

LightStream 2020 Statistics and Data Collection

Statistics are counters collected by a LightStream 2020 multiservice ATM switch (LS2020 switch). This chapter tells you how to monitor and evaluate the state and performance of your LS2020 switch by reviewing and sometimes processing the statistics.

Viewing Lightstream 2020 Statistics

LS2020 statistics are stored in MIB objects. Examples of statistics include the following MIB objects:

- ifInOctets
- ifInErrors
- ifOutErrors

Many statistics of interest for the LS2020 switch are defined in some of the standard MIBs and the LS2020 private MIB. For a list of standard MIBs and information on the LS2020 private MIB, see the *LightStream 2020 CLI Reference Manual*.

All statistics in an LS2020 switch are initialized at system startup. When they reach their maximum value (or if you do another startup), they are reset to 0. You can use the collector to collect statistics at regular intervals.

Viewing Port Statistics

To view port statistics, use the **show** command. All of the information displayed for port interfaces is generic and applies to all interfaces, regardless of the interface type.

Step 1 Verify that the target switch is set correctly by entering the following at the `cli>` prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, see the “Setting the Target Switch for CLI Commands” section in the chapter entitled “Command Line Interface.”

Step 2 Enter the following at the `cli>` prompt:

```
cli> show port <port#> statistics
```

where

<port#> is the number of the port for which statistics are displayed. The port number is in card.port format (card = 2 - 10; port = 0 - 7).

The following is an example of the information that displays when you enter **show port 2.0 statistics**:

```
cli> show port 2.0 statistics
Octets Rcvd:          372684
Normal Packets Rcvd:  432684
Multicast Packets Rcvd: 0
Discarded Rcvd Packets: 0
Receive Errors:       0
Unknown Protocols Rcvd: 0
Octets Sent:          1977615601
Normal Packets Sent:  20643144
Multicast Packets Sent: 0
Discarded Output Packets: 0
Output Errors:        1
```

where

Octets Rcvd is the total octets received from the media.

Normal Packets Rcvd is the number of unicast packets delivered (a portion of the total).

Multicast Packets Rcvd is the number of broadcast/multicast packets delivered (a portion of the total).

Discarded Rcvd Packets is the packets from the switch discarded due to resource limitation.

Receive Errors is the packets discarded due to format error.

Unknown Protocols Rcvd is the packets destined for unknown protocols.

Octets Sent is the total octets sent on the media.

Normal Packets Sent is the number of unicast packets sent (a portion of the total).

Multicast Packets Sent is the number of broadcast/multicast packets sent (a portion of the total).

Discarded Output Packets is the packets discarded due to resource limitation.

Output Errors is the packets discarded due to error.

The first time you request statistics for a particular port, you see a display like the one shown above. However, if you request statistics on that port again, the current value of each statistic displays with additional information. The additional information shows the amount that each statistic increased since your last request and the rate at which the statistic is changing. The following example shows the statistics that display when you enter **show port 2.0 statistics** command anytime after the first time.

```
cli> show port 2.0 statistics
Octets Rcvd:          372684          (Increase: 2058      Rate: 8.43/sec)
Normal Packets Rcvd:  432684          (Increase: 244      Rate: 1.00/sec)
Multicast Packets Rcvd: 0             (Increase: 0        Rate: 0.00/sec)
Discarded Rcvd Packets: 0             (Increase: 0        Rate: 0.00/sec)
Receive Errors:       0             (Increase: 0        Rate: 0.00/sec)
Unknown Protocols Rcvd: 0             (Increase: 0        Rate: 0.00/sec)
Octets Sent:          1977615601      (Increase: 11505204  Rate: 4152.47/sec)
Normal Packets Sent:  20643144        (Increase: 12009     Rate: 492.20/sec)
Multicast Packets Sent: 0             (Increase: 0        Rate: 0.00/sec)
Output Errors:        1             (Increase: 0        Rate: 0.00/sec)

Discarded Output Packets: 0
Output Errors:        1
```

Using Data Collection

The LS2020 switch provides a data collection facility called the collector. The collector allows you to define a set of MIB objects, collect their values at regular intervals, and save the information in a file. The set of MIB objects and their collected values is a collection.

The collector maintains a table called collectTable in the private section of the LS2020 MIB. It uses the information in this table to control the operations of its collections. Table 7-1 shows a sample collection table. Each row defines a collection.

Table 7-1 Sample Collection Table

Collection Number	Collection Status	Operational Status	Begin Time	End Time	Interval	File Name	File Size
1	Valid	Running	10:00:00 1/4/95	11:00:00 1/4/95	60	collect.1	100
2	Under creation	Waiting	12:30:15 1/14/95	12:00 1/28/95	360	collect.2	100
75	Valid	Running	00:01:00 4/2/95	00:01:00 4/3/95	3600	collect.75	50

You can use the **begintime** and **endtime** arguments to control the collection duration and the **filesize** argument to control the collection size. If you do not specify the **begintime** and **endtime** arguments, the collection runs continually.

The collector can run up to 25 collections simultaneously on a particular switch. If you attempt to start more than 25 collections, any additional collections have an operational status of waiting. When the number of running collections falls below 25, the next waiting collection begins running.

You can collect any set of MIB objects you want at various time intervals. For example, you can define a collection of ifInErrors and ifOutErrors objects for a particular set of ports to be collected once an hour or once a day. Once the collection starts, the specified information is collected at the correct interval and appended to the collection file.

The collector automatically generates the file and file name for every collection that you create. The file name is of the form collect.<collection number>. All collection files are placed in the /usr/tmp/collector directory. Sample names are /usr/tmp/collector/collect.6 and /usr/tmp/collector/collect.128.

Collection files are circular ASCII files containing the time each object was collected, the object name, and the object value. Figure 7-1 shows a sample collection file. When the circular file is full, the oldest data (including start date) is overwritten by newer data. Files are displayed in chronological order from the oldest entries to the newest. The default for the maximum file size is 100 kilobytes (KB).

Figure 7-1 Sample Collection File

12:13:37	ifInUcastPkts.7000	0
12:13:37	ifInUcastPkts.7001	7790663
12:13:37	ifInUcastPkts.9000	0
12:13:37	ifInUcastPkts.9001	0
12:13:37	ifInUcastPkts.9002	45347
12:13:37	ifInUcastPkts.9003	279082332
12:13:37	ifInUcastPkts.9004	45346
12:13:37	ifInUcastPkts.9005	278173235
12:13:37	ifInUcastPkts.9006	239398
12:13:37	ifInUcastPkts.9007	57462208
12:14:37	ifInUcastPkts.7000	2124113265
12:14:37	ifInUcastPkts.7001	1261205209
12:14:37	ifInUcastPkts.9000	123177948
12:14:37	ifInUcastPkts.9001	2063392
12:14:37	ifInUcastPkts.9002	740698
12:14:37	ifInUcastPkts.9003	90800255
12:14:37	ifInUcastPkts.9004	237879
12:14:37	ifInUcastPkts.9005	240500
12:14:37	ifInUcastPkts.9006	46828
12:14:37	ifInUcastPkts.9007	543157670

H3380

You can view and use the collection files as they are or you can process the data. (You may want to sort the data or make calculations to determine when changes in performance or state occur.)

You may want to perform these functions on your collections:

- Specify a start and end time
- View the data that has been collected
- List all collections that have been defined
- Stop a particular collection
- Delete a particular collection

To begin a new collection, follow these procedures in the order given here:

- 1 Create a collection
- 2 Set the collection time interval
- 3 Add or delete collection objects
- 4 Specify collection size
- 5 Start the collection
- 6 Specify collection start and end times

Each of these procedures is explained in the sections that follow.

Creating a Collection

These steps enable you to create a collection. They set up a row in the `collectTable` object for a collection with the number you specify.

- Step 1** Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

- Step 2** Enter the following at the `cli>` prompt:

```
cli> set collection <collection number> create
```

where

<collection number> is an integer between 1 and 128, inclusive.

- Step 3** To verify that the collection has been created, enter the following at the `cli>` prompt:

```
cli> show collection <collection number>
```

Setting the Collection Time Interval

These steps enable you to set the time interval for collecting data for a particular collection:

- Step 1** Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

- Step 2** Enter the following at the `cli>` prompt:

```
cli> set collection <collection number> frequency <interval>
```

where

<collection number> is the number of a collection for which you are setting the collection interval.

<interval> is the interval (in seconds) at which you want to collect the data. The default is 60 seconds.

Note If you set extremely short intervals to collect many objects, there is a noticeable difference in response time.

Step 3 To verify the collection interval for a particular collection, enter the following at the **cli>** prompt:

```
cli> show collection <collection number>
```

Check the information in the Collection Interval field.

Adding or Deleting Collection Objects

This procedure explains how to specify a MIB object and add it to or delete it from a collection. Each MIB object is added or deleted individually.

Adding Objects to a Collection

Step 1 Set the SNMP community to a read/write community by entering the following at the **cli>** prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

Step 2 If you do not know which MIB objects are available for collection, use the **browse** or the **walksnmp** command (described in the “SNMP Commands” chapter) to list the objects that are available.

Step 3 To add MIB objects to a collection, enter the following at the **cli>** prompt:

```
cli> set collection <collection number> addvar <MIB address>
```

where

<collection number> is the number of a collection for which you are adding objects.

<MIB address> is the MIB name or address of the object you want to add to the collection. The MIB address must include the instance identifier as described in the “SNMP Commands” chapter.

For example, to collect the ifInOctets MIB object for ports 1 and 2 on card 3, enter the following at the **cli>** prompt:

```
cli> set collection 10 addvar ifInOctets.3001
```

```
cli> set collection 10 addvar ifInOctets.3002
```

- Step 4** To verify that the object(s) you specified in step 3 are collected, enter the following from the `cli>` prompt:

```
cli> show collection <collection number>
```

Check the information in the Collection Items field. You should see the MIB object(s) listed that you added to the collection.

Deleting Objects from a Collection

- Step 1** Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

- Step 2** To delete objects from a collection, enter the following at the `cli>` prompt:

```
cli> set collection <collection number> delvar <MIB address>
```

where

<collection number> is the number of a collection for which you are deleting objects.

<MIB address> is the MIB name or address of the object you want to delete from the collection.

- Step 3** To display the current objects to be collected for the collection, enter the following at the `cli>` prompt:

```
cli> show collection <collection number>
```

Check the information in the Collection Items field. The MIB object you specified should not appear in the display.

Specifying Collection Size

To specify the size of a collection file, follow these steps:

- Step 1** Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

- Step 2** Enter the following at the `cli>` prompt:

```
cli> set collection <collection number> filesize <size>
```

where

<collection number> is the number of a collection for which you are specifying the collection file size.

<size> is the size of the collection file in kilobytes. The default is 100 KB. The actual file size is 100 x 1024 bytes.

Step 3 To display the size of the collection, enter the following from the `cli>` prompt:

```
cli> show collection <collection number>
```

Check the information in the Collection File Size.

Starting a Collection

Starting the collector causes the LS2020 switch to collect the specified MIB objects (at the defined times) and store the data in the collection file. To start a collection, follow these steps:

Step 1 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

Step 2 Enter the following at the `cli>` prompt:

```
cli> set collection <collection number> start
```

where

<collection number> is the number of a collection that you are starting.

Step 3 To confirm that the collection has been started, enter the following from the `cli>` prompt:

```
cli> show collection <collection number>
```

Look at the Collection Status field.

If the Collection Status field status is Valid, the collection has been started and is running. If the status is Waiting, you have more than 25 collections running on this switch, or the collection is scheduled to start at a future time.

Specifying Collection Start and End Times

This optional procedure enables you to specify a start and end time for a collection.

Specifying Start Time

Step 1 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

Step 2 To specify the start time, enter the following at the `cli>` prompt:

```
cli> set collection <collection number> begintime <time>
```

where

<collection number> is the number of a collection for which you are specifying the start time.

<time> is the time when you want the collection to start. Enter the time in [[yy:]mm:]dd:]hh:mm:ss format. The default is the current time and date.

Step 3 To determine collection start time, enter the following from the `cli>` prompt:

```
cli> show collection <collection number>
```

Look at the Start Time field.

Specifying End Time

Step 1 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

Step 2 To specify the stop time, enter the following from the `cli>` prompt:

```
cli> set collection <collection number> endtime <time>
```

where

<collection number> is the number of a collection for which you are specifying the end time.

<time> is the time when you want the collection to stop. Enter the time in [[yy:]mm:]dd:]hh:mm:ss format. The default is the time when the LS2020 system clock runs out (for example, Thu Dec 31 23:59:59 2037).

Viewing a Collection

You can view the collection file from the command line interface (CLI), or you can copy it to a third-party network management system (NMS) or workstation, where you can view it or process it.

Two procedures can be used to view the collection from CLI. The first is used if you are running CLI on the network processor (NP) of the LS2020 switch on which the collection is being run, and the second is used if you are running CLI on a remote NP or Sun SPARCstation.

Note You must be in protected mode to view a collection if CLI is running from a remote NP or Sun SPARCstation, or if you are moving the collection file to a third-party NMS or workstation. See the chapter entitled “Command Line Interface” for instructions on accessing protected mode. You must also have an account (user name and password) for the workstation or host to which you are moving the collection file. Your network administrator should provide you with this information.

Procedure 1: Viewing a Collection

Enter the following at the ***cli>** prompt:

```
*cli> shell "cbufpr /usr/tmp/collector/collect.<collection#>"
```

For more information on the circular buffer print (**cbufpr**) command, see the chapter entitled "Command Line Interface."

Procedure 2: Copying a Collection File for Viewing or Processing

Step 1 Enter the following at the **cli>** prompt:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

Enter password:

Step 3 Enter the following at the ***cli>** prompt:

```
*cli> shell "ftp <name or address of workstation or host to which you want to move the collection file>"
```

The LS2020 switch responds with a message similar to the following:

```
Connection to 127.1.41.22.  
220 NMS1 FTP server (version 4.162 Tue Nov 1 10:50:37 PST 1988) ready.
```

Step 4 When you see the following prompt, enter the user name for the account on the workstation or host to which you are moving the collection file:

Name (127.1.41.22:<user name>):

Step 5 Enter the password for the account when you see the following prompt:

Password:

If you enter the user name and password correctly, the workstation or host displays information similar to the following:

```
230 User <user name> logged in.
```

```
ftp>
```

Step 6 Enter the following at the **ftp>** prompt:

```
ftp> put /usr/tmp/mma/collectIndex.8 [<new name>]
```

where

<new name> is the name of the file that identifies the chassis or appropriate directory name for the file. For example, if you are moving a trap log for a switch called Light5, the new name could be `mma_Light5.traplog`.

The **put** command sends the log file to the specified workstation or host. The system tells you when the transfer is complete.

You can use a standard UNIX command such as **more** or **cat** to view the collection file on the workstation or host. You can also use the text file with any text processing tool you have available.

Step 7 To exit the file transfer program and return to CLI, enter **bye** or **quit** at the **ftp>** prompt:

Listing Defined Collections

To get a list of all defined collections and to determine the status and contents of any defined collection, follow these steps:

Step 1 Enter the following at the `cli>` prompt:

```
cli> walksnmp collectIndex
```

The `walksnmp` command displays a list of all collections that have been defined, as follows:

```
cli> walk collectIndex
Name: collectIndex.1      Value: 1
Name: collectIndex.2      Value: 2
Name: collectIndex.3      Value: 3
.
.
.
cli>
```

The numbers following the term `Value` are the collection numbers that have been defined.

Step 2 Once you know the numbers of the collections that have been defined, you can determine the contents and status of each collection by typing the following at the `cli>` prompt:

```
cli> show collection [<collection number>]
```

If you do not enter a collection number, all collections are displayed.

The following is an example of the output that displays when you enter **show collection 10**:

```
cli> show collection 10
Collection Record 10:
Collection Status:      Under Creation
Begin Time:             Mon Aug 2 15:43 1995
Ending Time:            Thu Dec 31 23:59:59 2037
Collection Interval:    30 sec
Collection File:         /usr/tmp/collector/collect.10
Collection File Size:    100 KB

Collection Items:

Name: collectDBObjectID.10.1  Value: ifInOctets.3000
Name: collectDBObjectID.10.2  Value: ifInOctets.3001
Name: collectDBObjectID.10.2  Value: ifInOctets.3005
cli>
```

Stopping a Collection

To temporarily stop a collection, follow these steps:

Step 1 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

`<community name>` is the name for the SNMP community with read/write privileges that you want to access.

Step 2 Enter the following at the `cli>` prompt:

```
cli> set collection <collection number> halt
```

where

<collection number> is the number of a collection you are stopping.

Step 3 To determine if a collection has been stopped, enter the following from the `cli>` prompt:

```
cli> show collection <collection number>
```

If you restart this collection at a later time, new data is appended to the existing collection file.

The **show collection <collection number>** command displays the Collection Status field. If the status is Under Creation, the collection has been stopped.

If a collection has been restarted, the Collection Status changes from Under Creation to Valid.

If you have 25 collections running and you stop one or more of them, collections with a Collection Status of Waiting may be started as a result. (In this case, you may not be able to restart the collection you just stopped, unless you halt another collection.)

Deleting a Collection

The following steps tell you how to delete the definition of a collection. The procedure deletes the definition of the collection from the `collectTable`, but it does not delete the collection file itself. Once a collection is deleted, you can no longer restart it because its definition no longer exists.

Step 1 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> is the name for the SNMP community with read/write privileges that you want to access.

Step 2 Enter the following at the `cli>` prompt:

```
cli> set collection <collection number> del
```

where

<collection number> is the number of a collection to be deleted.

Step 3 To list the collection numbers that have been defined for this switch, enter the following from the `cli>` prompt:

```
cli> walksnmp collectIndex
```

The **walksnmp collectIndex** command displays the collection numbers that have been defined for this switch. The number that you deleted should not appear on this list.

Sample Collection Scenario

The following shows a list of commands used to create, define, and then start a collection. This sample collection is defined to collect the `ifInErrors` MIB object for two ports on card 1 and two ports on card 3 of the target LS2020 switch. It is placed in `/usr/tmp/collector/collect.2`. The data is collected every hour starting when you issue the **set collection start** command and continuing until you stop it. (No end time has been defined.)

```
cli> set collection 2 create

cli> set collection 2 frequency 3600

cli> set collection 2 addvar ifInErrors.1000

cli> set collection 2 addvar ifInErrors.1001

cli> set collection 2 addvar ifInErrors.3000

cli> set collection 2 addvar ifInErrors.3001

cli> set collection 2 start
```

You can view the collection file by following the instructions in the section “Procedure 1: Viewing a Collection.” Figure 7-2 shows a sample display of the collection file.

Figure 7-2 **Sample Display of a Collection File**

```
06:14:23 ifInErrors.1000      0
06:14:23 ifInErrors.1001      0
06:14:23 ifInErrors.3000      0
06:14:23 ifInErrors.3001      0
07:14:23 ifInErrors.1000      0
07:14:23 ifInErrors.1001      3276418
07:14:23 ifInErrors.3000      0
07:14:23 ifInErrors.3001      631645383
08:14:23 ifInErrors.1000      2
08:14:23 ifInErrors.1001      0
08:14:23 ifInErrors.3000      94
08:14:23 ifInErrors.3001      0
...
```

H3382

