

管理OSPF与SNMP上下文的多个实例

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[SNMP环境感知](#)

[配置](#)

[验证](#)

[SNMPv2验证](#)

[SNMPv3验证](#)

[相关信息](#)

简介

本文档提供SNMPv2和SNMPv3的示例配置，这些配置描述如何使用SNMP环境管理开放最短路径优先(OSPF)的多个实例。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

背景信息

由IETF(RFC 1850 [\[1\]](#))定义的OSPF MIB设计为仅在给定路由器上使用一个OSPF进程/实例。

例如，只有一个 `ospfRouterId` 对象，而没有其表。为了处理多个实例，[RFC 4750](#) 建议您使用 SNMPv3 上下文来提供每个实例的视图。

SNMP环境感知

在使 IOS OSPF SNMP 代码情景感知之前，系统会在返回标量对象和某些表时选择或多或少随机的“默认”实例。在这些情况下，来自其他实例的信息无法通过 SNMP 获取。对于某些其他表，SNMP 会将所有实例的条目进行混合，而无法识别哪个条目是什么。在许多情况下，这可能导致不明确或重复的条目。在 PE-CE 配置中，IP 地址和邻居路由器 ID 可能不唯一，这种做法尤其不好。这使监控和排除单个 CE 实例的故障变得困难或不可能。

使用当前情景感知 IOS 代码（当未指定情景时），标量对象的旧行为仍然存在。唯一的变化是，它现在将所有表限制为与扩展器相同的“默认”OSPF 实例，而不是仅限部分表。当提供情景时，SNMP 查询可以针对特定 OSPF 实例，并且该实例的所有信息能够以一致且明确的方式检索。

如果使用 SNMPv3，则可以直接随轮询提供上下文字符串。SNMPv2c 不提供情景。但是，您可以将 SNMP 社区字符串映射到 IOS 配置中的情景，并且这些情景可用于将 SNMPv2 轮询定向到特定 OSPF 实例。

配置

此配置示例基于 SNMPv2:

```
路由器 1

Router1#

router ospf 1
  router-id 1.1.1.111
  log-adjacency-changes
  snmp context context1
!
router ospf 2
  router-id 4.4.4.111
  log-adjacency-changes
  snmp context context2
!--- Associates the SNMP context with the instance. !
snmp-server user u2 g2 v2c !--- Configures the user u2
to the SNMP group g2 and !--- specifies the group is
using the SNMPv2c security model. snmp-server group g2
v2c !--- Configures the SNMP group g2 and specifies !---
the group is using the SNMPv2c security model. snmp-
server group g2 v2c context context1 snmp-server group
g2 v2c context context2 snmp-server community public RO
!--- Community access string to permit access !--- to
the SNMP. snmp-server community cx1 RO snmp-server
community cx2 RO snmp-server context context1 snmp-
server context context2 snmp mib community-map cx1
context context1 security-name u2 !--- Associates the
SNMP community cx1 with !--- the context context 1. snmp
mib community-map cx2 context context2 security-name u2
```

此配置示例基于 SNMPv3:

```
路由器 1
```

```
Router1#  
  
router ospf 1  
  router-id 1.1.1.111  
  log-adjacency-changes  
  snmp context context1  
!  
router ospf 2  
  router-id 4.4.4.111  
  log-adjacency-changes  
  snmp context context2  
!  
snmp-server user u1 g1 v3  
snmp-server group g1 v3 noauth  
snmp-server group g1 v3 noauth context context1  
snmp-server group g1 v3 noauth context context2  
snmp-server context context1  
snmp-server context context2
```

注意：使用[命令查找工具](#)([仅注册客户](#))可查找有关本文档中使用的命令的详细信息。

验证

您可以在任何客户端计算机上使用snmpwalk命令来验证输出。

注意：[输出解释器工具](#)([仅注册客户](#))(OIT)支持某些show命令。使用 OIT 可查看对 show 命令输出的分析。

SNMPv2验证

```
SNMPv2  
  
linux>snmpwalk -c public -v2c irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111  
  
linux>snmpwalk -c cx1 -v2c irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 1.1.1.111  
  
linux>snmpwalk -c cx2 -v2c irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111
```

SNMPv3验证

```
SNMPv3  
  
linux>snmpwalk -u u1 -v3 irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111  
  
linux>snmpwalk -u u1 -v3 -n context1 irp-view14:7890  
OSPF-MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 1.1.1.111
```

```
linux>snmpwalk -u u1 -v3 -n context2 irp-view14:7890
OSPF-MIB::ospfRouterId.0
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111
```

[相关信息](#)

- [使用 SNMP 管理 OSPF 配置](#)
- [技术支持和文档 - Cisco Systems](#)