使用 SNMP 管理 OSPF 配置

目录

简介

开放最短路径优先(OSPF)路由协议由RFC 2328 OSPF第2版定义 。本白皮书的目标是提供一个过程框架，使组织能够实施配置管理程序，以根据OSPF设计计划检验OSPF部署，并定期审核OSPF部署，以确保与预期设计的长期一致性。

本文重点介绍ITU-T定义的FCAPS（故障、配置、记帐/库存、性能、安全）模型中的配置管理功能

。配置管理由ITU-T M.3400定义，它提供对NE（网络元素）进行控制、识别、收集数据和提供数据的功能。

本文提供的信息将在以下几个主要部分介绍。

OSPF背景部分提供OSPF技术概述，包括有关OSPF部署重要方面的背景信息。

“进程定义”部分概述了用于完成OSPF配置管理的进程定义。流程详细信息按目标、绩效指标、输入、输出和单个任务进行描述。

“任务定义”部分提供了详细的流程任务定义。每项任务都以目标、任务输入、任务输出、完成任务所需的资源以及任务实施者所需的工作技能来描述。

数据标识部分介绍OSPF的数据标识。数据标识考虑信息的来源或信息所在的位置。例如，系统在简单网络管理协议(SNMP)管理信息库(MIB)、系统日志生成的日志文件或只能通过命令行界面(CLI)访问的内部数据结构中包含信息。

本文的“数据收集”部分介绍OSPF数据的收集。数据的收集与数据的位置密切相关。例如，SNMP MIB数据的收集采用几大机制，例如陷井、远程监控(RMON)告警与事件、或轮询。由内部数据结构维护的数据由自动脚本或用户手动登录系统以发出CLI命令，然后记录输出来收集。

“数据演示”部分提供了如何以报告格式显示数据的示例。识别并收集数据后，对数据进行分析。本白皮书提供可用于记录和比较OSPF配置数据的示例报告。

商业和公共Internet监控工具、SNMP轮询数据和示例数据收集算法部分提供了有关开发工具以实施OSPF配置管理过程的信息。

OSPF 背景

OSPF是一种内部网关协议，设计用于单个自治系统中。OSPF使用基于链路状态或最短路径优先(SPF)的技术，与路由协议(如路由信息协议(RIP))中的距离矢量或贝尔曼 — 福特技术相比，OSPF采用的是这种技术。 单个链路状态通告(LSA)描述OSPF路由域（例如整个自治系统）的部分。这些LSA在路由域中泛洪，形成链路状态数据库。域中的每台路由器都有相同的链路状态数据库。链路状态数据库的同步使用可靠的泛洪算法进行维护。从链路状态数据库，每台路由器通过计算最短路径树来构建路由表，树的根是计算路由器本身。此计算通常称为Dijkstra算法。

LSA很小，每个LSA都描述OSPF路由域的一小部分，具体来说，是单台路由器的邻居、单个中转网络的邻居、单个区域间路由或单个外部路由。

下表定义了OSPF的主要功能：

功能	描述
邻接	当成对的OSPF路由器相邻时，两台路由器通过以OSPF数据库交换数据包的形式交换数据库摘要来同步其链路状态数据库。然后，相邻路由器通过可靠的泛洪算法维护其链路状态数据库的同步。通过串行线路连接的路由器始终是邻接的。在多路访问网络(Ethernet)上，连接到网络的所有路由器都与指定路由器(DR)和备用指定路由器(BDR)相邻。
指	当在所有多路访问网络上选举DR时，它会生成描述网

定路由器	络本地环境的网络LSA。在泛洪算法中，它也发挥了特殊作用，因为在泛洪过程中，网络上的所有路由器都通过向DR发送和接收LSA以及从DR接收LSA来同步其链路状态数据库。
备份指定路由器	当当前DR消失时，在多路访问网络上选择BDR以加速DR的过渡。当BDR接管时，它无需在局域网(LAN)上完成邻接过程。 BDR还使可靠泛洪算法能够在DR不存在时继续执行，然后发现DR消失。
非广播多路访问网络支持	OSPF将网络(如帧中继公共数据网络(PDN))视为LAN。但是，连接到这些网络的路由器需要额外的配置信息才能开始找到彼此。
OSPF配置管理区域	OSPF允许将自治系统分为多个区域。这提供了额外的路由保护级别，以便保护区域内的路由免受区域外部的所有信息的影响。此外，通过将自治系统划分为多个区域，Dijkstra过程的成本（以CPU周期而言）会降低。
虚拟链路	通过允许配置虚拟链路，OSPF消除了自治系统中区域布局的拓扑限制。
路由协议交换的身份验证	每次OSPF路由器收到路由协议数据包时，都可以选择在进一步处理数据包之前对数据包进行身份验证。
灵活的路由	在OSPF中，度量分配给出站路由器接口。路径的开销是路径的组件接口之和。默认情况下，路由度量源自链路的带宽。系统管理员可以指定它来指示延迟、带宽和开销等网络特征的任意组合。

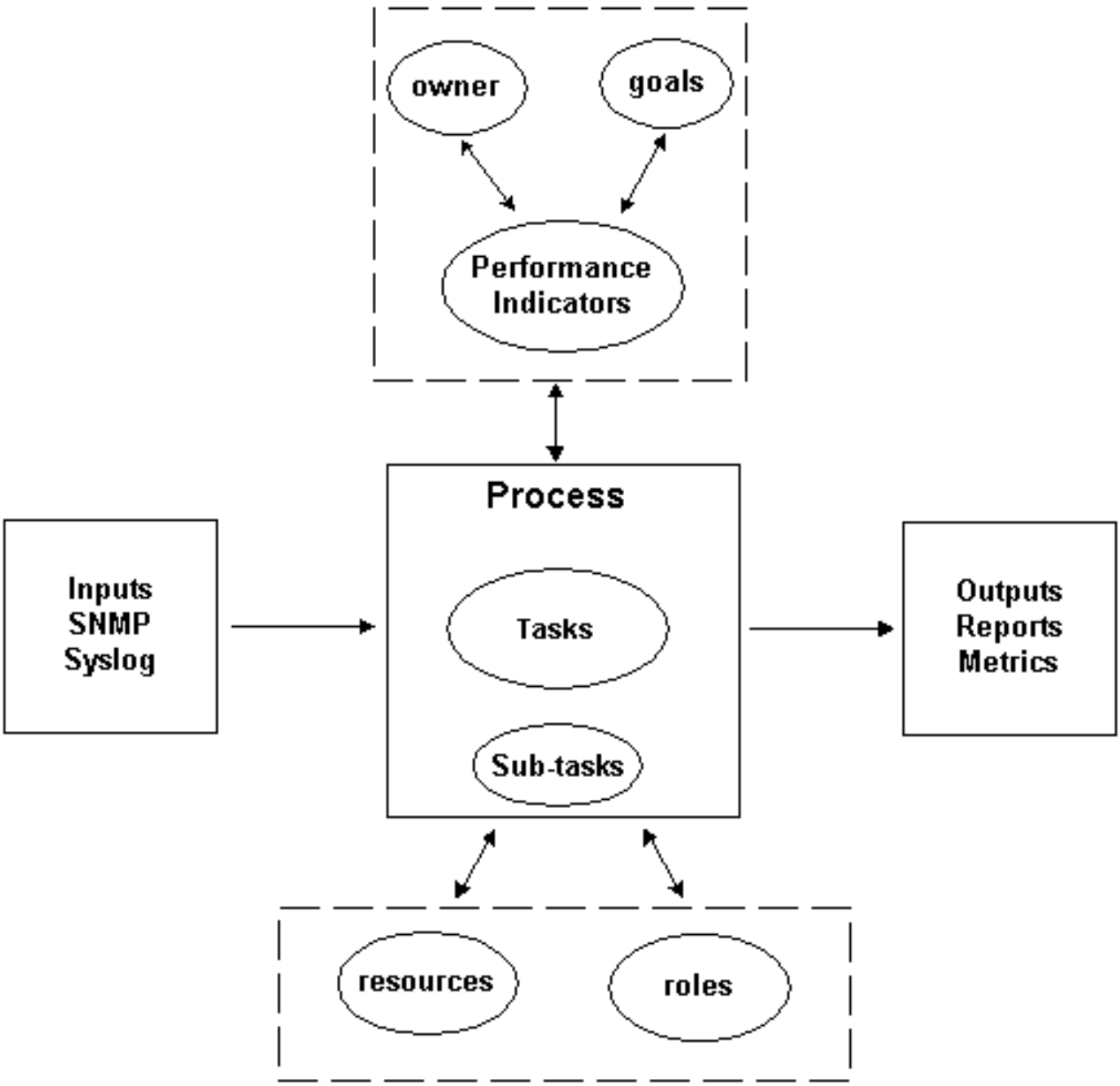
度量	
等价多路径	当存在到目的地的多条开销最高的路由时，OSPF会查找并使用它们来加载到目的地的共享流量。
可变长子网支持	通过携带每个通告目的地的网络掩码来支持可变长子网掩码。
末节区域支持	为了支持内存不足的路由器，可将区域配置为末节。外部LSA不会泛洪到末节区域及整个末节区域。末节区域中到外部目的地的路由完全基于默认值。

进程定义

进程定义是代理执行的一系列连续动作、活动和变化，目的是满足目的或达到目标。

进程控制是规划和调控过程，以便采用有效和高效的方式执行进程目标。

图形显示如下图所示。



流程的输出必须依照组织定义的操作规范，并且基于业务目标。如果程序依照某套标准，并且程序可以重复被执行、能被测量和被管理，并且对业务目标有用，程序则视为有效，如果以最小的努力进行活动，则该过程也被视为有效。

进程所有者

流程跨越各种组织边界。所以，拥有负责流程定义的单个进程所有者很重要。所有者是确定和报告流程是否有效且高效的焦点。如果该进程无效，那么进程拥有者加建将执行进程修改。流程的修改由变更控制和审核流程管理。

进程目标

建立流程目标以设置流程定义的方向和范围。目标也用于定义用来测量进程效果的尺度。

此过程的目标是提供一个框架，以根据预期设计验证OSPF实施的部署配置，并提供一种定期审核OSPF部署的机制，以确保在一段时间内与预期设计保持一致。

进程性能指示器

流程绩效指标用于衡量流程定义的有效性。绩效指标应可衡量和量化。下面列出的绩效指标为数字或按时间测量。OSPF配置管理进程的性能指标定义如下：

- 在整个流程中循环所需的时间。
- 在OSPF问题影响用户之前主动检测这些问题所需的执行频率。
- 与进程执行相关的网络负载。
- 流程建议的纠正措施数。
- 由此流程实施的纠正措施数。
- 实施纠正措施所需的时间长度。
- 实施纠正措施所需的时间长度。
- 纠正措施积压。
- 停机时间由OSPF相关问题引起。
- 在种子文件中添加、删除或修改的项目数。这表明了准确性和稳定性。

进程输入

流程输入用于定义流程的标准和先决条件。很多时候，识别流程输入提供有关外部依赖性的信息。下面提供了与OSPF配置管理相关的输入列表。

- OSPF设计文档
- 通过SNMP轮询收集的OSPF MIB数据
- 系统日志信息

进程输出

流程输出定义如下：

- OSPF配置报告在本白皮书的数据演示部分中定义
- OSPF配置建议，用于执行纠正措施

任务定义

以下各节定义与OSPF配置管理相关的初始化和迭代任务。

初始化任务

初始化任务在流程实施期间执行一次，不应随流程的每次迭代一起执行。

检验必备任务

检验必备任务时，如果确定任何某项任务不能实施，也不能提供充足的信息，来有效满足该程序的需求，程序所有者则应当文件记录该事实，并提交给管理人员。下表概述了必备的初始化任务。

前期任	描述

务	
任务目标和输入	1. 验证OSPF设计文档是否存在，以及网络设计文档中是否可以随时获取以下信息： 区域定义 — 名称、地址范围和区域类型区域边界路由器/自治系统边界路由器(ABR/ASBR)标识DR/BDR标识Internet注册(IR)节点和分配给区域的接口 2. 使用SNMP标准配置模板检验SNMP是否在网络中配置。 **注意：**稍后将其用作创建种子文件的输入。 3. 使用系统日志标准配置模板验证系统日志正在网络中部署。
任务输出	任务输出是有关必备任务状况的状态报告。如果任何支持任务被视为无效，流程所有者应提交请求以更新支持流程。如果支持流程无法更新，请对此流程的影响进行评估。
任务角色	网络工程师技能集

创建种子文件

OSPF配置管理过程需要使用种子文件来消除对网络发现功能的需求。种子文件记录由OSPF进程管理的一组路由器，并用作与组织中的变更管理进程协调的焦点。例如，如果新节点输入到网络中，则需要将其添加到OSPF种子文件。如果由于安全需求而变更SNMP属性名称，那些修改应反映在种子文件中。下表概述了创建种子文件的过程。

Process	**描述**
任务目标	创建用于初始化OSPF配置管理软件的种子文件。种子文件的格式取决于用于实施OSPF配置管理过程的资源。如果开发了自定义脚本，则种子文件的格式由软件设计定义。如果使用网络管理系统(NMS)，则种子文件的格式由NMS文档定义。
任务输入	1. 格式化种子文件。 2. 使用OSPF设计文档确定以下数据： 所有节点的IP地址SNMP 社区字符串Telnet和CLI登录帐户和密码 3. 安排网络变更管理流程的联系人姓名。
任务输出	OSPF配置管理进程的种子文件。
任	• 商业NMS系统

务资源	• 自定义开发的软件系统 • 手动进程 — 登录每个网络元素并发出命令行并记录输出。
任务角色	• NMS — 网络工程师、NMS管理员和NMS脚本技能集。 • 自定义脚本 — 网络工程师和NMS脚本技能集。 • 手动流程 — 网络工程师。

重复任务

迭代任务与过程的每次迭代一起执行，并且确定和修改其频率以改进性能指标。

维护种子文件

种子文件对有效实施OSPF配置管理过程至关重要。因此，必须主动管理种子文件的当前状态。影响种子文件内容的网络更改需要由OSPF配置管理进程所有者跟踪。

Process	描述
任务目标	1. 通过跟踪和与控制网络移动、添加、更改和/或网络配置修改的组织功能交互来维护种子文件的币种。 2. 维护种子文件的版本控制和备份控制。
任务输入	1. 来自更改管理的信息（如移动、添加和更改）会影响种子文件的内容。 2. 影响种子文件内容的工程/设计信息。
任务输出	1. 有关种子文件币种状态的每周报告。 2. 介绍种子文件备份的位置和恢复过程的定义和文档。
任务资源	• 商业NMS系统 • 自定义开发的软件系统 • 手动进程 — 登录每个网络元素并发出命令行并记录输出。
任务角色	• NMS — 网络工程师、NMS管理员和NMS脚本技能集。 • 自定义脚本 — 网络工程师和NMS脚本技能集。 • 手动流程 — 网络工程师。

执行OSPF扫描

执行OSPF扫描的两个步骤是：

1. 收集数据。
2. 分析数据。

根据过程的使用方式，这两个步骤的频率会有所不同。例如，此过程可用于验证安装修改。在这种情况下，数据收集在更改前后运行，数据分析在更改后进行，以确定更改的成功。

如果此过程用于验证OSPF配置管理设计记录，则数据收集和分析频率取决于网络中的更改率。例如，如果网络发生重大更改，则每周进行一次设计验证。如果网络变化很小，则每月不超过一次进行设计验证。

查看OSPF报告

OSPF配置管理报告的格式取决于用于实施OSPF配置管理过程的资源。下表提供建议的自定义开发报告格式。

报告	格式
任务输入	有关OSPF配置管理报告，请参阅本文档中的数据演示部分。
任务输出	如果在扫描报告和逻辑设计记录之间发现问题，则必须确定哪项正确，哪项不正确。应更正错误的项目。这可能涉及修改设计记录或网络更改顺序。
任务资源	• 商业NMS系统 • 自定义开发的软件系统 • 手动 — 登录每个网络元素并发出命令行并记录输出
任务角色	• NMS — 网络工程师、NMS管理员和NMS脚本技能集。 • 自定义脚本 — 网络工程师和NMS脚本技能集。 • 手动流程 — 网络工程师。

数据识别

一般数据特性

下表介绍可应用于OSPF配置管理的数据。

数据	描述
OSPF 区域	描述路由器连接区域的信息包括： • 区域 id • 区域验证 • SPF运行 • 区域中的ABR数 • 区域中的ASBR数 • 区域LSA计数 — 区域中路由器之间的一致性 • 区域LSA校验和 — 区域中路由器之间的一致性

	• 由于每个区域存在编址错误而丢弃数据包的频率 • 每个区域的路由进程丢弃协议数据包的频率 • 由于每个区域找不到路由条*件，路由数*据包丢弃频率
OSPF接口	从OSPF的角度描述一个接口，例如： • IP 地址 • 区域 id • 管理状态 • 分配给接口的OSPF度量 • 分配给接口的OSPF计时器 • OSPF状态
OSPF邻居状态	描述OSPF邻居。 • 邻居路由器 ID • 邻居状态 • 邻居事件 — 邻居关系更改状态或发生错误的次数。 • 邻居重新传输队列 — 重新传输队列的当前长度。

SNMP 数据识别

Cisco目前支持RFC 1253 OSPF第2版MIB 。RFC 1253不包含OSPF的SNMP陷阱定义。OSPF MIB的最新版本是RFC 1850 OSPF第2版 。RFC 1850中为OSPF定义了SNMP陷阱。思科实施OSPF MIB时不支持RFC 1850。

有关详细信息，请参阅本文档的SNMP轮询数据部分。

请参阅思科网络管理软件页面，了解哪些平台和代码版本支持哪些MIB的确定列表。

RMON 数据识别

此过程不需要RMON特定数据。

系统日志数据识别

通常，系统日志会为不同技术生成特定于服务的消息。虽然系统日志信息更适合于故障和性能管理，但此处提供的信息是参考。有关Cisco设备生成的OSPF系统日志信息的示例，请参见OSPF错误消息。

有关按设施列出的系统消息的完整列表，请参阅消息和恢复过程。

CISCO IOS CLI 数据识别

在此版本的OSPF配置管理过程中，不需要CLI数据。

数据收集

SNMP 数据收集

下表定义了SNMP数据收集的不同组件。

SNMP数据组件	定义
一般 SNMP配置	有关SNMP配置最佳实践的一般信息，请参阅配置SNMP。
服务特定SNMP配置	此过程不需要服务特定的SNMP配置。
SNMP MIB要求	请参阅上面的“数据标识”部分。
SNMP MIB轮询收集	SNMP轮询数据由商业系统(如hp OpenView)或自 定义脚本收集。有关收集算法的进一步讨论，请参阅本文档的数据收集算法示例部分。
SNMP MIB陷阱收集	思科设备支持的当前OSPF MIB版本不支持SNMP陷阱。此过程不需要SNMP陷阱。

RMON 数据收集

此版本的过程不需要RMON配置和数据。

系统日志数据收集

一般系统日志配置指南不在本文档的讨论范围之内。有关详细信息，请参阅使用单个内部网络配置和排除Cisco Secure PIX防火墙故障。

通过使用以下命令配置OSPF路由器，使用系统日志消息记录邻居更改，从而满足OSPF特定要求：

```
OSPF_ROUTER(config)# ospf log-adj-changes
```

CISCO IOS CLI 数据收集

通常，Cisco IOS CLI可以最直接地访问NE包含的原始信息。但是，CLI访问更适合故障排除步骤和变更管理活动，而非此步骤定义的全局配置管理。通过CLI的访问无法扩展以管理大型网络。在这些情况下，需要自动访问信息。

在此版本的OSPF配置管理过程中，不需要CLI配置和数据。

数据表示

OSPF 区域报告

以下是OSPF区域报告的示例格式。报告的格式由商业NMS的功能（如果使用）或自定义脚本的设计输出确定。

区域	数据字段	上次运行	此运行
区域ID #1	身份验证		
	SPF运行		
	ABR计数		
	ASBR计数		
	LSA计数		
	LSA校验和		
	地址错误		
	路由丢弃		
	未找到路由		
区域ID #n	身份验证		
	SPF运行		
	ABR计数		
	ASBR计数		
	LSA计数		
	LSA校验和		
	地址错误		
	路由丢弃		
	未找到路由		

OSPF 接口报告

以下是OSPF接口报告的示例格式。实际上，报告的格式由商业NMS（如果使用）的功能或定制脚本的设计输出确定。

区域	设备	接口	数据字段	上次运行	此运行
区域 ID #1	节点 ID #1	接口 ID #1	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		
		接口 ID #n	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		
	节点 ID #n	接口 ID #1	IP Address		
			区域 id		

			管理州		
			OSPF状态		
			指标/成本/计时器		
		接口ID #n	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		
区域ID #n	节点ID #1	接口ID #1	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		
		接口ID #n	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		
	节点ID #n	接口ID #1	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		
		接口ID #n	IP Address		
			区域 id		
			管理州		
			OSPF状态		
			指标/成本/计时器		

OSPF 邻居报告

以下是OSPF邻居报告的示例格式。实际上，报告的格式由商业NMS（如果使用）的功能或定制脚本的设计输出确定。

区域	设备	邻居	数据字段	上次运行	此运行
区域ID #1	节点ID #1	邻居ID #1	路由器 ID		
			路由器 IP 地址		
			状态		

			事件		
			雷特兰斯·克		
		邻居ID #n	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		
	节点ID #n	邻居ID #1	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		
		邻居ID #n	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		
区域ID #n	节点ID #1	邻居ID #1	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		
		邻居ID #n	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		
	节点ID #n	邻居ID #1	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		
		邻居ID #n	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			雷特兰斯·克		

商业和公共互联网监控工具

商业工具可帮助收集和处理系统日志信息以及收集一般SNMP MIB变量轮询。

没有商用或公共Internet监控工具支持本程序定义的OSPF配置管理。因此，需要本地自定义脚本和过程。

SNMP 轮询数据

路由表RFC 1213

对象名称	对象说明
ipRouteDest	路由的目的IP地址。值为0.0.0.0的条目被视为默认路由。到单个目的地的多条路由可能出现在表中，但对这些多条路由的访问取决于正在使用的网络管理协议定义的表访问机制。::= { ipRouteEntry 1 }对象标识符= 1.3.6.1.2.1.4.21.1.1
ipRouteMask	表示在与ipRouteDest字段中的值进行比较之前，掩码与目标地址是逻辑的。对于不支持任意子网掩码的系统，代理使用以下掩码网络之一，通过确定相应ipRouteDest字段的值是属于A类、B类还是C类网络来构建ipRouteMask的值： • A类= 255.0.0.0 • B类= 255.255.0.0 • C类= 255.255.255.0 如果ipRouteDest的值为0.0.0.0（默认路由），则掩码值也为0.0.0.0。 **注意：所**有IP路由子系统都隐式使用此机制。 ::= { ipRouteEntry 11 }对象标识符= 1.3.6.1.2.1.4.21.1.11
ipRouteNextHop	此路由下一跳的IP地址。如果路由绑定到通过广播介质实现的接口，则此字段的值是该接口上的座席IP地址。::= { ipRouteEntry 7 }对象标识符= 1.3.6.1.2.1.4.21.1.7
ipRouteIfIndex	唯一标识到达路由下一跳的本地接口的索引值。此接口与IfIndex值所标识的接口相同。::= { ipRouteEntry 2 }对象标识符= 1.3.6.1.2.1.4.21.1.2

RFC 1213其他对象

对象	对象说明

名称	
ipAdEntIfIndex	唯一标识适用于该条目的接口的索引值。此接口与IfIndex值所标识的接口相同。::= { ipAddrEntry 2 }对象标识符= 1.3.6.1.2.1.4.20.1.2
ipInAddrErrors	由于其IP报头中的IP地址是实体的无效目标字段而丢弃的输入数据报数。此计数包括无效地址(0.0.0.0)和不受支持的类地址（E类）。 对于非IP网关且不转发数据报的实体，计数器包括因目标地址不是本地地址而丢弃的数据报。{ ip 5 } object identifier = 1.3.6.1.2.1.4.5
ipRoutingDiscards	丢弃的有效路由条目数。丢弃此类条目的一个可能原因是为其他路由条目释放缓冲空间。{ ip 23 }对象标识符= 1.3.6.1.2.1.4.23
ipOutNoRoutes	由于找不到将IP数据报传输到其目的地的路由而丢弃的IP数据报数。{ ip 12 }对象标识符= 1.3.6.1.2.1.4.12

RFC 1253 OSPF区域表

对象名称	对象说明
ospfAreaID	唯一标识区域的32位整数。区域ID 0.0.0.0用于OSPF主干。::= { ospfAreaEntry 1 }对象标识符= 1.3.6.1.2.1.14.2.1.1
ospfAuthType	为此区域指定的身份验证类型。其他身份验证类型可以按区域在本地分配。默认值为0。 ::= { ospfAreaEntry 2 }对象标识符= 1.3.6.1.2.1.14.2.1.2
OspfSpfRuns	使用该区域的链路状态数据库计算区域内路由表的次数。对象标识符= 1.3.6.1.2.1.14.2.1.4
ospfAreaBdrRtrCount	此区域内可达的ABR总数。此值最初为0（默认值），并在每个SPF传递中计算。::= { ospfAreaEntry 5 }对象标识符= 1.3.6.1.2.1.14.2.1.5
ospfASBdrRtrCount	此区域内可达的ABSR总数。此值最初为0（默认值），在每个SPF传递中计算。::= { ospfAreaEntry 6 }对象标识符= 1.3.6.1.2.1.14.2.1.6
ospfAreaLSACount	区域的链路状态数据库（不包括外部LSA）中的LSA总数。默认值为0。 ::= { ospfAreaEntry 7 }对象标识符= 1.3.6.1.2.1.14.2.1.7
ospfAreaLSAChksumSum	区域链路状态数据库中包含的LSA LS校验和的32位无符号和。此总和不包括外部（LS第5类）LSA。该总和可用于确定路由器的链路状态数据库是否发生变化，以及比较两台路由器的链路状态数据库。默认值为0。 ::= { ospfAreaEntry 8 }对象标识符= 1.3.6.1.2.1.14.2.1.8

RFC 1253 OSPF接口表

对象名称	对象说明
OSPFifIpAddress	OSPF接口的IP地址。对象标识符= 1.3.6.1.2.1.14.7.1.1
OspfIfEvents	OSPF接口更改其状态或发生错误的次数。对象标识符= 1.3.6.1.2.1.14.7.1.15
OspfIfState	OSPF接口状态。对象标识符= 1.3.6.1.2.1.14.7.1.12

RFC 1253 OSPF邻居表

对象名称	对象说明
OSPFNBRIPaddr	此邻居的IP地址。::= { ospfNbrEntry 1 }对象标识符= 1.3.6.1.2.1.14.10.1.1
ospfNbrAddressLessIndex	Internet标准MIB中没有IP地址的索引的IfIndex对应值。在创建行时，可以从实例派生。::= { ospfNbrEntry 2 }对象标识符= 1.3.6.1.2.1.14.10.1.2
ospfNbrRtrId	一个32位整数，表示为IpAddress，唯一标识自治系统中的相邻路由器。默认值为0.0.0.0。 ::= { ospfNbrEntry 3 }对象标识符= 1.3.6.1.2.1.14.10.1.3
ospfNbrState	与邻居关系的状态。状态为： • 向下(1) • 尝试(2) • init(3) • 双向(4) • exchangeStart(5) • 交换(6) • 加载(7) • full(8) ::= { ospfNbrEntry 6 }对象标识符= 1.3.6.1.2.1.14.10.1.6
ospfNbrEvents	邻居关系更改状态或发生错误的次数。默认值为0. ::= { ospfNbrEntry 7 }对象标识符= 1.3.6.1.2.1.14.10.1.7
ospfNbrLSRetransQLen	重传队列的当前长度。默认值为0. ::= { ospfNbrEntry 8 }对象标识符= 1.3.6.1.2.1.14.10.1.8

数据收集算法示例

在本文的研究过程中，开发了一个原型的C程序。该程序名为oscan，是使用Microsoft Developer Studio 97和Visual C++ 5.0版编写的。有两个特定的库提供SNMP函数应用程序编程接口(API)。 这些库是snmpapi.lib和mgmtapi.lib

Microsoft API提供的功能分为三个主要类别，并列于下表。

座席功能	管理器功能	实用程序功能
SnmpExtensionInit	SnmpMgrClose SnmpMgrGetTra	SNMPutilMemAlloc SNMPutilMemFree

SnmpExtensionInitEx SnmpExtensionQuery SnmpExtensionTrap	p SnmpMgrOidToStr SnmpMgrOpen SnmpMgrRequest SnmpMgrStrToOid SnmpMgrTrapListen	SnmpUtilMemReAlloc SnmpUtilOidAppend SnmpUtilOidCmpSnmpUtilOidCpy SNMPUtilOidFree SnmpUtilOIDNCmpUtilPrintAsnAny SNMPutilVarBindCpy SNMPutilVarBindListCpy SNMPUtilVarBindFree SnmpUtilVarBindListFree

oscan原型代码封装了Microsoft API，并附带了下面列出的一组附加功能。

- snmpWalkStrOid
- snmpWalkAsnOid
- snmpWalkVarBind
- snmpWalkVarBindList

这些功能提供了通用API，允许访问用于维护OSPF配置数据的各种SNMP MIB表。将要访问的表的对象标识符(OID)连同表特定回调函数一起传递给oscan API。回叫功能具有对从表返回的数据执行操作的智能。

主例程

第一项任务是创建节点列表，这些节点将成为oscan程序的目标。为避免“设备发现”问题，需要种子文件来标识要扫描的节点。种子文件提供IP地址和SNMP只读社区字符串等信息。

oscan程序需要维护多个内部数据结构来存储从路由器收集的SNMP信息。通常，收集的每个SNMP MIB表都有内部数据结构。

```
Main
load node array based on information in the seed file.
while more entries in the node array
start SNMP session for this node
collect IP route table for this node
collect OSPF area table for this node
collect OSPF Neighbor table for this node
collect  sysName for this node
collect OSPF Interface table for this node
end SNMP session for this node
end while
```

IP 路由表

使用SNMP访问IP路由表时必须小心，因为在此操作期间路由器的CPU很容易过载。因此，oscan程序利用用户可配置的延迟参数。该参数提供每个SNMP请求之间的延迟。对于大型环境，这意味着收集信息的总时间可能非常长。

路由表包含oscan感兴趣的四条信息：

- ipRouteDest

- ipRouteMask
- ipRouteNextHop
- ipRouteIfIndex

路由表由ipRouteDest编制索引。因此，从SNMP get-request返回的每个**对象都**将ipRouteDest附加到OID。

对象ipRouteIfIndex是索引到IP地址表(ipAddrTable)的整数。 ipAddrTable使用ipAdEntAddr对象（接口的IP地址）进行索引。 要获取接口的IP地址，需要执行四步过程：

1. 从路由表收集ipRouteIfIndex。
2. 使用ipRouteIfIndex访问ipAddrTable以进行模式匹配。
3. 找到模式后，将OID转换为字符串并收集最后四个点分十进制字段，这些字段将是接口的IP地址。
4. 将接口的IP地址存储回IP路由表中。

访问IP路由表的一般算法如下所示。此时，仅存储ipRouteIfIndex的整数值。在后续过程中，当收集接口信息时，会访问ipAddrTable，并收集其余信息并将其放入内部IP路由表中。

```
OID List =
ipRouteDestOID,
ipRouteMaskOID,
ipRouteNextHopOID,
ipRouteIfIndexOID;

For each object returned by  SNMP route table walk
Sleep // user configurable polling delay.
check varbind oid against OID list
if OID is ipRouteDestOID
add new entry in the internal route table array
if OID is one of the others
search internal route array for matching index value
store information in array
```

所收集的信息以类似于下面路由器CLI的常见输出的表格表示。

```
ROUTE TABLE
*******************************************************
Destination     Mask                GW              Interface
10.10.10.4      255.255.255.252     10.10.10.5      10.10.10.5
10.10.10.16     255.255.255.252     10.10.10.6      10.10.10.5
10.10.10.24     255.255.255.252     10.10.10.25     10.10.10.25
10.10.10.28     255.255.255.252     10.10.11.2      10.10.11.1
10.10.10.36     255.255.255.252     10.10.10.6      10.10.10.5
10.10.11.0      255.255.255.0       10.10.11.1      10.10.11.1
10.10.13.0      255.255.255.0       10.10.11.2      10.10.11.1
```

OSPF 区域表

从OSPF区域表收集信息的方法是扫描OSPF区域表(ospfAreaTable)并在返回时处理数据。ospfAreaTable的索引是osfpAreaId。ospfAreaId以点分十进制格式存储，与IP地址相同。因此，此处可重复使用用于处理和搜索ipRouteTable和ipRouteIfIndex的相同子例程。

本部分包含的OSPF区域表中实际上没有几个数据项。例如，ipInAddrErrors、IpRoutingDiscards和ipOutNoRoute对象在MIB-2定义中，但不与OSPF区域关联。这些对象与路由器关联。因此，通过将区域中每个节点的值添加到区域计数器，这些计数器用作区域度量。例如，在OSPF区域报告中

，由于找不到路由而丢弃的数据包数实际上是该区域中所有路由器丢弃的数据包数的总和。这是一个高级度量，提供区域路由运行状况的一般视图。

```
OID List =
ipInAddrErrorsOID,
ipRoutingDiscardsOID,
ipOutNoRouteOID,
areaIdOID,
authTypeOID,
spfRunsOID,
abrCountOID,
asbrCountOID,
lsaCountOID,
lsaCksumSumOID;

For object returned from the SNMP walk of  the Area Table
Sleep // user configurable polling delay.
check varbind oid against OID list.
if OID is ospfAreaId
add new entry in the internal route table array
if OID one of the others
search internal array for matching index value
store information in array
end of for loop
get ipInAddrErrors, ipRoutingDiscards, ipOutNoRoute
add values to overall Area counters
```

收集的信息在下面的ASCII表中表示。

```
AREAS
*********************************************************
AREA = 0.0.0.0AREA = 0.0.0.2
authType = 0authType = 0
spfRuns = 38spfRuns = 18
abrCount = 2abrCount = 1
asbrCount = 0asbrCount = 0
lsaCount = 11lsaCount = 7
lsaCksumSum = 340985lsaCksumSum = 319204
ipInAddrErrors = 0   ipInAddrErrors = 0
ipRoutingDiscards = 0ipRoutingDiscards = 0
ipOutNoRoutes = 0ipOutNoRoutes = 0
```

OSPF 邻居表

邻居表的索引有两个值：

- ospfNbrIpAddr - ospfNbrIpAddr是邻居的IP地址。
- ospfNbrAddressLessIndex - ospfNbrAddressLessIndex可以是以下两个值之一：对于已分配IP地址的接口，它为零。对于未分配IP地址的接口，会解释为Internet标准MIB中的IfIndex。

由于索引有两个值，您需要调整之前用于附加到返回的OID的额外信息的算法。进行此调整后，可在此处重新使用用于处理和搜索ipRouteTable和ipRouteIfIndex的相同子例程。

```
OID List =
ospfNbrIpAddrOID,
ospfNbrAddressLessIndexOID,
```

```
ospfNbrRtrIdOID,
ospfNbrStateOID,
ospfNbrEventsOID,
ospfNbrLSRetransQLenOID,

For object returned from the SNMP walk of the Neighbor Table
Sleep // user configurable polling delay.
check varbind OID against OID list.
if OID matches ospfNbrIpAddr
add new entry in the internal neighbor table array
if OID matches one of the others
search array for matching index value
store information in array
```

收集的信息在下面的ASCII表中表示。

```
NEIGHBORS
************************************************************
NEIGHBOR #0NEIGHBOR #1
Nbr Ip Addr = 10.10.10.6Nbr Ip Addr = 10.10.11.2
Nbr Rtr Id = 10.10.10.17Nbr Rtr Id = 10.10.10.29
Nbr State = 8Nbr State = 8
Nbr Events = 6Nbr Events = 30
Nbr Retrans = 0Nbr Retrans = 0
```

相关信息

- OSPF配置指南
- RFC 1246使用OSPF协议的经验
- RFC 1245 OSPF协议分析
- RFC 1224异步生成警报管理技术
- OSPF 支持页
- IP 路由 支持页
- 技术支持 - Cisco Systems