SNMP

This chapter explains Simple Network Management Protocol (SNMP) as implemented by CPT.

Understanding SNMP

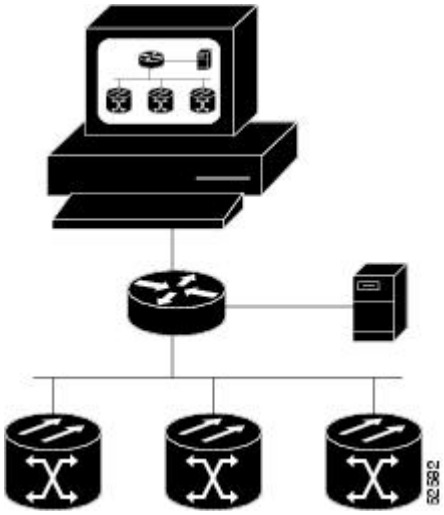
This chapter explains Simple Network Management Protocol (SNMP) as implemented by CPT.

SNMP is an application–layer communication protocol that allows network devices to exchange management information among these systems and with other devices outside the network. Through SNMP, network administrators can manage network performance, find and solve network problems, and plan network growth. SNMP makes network monitoring more cost effective and allows your network to be more reliable.

CPT supports SNMP Version 1 (SNMPv1), SNMP Version 2c (SNMPv2c), and SNMP Version 3 (SNMPv3). As compared to SNMPv1, SNMPv2c includes additional protocol operations and 64–bit performance monitoring support. SNMPv3 provides authentication, encryption, and message integrity and is more secure.

The following figure illustrates the basic layout idea of an SNMP-managed network.

Figure 1: Basic Network Managed by SNMP



The advantages of SNMP are as follows:

- SNMP is LAN based.
- SNMP is an open standard.
- SNMP can be easily extended.
- SNMP provides a common management platform for many different devices.

Understanding SNMP Components

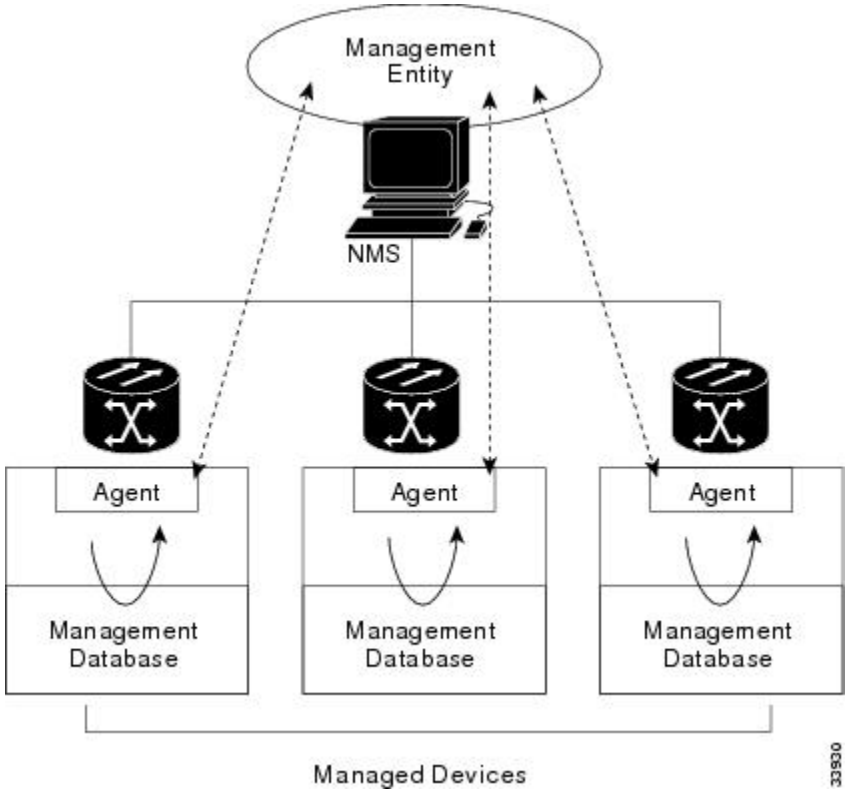
An SNMP–managed network consists of a manager, agents, and managed devices.

The manager provides the interface between the human network manager and the management system. The agent provides the interface between the manager and the physical device being managed

Management systems execute most of the management processes and provide the bulk of memory resources used for network management. A network might be managed by one or several management systems.

The following figure illustrates the relationship between the network manager, the SNMP agent, and the managed devices.

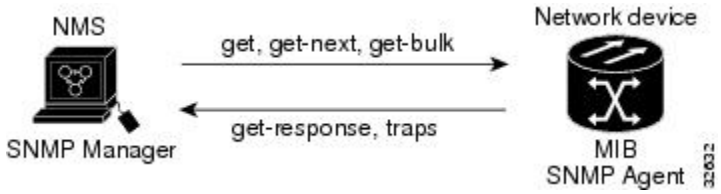
Figure 2: Example of the Primary SNMP Components



An agent residing on each managed device translates local management information data—such as performance information or event and error information—caught in software traps, into a readable form for the management system.

The following figure illustrates SNMP agent get–requests that transport data to the network management software.

Figure 3: Agent Gathering Data from a MIB and Sending Traps to the Manager



The SNMP agent captures data from MIBs, which are device parameter and network data repositories, or from error or change traps.

A managed element—such as a router, access server, switch, bridge, hub, computer host, or network element—is accessed through the SNMP agent. Managed devices collect and store management information, making it available through SNMP to other management systems having the same protocol compatibility.

Note It is recommended that the SNMP Manager timeout value be set to 60 seconds. Under certain conditions, if this value is lower than the recommended time, the TNC/TSC card can be reset. However, the response time depends on various parameters such as object being queried, complexity, number of hops in the node and so on.

Understanding MIB

The Management Information Base (MIB) is a data structure that describes SNMP network elements as a list of data objects. The SNMP manager must compile the MIB file for each equipment type in the network to monitor SNMP devices.

The manager and agent use a MIB and a relatively small set of commands to exchange information. The MIB is organized in a tree structure with individual variables being represented as leaves on the branches. A long numeric tag or object identifier (OID) is used to distinguish each variable uniquely in the MIB and in SNMP messages. The MIB associates each OID with a readable label and various other parameters related to the object. The MIB then serves as a data dictionary or codebook that is used to assemble and interpret SNMP messages.

When the SNMP manager wants to know the value of an object, such as the state of an alarm point, the system name, or the element uptime, it will assemble a GET packet that includes the OID for each object of interest. The element receives the request and looks up each OID in its code book (MIB). If the OID is found (the object is managed by the element), a response packet is assembled and sent with the current value of the object included. If the OID is not found, a special error response is sent that identifies the unmanaged object.

MIBs Supported in CPT

The following table lists the MIBs supported in CPT.

Table 1: MIBs Supported in CPT

MIB Module
BGP4-MIB.my
BRIDGE-MIB.my
CERENT-454.mib
CERENT-ENVMON-MIB.mib
CERENT-FC-MIB.mib
CERENT-GENERIC-PM-MIB.mib
CERENT-GLOBAL-REGISTRY.mib
CERENT-HC-RMON-MIB.mib
CERENT-IF-EXT-MIB.mib
CERENT-MSDWDM-MIB.mib

MIB Module
CERENT-OPTICAL-MONITOR-MIB.mib
CERENT-TC.mib
CISCO-CDP-MIB.my
CISCO-CLASS-BASED-QOS-MIB.my
CISCO-ENTITY-ASSET-MIB.my
CISCO-ENTITY-EXT-MIB.my
CISCO-ENTITY-VENDORTYPE-OID-MI
CISCO-FRAME-RELAY-MIB.my
CISCO-FTP-CLIENT-MIB.my
CISCO-HSRP-EXT-MIB.my
CISCO-HSRP-MIB.my
CISCO-IETF-PW-MIB
CISCO-IGMP-SNOOPING-MIB.mib
CISCO-IMAGE-MIB.my
CISCO-IPMROUTE-MIB.my
CISCO-IP-STAT-MIB.my
CISCO-MEMORY-POOL-MIB.my
CISCO-OPTICAL-MONITOR-MIB.mib
CISCO-PING-MIB.my
CISCO-PORT-QOS-MIB.my
CISCO-PROCESS-MIB.my
CISCO-PRODUCTS-MIB.my
CISCO-REP-MIB.my
CISCO-SMI.mib
CISCO-SYSLOG-MIB.my
CISCO-TC.my
CISCO-TCP-MIB.my
CISCO-VLAN-IFTABLE-RELATIONSHIP
entityMIB
entityx.mib
EtherLike-MIB-rfc2665.mib

MIB Module
HCNUM-TC.mib
HC-PerfHist-TC-MIB.my
HC-RMON-rfc3273.mib
IANAifType-MIB.mib
IANA-RTPROTO-MIB.my
IEEE8023-LAG-MIB.my
IEEE-802DOT17-RPR-MIB.my
IF-MIB-rfc2233.mib
IGMP-MIB.my
INET-ADDRESS-MIB.mib
IPMROUTE-STD-MIB.my
MPLS-TE-MIB
OLD-CISCO-TCP-MIB.my
OLD-CISCO-TS-MIB.my
OSPF-MIB.my
P-BRIDGE-MIB-rfc2674.mib
PerfHist-TC-MIB-rfc2493.mib
PIM-MIB.my
Q-BRIDGE-MIB-rfc2674.mib
RFC1155-SMI.my
RFC1213-MIB.mib
RFC1253-MIB-rfc1253.mib
RFC1315-MIB.my
RIPv2-MIB-rfc1724.mib
RMON2-MIB-rfc2021.mib
RMON-MIB-rfc2819.mib
RMONTOK-rfc1513.mib
SNMP-FRAMEWORK-MIB-rfc2571.mib
SNMP-MPD-MIB.mib
SNMP-NOTIFICATION-MIB.my
SNMP-NOTIFY-MIB-rfc3413.mib

MIB Module
SNMP-PROXY-MIB-rfc3413.mib
SNMP-TARGET-MIB-rfc3413.mib
SNMP-USER-BASED-SM-MIB-rfc3414.mib
SNMPv2-MIB-rfc1907.mib
SNMPv2-SMI.my
SNMPv2-TC.my
SNMP-VIEW-BASED-ACM-MIB-rfc3415.mib
TCP-MIB.my
TOKEN-RING-RMON-MIB.my
UDP-MIB.my

Understanding SNMP Traps

CPT uses SNMP traps to generate all the alarms and events. The traps contain the following information:

- Object IDs that uniquely identify each event with information about the generating entity.
- Severity and service effect of the alarm (critical, major, minor, or event; service-affecting or non-service-affecting).
- Date and time stamp showing when the alarm occurred.

Generic IETF Traps

CPT supports the generic IETF traps listed in the following table.

Table 2: Supported Generic IETF Traps

Trap	Description
coldStart	Agent up, cold start.
warmStart	Agent up, warm start.
authenticationFailure	Community string does not match.
newRoot	Sending agent is the new root of the spanning tree.
topologyChange	A port in a bridge has changed from Learning to Forwarding or Forwarding to Blocking.
entConfigChange	The entLastChangeTime value has changed.

Trap	Description
risingAlarm	The SNMP trap that is generated when an alarm entry crosses the rising threshold and the entry generates an event that is configured for sending SNMP traps.
fallingAlarm	The SNMP trap that is generated when an alarm entry crosses the falling threshold and the entry generates an event that is configured for sending SNMP traps.

Examples of IETF Traps

coldStart

```
DISMAN-EVENT-MIB::sysUpTimeInstance = 21775
SNMPv2-MIB::snmpTrapOID.0 = SNMPv2-MIB::coldStart
CERENT-454-MIB::cerent454NodeTime.0 = 20110705135346D
CERENT-454-MIB::cerent454AlarmState.1.1 = administrative
SNMP-COMMUNITY-MIB::snmpTrapAddress.0 = 10.64.106.142"
```

warmStart

```
DISMAN-EVENT-MIB::sysUpTimeInstance = 21775
SNMPv2-MIB::snmpTrapOID.0 = SNMPv2-MIB::warmStart
CERENT-454-MIB::cerent454NodeTime.0 = 20110705135346D
CERENT-454-MIB::cerent454AlarmState.1.1 = administrative
SNMP-COMMUNITY-MIB::snmpTrapAddress.0 = 10.64.106.142"
```

authenticationFailure

```
DISMAN-EVENT-MIB::sysUpTimeInstance = 6335948
SNMPv2-MIB::snmpTrapOID.0 = SNMPv2-MIB::authenticationFailure
CERENT-454-MIB::cerent454NodeTime.0 = 20110705121300D
CERENT-454-MIB::cerent454AlarmState.1.1 = administrative
SNMP-COMMUNITY-MIB::snmpTrapAddress.0 = 10.64.106.142"
```

newRoot

```
RFC1213-MIB::sysUpTime.0 = 255172
SNMPv2-MIB::snmpTrapOID.0 = BRIDGE-MIB::newRoot
CERENT-454-MIB::cerent454NodeTime.0 = 20000125062804S
CERENT-454-MIB::cerent454AlarmState.1.1 = notAlarmedNonServiceAffecting
CERENT-454-MIB::cerent454AlarmSeverity.1.1 = notAlarmed
CERENT-454-MIB::cerent454AlarmStatus.1.1 = transient
CERENT-454-MIB::cerent454AlarmServiceAffecting.1.1 = nonServiceAffecting
SNMPv2-SMI::snmpModules.18.1.3.0 = 10.64.104.11
```

topologyChange

```
RFC1213-MIB::sysUpTime.0 = 254973
SNMPv2-MIB::snmpTrapOID.0 = BRIDGE-MIB::topologyChange
CERENT-454-MIB::cerent454NodeTime.0 = 20000125062802S
CERENT-454-MIB::cerent454AlarmState.1.1 = notAlarmedNonServiceAffecting
CERENT-454-MIB::cerent454AlarmSeverity.1.1 = notAlarmed
CERENT-454-MIB::cerent454AlarmStatus.1.1 = transient
CERENT-454-MIB::cerent454AlarmServiceAffecting.1.1 = nonServiceAffecting
SNMPv2-SMI::snmpModules.18.1.3.0 = 10.64.104.11
```

entConfigChange

```
DISMAN-EVENT-MIB::sysUpTimeInstance = 6246394
SNMPv2-MIB::snmpTrapOID.0 = ENTITY-MIB::entConfigChange
CERENT-454-MIB::cerent454NodeTime.0 = 20110705115804D
CERENT-454-MIB::cerent454AlarmState.4096.1 = administrative
SNMP-COMMUNITY-MIB::snmpTrapAddress.0 = 10.64.106.142"
```

risingAlarm

```
SNMPv2-MIB::sysUpTime.0 = 39547235
SNMPv2-MIB::snmpTrapOID.0 = RMON-MIB::risingAlarm
RMON-MIB::alarmIndex.1 = 1
RMON-MIB::alarmVariable.1 = IF-MIB::ifInOctets.16409
RMON-MIB::alarmSampleType.1 = absoluteValue
RMON-MIB::alarmValue.1 = 0
RMON-MIB::alarmRisingThreshold.1 = 100
CERENT-454-MIB::cerent454NodeTime.0 = 20090402234612D
CERENT-454-MIB::cerent454AlarmState.16409.1 = administrative
SNMP-COMMUNITY-MIB::snmpTrapAddress.0 = 10.64.105.113"
```

fallingAlarm

```
SNMPv2-MIB::sysUpTime.0 = 39463718
SNMPv2-MIB::snmpTrapOID.0 = RMON-MIB::fallingAlarm
RMON-MIB::alarmIndex.7 = 7
RMON-MIB::alarmVariable.7 = EtherLike-MIB::dot3StatsFCSErrors.16409
RMON-MIB::alarmSampleType.7 = deltaValue
RMON-MIB::alarmValue.7 = 0
RMON-MIB::alarmFallingThreshold.7 = 500
CERENT-454-MIB::cerent454NodeTime.0 = 20090402233217D
CERENT-454-MIB::cerent454AlarmState.16409.1 = administrative
SNMP-COMMUNITY-MIB::snmpTrapAddress.0 = 10.64.105.113"
```

SNMP Traps Supported in CPT

The following table lists the SNMP traps supported in CPT.

Table 3: SNMP Traps Supported in CPT

MIB Module
pseudowireDown
workingPseudowireControlPlainDown
protectPseudowireControlPlainDown
workingPseudowireConnectivityCheckDown
protectPseudowireConnectivityCheckDown
pseudowireTrafficSwitchedToProtection
workingPseudowireLocalAcTxPortFault
protectPseudowireLocalAcTxPortFault
workingPseudowireLocalAcRxPortFault
protectPseudowireLocalAcRxPortFault
workingPseudowireRemoteAcTxPortFault
protectPseudowireRemoteAcTxPortFault
workingPseudowireRemoteAcRxPortFault
protectPseudowireRemoteAcRxPortFault

MIB Module
workingRemotePseudowireNotForwarding
protectRemotePseudowireNotForwarding
tpTunnelDown
workingLabelSwitchedPathDown
protectLabelSwitchedPathDown
bidirectionalForwardDetectionDown
tpTrafficSwitchedFromWorkingToProtection
workingTpLockout
protectTpLockout
ethernetFlowPointFailed
teTunnelDown
macSystemLimitReached
macBridgeDomainLimitReached
packetTransportServiceFailed
satellitePanelDiscoveryFailure
satellitePanelActiveLinkFailure
satellitePanelCommunicationFailure
satellitePanelImproperConfiguration
satellitePanelFanMismatchOfEquipmentAndAttributes
satellitePanelFanFailure
satellitePanelPartialFanFailure
satellitePanelFANManufacturingDataMemoryEEPROMFailure
satellitePanelFANUnitIsMissing
satellitePanelIndustrialHighTemperature
satellitePanelHighTemperature
satellitePanelBatteryFailureA
protectionCardConfigurationMismatch
routerProcessorSwitchOver
runningLowOnResources
noMoreResourcesAreAvailable
licenseWillExpireWithin24Hours

MIB Module
licenseWillExpireAnytimeAfter1DayButBefore14Days
licenseIsExpired
temporaryLicenseIsInUse
evaluationLicenseIsInUse
licenseIsMissing
SMBackwardIncomingAlignmentError
resourceAllocationFailed
workingLabelSwitchedPathLinkDownIndication
protectLabelSwitchedPathLinkDownInication
workingLabelSwitchedPathLockReport
protectLabelSwitchedPathLockReport
satellitePanelBatteryFailureB
coolingProfileMismatch
trunkOduAlarmIndicationSignal
companionCardMissing
powerConsumptionLimitHasCrossed
controlPlaneUnverifiedClearedAlarmsPresent
singleSpanFail
multipleSpanFail
topoMisConfig
dbSyncFail
dbLoss
DUALHOME_STATE_CHANGE_TRAP
fastAutomaticProtectionSwitchingConfigMismatch
ftaMismatch
frontPortLinkLoss
workQueueFull
equipmentPowerFailureAtConnectorA
equipmentPowerFailureAtConnectorB
equipmentPowerFailureAtReturnConnectorA
equipmentPowerFailureAtReturnConnectorB

MIB Module
openIOSlots
voaControlLoopDisableDueToExcessiveCounterPropagationLight
eqptDegrade
plannedSwitchOver
licenseCountViolation
slaThresholdCrossAlert
primarySynchronizationReferenceFailure
secondarySynchronizationReferenceFailure
thirdSynchronizationReferenceFailure
regeneratorSectionTraceIdentifierMismatch
workQueueFull
SMBackwardIncomingAlignmentError
lossOfSynchronization
outOfSynchronization
failedToReceiveSynchronizationStatusMessage
synchronizationStatusMessagesAreDisabledOnThisInterface
stratum1PrimaryReferenceSourceTraceable
stratum2Traceable
stratum3Traceable
stratum3ETraceable
stratum4Traceable
synchronizedTraceabilityUnknown
transitNodeClockTraceable
sonetMinimumClockTraceable
doNotUseForSynchronization
reservedForNetworkSynchronizationUse
automaticSystemReset

Understanding SNMP Community Names

Community names are used to group SNMP trap destinations. All the trap destinations can be provisioned as part of SNMP communities in CTC. When community names are assigned to traps, the request is treated as valid if the community name matches one that is provisioned in CTC. In this case, all agent-managed MIB variables are accessible to that request. If the community name does not match the provisioned list, SNMP drops the request.

Accessing Fabric Card Through SNMP

Each fabric card runs a separate instance of SNMP. SNMP requests are relayed to the individual fabric card based on the community string. The community string uses the following format:

com_str_configured_from_CTC@fabric_card_slot_number

Understanding SNMP Messages

SNMP uses the following messages to communicate between the manager and the agent.

- Get
- GetNext
- GetResponse
- Set
- Trap

The Get and GetNext messages allow the manager to request information for a specific variable. The agent, upon receiving a Get or GetNext message, will issue a GetResponse message to the manager with either the information requested or an error indication as to why the request cannot be processed.

A Set message allows the manager to request a change be made to the value of a specific variable in the case of an alarm remote that will operate a relay. The agent will then respond with a GetResponse message indicating the change has been made or an error indication as to why the change cannot be made.

The Trap message allows the agent to inform the manager of an important event. An SNMP Trap is a change–of–state (COS) message—it could mean an alarm, a clear or simply a status message.