Understanding Simple Network Management Protocol (SNMP) Traps

Document ID: 7244

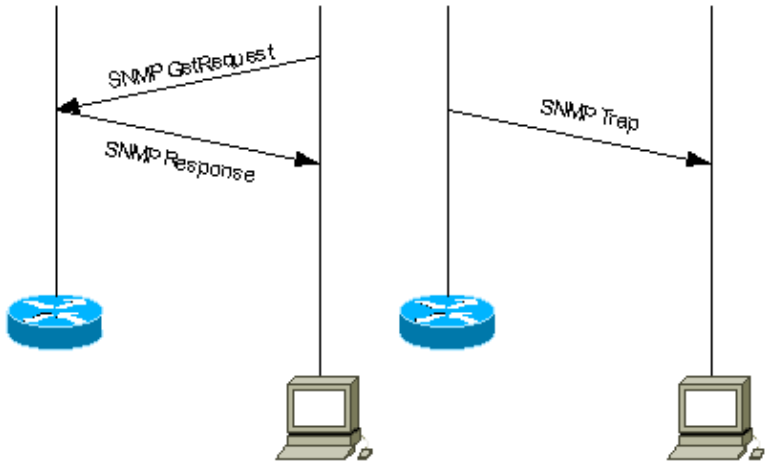
Contents

Introduction

This document provides an introduction to SNMP traps. It shows how SNMP traps are used and the role they play in the management of a data network.

SNMP traps enable an agent to notify the management station of significant events by way of an unsolicited SNMP message.

In this diagram, the setup on the left shows a network management system that polls information and gets a response. The setup on the right shows an agent that sends an unsolicited or asynchronous trap to the network management system (NMS).



Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Use SNMP Traps

SNMPv1 (Simple Network Management Protocol) and SNMPv2c, along with the associated Management Information Base (MIB), encourage trap−directed notification.

The idea behind trap−directed notification is that if a manager is responsible for a large number of devices, and each device has a large number of objects, it is impractical for the manager to poll or request information from every object on every device. The solution is for each agent on the managed device to notify the manager without solicitation. It does this by sending a message known as a trap of the event.

After the manager receives the event, the manager displays it and can choose to take an action based on the event. For instance, the manager can poll the agent directly, or poll other associated device agents to get a better understanding of the event.

Trap−directed notification can result in substantial savings of network and agent resources by eliminating the need for frivolous SNMP requests. However, it is not possible to totally eliminate SNMP polling. SNMP requests are required for discovery and topology changes. In addition, a managed device agent can not send a trap, if the device has had a catastrophic outage.

SNMPv1 traps are defined in RFC 1157, with these fields:

- *Enterprise* Identifies the type of managed object that generates the trap.
- *Agent address* Provides the address of the managed object that generates the trap.
- *Generic trap type* Indicates one of a number of generic trap types.
- *Specific trap code* Indicates one of a number of specific trap codes.
- *Time stamp* Provides the amount of time that has elapsed between the last network reinitialization and generation of the trap.
- *Variable bindings* The data field of the trap that contains PDU. Each variable binding associates a particular MIB object instance with its current value.

Standard generic traps are: coldStart, warmStart, linkDown, linkUp, authenticationFailure, egpNeighborLoss. For generic SNMPv1 traps, *Enterprise* field contains value of sysObjectID of the device that sends trap. For vendor specific traps, *Generic trap type field* is set to enterpriseSpecific(6). Cisco implemented its own specific traps in a non−conventional way. Instead of having the trap *Enterprise* field still the sysObjectID and having the *Specific trap code* to identify all specific traps supported by all Cisco devices, Cisco implemented trap identification using various trap Enterprise and Specific trap code fields. You can see the actual values from the SNMP Object Navigator. Also, Cisco redefined some generic traps in CISCO−GENERAL−TRAPS MIB with the addition of more bound variables. For these traps, *Generic trap type* is kept the same and not set to enterpriseSpecific(6).

In SNMPv2c trap is defined as NOTIFICATION and formatted differently compared to SNMPv1. It has these parameters:

- *sysUpTime* This is the same as Time stamp in SNMPv1 trap.

- *snmpTrapOID* Trap identification field. For generic traps, values are defined in RFC 1907, for vendor specific traps *snmpTrapOID* is essentially a concatenation of the SNMPv1 *Enterprise* parameter and two additional sub−identifiers, '0', and the SNMPv1 *Specific trap code* parameter.
- *VarBindList* This is a list of variable−bindings.

In order for a management system to understand a trap sent to it by an agent, the management system must know what the object identifier (OID) defines. Therefore, it must have the MIB for that trap loaded. This provides the correct OID information so that the network management system can understand the traps sent to it.

For traps that are supported by Cisco devices in specific MIBs, refer to the Cisco SNMP Object Navigator. This lists the traps available for a specific MIB. In order to receive one of these traps, your Cisco IOS® Software Release must support the MIB listed. In order to find out which MIBs are supported on your Cisco device, visit www.cisco.com/go/mibs. The MIB must be loaded into your network management system. This is commonly referred to as compiling. See your Network Management System (for instance, HP OpenView or NetView) user guide about MIB compiling on your NMS platform. Also refer to SNMP: Frequently Asked Questions About MIBs and MIB Compilers and Loading MIBs.

Additionally, a device does not send a trap to a network management system unless it is configured to do so. A device must know that it should send a trap. The trap destination is usually defined by an IP address, but can be a host name, if the device is set up to query a Domain Name System (DNS) server. In later versions of Cisco IOS software, device administrators can choose which traps they would like send. For information on how to configure a Cisco device for SNMP, and how to send traps, refer to correspondent device configuration guides and Basic Dial NMS Implementation Guide, Cisco IOS SNMP Traps Supported and How to Configure Them and How−To Support and Configure Cisco CatalystOS SNMP Traps.

Note: The manager typically receives SNMP notifications (TRAPs and INFORMs) on UDP port number 162.

Examples of Traps Sent by Cisco IOS

This section contains some examples of traps sent by Cisco IOS, taken with **debug snmp packet**.

SNMPv1 generic trap, redefined by Cisco:

```
Nov 21 07:44:17: %LINK−3−UPDOWN: Interface Loopback1, changed state to up
4d23h: SNMP: Queuing packet to 172.17.246.162
4d23h: SNMP: V1 Trap, ent products.45, addr 172.17.246.9, gentrap 3, spectrap 0
 ifEntry.1.23 = 23
 ifEntry.2.23 = Loopback1
 ifEntry.3.23 = 24
 lifEntry.20.23 = up
```

This output shows the Cisco redefined linkUp trap from CISCO−GENERAL−TRAPS MIB with four bound variables. It has these fields:

- *Enterprise* = products.45 (sysObjectID of the device sending trap, in this example, it is c7507 router)
- *Generic trap type* = 3 (linkUp)
- *Specific trap code* = 0

SNMPv1 Cisco specific trap:

```
4d23h: SNMP: Queuing packet to 172.17.246.162
4d23h: SNMP: V1 Trap, ent ciscoSyslogMIB.2, addr 172.17.246.9, gentrap 6, spectrap 1
 clogHistoryEntry.2.954 = LINK
 clogHistoryEntry.3.954 = 4
```

```
 clogHistoryEntry.4.954 = UPDOWN
 clogHistoryEntry.5.954 = Interface Loopback1, changed state to up
 clogHistoryEntry.6.954 = 43021184
```

This output shows the Cisco specific clogMessageGenerated trap from CISCO−SYSLOG−MIB with five bound variables. It has these fields:

- *Enterprise* = Enterprise value of clogMessageGenerated trap
- *Generic trap type* = 6 (enterpriseSpecific)
- *Specific trap code* = 1 (specific trap code of clogMessageGenerated)

SNMPv2c Cisco specific trap:

```
4d23h: SNMP: Queuing packet to 172.17.246.162
4d23h: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
 sysUpTime.0 = 43053404
 snmpTrapOID.0 =
 clogHistoryEntry.2.958 = SYS
 clogHistoryEntry.3.958 = 6
 clogHistoryEntry.4.958 = CONFIG_I
 clogHistoryEntry.5.958 = Configured from console by vty0 (10.10.10.10)
 clogHistoryEntry.6.958 = 43053403
```

This output shows the Cisco specific ciscoConfigManEvent SNMPv2c notification from CISCO−CONFIG−MAN−MIB with three bound variables:

- ccmHistoryEventCommandSource
- ccmHistoryEventConfigSource
- ccmHistoryEventConfigDestination

This trap can be used if there has been any changes done to the device's configuration. The values of last two components determine if a **show** command was issued or if the configuration was touched.

```
6506E#term mon
6506E#debug snmp packet
SNMP packet debugging is on

6506E#sh run
Building configuration...
...
6506E#
19:24:18: SNMP: Queuing packet to 10.198.28.80
19:24:18: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
sysUpTime.0 = 6981747
snmpTrapOID.0 = ciscoConfigManMIB.2.0.1
ccmHistoryEventEntry.3.100 = 1

!--- 1 -> commandLine. Executed via CLI.

ccmHistoryEventEntry.4.100 = 3

!--- 3 -> running

ccmHistoryEventEntry.5.100 = 2

!--- 2 -> commandSource. Show command was executed.


6506E#term mon
6506E#debug snmp packet
SNMP packet debugging is on
```

```
6506E#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
6506E(config)#exit

22:57:37: SNMP: Queuing packet to 10.198.28.80
22:57:37: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
 sysUpTime.0 = 8261709
 snmpTrapOID.0 = ciscoConfigManMIB.2.0.1
 ccmHistoryEventEntry.3.108 = 1

!--- 1 -> commandLine. Executed via CLI.

 ccmHistoryEventEntry.4.108 = 2

!--- 2 -> commandSource

 ccmHistoryEventEntry.5.108 = 3

!--- 3 -> running. Change was destined to the running configuration.
```

Related Information

- **Technical Support & Documentation − Cisco Systems**

Updated: Oct 10, 2006 Document ID: 7244