

使用 SNMP 管理 OSPF 配置

目录

[简介](#)

[OSPF 背景](#)

[进程定义](#)

[进程所有者](#)

[进程目标](#)

[进程性能指示器](#)

[进程输入](#)

[进程输出](#)

[任务定义](#)

[初始化任务](#)

[重复任务](#)

[数据识别](#)

[一般数据特性](#)

[SNMP 数据识别](#)

[RMON 数据识别](#)

[系统日志数据识别](#)

[CISCO IOS CLI 数据识别](#)

[数据收集](#)

[SNMP 数据收集](#)

[RMON 数据收集](#)

[系统日志数据收集](#)

[CISCO IOS CLI 数据收集](#)

[数据表示](#)

[OSPF 区域报告](#)

[OSPF 接口报告](#)

[OSPF 邻居报告](#)

[商业和公共互联网监控工具](#)

[SNMP 轮询数据](#)

[数据收集算法示例](#)

[相关信息](#)

简介

开放最短路径优先(OSPF)路由协议由[RFC 2328 OSPF第2版](#)定义。本文目标是提供使组织实现配置管理过程验证OSPF配置OSPF设计规划的一个程序框架和周期地审计OSPF部署保证与刻意的设计的长期一致性。

本文着重从ITU-T定义的FCAPS (故障，配置，认为/库存、性能，安全)型号的配置管理功能。配置

管理由ITU-T M.3400定义，当提供作用练习控制，识别，收集数据从和提供数据给NEs (网元)。

本文提供的信息在下述几个的主要部分被提交。

[OSPF背景](#)部分在OSPF部署的重要方面提供OSPF技术概述包括背景信息。

[进程定义](#)区分提供用于的进程定义的概述完成OSPF配置管理。进程详细资料描述根据目标、指示器、输入、输出和单个任务。

[任务定义](#)区分提供详细的流程任务定义。每任务描述根据目标，任务输入，任务输出，需的资源完成成为任务实施者需要的任务和工作技能。

[数据标识部分](#)描述OSPF的数据识别。数据识别考虑信息的来源或查找的地方。例如，信息由在简单网络管理协议(SNMP)管理信息库(MIB)， Syslog生成的日志的系统包含可能由命令行界面(CLI)只访问的日志文件或者内部数据结构。

本文[数据收集部分](#)描述OSPF数据的集。数据的收藏是密切相关对数据的位置。例如，SNMP MIB数据的收集采用几大机制，例如陷阱、远程监控(RMON)告警与事件、或轮询。内部数据结构维护的数据收集由自动脚本或由用户手册登录系统发出CLI命令然后记录输出的。

[Data Presentation部分](#)提供示例数据如何在报告格式被提交。在数据识别并且收集后，分析。本文提供可能使用记录和比较OSPF配置数据的示例报告。

[商业和公共互联网监控工具](#)、[SNMP轮询数据](#)和[数据收集算法示例](#)部分在工具的开发提供信息实现OSPF配置管理过程。

OSPF 背景

OSPF是设计的内部网关协议在单个自治系统内使用。OSPF用途连接状态或Shortest Path First (SPF) -基础技术，与在路由协议或Bellman-Ford技术比较找到的距离矢量例如路由信息协议(RIP)。单个Link State Advertisement (LSA)描述OSPF路由域的片段，例如，整个自治系统。这些LSA被充斥在路由域中，形成链路状态数据库。在域的每个路由器有一个相同的链路状态数据库。链路状态数据库的同步保持与可靠的扩散算法。从链路状态数据库，每个路由器通过计算一个最短路径树构件路由表，用的是树的根计算的的路由器。此计算通常指Dijkstra算法。

LSA小，并且每个LSA描述一小块OSPF路由域，特别地，单个路由器的网络邻居，单个转接网络的网络邻居，单个区域间路由或者单个外部路由。

此表定义了OSPF:关键特性

功能	说明
邻接	当对OSPF路由器变得相邻时，以OSPF数据库交换信息包的形式，两路由器通过交换数据库汇总同步他们的链路状态数据库。邻接路由器通过可靠的扩散算法然后保持他们的链路状态数据库的同步。由串行线路的路由器连接总是变得相邻。在多路访问网络(以太网)，所有路由器附加对网络在指定路由器(DR)和备份指定路由器(BDR)附近变为。
指定	当DR在所有多路访问网络时选择，产生描述网络的本地环境的网络LSA。它在扩散算法也扮演特殊作用

路由器	，因为网络的所有路由器通过发送和接收LSA同步他们的链路状态数据库到/从DR在溢流过程中。
备份指定路由器	当当前DR消失时，BDR在多路访问网络决定加速DR的转换。当BDR接管时，不需要通过在局域网的邻接进程。在DR的失踪被注意前，BDR在DR的缺乏也使可靠的扩散算法继续。
非广播多路访问网络支持	OSPF对待网络，例如帧中继公共数据网(PDN)，好象他们LAN。然而，更多的配置信息为路由器是需要的附加对这些网络最初互相找到。
OSPF配置管理区域	OSPF允许将分类为的自治系统区域。这提供一个额外的级别路由保护，以便路由在区域内从所有信息外部保护到区域。并且，通过拆分一自治系统到区域里，根据CPU周期，降低Dijkstra步骤的开销。
虚拟链路	通过允许虚链路的配置，OSPF删除在区域布局的拓扑限制在自治系统。
路由协议交换的验证	在OSPF路由器收到路由协议信息包时候，能或者在进一步处理它前验证数据包。
灵活路由度量	在OSPF中，量度分配到出站路由器接口。路径的开销是路径的组分接口的总和。默认情况下，路由度量从链路的带宽派生。它可以由系统管理员分配指示网络特性的所有组合例如延迟、带宽和开销。
相等	当对目的地的多个最佳开销路由存在时，OSPF查找并且使用他们对负载共享流量对目的地。

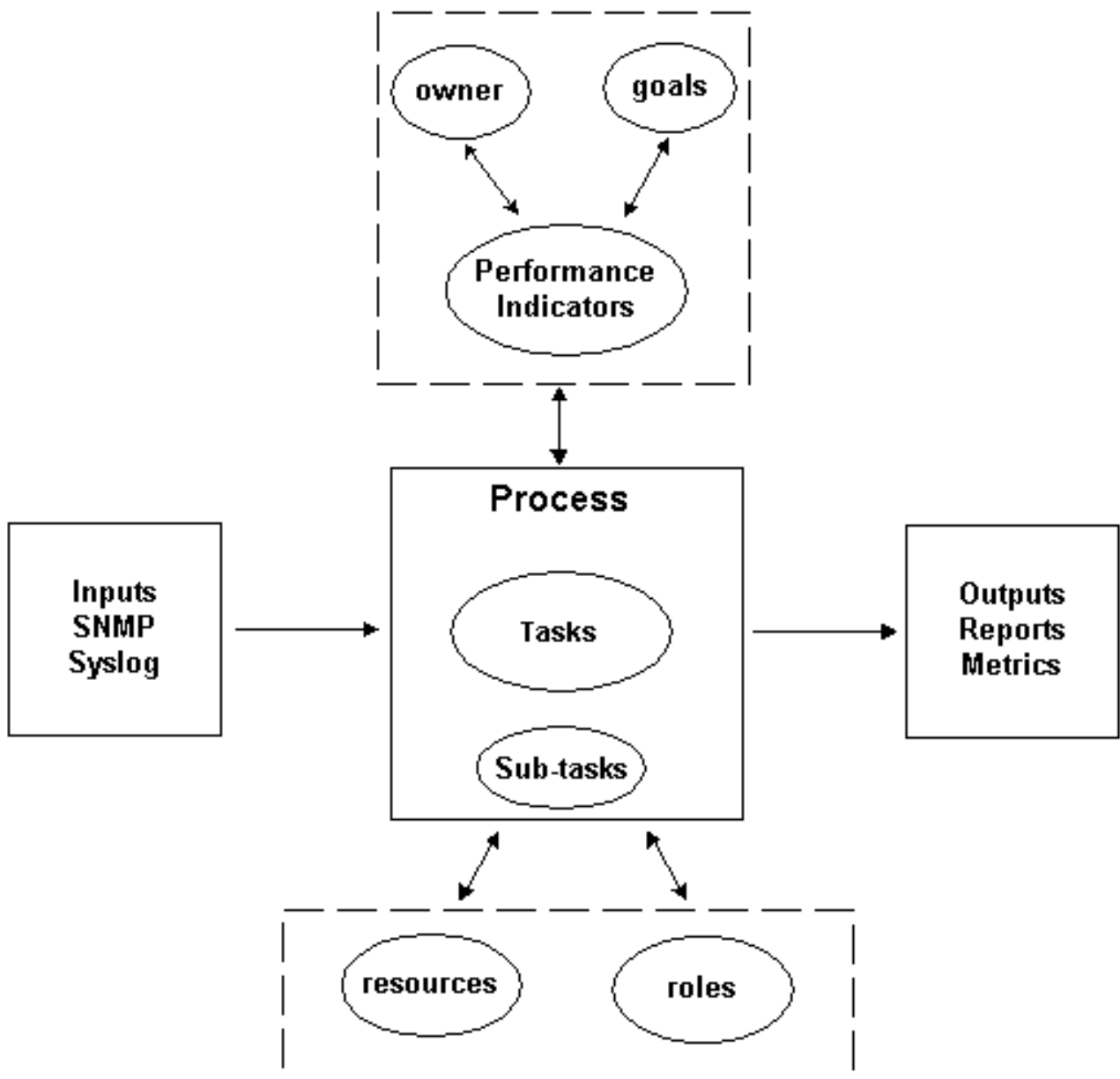
代价多重通道	
可变长度子网支持	通过运载与每个建议的目的地的一个网络掩码支持可变长度子网掩码。
末节区域支持	要支持有的路由器内存不足，区域可以配置作为残余部分。外部LSA没有被充斥到和在末节区域中。对外部目的地的路由末节区域的根据默认独自地。

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。

如图解，这在下面图显示。



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，如果活动以最少的力气进行，进程也被认为高效率。

进程所有者

进程跨过多种组织边界。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，那么进程所有者加建将执行进程修改。进程的修改由更改控制和复核进程管理。

进程目标

进程目标设立设置方向和范围进程定义的。目标也用于定义用来测量进程效果的尺度。

此进程目标是提供框架验证OSPF实施的部署的配置刻意的设计和提供机制周期地审计OSPF部署随着时间的推移保证一致性关于刻意的设计。

进程性能指示器

进程性能指示器用于测量进程定义的效果。指示器应该是可测量和可计量的。如下所示的指示器是数字或测量在时间之前。OSPF配置管理进程的指示器定义如下：

- 要求的时间长度通过整个过程循环。
- 执行要求的频率为了主动地检测OSPF问题，在他们影响用户前。
- 网络负载关联与进程的执行。
- 进程推荐的纠正措施数量。
- 由于进程实现的纠正措施数量。
- 要求的时间长度实现纠正措施。
- 要求的时间长度实现纠正措施。
- 纠正措施积压。
- 停机时间归因于OSPF相关问题。
- 在种子文件添加，删除或者修改的项目数量。这是准确性和稳定性的征兆。

进程输入

进程输入用于定义标准和前提对于进程。许多次，进程输入的认识在外部条件提供信息。下面提供与OSPF配置管理涉及的输入列表。

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

进程输出

进程输出定义如下：

- 在本文的[Data Presentation部分](#)定义的OSPF配置报告
- 纠正措施的OSPF配置推荐能执行

任务定义

以下部分定义了初始化和重复任务关联与OSPF配置管理。

初始化任务

初始化任务一次在进程的实施时完成，并且不应该执行与进程的每迭代。

验证必备任务

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。表下面的概述事先需要的初始化任务。

前期	说明
----	----

任务	
任务目的和输入	<ol style="list-style-type: none"> 1. 验证OSPF设计文件存在，并且以下信息是可用的在网络设计文档：区域定义—名称、地址范围和区域类型局域边界路由器/自治系统边界路由器(ABR/ASBR)识别DR/BDR识别互联网注册(IR)节点和接口分配到区域 2. 请使用一个SNMP标准配置模板验证SNMP在网络配置。注意：这创建的种子文件是使用的以后作为输入。 3. 请使用一个Syslog标准配置模板验证Syslog在网络部署。
任务输出	任务输出是关于必备任务的条件的一个状态报告。如果其中任一支持的任务视为作为无效，程序所有者应该提交请求安排支持进程更新。如果支持进程不可能更新，请进行在影响的一评估对此进程。
任务角色	网络工程师技能集

创建种子文件

OSPF配置管理进程要求使用种子文件取消需要对于网络发现功能。种子文件记录由OSPF程序管理和也用于作为汇聚点协调与在组织的变更管理流程的一组路由器。例如，如果新节点被输入到网络，他们需要被添加到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。表下面的概述创建的种子文件进程。

进程	说明
任务目的	创建将使用初始化OSPF配置管理软件的种子文件。种子文件的格式化取决于用于的资源实现OSPF配置管理进程。如果自定义脚本开发，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)，种子文件的格式由NMS文档定义。
任务输入	<ol style="list-style-type: none"> 1. 安排种子文件的格局。 2. 请使用OSPF设计文档识别以下数据：所有节点的IP地址SNMP团体字符串Telnet和CLI登录帐户和密码 3. 日程和联系人名称对于网络更改管理进程。
任务输出	OSPF配置管理进程的一个种子文件。
任务资源	<ul style="list-style-type: none"> • 商业NMS系统 • 定制开发的软件系统 • 手动程序—登录每网元并且发出命令行并且记录输出。

任务角色	<ul style="list-style-type: none"> • 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。 • 手动程序—网络工程师。
------	---

重复任务

重复任务完成与进程的每迭代，并且确定他们的频率并且被修改为了改善指示器。

维护种子文件

种子文件为OSPF配置管理进程的有效的应用是关键。所以，种子文件的当前状态必须积极地被管理。变成影响种子文件需要内容由OSPF配置管理进程所有者跟踪的网络。

进程	说明
任务目的	<ol style="list-style-type: none"> 1. 通过跟踪保存种子文件的货币，并且与控制网络移动的组织功能的交互作用，添加，更改，并且/或者网络配置修改。 2. 保持版本控制并且备份种子文件的控制。
任务输入	<ol style="list-style-type: none"> 1. 从变更管理的信息，例如移动、新增内容和更改，影响种子文件的内容。 2. 从影响种子文件的内容的工程/设计的信息。
任务输出	<ol style="list-style-type: none"> 1. 关于种子文件货币的状况的每周报告。 2. 描述种子文件备份的定义和文档位置和恢复步骤。
任务资源	<ul style="list-style-type: none"> • 商业NMS系统 • 定制开发的软件系统 • 手动程序—登录每网元并且发出命令行并且记录输出。
任务角色	<ul style="list-style-type: none"> • 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。 • 手动程序—网络工程师。

执行OSPF扫描

用于的两个步骤执行OSPF扫描是：

1. 收集数据。
2. 分析数据。

根据如何使用进程，频率这两个步骤将变化。例如，此进程可以用于验证安装修改。在这种情况下，在更改和数据分析在更改以后执行确定更改的成功前后，数据收集运行。

如果此进程用于验证OSPF配置管理设计记录，数据收集和分析频率依靠变化率在网络的。例如，如果有在网络上的巨大数量的变化，设计验证每星期一次执行。如果有非常在网络上的少量变化

，设计验证没有更多比每月一次执行。

查看OSPF报告

OSPF配置管理报告的格式依靠用于的资源实现OSPF配置管理进程。下表提供建议的按客户需要开发的报告格式。

报告	格式
任务输入	关于OSPF配置管理报告，请参阅在本文内的 Data Presentation部分 。
任务输出	如果问题被找到在扫描报告和逻辑设计记录之间，必须做出决定项目是正确，并且不正确。应该更正不正确项目。这可能包括设计记录或网络更改更改单的修改。
任务资源	<ul style="list-style-type: none"> • 商业NMS系统 • 定制开发的软件系统 • 手工—登录每网元并且发出命令行并且记录输出
任务角色	<ul style="list-style-type: none"> • 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。 • 手动程序—网络工程师。

数据识别

一般数据特性

下表描述可以应用到OSPF配置管理的数据。

数据	说明
OSPF区域	描述的信息路由器的连接的区域包括： <ul style="list-style-type: none"> • 区域 id • 区域验证 • SPF运行 • ABR编号在区域 • ASBRs编号在区域 • 区域LSA计数—在路由器间的一致性在区域 • 区域LSA校验和—在路由器间的一致性在区域 • 频率信息包丢弃由于寻址错误每个区域 • 频率由路由进程的协议信息包丢弃每个区域 • 频率路由信息包丢弃由于没有路由被找

	到的情况每个区域
OSPF接口	从OSPF的观点描述一个接口例如： <ul style="list-style-type: none"> • IP 地址 • 区域 id • 管理状态 • OSPF度量分配到接口 • OSPF计时器分配到接口 • OSPF状态
OSPF邻居状态	描述OSPF邻居。 <ul style="list-style-type: none"> • 邻接路由器ID • 邻居状态 • 邻接事件—次数邻接关系更改状态，或者错误出现。 • 邻接重传队列—重传队列的当前长度。

[SNMP 数据识别](#)

Cisco当前支持[RFC 1253 OSPF第2版MIB](#)。RFC 1253不包含OSPF的SNMP陷阱定义。OSPF MIB的新版本是[RFC 1850 OSPF版本2](#)。SNMP陷阱为在RFC 1850的OSPF定义。[OSPF MIB的Cisco的实施不支持RFC 1850。](#)

请参考本文的[SNMP Polling Data部分](#)关于更详细的资料。

请参考MIB支持平台和代码版本的一明确列表的[Cisco网络管理软件页](#)。

[RMON 数据识别](#)

没有为此步骤要求的RMON特定数据。

[系统日志数据识别](#)

一般来说，Syslog生成另外技术的服务特定的消息。虽然系统日志信息为故障和性能管理是适当，被提供的信息此处是参考。关于Cisco设备生成的OSPF系统日志信息示例，请参阅[OSPF错误消息](#)。

对于系统消息完整列表由设备的，请参考[消息和恢复流程](#)。

[CISCO IOS CLI 数据识别](#)

在OSPF配置管理过程的此版本中，没有要求的CLI数据。

[数据收集](#)

[SNMP 数据收集](#)

下面的表定义了SNMP数据收集不同的组件。

SNMP数据组件	定义
一般SNMP配置	参考 配置SNMP 关于SNMP配置最佳实践的一般信息。
有服务特效的SNMP配置	没有为此步骤要求的服务特定的SNMP配置。
SNMP MIB需求	请参阅上面 数据标识部分 。
轮询集的SNMP MIB	SNMP轮询数据收集由一个商业系统例如 HP OpenView 或由自定义脚本。 关于收集算法进一步讨论 ，请参阅本文的 Example Data Collection Algorithms部分 。
SNMP MIB陷阱集	支持Cisco设备OSPF MIB当前版本不支持SNMP陷阱。没有为此步骤要求的SNMP陷阱。

RMON 数据收集

没有在步骤的此版本和数据要求的RMON配置。

系统日志数据收集

在本文的范围之外，一般Syslog配置指南是。参考的[配置和排除故障与一个内部网络的Cisco Secure PIX防火墙](#)欲知更多信息。

使用以下命令， OSPF特定需求通过配置OSPF路由器记录与系统消息的邻接更改解决：

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

CISCO IOS CLI 数据收集

一般来说， Cisco IOS CLI提供最直接访问给NE包含的原始信息。然而， CLI访问更好地适合故障排除程序和变更管理活动比对于全局配置配置管理如定义由此步骤。访问通过CLI不会为大型网络的管理扩展。在这些情况下，自动化的信息存取要求。

在OSPF配置管理过程的此版本中，没有要求的CLI配置和数据。

数据表示

OSPF 区域报告

下列是OSPF区域报告的一种示例格式。报告的格式取决于商业NMS的功能，如果使用一个，或者自定义脚本的有针对的输出。

区域	数据域	为时运行	此运行
区域ID #1	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		
	没有找到的路由		
区域ID #n	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		
	没有找到的路由		

OSPF 接口报告

下列是OSPF接口报告的一种示例格式。实际上，报告的格式取决于商业NMS的功能，如果使用一个，或者自定义脚本的有针对性的输出。

区域	设备	接口	数据域	为时运行	此运行
区域ID #1	节点ID #1	接口ID #1	IP Address		
			区域 id		
			管理状态		
			OSPF状态		
		/费用的量度/计时器			
		接口ID #n	IP Address		
			区域 id		
			管理状态		
	OSPF状态				
	/费用的量度/计时器				
	节点ID #n	接口ID #1	IP Address		
			区域 id		
			管理状态		
			OSPF状态		
/费用的量度/计时器					
接口ID #n		IP Address			
	区域 id				

			管理状态		
			OSPF状态		
			/费用的量度/计时器		
区域 ID #n	节点 ID #1	接口 ID #1	IP Address		
			区域 id		
			管理状态		
			OSPF状态		
			/费用的量度/计时器		
		接口 ID #n	IP Address		
			区域 id		
			管理状态		
			OSPF状态		
			/费用的量度/计时器		
	节点 ID #n	接口 ID #1	IP Address		
			区域 id		
			管理状态		
			OSPF状态		
			/费用的量度/计时器		
		接口 ID #n	IP Address		
			区域 id		
			管理状态		
			OSPF状态		
			/费用的量度/计时器		

OSPF 邻居报告

下列是OSPF邻居报告的一种示例格式。实际上，报告的格式取决于商业NMS的功能，如果使用一个，或者自定义脚本的有针对的输出。

区域	设备	邻居	数据域	为时运行	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		
			Retrans Que		
		邻居ID #n	路由器 ID		

			路由器 IP 地址				
			状态				
			事件				
			Retrans Que				
	节点ID #n	邻居ID #1		路由器 ID			
				路由器 IP 地址			
				状态			
				事件			
		邻居ID #n			路由器 ID		
					路由器 IP 地址		
状态							
事件							
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				
			Retrans Que				
	邻居ID #n			路由器 ID			
				路由器 IP 地址			
				状态			
				事件			
				Retrans Que			
节点ID #n	邻居ID #1		路由器 ID				
			路由器 IP 地址				
			状态				
			事件				
			Retrans Que				

		邻居ID #n	路由器 ID		
			路由器 IP 地址		
			状态		
			事件		
			Retrans Que		

商业和公共互联网监控工具

商业工具在集和处理存在帮助系统日志信息和一般SNMP MIB变量集?的。

没有商业或公共互联网监控工具知道支持OSPF配置管理如定义由此步骤。所以，本地定制脚本和步骤要求。

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。与值的一个条目0.0.0.0认为默认路由。对单个目的地的多个路由在表里能出现，但是对这样多个条目的访问依靠网络管理协议定义的表存取机制在使用中。 : := {iprouteentry 1}对象标识符= 1.3.6.1.2.1.4.21.1.1
ipRouteMask	在URL域指示掩码是逻辑与目的地址在与值比较前。对于不支持任意子网掩码的那些系统，代理程序修建值ipRouteMask通过确定对应的URL域的值是否属于A类， B或者C网络，使用以下掩码网络之一， : <ul style="list-style-type: none"> • A类= 255.0.0.0 • B类= 255.255.0.0 • C类= 255.255.255.0 如果ipRouteDest的值是0.0.0.0，默认路由，掩码值也是0.0.0.0。 注意： 所有IP路由子系统隐含地使用此机制。 : := {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。一旦路由一定对认识到与广播价质的接口，值此字段是在接口的代理程序的IP地址。 : := {iprouteentry 7}对象标识符= 1.3.6.1.2.1.4.21.1.7
ipRoute	独特识别本地接口路由下一跳被到达的索引值。此接口是IfIndex值识别的同一个接口。 : :=

elfl nde x	{iprouteentry 2}对象标识符= 1.3.6.1.2.1.4.21.1.2
------------------	---

RFC 1213其他对象

对象名称	对象说明
ip Ad En tlfl nd ex	独特识别接口可适用对条目的索引值。此接口是 IflIndex值识别的同一个接口。 : := {ipaddentry 2}对象标识符= 1.3.6.1.2.1.4.20.1.2
ipl nA dd rEr ror s	输入数据包数量丢弃的，因为在他们的IP报头的IP地址是实体的一个无效目标字段。此计数包括无效的地址(0.0.0.0)和不支持的类地址(类E)。对于不是IP网关，并且不转发数据包的实体，计数器包括丢弃的数据包，因为目的地址不是本地地址。{ip 5}对象标识符= 1.3.6.1.2.1.4.5
ip Ro uti ng Di sc ar ds	丢弃的有效路由条目数量。丢弃的这样条目一个可能的来源是释放其他路由条目的缓冲空间。{ip 23}对象标识符= 1.3.6.1.2.1.4.23
ip Ou tN oR ou tes	IP数据包数量丢弃的，因为不可能发现路由传达他们给他们的目的地。{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
ospfArea ld	独特识别区域的32位整数。区域ID 0.0.0.0使用 OSPF骨干网。 : := {ospfareaentry 1}对象标识符= 1.3.6.1.2.1.14.2.1.1
ospfAuth Type	为此区域指定的认证类型。另外的认证类型可能分配本地根据一个每区域基本类型。默认值是0。 : := {ospfareaentry 2}对象标识符= 1.3.6.1.2.1.14.2.1.2
OspfSpf Runs	使用此区域的链路状态数据库，次数区域内路由表计算。对象标识符= 1.3.6.1.2.1.14.2.1.4
ospfArea BdrRtrC ount	ABR总数可及的在此区域内。这是最初0，默认值和计算在每张SPF通行证。 : := {ospfareaentry 5}对象标识符= 1.3.6.1.2.1.14.2.1.5

ospfASBdrRtrCount	ABSRs总数可及的在此区域内。这最初是0 (默认值)和计算在每张SPF通行证。 : := {ospfareaentry 6}对象标识符= 1.3.6.1.2.1.14.2.1.6
ospfAreaLSACount	LSA总数在区域的链路状态数据库的，不包括外部LSA。默认值是0。 : := {ospfareaentry 7}对象标识符= 1.3.6.1.2.1.14.2.1.7
ospfAreaLSACKsumSum	在区域的链路状态数据库包含的LSA的LS校验和的32位未签名的总和。此总和排除外部(LS类型5) LSA。总和可以用于确定是否有在路由器的链路状态数据库上的一个变化和比较两路由器链路状态数据库。默认值是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	次数OSPF接口更改其状态，或者错误出现。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
OspfNbrIpAddress	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	对应的值在互联网标准MIB的IfIndex在没有一个IP地址的索引。在行创建，这可以从实例派生。 : := {ospfnbrentry 2}对象标识符= 1.3.6.1.2.1.14.10.1.2
ospfNbrRtrId	32位整数，代表作为IP地址，独特识别自治系统的相邻路由器。默认值是0.0.0.0。 : := {ospfnbrentry 3}对象标识符= 1.3.6.1.2.1.14.10.1.3
ospfNbrState	关系的状态与邻居的。状态是： <ul style="list-style-type: none"> • 在(1)下 • 尝试(2) • init (3) • 双向(4) • exchangeStart (5) • 交换(6) • 加载(7) • 全双工(8) : := {ospfnbrentry 6}对象标识符= 1.3.6.1.2.1.14.10.1.6
ospfNbrEvents	次数邻接关系更改状态，或者错误出现。默认值是0。 : := {ospfnbrentry 7}对象标识符= 1.3.6.1.2.1.14.10.1.7

ospfNbrLS	重传队列的当前长度。默认值是0。 : :=
RetransQL	{ospfnbrentry 8}对象标识符=
en	1.3.6.1.2.1.14.10.1.8

数据收集算法示例

在对本文的调查时，原型‘C’程序开发。使用有Visual C++版本5.0的，Microsoft Developer Studio 97程序，呼叫oscan，写入。有提供SNMP功能Application Programming Interface (API)的两个特定库。那些库是snmpapi.lib和mgmtapi.lib

Microsoft API提供的功能在下表分组到三个主要类别并且列出。

代理程序功能	管理器功能	实用程序函数
SnmpExtensionInit	SnmpMgrClose	SnmpUtilMemAlloc
SnmpExtensionInitEx	SnmpMgrGetTrap	SnmpUtilMemFree
SnmpExtensionQuery	SnmpMgrOidToStr	SnmpUtilMemReAlloc
SnmpExtensionTrap	SnmpMgrOpen	SnmpUtilOidAppend
	SnmpMgrRequest	SnmpUtilOidCmp
	SnmpMgrStrToOid	SnmpUtilOidCpy
	SnmpMgrTrapListen	SnmpUtilOidFree
		SnmpUtilOidNCmp
		SnmpUtilPrintAsnAny
		SnmpUtilVarBindCpy
		SnmpUtilVarBindListCpy
		SnmpUtilVarBindFree
		SnmpUtilVarBindListFree

oscan原型代码封装与如下所示的一套的Microsoft API另外的功能。

- snmpWalkStrOid
- snmpWalkAsnOid
- snmpWalkVarBind
- snmpWalkVarBindList

这些功能提供对多种SNMP MIB表的允许使用维护OSPF配置数据的通用的API。能将访问的表的Object Identifier (OID)通过对oscan API与表特定呼叫上一步功能一起。呼叫上一步功能在从表返回的数据上有智能对起作用。

主例程

首要任务创建将是OSCAN程序的目标节点的列表。为了避免“设备发现”问题，种子文件要求识别将被扫描的节点。种子文件提供信息例如IP地址和SNMP只读属性字段。

OSCAN程序需要维护几个内部数据结构存储从路由器收集的SNMP信息。一般来说，有收集的每个SNMP MIB表的一个内部数据结构。

```

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

必须保重，当访问与SNMP时的IP路由表，因为简单在此操作时超载路由器CPU。所以，OSCAN程序使用一个用户可配置延迟参数。参数提供每SNMP请求之间的延迟。对于大环境，这意味着总时间收集信息可以是非常重大的。

路由表包含四个信息oscan是感兴趣在：

- ipRouteDest
- ipRouteMask
- ipRouteNextHop
- ipRouteIfIndex

ipRouteDest标注路由表。所以，从SNMP获取请求返回的每个对象有ipRouteDest被添附对OID。

对象ipRouteIfIndex是整数该索引到IP地址表(ipAddrTable)。ipAddrTable被标注使用ipAdEntAddr对象(接口的IP地址)。为了获得接口的IP地址，一四步进程要求：

1. 收集从路由表的ipRouteIfIndex。
2. 使用模式匹配的，ipRouteIfIndex访问ipAddrTable。
3. 当找到时模式，请转换OID对字符串并且收集将是接口的IP地址的最后四个点分的十进制字段。
4. 存储接口的IP地址回到IP路由表。

访问的IP路由表一般算法如下所示。这时，ipRouteIfIndex的仅整数值存储。以后在进程，当收集接口信息，ipAddrTable访问时和剩余信息收集并且被放置到内部IP路由表。

```
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
```

收集的信息在类似于从下面路由器CLI的熟悉输出的表里代表。

```
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区域表的信息收集由扫描OSPF区域表(ospfAreaTable)和处理数据完成，当返回。对ospfAreaTable的索引是ospfAreaId。ospfAreaId在与IP地址是相同的点分十进制格式存储。所以，使用处理和搜索ipRouteTable和ipRouteIfIndex的同样子例行程序可以重新使用此处。

有实际上不在OSPF区域表在此部分包括的几数据项。例如，ipInAddrErrors，IpRoutingDiscards，并且ipOutNoRoute对象在MIB-2定义，但是没有关联与OSPF区域。这些对象用路由器关联。所以，这些计数器使用作为区域量度通过增加值每个节点在区域到区域计数器。例如，在OSPF区域报告，编号数据包丢弃的由于没有找到的路由实际上是所有丢弃的数据包的总和和路由器在该区域。这是提供区域的路由健康全视图的一高层次量度。

```
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
```

收集的信息在下表ASCII表代表。

```
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](#)

邻居表的索引是两个值：

- ospfNbrIpAddr — ospfNbrIpAddr是邻居的IP地址。
- ospfNbrAddressLessIndex — ospfNbrAddressLessIndex可以是两个值之一：对于有分配的一个IP地址的接口，它是零。对于没有分配的一个IP地址的接口，它解释作为从互联网标准MIB的IfIndex。

由于有索引的两个值，您需要调节用于额外信息的前算法被添附对返回的OIDs。以后进行此调整，使用处理，并且可以重新使用此处搜索ipRouteTable和ipRouteIfIndex的同样子例行程序。

```
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
```

收集的信息在下表ASCII表代表。

```
NEIGHBORS
*****
NEIGHBOR #ONEIGHBOR #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](#)
- [与OSPF协议的RFC 1246体验](#)
- [RFC 1245 OSPF协议分析](#)
- [管理异步地生成的警报的RFC 1224技术](#)
- [OSPF 支持页](#)
- [IP 路由支持页](#)
- [技术支持 - Cisco Systems](#)