CHAPTER 7

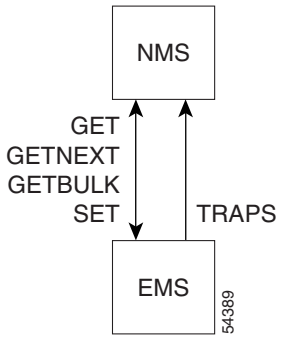
SNMP Interface

Revised: July 21, 2009, OL-4495-10

Introduction

The Cisco BTS 10200 Softswitch supports Simple Network Management Protocol (SNMP) operations that allow communications between the Element Management System (EMS) and a service provider's network management system (NMS). The EMS sends SNMP traps to the NMS, and the NMS can query the EMS for specific data elements (see Figure 7-1). Status and control operations as well as traffic and statistics query operations are supported.

Figure 7-1 NMS/EMS Interaction Via SNMP



Element Management System (SNMP Agent)

The Softswitch SNMP Agent supports SNMPv2c operations defined by the opticall.mib Management Information Base (MIB). The MIB is located in the directory /opt/BTSsnmp/etc on the EMS. The MIB opticall.mib uses variables from three other MIBs: IPCELL-TC, SNMPv2-TC, and SNMPv2-SMI. The NMS needs to load the main MIB (opticall.mib), that will in turn import the three other MIBs.

SNMP Agent Functions

The following functions are supported by the Softswitch SNMP Agent:

- MIB-II System branch information
- Collection of statistics and traffic management data
- Status and control
- Bulk Status and control
- SNMP trap reports
- SNMP trap retransmission

Read access to the SNMP agent is required for the statistics and traffic management queries and for status queries. Write access is required for the control commands. Trap reports do not involve read/write access.

Read/write access to the SNMP agent is restricted by requiring the NMS to pass a valid community string to the agent. The community string passed on by the NMS message is authenticated against a list of community strings maintained by the SNMP agent. The SNMP agent uses each string as a password, and disallows access if the password is not valid.

In addition, to perform Status/Control via SNMP, the community string used must also be at a minimum level to perform those commands as defined by the BTS 10200 Command Line Interface (CLI) security privileges. For example, if the community string used to control a termination in service (INS) is below the minimum level in the CLI, then the SNMP request fails with *General Error.*

The SNMP community table in the Softswitch database provides persistent storage of community strings for the SNMP agent. The default value for both the read and write communities is “public”. This default value can be deleted by the user and replaced with specific communities using the following CLI commands:

- To show all read communities-show snmpconfig type=readcommunity
- To show all write communities-show snmpconfig type=writecommunity
- To add read community-add snmpconfig type=readcommunity; value=.....; key1=command_level; value1=8;
- To add write community-add snmpconfig type=writecommunity; value=.....; key1=command_level; value1=8;
- To delete read community-delete snmpconfig type=readcommunity; value=.....
- To delete write community-delete snmpconfig type=writecommunity; value=.....

The provisioned values must be ASCII strings and can be up to 64 characters long.

Statistics/Traffic Measurement

Statistical data (traffic measurements) are collected for the following components of the Softswitch:

- AINSVC
- Announcement
- Audit
- Billing
- Call Processing
- DQOS
- Element Manager
- H323

- INAP
- ISDN
- ISUP SGA
- M3UA
- MGCP Adapter
- POTS-Feature Server
- SCCP
- SCTP
- SIA
- SIM
- SNMP
- SUA
- TCAP
- Trunk group usage
- TSA

SNMP Trap Reports

Traps are sent from the Softswitch SNMP agent to the NMS. Traps are mapped to all alarms generated from the EMS. Any alarms that cannot be mapped to a specific trap are mapped to a generic trap. Mapped traps and generic traps contain one or more of the following information types, depending upon availability of the information:

- Severity level
- Alarm ID associated with the trap
- Alarm category
- Set/Cleared flag
- Component (instance) ID
- Component type
- Details of the trap
- Time that trap was generated

An operator of an NMS who would like to receive traps from the SNMP agent needs to add an entry to SNMPTRAPDEST via CLI. For the specific CLI command, refer to the *CLI Reference Guide*.

- IP address or hostname of the NMS
- Port number on which to receive traps
- Community string (currently not used)
- Owner string (currently not used)
- Filter Types - Ranges from 0-32767. Default is 32767. A bitmask that specifies which subsystem types of the events to filter or permitted to be sent to this address. This is used in combination with Filterlevels to provide a granular filter for traps from the SNMP Agent side. From right to left the following bits specifies the following subsystem types: Bit #1: BILLING (right-most bit) Bit #2:

CALLP Bit #3: CONFIG Bit #4: DATABASE Bit #5: MAINTENANCE Bit #6: OSS Bit #7: SECURITY Bit #8: SIGNALING Bit #9: STATISTICS Bit #10: SYSTEM Bit #11: AUDIT (left-most bit) For example, in order to receive only CONFIG, DATABASE, and SIGNALING traps, then the filter in binary would be (0010001100) which converts to integer value of (140) to be entered as the token value. If all types are to be received then the binary would be (1111111111) which converts to (1023) as the token value. And likewise if no types are to be received then the binary would be (0000000000) which converts to (0) as the token value.

- Filter Levels - Ranges from 0-63. Default is 56. A bitmask that specifies which levels of the events to filter or permitted to be sent to this address. This is used in combination with Filtertypes to provide a granular filter for traps from the SNMP Agent side. From right to left the following bits specifies the following levels: Bit #1: DEBUG (right-most bit) Bit #2: INFO Bit #3: WARNING Bit #4: MINOR Bit #5: MAJOR Bit #6: CRITICAL (left-most bit). For example, in order to receive only INFO, MINOR, and MAJOR traps, then the filter in binary would be (011010) which converts to integer value of (26) to be entered as the token value. If all levels are to be received then the binary would be (111111) which converts to (63) as the token value. And likewise if no levels are to be received then the binary would be (000000) which converts to (0) as the token value. WARNING: Any filter that permits levels DEBUG and/or INFO traps will cause a high number of traps to be sent and tax the system resources on the SNMP Agent on the EMS; care must be taken when specifying numbers that converts to binary containing bits 1 or 2 turned on such as (49-51), (57-59), (61-63), etc.

Once this is done, the NMS will start receiving traps.

Note It is the responsibility of the NMS operator to filter the traps that are displayed on the NMS and those that are discarded.

Status and Controls

Querying and Controlling EMS, BDMS, CA and FS

Status queries on the following components can be performed by simple GET/GETNEXT operations:

- Primary and secondary EMS
- Primary and secondary BDMS
- Primary and secondary CA
- Primary and secondary POTS/Centrex/Tandem FS
- Primary and secondary AIN FS

Controls can be performed on these components using the SET operation, but only on the primary component (primary EMS, primary CA and primary FS). The primary component, in turn, controls the secondary component. If the operator tries to perform a SET operation on the secondary component, the agent returns an error.

Querying and Controlling Various Components

Status queries on the current status of the following components can be performed by GET/GETNEXT operations on the variousState columns in the MIB.

- Media Gateway (MGW)

- Trunk Group (TG)
- Subscriber Termination
- Trunk Termination
- SGP
- DPC
- SCTP Association

In addition, controls can be performed on the following components by SETs on various columns in the MIB.

- Media Gateway (MGW)
- Trunk Group (TG)
- Subscriber Termination
- Trunk Termination
- SCTP Association

Controls can be performed as follows:

Step 1 Perform SET operations on all the necessary fields (Mode column and TargetState column, and so forth).

Step 2 Perform a SET operation on the ControlState column, using the value of **1** (commit) to actually put the component into its target state.

If you perform a GET/GETNEXT operation on the ControlState when all necessary fields are NOT set, then a value of **2** (insufficient-data) is returned. If all necessary fields ARE set, then a value of **3** (ready-to-commit) is returned.

Querying and Controlling Bulk Status of Various Components

Bulk Status queries on the current status of the following components can be performed by GET/GETNEXT operations on the following branch: .iso.org.dod.internet.private.enterprises.ipcell.opticall.statusControlBulk. The results return from querying these components are a Page Number column and a Status Value column. The Status Value column specifies each component id and the statuses in enumerated value correlating to the statuses of those components above.

- Media Gateway (MGW) - The Status Value column specifies delimited string of X number of MGWs. The delimiters are: ';' = delimits each MGW. '|' = delimits status fields. The protocol of which is: 'MGW_ID|admin_state|oper_state;MGW_ID|admin_state...' And example: 'MGW_ABC|1|3;MGW_XYZ|1|3...' The enumerated states are the same as that of mediaGatewayOAMPTable.
- Trunk Group (TG) - The Status Value column specifies delimited string of X number of Trunk Groups. The delimiters are: ';' = delimits each Trunk Group. '|' = delimits status fields. The protocol of which is: 'TGN_ID|admin_state|oper_state;TGN_ID|admin_state...' And example: '232|1|3;233|1|3...' The enumerated states are the same as that of trunkGroupOAMPTable.

- Subscriber Termination - The Status Value column specifies delimited string of X number of Subscriber Terminations. The delimiters are: ';' = delimits each Subscriber Termination. '|' = delimits status fields. The protocol of which is: 'SUB_ID|admin_state|oper_state;SUB_ID|admin_state...' And example: 'SUB_ABC|1|3;SUB_XYZ|1|3...' The enumerated states are the same as that of subscrLineTermOAMPTable.
- Trunk Termination - The Status Value column specifies delimited string of X number of Trunk Terminations. The delimiters are: ';' = delimits each Trunk Termination. '|' = delimits status fields. '.' = delimits CIC and TGN_ID. The protocol of which is: 'TGN_ID.CIC|admin_state|oper_state|static_state|dynamic_state;CIC.TGN_ID|admin_state...' And example: '232.22|1|3|1|1;233.22|1|3|1|1...' The enumerated states are the same as that of trunkTermOAMPTable.

Accessing the Sun Solaris SNMP Agent

There are two possible methods to access and query the Sun Solaris SNMP Agent:

1. Directly through a non-standard SNMP port.

Note The Cisco BTS 10200 Softswitch SNMP Master Agent cannot be full when accessing or querying through a non-standard SNMP port.

2. Through the Cisco BTS 10200 Softswitch SNMP Master Agent using standard port 161.

Note You must be SNMPv2c compliant because the Cisco BTS 10200 Softswitch SNMP Agent proxies queries to the Sun Solaris Agent, which only supports SNMPv1. You must specify SNMPv1 as the SNMP version when accessing or querying using standard port 161.

Accessing and Querying a Non-Standard SNMP Port Directly

Perform the following steps to access and query the Sun Solaris SNMP Agent using a non-standard SNMP port:

Step 1 Modify following parameters in /etc/snmp/conf/snmpd.conf file:

a. read-community

- specify a user-defined community string for read access
- enter only one value

b. managers

- specify the IP or hostname for querying the entity (NMS)
- you can specify multiple entries delimited by spaces

Note You must retain the localhost as one of the entries in order to retain communication with BTS SNMP Master Agent.

Step 2 Restart Sun Solaris SNMP Agent

```
/etc/init.d/S98mibiisa stop
/etc/init.d/S98mibiisa start
```

Step 3 Begin directly querying Sun Solaris SNMP Agent with the specified read-community string using port 13230.

Accessing and Querying the Sun Solaris SNMP Agent

Perform the following steps to access and query the Sun Solaris SNMP Agent through the Cisco BTS 10200 Softswitch SNMP Master Agent Using Port 161:

Step 1 Modify following parameters in /etc/snmp/conf/snmpd.conf file:

a. read-community

- specify a user-defined community string for read access
- enter only one value

b. managers

- verify that the localhost is one of the entries

Step 2 Modify the SNMP configuration type and value:

```
add snmpconfig type=SETTING; value=COUPLE_SUN_AGENT
```

Step 3 Restart BTS SNMP Master Agent.

Step 4 Log in as root.

```
kill `ps -ef | grep -i sad | grep -v grep | awk '{print $2}'`
```

Step 5 Begin querying the Sun Solaris SNMP Agent object id (OID) with the specified read-community string using standard port 161.

Enabling NMS to Query/Poll Solaris SNMP Agent

The EMS runs two SNMP agents as follows:

- SAD (SNMP agent adapter)
- Solaris SNMP agent

The active EMS node runs the SAD process, which converts the BTS 10200 specific events/alarms into SNMP traps and sends them to the configured SNMP Trap listeners or the NMSes. The SAD process handles the SNMPWALK/GET/GETNEXT/SET on the OIDs that are defined in the opticall.mib file. The SAD process also runs on the standby EMS, but does not perform any function.

Note The SAD process does not run on the CA nodes.

Note The CA runs only the standard Solaris SNMP agent.

The standard Solaris SNMP agent runs on both the active and standby EMS and CA nodes. Therefore, all the four nodes generate the solaris-level traps. The name of the standard Solaris SNMP agent is **mibiisa**, which runs on port number 13230. The Solaris SNMP agent can be used to collect the sun box related statistics and/or traps. Note that the **mibiisa** supports only those OIDs (object identifiers) that are defined in the SUN MIB.

Note The active/standby EMS and active/standby CA nodes generate the solaris-level traps, whereas only the active EMS generates BTS-specific traps and sends them to NMS. The NMS can query/poll all the four nodes to receive the generated traps.

To enable the NMS to directly query the Solaris SNMP agent for a range of OIDs specified by SUN MIBs, and receive Solaris box-level traps, do the following:

1. Open the /etc/snmp/conf/snmpd.conf file.
2. Define the read-community as “public”.
3. In the “Managers” field, enter the IP address or hostname of the NMS from where the user needs to send the SNMP query. Enter multiple addresses separated by spaces, but leave the “localhost” entry as is.
4. In the Trap field, configure the IP address or hostname of the NMS where the traps have to be sent.
5. Restart the SNMP agent, enter:

```
/etc/init.d/S98mibiisa stop
/etc/init.d/S98mibiisa start
```

6. Query the SNMP agent (using SNMPGET/SNMPWALK) from the Manager using the read-community and port 13230. For example, to get the system up time, enter the following command:

```
snmpwalk -c public -p 13230 prica07 system
```

The output appears as given below:

```
system.sysDescr.0 = Sun SNMP Agent
system.sysObjectID.0 = OID: enterprises.42.2.1.1
system.sysUpTime.0 = Timeticks: (279199168) 32 days, 7:33:11.68
system.sysContact.0 = System administrator
system.sysName.0 = prica07
system.sysLocation.0 = System administrators office
system.sysServices.0 = 72
```