送的 OSPF 包后追加了“信息摘要”；接收路由器使用共享的密码和接收到的摘要来验证每个收到的 OSPF 包。

密码验证所提供的安全性取决于信息摘要算法的强壮性（当前仅指定了 MD5 算法）；所使用密码的强壮性；以及相互通讯的 OSPF 实现之间的安全机制。同时也需要各方都保持密码的安全。

OSPF 验证不提供机密性，也不能抵御流量分析。密码的管理不在本备忘录的范围之内。

更多信息，见第 8.1、8.2 节和附录 D。

作者的地址

John Moy
Ascend Communications, Inc.
1 Robbins Road
Westford, MA 01886

Phone: 978-952-1367
Fax: 978-392-2075
EMail: jmoy@casc.com

译者的通讯方式

马儿快跑

EMail: raymon@itpub.net

另：由于还没有完全阅读 **RFC 1583**，对于其中 **TOS** 值、点对点接口设定的理解还不明确。对应部分的翻译不够理想，欢迎大家指正。

完整的版权声明

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.