TN3270E Working Group
INTERNET DRAFT: <draft-ietf-tn3270e-rt-mib-07.txt>
Expiration Date: January, 1999

Kenneth White
IBM Corp.
Robert Moore
IBM Corp.

July 1998

Definitions of Protocol and Managed Objects for TN3270E Response Time Collection Using SMIv2 (TN3270E-RT-MIB)

<draft-ietf-tn3270e-rt-mib-07.txt>

Status of this Memo

Copyright Notice

Abstract

This memo defines the protocol and the Management Information Base (MIB) for performing response time data collection on TN3270 and TN3270E sessions by a TN3270E server. The response time data collected by a TN3270E server is structured to support both validation of service level agreements and performance monitoring of TN3270 and TN3270E Sessions. This MIB has as a prerequisite the TN3270E-MIB, reference [20].

TN3270E, defined by RFC 2355 [19], refers to the enhancements made to the Telnet 3270 (TN3270) terminal emulation practices. Refer to RFC 1041 [18], RFC 854 [16], and RFC 860 [17] for a sample of what is meant by TN3270 practices.

Table of Contents

1.0 Introduction

This document is a product of the TN3270E Working Group. It defines a protocol and a MIB module to enable a TN3270E server to collect and keep track of response time data for both TN3270 and TN3270E clients. Basis for implementing this MIB:

- TN3270E-MIB, Base Definitions of Managed Objects for TN3270E Using SMIv2 [20]

- TN3270E RFCs

- Telnet Timing Mark Option RFC [17].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, reference [23].

2.0 The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2271 [1].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [2], RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in RFC 1902 [5], RFC 1903 [6] and RFC 1904 [7].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2272 [11] and RFC 2274 [12].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- A set of fundamental applications described in RFC 2273 [14] and the view-based access control mechanism described in RFC 2275 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3.0 Response Time Collection Methodology

This section explains the methodology and approach used by the MIB defined by this memo for response time data collection by a TN3270E server.

3.1 General Response Time Collection

Two primary methods exist for measuring response times in SNA networks:

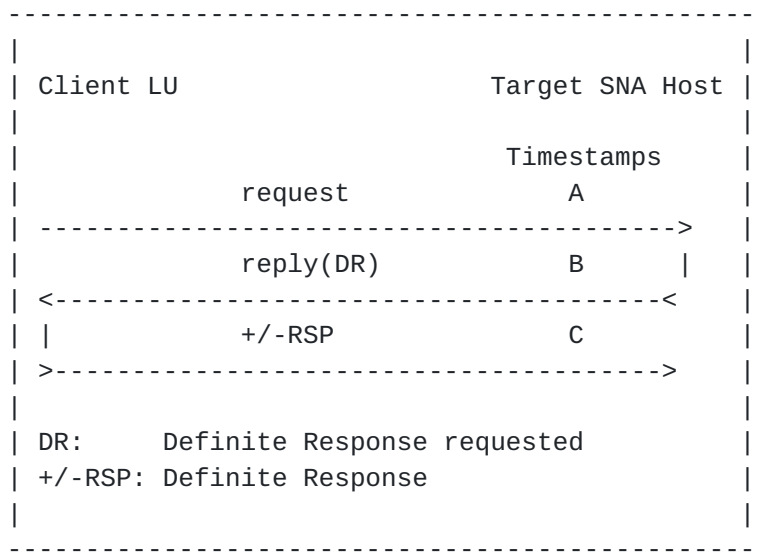
o The Systems Network Architecture Management Services (SNA/MS) Response Time Monitoring (RTM) function.

o Timestamping using definite response flows.

This memo defines an approach using definite responses to timestamp the flows between a client and its TN3270E server, rather than by use of the RTM method. Extensions to the SNA/MS RTM flow were considered, but this approach was deemed unsuitable since not all TN3270E server implementations have access to their underlying SNA stacks. The RTM concepts of keeping response time buckets for service level agreements and of interval-based response time collection for performance monitoring are preserved in the MIB module defined in this memo.

As mentioned, this memo focuses on using definite responses to timestamp the flows between a client and its TN3270E server for generating performance data. Use of a definite response flow requires that the client supports TN3270E with the RESPONSES function negotiated. The TN3270 TIMING-MARK option can be used instead of definite response for supporting TN3270 clients or TN3270E clients that don't support RESPONSES. This document focuses first on defining the protocol and methods for generating performance data using definite responses, and then describes how the TIMING-MARK option can be used instead of definite response.

In an SNA network, a transaction between a client Logical Unit (LU) and a target host in general looks as follows:



This transaction is a simple one, and is being used only to illustrate how timestamping at a target SNA host can be used to generate response times. An IBM redbook [12] provides a more detailed description of

response time collection for a transaction of this type. Note that for the purpose of calculating an approximation for network transit time, it doesn't matter if the response is positive or negative. Two response time values are typically calculated:

o Host Transit Time: Timestamp B - Timestamp A

o Network Transit Time: Timestamp C - Timestamp B

Network transit time is an approximation for the amount of time that a transaction requires to flow across a network, since the response flow is being substituted for the request flow at the start of the transaction. Network transit time, timestamp C - timestamp B, is the amount of time that the definite response request and its response required. Host time, timestamp B - timestamp A, is the actual time that the host required to process the transaction. Experience has shown that using the response flow to approximate network transit times is useful, and does correlate well with actual network transit times.

A client SHOULD respond to a definite response request when it completes processing the transaction. This is important since it increases the accuracy of a total response time. Clients that immediately respond to a definite response request will be attributed with lower total response times then those that actually occurred.

The TN3270E-RT-MIB describes a method of collecting performance data that is not appropriate for printer (LU Type 1 or LU Type 3) sessions; thus collection of performance data for printer sessions is excluded from this MIB. This exclusion of printer sessions is not considered a problem, since these sessions are not the most important ones for response time monitoring, and since historically they were excluded from SNA/MS RTM collection. The tn3270eTcpConnResourceType object in a tn3270eTcpConnEntry (in the TN3270E-MIB) can be examined to determine if a client session is ineligible for response time data collection for this reason.

3.2 TN3270E Server Response Time Collection

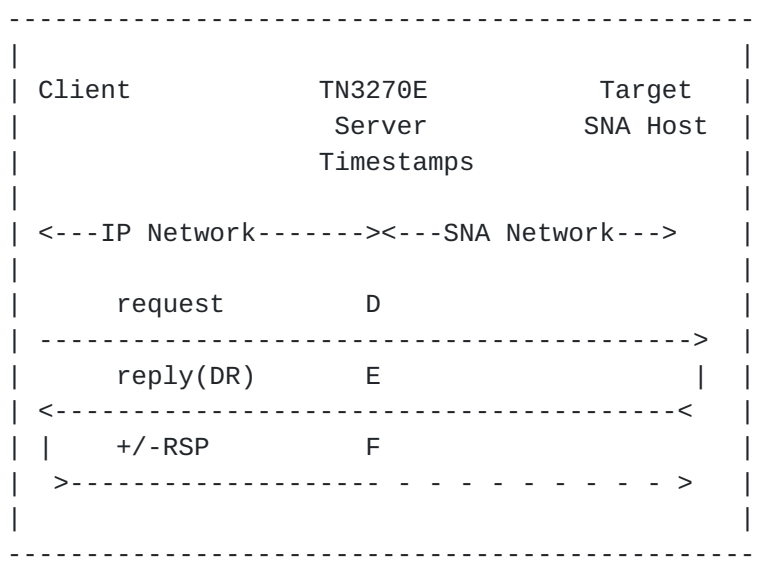
A TN3270E server connects a Telnet client performing 3270 emulation to a target SNA host over both a client-side network (client to TN3270E server) and an SNA Network (TN3270E server to target SNA host). The client-side network is typically TCP/IP, but it need not be. For ease of exposition this document uses the term "IP network" to refer to the client-side network, since IP is by far the most common protocol for these networks.

A TN3270E server can use SNA definite responses and the TN3270 Enhancement (RFC 2355 [19]) RESPONSES function to calculate response times for a transaction, by timestamping when a client request arrives at the server, when the reply arrives from the target host, and when the response acknowledging this reply arrives from the client.

Section 3.4, Timestamp Calculation, provides specifics on when in the sequence of flows between a TN3270E client and its target SNA host a TN3270E server takes the required timestamps. In addition, it provides information on how a TN3270 TIMING-MARK request/response flow can be

used instead of DR for approximating IP network transit times.

The following figure adds a TN3270E server between the client, in this case a TN3270E client and the target SNA host:



A TN3270E server can save timestamp D when it receives a client request, save timestamp E when the target SNA host replies, and save timestamp F when the client responds to the definite response request that flowed with the reply. It doesn't matter whether the target SNA host requested a definite response on its reply: if it didn't, the TN3270E server makes the request on its own, to enable it to produce timestamp F. In this case the TN3270E server does not forward the response to the target SNA host, as the dotted line in the figure indicates.

Because it is a special case, a transaction in which a target SNA host returns an UNBIND in response to a client's request, and the TN3270E server forwards the UNBIND to the client, is not included in any response time calculations.

In order to generate timestamp F, a TN3270E server MUST insure that the transaction specifies DR, and that the TN3270E RESPONSES function has been negotiated between itself and the client. Negotiation of the TN3270E RESPONSES function occurs during the client's TN3270E session initialization. The TN3270E servers that the authors are aware of do request the RESPONSES function during client session initialization. TN3270E clients either automatically support the RESPONSES function, or can be configured during startup to support it.

Using timestamps D, E, and F the following response times can be calculated by a TN3270E server:

o Total Response time: Timestamp F - Timestamp D
o IP Network Transit Time: Timestamp F - Timestamp E

Just as in the SNA case presented above, these response times are also approximations, since the final +/- RSP from the client is being

substituted for the request from the client that began the transaction.

The MIB provides an object, tn3270eRtCollCtlType, to control several aspects of response time data collection. One of the available options in setting up a response time collection policy is to eliminate the IP-network component altogether. This might be done because it is

determined either that the additional IP network traffic would not be desirable, or that the IP-network component of the overall response times is not significant.

Excluding the IP-network component from response times also has an implication for the way in which response time data is aggregated. A TN3270E server may find that some of its clients simply don't support any of the functions necessary for the server to calculate the IP-network component of response times. For these clients, the most that the server can calculate is the SNA-network component of their overall response times; the server records this SNA-network component as the TOTAL response time each of these clients' transactions. If a response time collection is aggregating data from a number of clients, some of which have the support necessary for including the IP-network component in their total response time calculations, and some of which do not, then the server aggregates the data differently depending on whether the collection has been defined to include or exclude the IP-network component:

- If the IP-network component is included, then transactions for the clients that don't support calculation of the IP-network component of their response times are excluded from the aggregation altogether.

- If the IP-network component is excluded, then total response times for ALL clients include only the SNA-network component, even though the server could have included an IP-network component in the overall response times for some of these clients. The server does this by setting timestamp F, which marks the end of a transaction's total response time, equal to timestamp E, the end of the transaction's SNA-network component.

The principle here is that all the transactions contributing their response times to an aggregated value MUST make the same contribution. If the aggregation specifies that an IP-network component MUST be included in the aggregation's response times, then transactions for which an IP-network component cannot be calculated aren't included at all. If the aggregation specifies that an IP-network component is not to be included, then only the SNA-network component is used, even for those transactions for which an IP-network component could have been calculated.

There is one more complication here: the MIB allows a management application to enable or disable dynamic definite responses for a response time collection. Once again the purpose of this option is to give the network operator control over the amount of traffic introduced into the IP network for response time data collection. A DYNAMIC definite response is one that the TN3270E server itself adds to a reply, in a transaction for which the SNA application at the target SNA host

did not specify DR in its reply. When the +/-RSP comes back from the client, the server uses this response to calculate timestamp F, but then it does not forward the response on to the SNA application (since the application is not expecting a response to its reply).

The dynamic definite responses option is related to the option of including or excluding the IP-network component of response times (discussed above) as follows:

- If the IP-network component is excluded, then there is no reason for enabling dynamic definite responses: the server always sets timestamp F equal to timestamp E, so the additional IP-network traffic elicited by a dynamic definite response would serve no purpose.

- If the IP-network component is included, then enabling dynamic definite responses causes MORE transactions to be included in the aggregated response time values:

  - For clients that do not support sending of responses, timestamp F can never be calculated, and so their transactions are never included in the aggregate.

  - For clients that support sending of responses, timestamp F will always be calculated for transactions in which the host SNA application specifies DR in its reply, and so these transactions will always be included in the aggregate.

  - For clients that support sending of responses, having dynamic definite responses enabled for a collection results in the inclusion of additional transactions in the aggregate: specifically, those for which the host SNA application did not specify DR in its reply.

A TN3270E server also has the option of substituting TIMING-MARK processing for definite responses in calculating the IP-network component of a transaction's response time. Once again, there is no reason for the server to do this if the collection has been set up to exclude the IP-network component altogether in computing response times.

The MIB is structured to keep counts and averages for total response times (F - D) and their IP-network components (F - E). A management application can obviously calculate from these two values an average SNA-network component (E - D) for the response times. This SNA-network component includes the SNA node processing time at both the TN3270E server and at the target application.

A host TN3270E server refers to an implementation where the TN3270E server is collocated with the Systems Network Architecture (SNA) System Services Control Point (SSCP) for the dependent Secondary Logical Units (SLUs) that the server makes available to its clients for connecting into an SNA network. A gateway TN3270E server resides on an SNA node other than an SSCP, either an SNA type 2.0 node, a boundary-function-attached type 2.1 node, or an APPN node acting in the

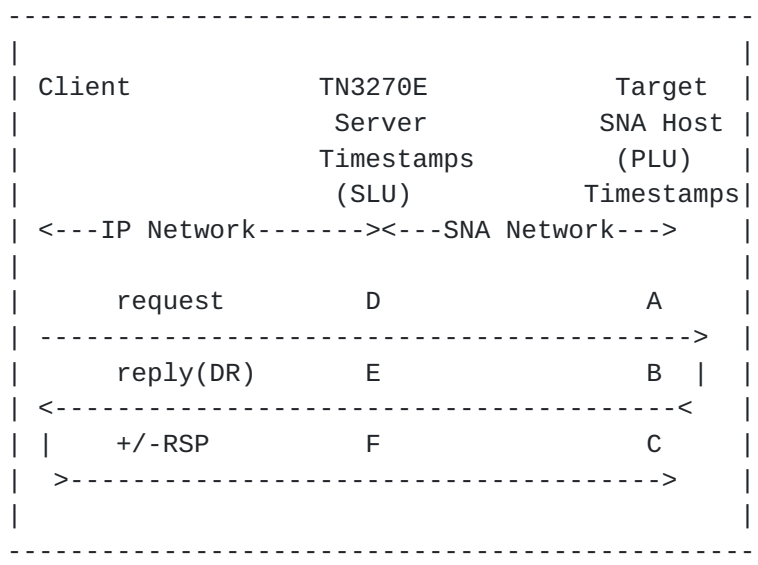
role of a Dependent LU Requester (DLUR). Host and gateway TN3270E server implementations typically differ greatly as to their internal implementation and System Definition (SYSDEF) requirements.

If a host TN3270E server is in the same SNA host as the target application, then the SNA-network component of a transaction's response

time will approximately equal the host transit time (B - A) described previously. A host TN3270E server implementation can, however, typically support the establishment of sessions to target applications in SNA hosts remote from itself. In this case the SNA-network component of the response time equals the actual SNA-network transit time plus two host transit times.

3.3 Correlating TN3270E Server and Host Response Times

It is possible that response time data is collected from TN3270E servers at the same time as a management application is monitoring the SNA sessions at a host. For example, a management application can be monitoring a secondary logical unit (SLU) while retrieving data from a TN3270E server. Consider the following figure:



The following response times are available:

- Target SNA host transit time: Timestamp B - Timestamp A
- Target SNA host network transit time: Timestamp C - Timestamp B
- TN3270E server total response time: Timestamp F - Timestamp D
- TN3270E server IP-network component: Timestamp F - Timestamp E

The value added by the TN3270E server in this situation is its approximation of the IP-network component of the overall response time. The IP-network component can be subtracted from the total network transit time (which can be captured at an SSCP monitoring SNA traffic from/to the SLU) to see the actual SNA versus IP network transit times.

The MIB defined by this memo does not specifically address correlation of the data it contains with response time data collected by direct monitoring of SNA resources: its focus is exclusively response time

data collection from a TN3270E server perspective. It has, however, in conjunction with the TN3270E-MIB [10], been structured to provide the information necessary for correlation between TN3270E server-provided response time information and that gathered from directly monitoring SNA resources.

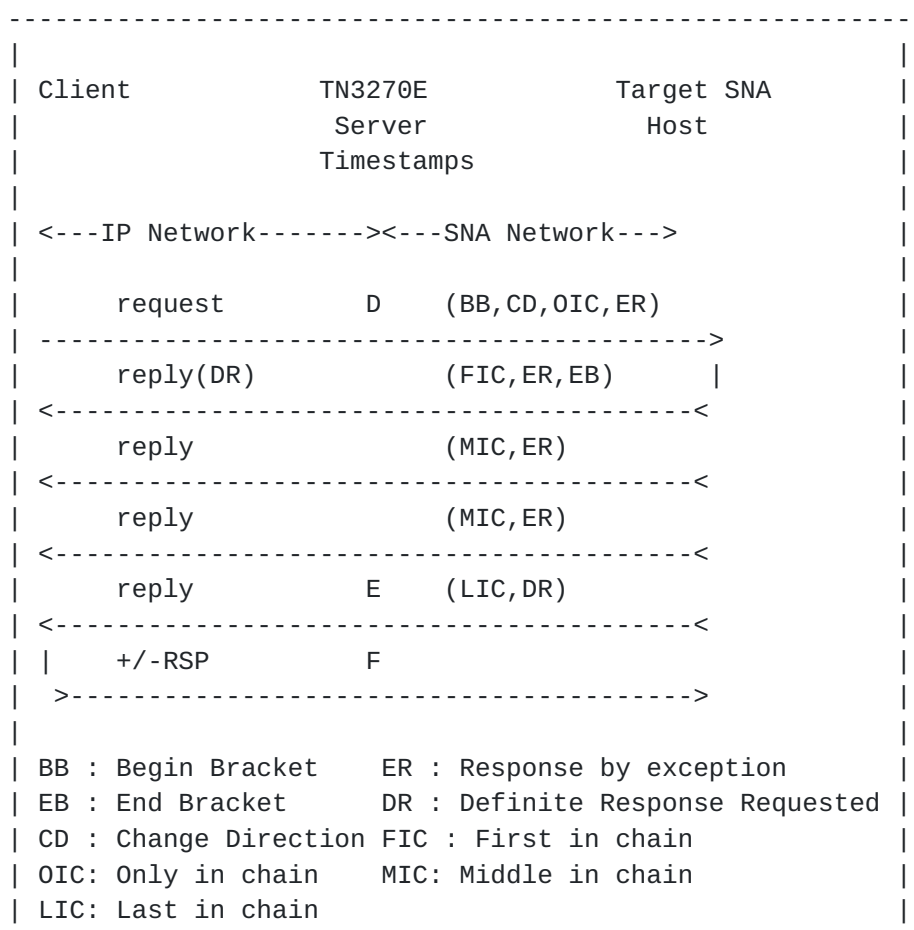
A management application attempting to correlate SNA resource usage to Telnet clients can monitor either the tn3270eResMapTable or the tn3270eTcpConnTable to determine resource-to-client address mappings. Both of these tables are defined by the TN3270E-MIB [10]. Another helpful table is the tn3270eSnaMapTable, which provides a mapping between SLU names as they are known at the SSCP (VTAM) and their local names at the TN3270E server. Neither the tn3270eClientGroupTable, the tn3270eResPoolTable, nor the tn3270eClientResMapTable from the TN3270E-MIB can be used for correlation, since the mappings defined by these tables can overlap, and may not provide one-to-one mappings.

3.4 Timestamp Calculation

This section goes into more detail concerning when the various timestamps can be taken as the flows between a TN3270E client and its target SNA host pass through a TN3270E server. In addition, information is provided on how the TN3270 TIMING-MARK request/response flow can be used in place of DR for approximating IP network transit times.

3.4.1 DR Usage

Consider the following flow:



Timestamp D is taken at the TN3270E server when the server has received data from a client for forwarding to its target SNA host, and the direction of the SNA session allows the server to forward the data immediately (either the direction is inbound towards the SNA host, or

the session is between brackets). This is most likely when the server finds the end of record indicator in the TCP data received from the client.

The target SNA application returns its reply in one or more SNA Request Units (RUs); in this example there are four RUs in the reply. The first RU is marked as first in chain (FIC), the next two are marked as middle in chain (MIC), and the last is marked as last in chain (LIC). If the SNA host sends a multiple-RU chain, the server does not know until the last RU is received whether DR is being requested. The server's only chance to request DR from the client, however, comes when it forwards the FIC RU, since this is the only time that the TN3270E header is included. Since a server may forward the FIC RU to the client before it receives the LIC RU from the SNA host, some servers routinely specify DR on all FIC RUs.

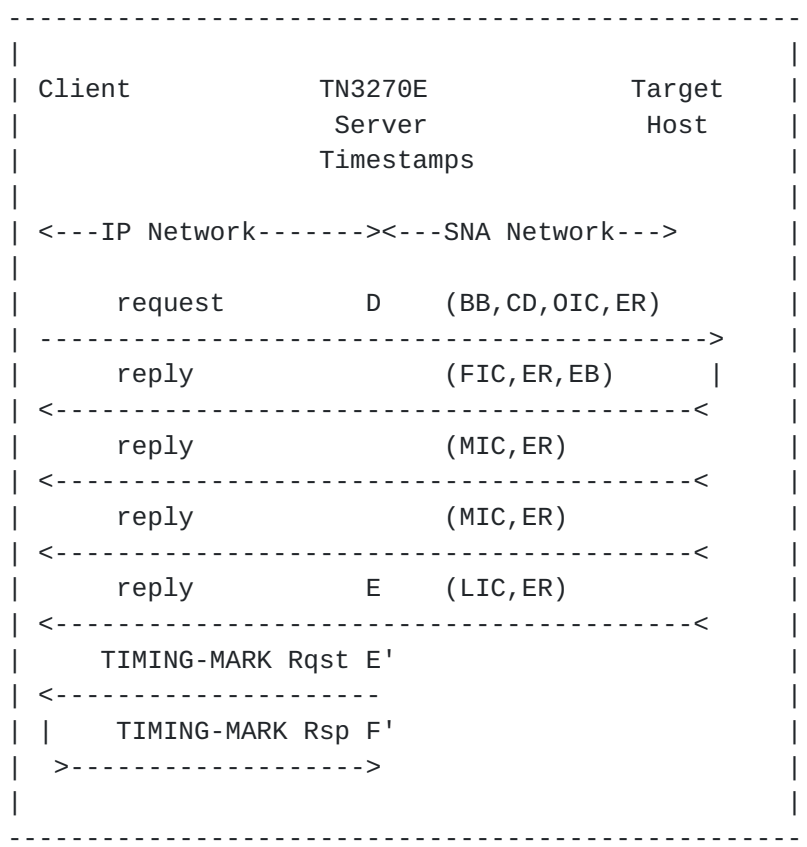
If the server has specified DR on the TN3270E request for the FIC RU in a chain, it takes timestamp E when it forwards the LIC RU to the client. Since timestamp E is used for calculating the IP-network time for the transaction, the server SHOULD take timestamp E as close as possible to its "Telnet edge". The server takes timestamp F when it receives the RESPONSES response from the client.

A target SNA application doesn't necessarily return data to a client in a transaction; it may, for example, require more data from the client before it can formulate a reply. In this case the application may simply return to the TN3270E server a change of direction indicator. At this point the server must send something to the client (typically a Write operation with a WCC) to unlock the keyboard. If the server specifies DR on the request to the client triggered by its receipt of the change of direction indicator from the SNA application, then timestamps E and F can be taken, and the usual response times can be calculated. When the client sends in the additional data and gets a textual response from the SNA application, the server treats this as a separate transaction from the one involving the change of direction.

3.4.2 TIMING-MARK Usage

It is possible for a TN3270E server to use the TIMING-MARK flow for approximating IP network transit times. Using TIMING-MARKs would make it possible for a server to collect performance data for TN3270 clients, as well as for TN3270E clients that do not support the RESPONSES function. In order for TIMING-MARKs to be used in this way, a client can't have the NOP option enabled, since responses are needed to the server's TIMING-MARK requests. An IP network transit time approximation using a TIMING-MARK is basically the amount of time it takes for a TN3270 server to receive from a client a response to a TIMING-MARK request.

To get an estimate for IP network transit time, a TN3270E server sends a TIMING-MARK request to a client after a LIC RU has been received, as a means of approximating IP network transit time:



The response times can then be calculated as follows:

o TN3270E server total response time:
(Timestamp E - Timestamp D) + (Timestamp F' - Timestamp E')

o TN3270E server IP network time: Timestamp F' - Timestamp E'

If a TN3270E server is performing the TIMING-MARK function (independent of the response time monitoring use of the function discussed here), then it most likely has a TIMING-MARK interval for determining when to examine client sessions for sending the TIMING-MARK request. This interval, which is ordinarily a global value for an entire TN3270E server, is represented in the TN3270E-MIB by the tn3270eSrvrConfTmNopInterval object. A TIMING-MARK request is sent only if, when it is examined, a client session is found to have had no activity for a different fixed length of time, represented in the TN3270E-MIB by the tn3270eSrvrConfTmNopInactTime object.

Servers that support a large number of client sessions should spread out the TIMING-MARK requests they send to these clients over the activity interval, rather than sending them all in a single burst, since otherwise the network may be flooded with TIMING-MARK requests. When a server uses TIMING-MARKs for approximating response times, this tends to introduce a natural spreading into its TIMING-MARK requests, since the

requests are triggered by the arrival of traffic from an SNA host.

A TN3270E server MUST integrate its normal TIMING-MARK processing with its use of TIMING-MARKs for computing response times. In particular, it MUST NOT send a second TIMING-MARK request to a client while waiting for the first to return, since this is ruled out by the TIMING-MARK protocol

itself. If a TIMING-MARK flow has just been performed for a client shortly before the LIC RU arrives, the server MAY use the interval from this flow as its approximation for IP network transit time, (in other words, as its (F' - E') value) when calculating its approximation for the transaction's total response time, rather than sending a second TIMING-MARK request so soon after the preceding one.

Regardless of when the server sends its TIMING-MARK request, the accuracy of its total response time calculation depends on exactly when the client responds to the TIMING-MARK request.

3.5 Performance Data Modelling

The following two subsections detail how the TN3270E-RT-MIB models and controls capture of two types of response time data: average response times and response time buckets.

3.5.1 Averaging Response Times

Average response times play two different roles in the MIB:

- They are made available for management applications to retrieve.
- They serve as triggers for emitting notifications.

Sliding-window averages are used rather than straight interval-based averages, because they are often more meaningful, and because they cause less notification thrashing. Sliding-window average calculation can, if necessary, be disabled, by setting the sample period multiplier, tn3270eRtCollCtlSPMult, to 1, and setting the sample period, tn3270eRtCollCtlSPeriod, to the required collection interval.

In order to calculate sliding-window averages, a TN3270E server MUST:

- Select a fixed, relatively short, sample period SPeriod; the default value for SPeriod in the MIB is 20 seconds.

- Select an averaging period multiplier SPMult. The actual collection interval will then be SPMult times SPeriod. The default value for SPMult in the MIB is 30, yielding a default collection interval of 10 minutes. Note that the collection interval (SPMult*SPeriod) is always a multiple of the sample period.

  Clearly, SPMult*SPeriod should not be thought of as literally the averaging period. The average calculated will include contributions older than that time, and does not weight equally all contributions since that time. In fact, it gives a smoother result than a traditional sliding average, as used in finance. More subtly, it is best to think of the effective averaging period as being 2*SPMult*SPeriod. To see this, consider how long the contribution to the result made by a particular transaction lasts. With a

traditional sliding average, it lasts exactly the averaging period. With the aging mechanism described here, it has a half-life of SPMult*SPeriod.

o Maintain the following counters to keep track of activity within the current sample period; these are internal counters, not made visible to a management application via the MIB.

  - T (number of transactions in the period)

  - TotalRts (sum of the total response times for all transactions in the period)

  - TotalIpRts (sum of the IP network transit times for all transactions in the period; note that if IP network transit times are being excluded from the response time collection, this value will always be 0).

o Also maintain sliding counters, initialized to zero, for each of the quantities being counted:

  - AvgCountTrans (sliding count of transactions)
  - TotalRtsSliding (sliding count of total response times)
  - TotalIpRtsSliding (sliding count of IP network transit times)

o At the end of each sample period, update the sliding interval counters, using the following floating-point calculations:

```
AvgCountTrans = AvgCountTrans + T
     - (AvgCountTrans / SPMult)

TotalRtsSliding = TotalRtsSliding + TotalRts
     - (TotalRtsSliding / SPMult)

TotalIpRtsSliding = TotalIpRtsSliding + TotalIpRts
     - (TotalIpRtsSliding / SPMult)
```

  Then reset T, TotalRts, and TotalIpRts to zero for use during the next sample period.

o At the end of a collection interval, update the following MIB objects as indicated; the floating-point numbers are rounded rather than truncated.

```
tn3270eRtDataAvgCountTrans = AvgCountTrans
tn3270eRtDataAvgRt = TotalRtsSliding / AvgCountTrans
tn3270eRtDataAvgIpRt = TotalIpRtsSliding / AvgCountTrans
```

  As expected, if IP network transit times are being excluded from response time collection, then tn3270eRtDataAvgIpRt will always return 0.

The sliding transaction counter AvgCountTrans is not used for updating the MIB object tn3270eRtDataCountTrans: this object is an ordinary SMI

Counter32, which maintains a total count of transactions since its last discontinuity event. The sliding counters are used only for calculating averages.

Two mechanisms are present in the MIB to inhibit the generation of an excessive number of notifications related to average response times. First, there are high and low thresholds for average response times. A tn3270eRtExceeded notification is generated the first time a statistically significant average response time is found to have exceeded the high threshold. (The test for statistical significance is described below.) After this, no other tn3270eRtExceeded notifications are generated until an average response time is found to have fallen below the low threshold.

The other mechanism to limit notifications is the significance test for a high average response time. Intuitively, the significance of an average is directly related to the number of samples that go into it; so we might be inclined to use a rule such as "for the purpose of generating tn3270eRtExceeded notifications, ignore average response times based on fewer than 20 transactions in the sample period."

In the case of response times, however, the number of transactions sampled in a fixed sampling period is tied to these transactions' response times. A few transactions with long response times can guarantee that there will not be many transactions in a sample, because these transactions "use up" the sampling time. Yet this case of a few transactions with very poor response times should obviously be classified as a problem, not as a statistical anomaly based on too small a sample.

The solution is to make the significance level for a sample a function of the average response time. A value IdleCount is specified, which is used to qualify an sample as statistically significant. In order to determine at a collection interval whether to generate a tn3270eRtExceeded notification, a TN3270E server uses the following algorithm:

```
if AvgCountTrans * ((AvgRt/ThreshHigh - 1) ** 2) >=  IdleCount
then generate the notification,
```

where AvgRt is the value that would be returned by the object tn3270eRtDataAvgRt at the end of the interval, and the "**" notation indicates exponientiation.

Two examples illustrate how this algorithm works. Suppose that IdleCount has been set to 20 transactions, and the high threshold to 200 msecs per transaction. If the average observed response time is 300 msecs, then a notification will be generated only if AvgCountTrans >= **80**. **If, however, the observed response time is 500 msecs, then a** notification is generated if AvgCountTrans >= 9.

There is no corresponding significance test for the tn3270eRtOkay notification: this notification is generated based on an average

response time that falls below the low threshold, regardless of the sample size behind that average.

3.5.2 Response Time Buckets

The MIB also supports collection of response time data into a set of five buckets. This data is suitable either for verification of service level agreements, or for monitoring by a management application to identify performance problems. The buckets provide counts of transactions whose total response times fall into a set of specified ranges.

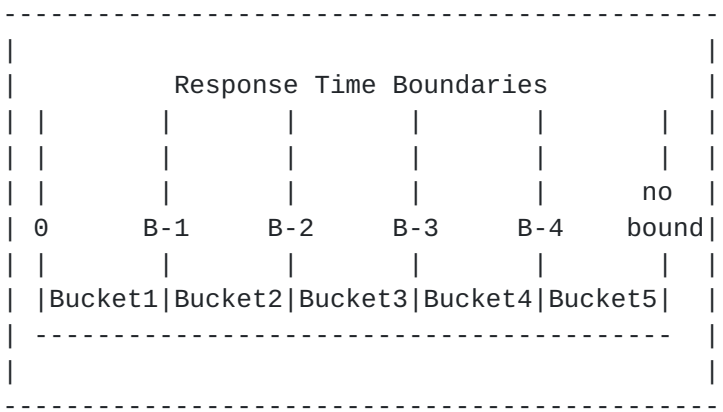
Like everything for a collection, the "total" response times collected in the buckets are governed by the specification of whether IP network transit times are to be included in the totals. Depending on how this option is specified, the response times being counted in the buckets will either be total response times (F - D), or only SNA network transit times (effectively E - D, because when it is excluding the IP-network component of transactions, a server makes timestamp F identical to timestamp E).

Four bucket boundaries are specified for a response time collection, resulting in five buckets. The first response time bucket counts those transactions whose total response times were less than or equal to Boundary 1, the second bucket counts those whose response times were greater than Boundary 1 but less than or equal to Boundary 2, and so on. The fifth bucket is unbounded on the top, counting all transactions whose response times were greater than Boundary 4.

The four bucket boundaries have default values of: 1 second, 2 seconds, **5 seconds, and 10 seconds, respectively.** These values are the defaults in the 3174 controller's implementation of the SNA/MS RTM function, and are thought to be appropriate for this MIB as well.

In SNA/MS the counter buckets were (by today's standards) relatively small, with a maximum value of 65,535. The bucket objects in the MIB are all Counter32's.

The following figure represents the buckets pictorially:



4.0 Structure of the MIB

The TN3270E-RT-MIB has the following components:

- tn3270eRtCollCtlTable
- tn3270eRtDataTable

o Notifications
o Advisory Spin Lock Usage

4.1 tn3270eRtCollCtlTable

The tn3270eRtCollCtlTable is indexed by tn3270eSrvrConfIndex and tn3270eClientGroupName imported from the TN3270E-MIB. tn3270eSrvrConfIndex identifies within a host a particular TN3270E server. tn3270eClientGroupName identifies a collection of IP clients for which response time data is to be collected. The set of clients is defined using the tn3270eClientGroupTable from the TN3270E-MIB.

A tn3270eRtCollCtlEntry contains the following objects:

1st Index	tn3270eSrvrConfIndex	Unsigned32
2nd Index	tn3270eClientGroupName	Utf8String
	tn3270eRtCollCtlType	BITS
	tn3270eRtCollCtlSPeriod	Unsigned32
	tn3270eRtCollCtlSPMult	Unsigned32
	tn3270eRtCollCtlThreshHigh	Unsigned32
	tn3270eRtCollCtlThreshLow	Unsigned32
	tn3270eRtCollCtlIdleCount	Unsigned32
	tn3270eRtCollCtlBucketBndry1	Unsigned32
	tn3270eRtCollCtlBucketBndry2	Unsigned32
	tn3270eRtCollCtlBucketBndry3	Unsigned32
	tn3270eRtCollCtlBucketBndry4	Unsigned32
	tn3270eRtCollCtlRowStatus	RowStatus

The tn3270eRtCollCtlType object controls the type(s) of response time collection that occur, the granularity of the collection, whether dynamic definite responses SHOULD be initiated, and whether notifications SHOULD be generated. This object is of BITS SYNTAX, and thus allows selection of multiple options.

The BITS in the tn3270eRtCollCtlType object have the following meanings:

o aggregate(0) - If this bit is set to 1, then data SHOULD be aggregated for the whole client group. In this case there will be only one row created for the collection in the tn3270eRtDataTable. The first two indexes for this row, tn3270eSrvrConfIndex and tn3270eClientGroupName, will have the same values as the indexes for the corresponding tn3270eRtCollCtlEntry. The third and fourth indexes of an aggregated tn3270eRtDataEntry have the values unknown(0) (tn3270eRtDataClientAddrType) and a zero-length octet string (tn3270eRtDataClientAddress). The fifth index, tn3270eRtDataClientPort, has the value 0.

If this bit is set to 0, then a separate entry is created in the tn3270eRtDataTable for each member of the client group. In this case tn3270eRtDataClientAddress contains the client's actual IP Address, tn3270eRtDataClientAddrType indicates the address type, and tn3270eRtDataClientPort contains the number of the port the client is using for its TN3270/TN3270E session.

- excludeIpComponent(1) - If this bit is set to 1, then the server SHOULD exclude the IP-network component from all the response times for this collection. If the target SNA application specifies DR in any of its replies, this DR will still be passed down to the client, and the client's response will still be forwarded to the application. But this response will play no role in the server's response time calculations.

  If this bit is set to 0, then the server includes in the collection only those transactions for which it can include an (approximate) IP-network component in the total response time for the transaction. This component MAY be derived from a "natural" DR (if the client supports the RESPONSES function), from a dynamic DR introduced by the server (if the client supports the RESPONSES function and the ddr(2) bit has been set to 1), or from TIMING-MARK processing (if the client supports TIMING-MARKs).

  If this bit is set to 1, then the ddr(2) bit is ignored, since there is no reason for the server to request additional responses from the client(s) in the group.

- ddr(2) - If this bit is set to 1, then the server SHOULD, for those clients in the group that support the RESPONSES function, add a DR request to the FIC reply in each transaction, and use the client's subsequent response for calculating an (approximate) IP-network component to include in the transaction's total response times.

  If this bit is set to 0, then the server does not add a DR request that it was not otherwise going to add to any replies from the target SNA application.

  If the excludeIpComponent(1) bit is set to 1, then this bit is ignored by the server.

- average(3) - If this bit is set to 1, then the server SHOULD calculate a sliding-window average for the collection, based on the parameters specified for the group.

  If this bit is set to 0, then an average is not calculated. In this case the tn3270eRtExceeded and tn3270eRtOkay notifications are not generated, even if the traps(5) bit is set to 1.

- buckets(4) - If this bit is set to 1, then the server SHOULD create and increment response time buckets for the collection, based on the parameters specified for the group.

  If this bit is set to 0, then response time buckets are not created.

- traps(5) - If this bit is set to 1, then a TN3270E Server is enabled to generate notifications pertaining to an tn3270eCollCtlEntry.

tn3270CollStart and tn3270CollEnd generation is enabled simply by traps(5) being set to 1. tn3270eRtExceeded and tn3270eRtOkay generation enablement requires that average(3) be set to 1 in addition to the traps(5) requirement.

If traps(5) is set to 0, then none of the notifications defined in this MIB are generated for a particular tn3270eRtCollCtlEntry.

Either the average(3) or the buckets(4) bit MUST be set to 1 in order for response time data collection to occur; both bits MAY be set to 1. If the average(3) bit is set to 1, then the following objects have meaning, and are used to control the calculation of the averages, as well as the generation of the two notifications related to them:

- tn3270eRtCollCtlSPeriod
- tn3270eRtCollCtlSPMult
- tn3270eRtCollCtlThreshHigh
- tn3270eRtCollCtlThreshLow
- tn3270eRtCollCtlIdleCount

The previous objects' values are meaningless if the associated average(3) bit is not set to 1.

If the buckets(4) bit is set to 1, then the following objects have meaning, and specify the bucket boundaries:

- tn3270eRtCollCtlBucketBndry1
- tn3270eRtCollCtlBucketBndry2
- tn3270eRtCollCtlBucketBndry3
- tn3270eRtCollCtlBucketBndry4

The previous objects' values are meaningless if the associated buckets(4) bit is not set to 1.

If an entry in the tn3270RtCollCtlTable has the value active(1) for its RowStatus, then an implementation SHALL NOT allow Set operations for any objects in the entry except:

- tn3270eRtCollCtlThreshHigh
- tn3270eRtCollCtlThreshLow
- tn3270eRtCollCtlRowStatus

4.2 tn3270eRtDataTable

Either a single entry or multiple entries are created in the tn3270eRtDataTable for each tn3270eRtCollCtlEntry, depending on whether tn3270eRtCollCtlType in the control entry has aggregate(0) selected. The contents of an entry in the tn3270eRtDataTable depend on the contents of the corresponding entry in the tn3270eRtCollCtlTable: as described above, some objects in the data entry return meaningful values only when the average(3) option is selected in the control entry, while others return meaningful values only when the buckets(4) option is selected. If both options are selected, then all the objects return meaningful values. When an object is not specified to return a meaningful value, an implementation may return any syntactically valid

value in response to a Get operation.

The following objects return meaningful values if and only if the average(3) option was selected in the corresponding tn3270eRtCollCtlEntry:

- tn3270eRtDataAvgRt
- tn3270eRtDataAvgIpRt
- tn3270eRtDataAvgCountTrans
- tn3270eRtDataIntTimeStamp
- tn3270eRtDataTotalRts
- tn3270eRtDataTotalIpRts
- tn3270eRtDataCountTrans
- tn3270eRtDataCountDrs
- tn3270eRtDataElapsRndTrpSq
- tn3270eRtDataElapsIpRtSq

The first three objects in this list return values derived from the sliding-window average calculations described earlier. The time of the most recent sample for these calculations is returned in the tn3270eRtDataIntTimeStamp object. The next four objects are normal Counter32 objects, maintaining counts of total response time and total transactions. The last two objects return sum of the squares values, to enable variance calculations by a management application.

The following objects return meaningful values if and only if the buckets(4) option was selected in the corresponding tn3270eRtCollCtlEntry:

- tn3270eRtDataBucket1Rts
- tn3270eRtDataBucket2Rts
- tn3270eRtDataBucket3Rts
- tn3270eRtDataBucket4Rts
- tn3270eRtDataBucket5Rts

A discontinuity object, tn3270eRtDataDiscontinuityTime, can be used by a management application to detect when the values of the counter objects in this table may have been reset, or otherwise experienced a discontinuity. A possible cause for such a discontinuity is the TN3270E server's being stopped or restarted. This object returns a meaningful value regardless of which collection control options were selected.

An object, tn3270eRtDataRtMethod, identifies whether the IP Network Time was calculated using either the definite response or TIMING-MARK approach.

When an entry is created in the tn3270eRtCollCtlTable with its tn3270eRtCollCtlType aggregate(0) bit set to 1, an entry is automatically created in the tn3270eRtDataTable; this entry's tn3270eRtDataClientAddress has the value of a zero-length octet string, its tn3270eRtDataClientAddrType has the value of unknown(0), and its tn3270eRtDataClientPort has the value 0.

When an entry is created in the tn3270eRtCollCtlTable with its tn3270eRtCollCtlType aggregate(0) bit set to 0, a separate entry is

created in the tn3270eRtDataTable for each member of the client group that currently has a session with the TN3270E server. Entries are subsequently created for clients that the TN3270E server determines to be members of the client group when these clients establish sessions with the server. Entries are also created when clients with existing sessions are added to the group.

All entries associated with a tn3270eRtCollCtlEntry are deleted from the tn3270eRtDataTable when that entry is deleted from the tn3270eRtCollCtlTable. An entry for an individual client in a client group is deleted when its TCP connection terminates. Once it has been created, a client's entry in the tn3270eRtDataTable remains active as long as the collection's tn3270eRtCollCtlEntry exists, even if the client is removed from the client group for the tn3270eRtCollCtlEntry.

4.3 Notifications

This MIB defines four notifications related to a tn3270eRtDataEntry. If the associated tn3270eRtCollCtlType object's traps(5) bit is set to 1, then the tn3270RtCollStart and tn3270RtCollEnd notifications are generated when, respsectively, the tn3270eRtDataEntry is created and deleted. If, in addition, this tn3270eRtCollCtlType object's average(3) bit is set to 1, then the the tn3270eRtExceeded and tn3270eRtOkay notifications are generated when the conditions they report occur.

The following notifications are defined by this MIB:

- tn3270eRtExceeded - The purpose of this notification is to signal that a performance problem has been detected. If average(3) response time data is being collected, then this notification is generated whenever (1) an average response time is first found, on a collection interval boundary, to have exceeded the high threshold tn3270eRtCollCtlThreshHigh specified for the client group, AND (2) the sample on which the average is based is determined to have been a significant one, via the significance algorithm described earlier. This notification is not generated again for a tn3270eRtDataEntry until an average response time falling below the low threshold tn3270eRtCollCtlThreshLow specified for the client group has occurred for the entry.

- tn3270eRtOkay - The purpose of this notification is to signal that a previously reported performance problem has been resolved. If average(3) response time data is being collected, then this notification is generated whenever (1) a tn3270eRtExceeded notification has already been generated, AND (2) an average response time is first found, on a collection interval boundary, to have fallen below the low threshold tn3270eRtCollCtlThreshLow specified for the client group. This notification is not generated again for a tn3270eRtDataEntry until an average response time exceeding the high threshold tn3270eRtCollCtlThreshHigh specified for the client group has occurred for the entry.

Taken together, the two preceding notifications serve to minimize the generation of an excessive number of traps in the case of an average response time that oscillates about its high threshold.

o tn3270eRtCollStart - This notification is generated whenever data collection begins for a client group, or when a new tn3270eRtDataEntry becomes active. The primary purpose of this notification is signal to a management application that a new client TCP session has been established, and to provide the IP-to-resource mapping for the session. This notification is not critical when

average(3) data collection is not being performed for the client group.

o tn3270eRtCollEnd - This notification is generated whenever a data collection ends. For an aggregate collection, this occurs when the corresponding tn3270eRtCollCtlEntry is deleted. For an individual collection, this occurs either when the tn3270eRtCollCtlEntry is deleted, or when the client's TCP connection terminates. The purpose of this notification is to enable a management application to complete a monitoring function that it was performing, by returning final values for the collection's data objects.

4.4 Advisory Spin Lock Usage

Within the TN3270E-RT-MIB, tn3270eRtSpinLock is defined as an advisory lock that allows cooperating TN3270E-RT-MIB applications to coordinate their use of the tn3270eRtCollCtlTable. When creating a new entry or altering an existing entry in the tn3270eRtCollCtlTable, an application SHOULD make use of tn3270eRtSpinLock to serialize application changes or additions. Since this is an advisory lock, its use by management applications SHALL NOT be enforced by agents. Agents MUST, however, implement the tn3270eRtSpinLock object.

5.0 Definitions

```
  TN3270E-RT-MIB DEFINITIONS ::= BEGIN

  IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
      Counter32, Unsigned32, Gauge32
                  FROM SNMPv2-SMI
      RowStatus, DateAndTime, TimeStamp, TestAndIncr
                  FROM SNMPv2-TC
      MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
                  FROM SNMPv2-CONF
      tn3270eSrvrConfIndex, tn3270eClientGroupName,
      tn3270eResMapElementType
                  FROM TN3270E-MIB
      IANATn3270eAddrType, IANATn3270eAddress
                  FROM IANATn3270eTC-MIB
      snanauMIB
                  FROM SNA-NAU-MIB;

    tn3270eRtMIB   MODULE-IDENTITY
        LAST-UPDATED "9807270000Z" -- July 27, 1998
        ORGANIZATION "TN3270E Working Group"
        CONTACT-INFO
          "Kenneth White (kennethw@vnet.ibm.com)
```

IBM Corp. - Dept. BRQA/Bldg. 501/G114
P.O. Box 12195
3039 Cornwallis
RTP, NC 27709-2195

Robert Moore (remoore@us.ibm.com)

```
           IBM Corp. - Dept. BRQA/Bldg. 501/G114
           P.O. Box 12195
           3039 Cornwallis
           RTP, NC 27709-2195
           (919) 254-4436"
       DESCRIPTION
          "This module defines a portion of the management
          information base (MIB) that enables monitoring of
          TN3270 and TN3270E clients' response times by a
          TN3270E server."
       REVISION  "9807270000Z" -- July 27, 1998
       DESCRIPTION
           "RFC nnnn (Proposed Standard)" -- RFC Editor to fill in
  ::= { snanauMIB 9 }
  -- snanauMIB ::= { mib-2 34 }

  -- Top level structure of the MIB

  tn3270eRtNotifications   OBJECT IDENTIFIER  ::= { tn3270eRtMIB 0 }
  tn3270eRtObjects         OBJECT IDENTIFIER  ::= { tn3270eRtMIB 1 }
  tn3270eRtConformance     OBJECT IDENTIFIER  ::= { tn3270eRtMIB 3 }

  -- MIB Objects

  -- Response Time Control Table

  tn3270eRtCollCtlTable  OBJECT-TYPE
      SYNTAX       SEQUENCE OF Tn3270eRtCollCtlEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
        "The response time monitoring collection control table,
        which allows a management application to control the
        types of response time data being collected, and the
        clients for which it is being collected.

        This table is indexed by tn3270eSrvrConfIndex and
        tn3270eClientGroupName imported from the
        TN3270E-MIB.  tn3270eSrvrConfIndex indicates within
        a host which TN3270E server an entry applies to.
        tn3270eClientGroupName it identifies the set of IP
        clients for which response time data is being collected.
        The particular IP clients making up the set are identified
        in the tn3270eClientGroupTable in the TN3270E-MIB."
      ::= { tn3270eRtObjects 1}

  tn3270eRtCollCtlEntry    OBJECT-TYPE
      SYNTAX        Tn3270eRtCollCtlEntry
      MAX-ACCESS    not-accessible
```

```
      STATUS       current
      DESCRIPTION
        "An entry in the TN3270E response time monitoring collection
        control table.  To handle the case of multiple TN3270E
        servers on the same host, the first index of this table is
        the tn3270eSrvrConfIndex from the TN3270E-MIB."
```

```
      INDEX {
        tn3270eSrvrConfIndex,    -- Server's index
        tn3270eClientGroupName } -- What to collect on
      ::= { tn3270eRtCollCtlTable 1 }

  Tn3270eRtCollCtlEntry ::= SEQUENCE {
      tn3270eRtCollCtlType              BITS,
      tn3270eRtCollCtlSPeriod           Unsigned32,
      tn3270eRtCollCtlSPMult            Unsigned32,
      tn3270eRtCollCtlThreshHigh        Unsigned32,
      tn3270eRtCollCtlThreshLow         Unsigned32,
      tn3270eRtCollCtlIdleCount         Unsigned32,
      tn3270eRtCollCtlBucketBndry1      Unsigned32,
      tn3270eRtCollCtlBucketBndry2      Unsigned32,
      tn3270eRtCollCtlBucketBndry3      Unsigned32,
      tn3270eRtCollCtlBucketBndry4      Unsigned32,
      tn3270eRtCollCtlRowStatus         RowStatus   }

  -- The OID { tn3270eRtCollCtlEntry 1 } is not used

  tn3270eRtCollCtlType  OBJECT-TYPE
      SYNTAX    BITS {
                       aggregate(0),
                       excludeIpComponent(1),
                       ddr(2),
                       average(3),
                       buckets(4),
                       traps(5)
                     }
      MAX-ACCESS   read-create
      STATUS       current
      DESCRIPTION
        "This object controls what types of response time data to
         collect, whether to summarize the data across the members
         of a client group or keep it individually, whether to
         introduce dynamic definite responses, and whether to
         generate traps.

         aggregate(0)          - Aggregate response time data for the
                                 client group as a whole.  If this bit
                                 is set to 0, then maintain response
                                 time data separately for each member
                                 of the client group.
         excludeIpComponent(1) - Do not include the IP-network
                                 component in any response times.
         ddr(2)                - Enable dynamic definite response.
         average(3)            - Produce an average response time
                                 based on a specified collection
                                 interval.
```

buckets(4) - Maintain tn3270eRtDataBucket values in a corresponding tn3270eRtDataEntry, based on the bucket boundaries specified in the tn3270eRtCollCtlBucketBndry objects .

traps(5) - generate the notifications specified

```
                                in this MIB module.  The
                                tn3270eRtExceeded and tn3270eRtOkay
                                notifications are generated only if
                                average(3) is also specified."
        ::= { tn3270eRtCollCtlEntry 2 }

    tn3270eRtCollCtlSPeriod OBJECT-TYPE
        SYNTAX  Unsigned32 (15..86400) -- 15 second min, 24 hour max
        UNITS   "seconds"
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
          "The number of seconds that defines the sample period.
           The actual interval is defined as tn3270eRtCollCtlSPeriod
           times tn3270eRtCollCtlSPMult.

           The value of this object is used only if the corresponding
           tn3270eRtCollCtlType has the average(3) setting."
        DEFVAL   {20}    -- 20 seconds
        ::= { tn3270eRtCollCtlEntry 3 }

    tn3270eRtCollCtlSPMult OBJECT-TYPE
        SYNTAX  Unsigned32 (1..5760) -- 5760 x SPeriod of 15 is 24 hours
        UNITS   "period"
        MAX-ACCESS   read-create
        STATUS       current
        DESCRIPTION
          "The sample period multiplier; this value is multiplied by
          the sample period, tn3270eRtCollCtlSPeriod, to determine
          the collection interval.

          Sliding-window average calculation can, if necessary, be
          disabled, by setting the sample period multiplier,
          tn3270eRtCollCtlSPMult, to 1, and setting the sample
          period, tn3270eRtCollCtlSPeriod, to the required
          collection interval.

          The value of this object is used only if the corresponding
          tn3270eRtCollCtlType has the average(3) setting."
        DEFVAL   { 30 }    -- yields an interval of 10 minutes when
                           -- used with the default SPeriod value
        ::= { tn3270eRtCollCtlEntry 4 }

    tn3270eRtCollCtlThreshHigh  OBJECT-TYPE
        SYNTAX            Unsigned32
        UNITS             "seconds"
        MAX-ACCESS        read-create
        STATUS            current
        DESCRIPTION
```

"The threshold for generating a tn3270eRtExceeded
notification, signalling that a monitored total response
time has exceeded the specified limit. A value of zero
for this object suppresses generation of this notification.
The value of this object is used only if the corresponding
tn3270eRtCollCtlType has average(3) and traps(5) selected.

```
        A tn3270eRtExceeded notification is not generated again for a
        tn3270eRtDataEntry until an average response time falling below
        the low threshold tn3270eRtCollCtlThreshLow specified for the
        client group has occurred for the entry."

      DEFVAL   { 0 }   -- suppress notifications
      ::= { tn3270eRtCollCtlEntry 5 }

  tn3270eRtCollCtlThreshLow   OBJECT-TYPE
      SYNTAX            Unsigned32
      UNITS             "seconds"
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
        "The threshold for generating a tn3270eRtOkay notification,
        signalling that a monitored total response time has fallen
        below the specified limit.  A value of zero for this object
        suppresses generation of this notification.  The value of
        this object is used only if the corresponding
        tn3270eRtCollCtlType has average(3) and traps(5) selected.

        A tn3270eRtOkay notification is not generated again for a
        tn3270eRtDataEntry until an average response time
        exceeding the high threshold tn3270eRtCollCtlThreshHigh
        specified for the client group has occurred for the entry."
      DEFVAL   { 0 }   -- suppress notifications
      ::= { tn3270eRtCollCtlEntry 6 }

  tn3270eRtCollCtlIdleCount   OBJECT-TYPE
      SYNTAX            Unsigned32
      UNITS             "transactions"
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
        "The value of this object is used to determine whether a
        sample that yields an average response time exceeding the
        value of tn3270eRtCollCtlThreshHigh was a statistically
        valid one.  If the following statement is true, then the
        sample was statistically valid, and so a tn3270eRtExceeded
        notification should be generated:

          AvgCountTrans * ((AvgRt/ThreshHigh - 1) ** 2) >=  IdleCount

        This comparison is done only if the corresponding
        tn3270eRtCollCtlType has average(3) and traps(5) selected."
      DEFVAL { 1 }
      ::= { tn3270eRtCollCtlEntry 7 }
```

```
  tn3270eRtCollCtlBucketBndry1   OBJECT-TYPE
      SYNTAX            Unsigned32
      UNITS             "tenths of seconds"
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
```

```
        "The value of this object defines the range of transaction
         response times counted in the Tn3270eRtDataBucket1Rts
         object: those less than or equal to this value."
      DEFVAL { 10 }
      ::= { tn3270eRtCollCtlEntry 8 }

  tn3270eRtCollCtlBucketBndry2   OBJECT-TYPE
      SYNTAX            Unsigned32
      UNITS             "tenths of seconds"
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
        "The value of this object, together with that of the
        tn3270eRtCollCtlBucketBndry1 object, defines the range
        of transaction response times counted in the
        Tn3270eRtDataBucket2Rts object: those greater than the
        value of the tn3270eRtCollCtlBucketBndry1 object, and
        less than or equal to the value of this object."
      DEFVAL { 20 }
      ::= { tn3270eRtCollCtlEntry 9 }

  tn3270eRtCollCtlBucketBndry3   OBJECT-TYPE
      SYNTAX            Unsigned32
      UNITS             "tenths of seconds"
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
        "The value of this object, together with that of the
        tn3270eRtCollCtlBucketBndry2 object, defines the range of
        transaction response times counted in the
        Tn3270eRtDataBucket3Rts object:  those greater than the
        value of the tn3270eRtCollCtlBucketBndry2 object, and less
        than or equal to the value of this object."
      DEFVAL { 50 }
      ::= { tn3270eRtCollCtlEntry 10 }

  tn3270eRtCollCtlBucketBndry4   OBJECT-TYPE
      SYNTAX            Unsigned32
      UNITS             "tenths of seconds"
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
        "The value of this object, together with that of the
        tn3270eRtCollCtlBucketBndry3 object, defines the range
        of transaction response times counted in the
        Tn3270eRtDataBucket4Rts object: those greater than the
        value of the tn3270eRtCollCtlBucketBndry3 object, and
        less than or equal to the value of this object.
```

```
        The value of this object also defines the range of
        transaction response times counted in the
        Tn3270eRtDataBucket5Rts object: those greater than the
        value of this object."
      DEFVAL { 100 }
      ::= { tn3270eRtCollCtlEntry 11 }
```

```
  tn3270eRtCollCtlRowStatus  OBJECT-TYPE
      SYNTAX            RowStatus
      MAX-ACCESS        read-create
      STATUS            current
      DESCRIPTION
        "This object allows entries to be created and deleted
         in the tn3270eRtCollCtlTable.  An entry in this table
         is deleted by setting this object to destroy(6).
         Deleting an entry in this table has the side-effect
         of removing all entries from the tn3270eRtDataTable
         that are associated with the entry being deleted."
      ::= { tn3270eRtCollCtlEntry 12 }


  -- TN3270E Response Time Data Table

  tn3270eRtDataTable  OBJECT-TYPE
      SYNTAX       SEQUENCE OF Tn3270eRtDataEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
        "The response time data table.  Entries in this table are
         created based on entries in the tn3270eRtCollCtlTable."
      ::= { tn3270eRtObjects 2 }

  tn3270eRtDataEntry  OBJECT-TYPE
      SYNTAX        Tn3270eRtDataEntry
      MAX-ACCESS    not-accessible
      STATUS        current
      DESCRIPTION
        "Entries in this table are created based upon the
        tn3270eRtCollCtlTable.  When the corresponding
        tn3270eRtCollCtlType has aggregate(0) specified, a single
        entry is created in this table, with a tn3270eRtDataClientAddrType
        of unknown(0), a zero-length octet string value for
        tn3270eRtDataClientAddress, and a tn3270eRtDataClientPort value of
        0.  When aggregate(0) is not specified, a separate entry is
        created for each client in the group.

        Note that the following objects defined within an entry in this
        table can  wrap:
            tn3270eRtDataTotalRts
            tn3270eRtDataTotalIpRts
            tn3270eRtDataCountTrans
            tn3270eRtDataCountDrs
            tn3270eRtDataElapsRnTrpSq
            tn3270eRtDataElapsIpRtSq
            tn3270eRtDataBucket1Rts
```

```
            tn3270eRtDataBucket2Rts
            tn3270eRtDataBucket3Rts
            tn3270eRtDataBucket4Rts
            tn3270eRtDataBucket5Rts"
         INDEX {
            tn3270eSrvrConfIndex,      -- Server's local index
```

```
        tn3270eClientGroupName,    -- Collection target
        tn3270eRtDataClientAddrType,
        tn3270eRtDataClientAddress,
        tn3270eRtDataClientPort }
    ::= { tn3270eRtDataTable 1 }

  Tn3270eRtDataEntry ::= SEQUENCE {
        tn3270eRtDataClientAddrType        IANATn3270eAddrType,
        tn3270eRtDataClientAddress         IANATn3270eAddress,
        tn3270eRtDataClientPort            Unsigned32,
        tn3270eRtDataAvgRt                 Gauge32,
        tn3270eRtDataAvgIpRt               Gauge32,
        tn3270eRtDataAvgCountTrans         Gauge32,
        tn3270eRtDataIntTimeStamp          DateAndTime,
        tn3270eRtDataTotalRts              Counter32,
        tn3270eRtDataTotalIpRts            Counter32,
        tn3270eRtDataCountTrans            Counter32,
        tn3270eRtDataCountDrs              Counter32,
        tn3270eRtDataElapsRndTrpSq         Unsigned32,
        tn3270eRtDataElapsIpRtSq           Unsigned32,
        tn3270eRtDataBucket1Rts            Counter32,
        tn3270eRtDataBucket2Rts            Counter32,
        tn3270eRtDataBucket3Rts            Counter32,
        tn3270eRtDataBucket4Rts            Counter32,
        tn3270eRtDataBucket5Rts            Counter32,
        tn3270eRtDataRtMethod              INTEGER,
        tn3270eRtDataDiscontinuityTime     TimeStamp
     }

  tn3270eRtDataClientAddrType   OBJECT-TYPE
      SYNTAX    IANATn3270eAddrType
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
        "Indicates the type of address represented by the value
        of tn3270eRtDataClientAddress.  The value unknown(0) is
        used if aggregate data is being collected for the client
        group."
      ::= { tn3270eRtDataEntry 1 }

  tn3270eRtDataClientAddress   OBJECT-TYPE
      SYNTAX    IANATn3270eAddress
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
        "Contains the IP address of the TN3270 client being
        monitored.  A zero-length octet string is used if
        aggregate data is being collected for the client group."
      ::= { tn3270eRtDataEntry 2 }
```

```
  tn3270eRtDataClientPort   OBJECT-TYPE
      SYNTAX       Unsigned32(0..65535)
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
```

```
        "Contains the client port number of the TN3270 client being
        monitored.  The value 0 is used if aggregate data is being
        collected for the client group, or if the
        tn3270eRtDataClientAddrType identifies an address type that
        does not support ports."
      ::= { tn3270eRtDataEntry 3 }

  tn3270eRtDataAvgRt OBJECT-TYPE
      SYNTAX       Gauge32
      UNITS        "tenths of seconds"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The average total response time measured over the last
        collection interval."
      DEFVAL { 0 }
      ::= { tn3270eRtDataEntry 4 }

  tn3270eRtDataAvgIpRt OBJECT-TYPE
      SYNTAX       Gauge32
      UNITS        "tenths of seconds"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The average IP response time measured over the last
        collection interval."
      DEFVAL { 0 }
      ::= { tn3270eRtDataEntry 5 }

  tn3270eRtDataAvgCountTrans   OBJECT-TYPE
      SYNTAX       Gauge32
      UNITS        "transactions"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The sliding transaction count used for calculating the
        values of the tn3270eRtDataAvgRt and tn3270eRtDataAvgIpRt
        objects.  The actual transaction count is available in
        the tn3270eRtDataCountTrans object.

        The initial value of this object, before any averages have
        been calculated, is 0."
      ::= { tn3270eRtDataEntry 6 }

  tn3270eRtDataIntTimeStamp   OBJECT-TYPE
      SYNTAX       DateAndTime
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
```

"The date and time of the last interval that
tn3270eRtDataAvgRt, tn3270eRtDataAvgIpRt, and
tn3270eRtDataAvgCountTrans were calculated.

Prior to the calculation of the first interval
averages, this object returns the value

```
        0x00000000000000000000.  When this value is
        returned, the remaining objects in the entry have
        no significance."
      ::= { tn3270eRtDataEntry 7 }

  tn3270eRtDataTotalRts   OBJECT-TYPE
      SYNTAX       Counter32
      UNITS        "tenths of seconds"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The count of the total response times collected.

        A management application can detect discontinuities in this
        counter by monitoring the tn3270eRtDataDiscontinuityTime
        object."
      ::= { tn3270eRtDataEntry 8 }

  tn3270eRtDataTotalIpRts   OBJECT-TYPE
      SYNTAX       Counter32
      UNITS        "tenths of seconds"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The count of the total IP-network response times
        collected.

        A management application can detect discontinuities in this
        counter by monitoring the tn3270eRtDataDiscontinuityTime
        object."
      ::= { tn3270eRtDataEntry 9 }

  tn3270eRtDataCountTrans   OBJECT-TYPE
      SYNTAX       Counter32
      UNITS        "transactions"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The count of the total number of transactions detected.

        A management application can detect discontinuities in this
        counter by monitoring the tn3270eRtDataDiscontinuityTime
        object."
      ::= { tn3270eRtDataEntry 10 }

  tn3270eRtDataCountDrs   OBJECT-TYPE
      SYNTAX       Counter32
      UNITS        "definite responses"
      MAX-ACCESS   read-only
```

```
      STATUS       current
      DESCRIPTION
        "The count of the total number of definite responses
        detected.

        A management application can detect discontinuities in this
```

```
       counter by monitoring the tn3270eRtDataDiscontinuityTime
       object."
     ::= { tn3270eRtDataEntry 11 }

  tn3270eRtDataElapsRndTrpSq   OBJECT-TYPE
     SYNTAX       Unsigned32
     UNITS        "tenths of seconds squared"
     MAX-ACCESS   read-only
     STATUS       current
     DESCRIPTION
       "The sum of the elapsed round trip time squared.  The sum
       of the squares is kept in order to enable calculation of
       a variance."
     DEFVAL { 0 }
     ::= { tn3270eRtDataEntry 12 }

  tn3270eRtDataElapsIpRtSq   OBJECT-TYPE
     SYNTAX       Unsigned32
     UNITS        "tenths of seconds squared"
     MAX-ACCESS   read-only
     STATUS       current
     DESCRIPTION
       "The sum of the elapsed IP round trip time squared.
       The sum of the squares is kept in order to enable
       calculation of a variance."
     DEFVAL { 0 }
     ::= { tn3270eRtDataEntry 13 }

  tn3270eRtDataBucket1Rts   OBJECT-TYPE
     SYNTAX       Counter32
     MAX-ACCESS   read-only
     STATUS       current
     DESCRIPTION
       "The count of the response times falling into bucket 1.

       A management application can detect discontinuities in this
       counter by monitoring the tn3270eRtDataDiscontinuityTime
       object."
     ::= { tn3270eRtDataEntry 14 }

  tn3270eRtDataBucket2Rts   OBJECT-TYPE
     SYNTAX       Counter32
     MAX-ACCESS   read-only
     STATUS       current
     DESCRIPTION
       "The count of the response times falling into bucket 2.

       A management application can detect discontinuities in this
       counter by monitoring the tn3270eRtDataDiscontinuityTime
```

```
        object."
      ::= { tn3270eRtDataEntry 15 }

  tn3270eRtDataBucket3Rts   OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
```

```
      STATUS       current
      DESCRIPTION
        "The count of the response times falling into bucket 3.

        A management application can detect discontinuities in this
        counter by monitoring the tn3270eRtDataDiscontinuityTime
        object."
      ::= { tn3270eRtDataEntry 16 }

  tn3270eRtDataBucket4Rts  OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The count of the response times falling into bucket 4.

        A management application can detect discontinuities in this
        counter by monitoring the tn3270eRtDataDiscontinuityTime
        object."
      ::= { tn3270eRtDataEntry 17 }

  tn3270eRtDataBucket5Rts  OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The count of the response times falling into bucket 5.

        A management application can detect discontinuities in this
        counter by monitoring the tn3270eRtDataDiscontinuityTime
        object."
      ::= { tn3270eRtDataEntry 18 }

  tn3270eRtDataRtMethod OBJECT-TYPE
      SYNTAX       INTEGER {
                             none(0),
                             responses(1),
                             timingMark(2)
                           }
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
        "The value of this object indicates the method that was
        used in calculating the IP network time.

        The value 'none(0) indicates that response times were not
        calculated for the IP network."
      ::= { tn3270eRtDataEntry 19 }
```

```
  tn3270eRtDataDiscontinuityTime OBJECT-TYPE
      SYNTAX      TimeStamp
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The value of sysUpTime on the most recent occasion at
```

```
          which one or more of this entry's counter objects
          suffered a discontinuity.  This may happen if a TN3270E
          server is stopped and then restarted, and local methods
          are used to set up collection policy
          (tn3270eRtCollCtlTable entries)."
      ::= { tn3270eRtDataEntry 20 }


  tn3270eRtSpinLock OBJECT-TYPE
      SYNTAX      TestAndIncr
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
        "An advisory lock used to allow cooperating TN3270E-RT-MIB
        applications to coordinate their use of the
        tn3270eRtCollCtlTable.

        When creating a new entry or altering an existing entry
        in the tn3270eRtCollCtlTable, an application should make
        use of tn3270eRtSpinLock to serialize application changes
        or additions.

        Since this is an advisory lock, the use of this lock is
        not enforced."
      ::= { tn3270eRtObjects 3 }

  -- Notifications

  tn3270eRtExceeded   NOTIFICATION-TYPE
      OBJECTS {
         tn3270eRtDataIntTimeStamp,
         tn3270eRtDataAvgRt,
         tn3270eRtDataAvgIpRt,
         tn3270eRtDataAvgCountTrans,
         tn3270eRtDataRtMethod
      }
      STATUS  current
      DESCRIPTION
        "This notification is generated when the average response
        time, tn3270eRtDataAvgRt, exceeds
        tn3270eRtCollCtlThresholdHigh at the end of a collection
        interval specified by tn3270eCollCtlSPeriod
        times tn3270eCollCtlSPMult.  Note that the corresponding
        tn3270eCollCtlType must have traps(5) and average(3) set
        for this notification to be generated.  In addition,
        tn3270eRtDataAvgCountTrans, tn3270eRtCollCtlThreshHigh, and
        tn3270eRtDataAvgRt are algorithmically compared to
        tn3270eRtCollCtlIdleCount for determination if this
        notification will be suppressed."
```

```
      ::= { tn3270eRtNotifications 1 }

  tn3270eRtOkay   NOTIFICATION-TYPE
      OBJECTS {
         tn3270eRtDataIntTimeStamp,
         tn3270eRtDataAvgRt,
```

```
        tn3270eRtDataAvgIpRt,
        tn3270eRtDataAvgCountTrans,
        tn3270eRtDataRtMethod
    }
    STATUS  current
    DESCRIPTION
      "This notification is generated when the average response
      time, tn3270eRtDataAvgRt, falls below
      tn3270eRtCollCtlThresholdLow at the end of a collection
      interval specified by tn3270eCollCtlSPeriod times
      tn3270eCollCtlSPMult, after a tn3270eRtExceeded
      notification was generated.  Note that the corresponding
      tn3270eCollCtlType must have traps(5) and average(3)
      set for this notification to be generated."
    ::= { tn3270eRtNotifications 2 }

tn3270eRtCollStart NOTIFICATION-TYPE
    OBJECTS {
       tn3270eRtDataRtMethod,       -- type of collection
       tn3270eResMapElementType     -- type of resource
    }
    STATUS  current
    DESCRIPTION
      "This notification is generated when response time data
      collection is enabled for a member of a client group.
      In order for this notification to occur the corresponding
      tn3270eRtCollCtlType must have traps(5) selected.

      tn3270eResMapElementType contains a valid value only if
      tn3270eRtDataClientAddress contains a valid address
      (rather than a zero-length octet string)."
    ::= { tn3270eRtNotifications 3 }

tn3270eRtCollEnd   NOTIFICATION-TYPE
    OBJECTS {
       tn3270eRtDataDiscontinuityTime,
       tn3270eRtDataAvgRt,
       tn3270eRtDataAvgIpRt,
       tn3270eRtDataAvgCountTrans,
       tn3270eRtDataIntTimeStamp,
       tn3270eRtDataTotalRts,
       tn3270eRtDataTotalIpRts,
       tn3270eRtDataCountTrans,
       tn3270eRtDataCountDrs,
       tn3270eRtDataElapsRndTrpSq,
       tn3270eRtDataElapsIpRtSq,
       tn3270eRtDataBucket1Rts,
       tn3270eRtDataBucket2Rts,
       tn3270eRtDataBucket3Rts,
```

```
         tn3270eRtDataBucket4Rts,
         tn3270eRtDataBucket5Rts,
         tn3270eRtDataRtMethod
      }
      STATUS  current
      DESCRIPTION
```

```
        "This notification is generated when an tn3270eRtDataEntry
        is deleted after being active (actual data collected), in
        order to enable a management application monitoring an
        tn3270eRtDataEntry to get the entry's final values.  Note
        that the corresponding tn3270eCollCtlType must have traps(5)
        set for this notification to be generated."
      ::= { tn3270eRtNotifications 4 }

  -- Conformance Statement

  tn3270eRtGroups       OBJECT IDENTIFIER ::= { tn3270eRtConformance 1 }
  tn3270eRtCompliances  OBJECT IDENTIFIER ::= { tn3270eRtConformance 2 }

  -- Compliance statements

  tn3270eRtCompliance     MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
        "The compliance statement for agents that support the
        TN327E-RT-MIB."
      MODULE   -- this module
         MANDATORY-GROUPS { tn3270eRtGroup, tn3270eRtNotGroup }

      OBJECT tn3270eRtCollCtlType
         MIN-ACCESS  read-only
         DESCRIPTION
            "The agent is not required to support a SET operation to
            this object in the absence of adequate security."

      OBJECT tn3270eRtCollCtlSPeriod
         MIN-ACCESS  read-only
         DESCRIPTION
            "The agent is not required to allow the user to change
            the default value of this object, and is allowed to
            use a different default."

      OBJECT tn3270eRtCollCtlSPMult
         MIN-ACCESS  read-only
         DESCRIPTION
            "The agent is not required to support a SET operation
            to this object in the absence of adequate security."

      OBJECT tn3270eRtCollCtlThreshHigh
         MIN-ACCESS  read-only
         DESCRIPTION
            "The agent is not required to support a SET operation
            to this object in the absence of adequate security."

      OBJECT tn3270eRtCollCtlThreshLow
         MIN-ACCESS  read-only
```

```
         DESCRIPTION
            "The agent is not required to support a SET operation
            to this object in the absence of adequate security."

      OBJECT tn3270eRtCollCtlIdleCount
```

```
        MIN-ACCESS  read-only
        DESCRIPTION
           "The agent is not required to support a SET operation
           to this object in the absence of adequate security."

     OBJECT tn3270eRtCollCtlBucketBndry1
        MIN-ACCESS  read-only
        DESCRIPTION
           "The agent is not required to support a SET operation
           to this object in the absence of adequate security."

     OBJECT tn3270eRtCollCtlBucketBndry2
        MIN-ACCESS  read-only
        DESCRIPTION
           "The agent is not required to support a SET operation
           to this object in the absence of adequate security."

     OBJECT tn3270eRtCollCtlBucketBndry3
        MIN-ACCESS  read-only
        DESCRIPTION
           "The agent is not required to support a SET operation
           to this object in the absence of adequate security."

     OBJECT tn3270eRtCollCtlBucketBndry4
        MIN-ACCESS  read-only
        DESCRIPTION
           "The agent is not required to support a SET operation
           to this object in the absence of adequate security."

     OBJECT tn3270eRtCollCtlRowStatus
        SYNTAX   INTEGER {
                          active(1) -- subset of RowStatus
                         }
        MIN-ACCESS read-only
        DESCRIPTION
           "Write access is not required, and only one of the six
           enumerated values for the RowStatus textual convention
           need be supported, specifically: active(1)."

     ::= {tn3270eRtCompliances 1 }

  -- Group definitions

  tn3270eRtGroup         OBJECT-GROUP
     OBJECTS {
         tn3270eRtCollCtlType,
         tn3270eRtCollCtlSPeriod,
         tn3270eRtCollCtlSPMult,
         tn3270eRtCollCtlThreshHigh,
```

```
          tn3270eRtCollCtlThreshLow,
          tn3270eRtCollCtlIdleCount,
          tn3270eRtCollCtlBucketBndry1,
          tn3270eRtCollCtlBucketBndry2,
          tn3270eRtCollCtlBucketBndry3,
          tn3270eRtCollCtlBucketBndry4,
```

```
          tn3270eRtCollCtlRowStatus,
          tn3270eRtDataDiscontinuityTime,
          tn3270eRtDataAvgRt,
          tn3270eRtDataAvgIpRt,
          tn3270eRtDataAvgCountTrans,
          tn3270eRtDataIntTimeStamp,
          tn3270eRtDataTotalRts,
          tn3270eRtDataTotalIpRts,
          tn3270eRtDataCountTrans,
          tn3270eRtDataCountDrs,
          tn3270eRtDataElapsRndTrpSq,
          tn3270eRtDataElapsIpRtSq,
          tn3270eRtDataBucket1Rts,
          tn3270eRtDataBucket2Rts,
          tn3270eRtDataBucket3Rts,
          tn3270eRtDataBucket4Rts,
          tn3270eRtDataBucket5Rts,
          tn3270eRtDataRtMethod,
          tn3270eRtSpinLock }
      STATUS  current
      DESCRIPTION
        "This group is mandatory for all implementations that
        support the TN3270E-RT-MIB. "
      ::= { tn3270eRtGroups 1 }

  tn3270eRtNotGroup         NOTIFICATION-GROUP
      NOTIFICATIONS {
          tn3270eRtExceeded,
          tn3270eRtOkay,
          tn3270eRtCollStart,
          tn3270eRtCollEnd
       }
      STATUS  current
      DESCRIPTION
        "The notifications that must be supported when the
        TN3270E-RT-MIB is implemented. "
      ::= { tn3270eRtGroups 2 }

  END
```

6.0 Security Considerations

Certain management information defined in this MIB may be considered sensitive in some network environments. Therefore, authentication of received SNMP requests and controlled access to management information SHOULD be employed in such environments. An authentication protocol is defined in [12]. A protocol for access control is defined in [15].

Several objects in this MIB allow write access or provide for row creation. Allowing this support in a non-secure environment can have a negative effect on network operations. It is RECOMMENDED that implementers seriously consider whether set operations or row creation SHOULD be allowed without providing, at a minimum, authentication of

request origin. It is RECOMMENDED that without such support that the following objects be implemented as read-only:

- tn3270eRtCollCtlType
- tn3270eRtCollCtlSPeriod
- tn3270eRtCollCtlSPMult
- tn3270eRtCollCtlThreshHigh
- tn3270eRtCollCtlThreshLow
- tn3270eRtCollCtlIdleCount
- tn3270eRtCollCtlBucketBndry1
- tn3270eRtCollCtlBucketBndry2
- tn3270eRtCollCtlBucketBndry3
- tn3270eRtCollCtlBucketBndry4
- tn3270eRtCollCtlRowStatus

The administrative method to use to create and manage the tn3270eRtCollCtlTable when SET support is not allowed is outside of the scope of this memo.

7.0 Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

8.0 Acknowledgments

This document is a product of the TN3270E Working Group. Special thanks are due to Derek Bolton and Michael Boe of Cisco Systems for their numerous comments and suggestions for improving the structure of this MIB. Thanks also to Randy Presuhn of BMC Software for his valuable review comments on several versions of the document.

9.0 References

[1] Harrington D., Presuhn, R., Wijnen, B., "An Architecture for Describing SNMP Management Frameworks", RFC 2271, Cabletron Systems, Inc., BMC Software, Inc., IBM T.J. Watson Research, January 1998.

[2] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990

[3] Rose, M., and K. McCloghrie, "Concise MIB Definitions", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991

[4] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, Performance Systems International, March 1991

[5] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, SNMP Research,Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[6] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[7] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[8] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[9] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[10] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[11] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message

Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, SNMP Research, Inc., Cabletron Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, January 1998.

[12] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, IBM T. J. Watson Research, January 1998.

[13] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[14] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2273, SNMP Research, Inc., Secure Computing Corporation, Cisco Systems, January 1998

[15] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, IBM T. J. Watson Research, BMC Software, Inc., Cisco Systems, Inc., January 1998

[16] Postel, J., and Reynolds, J., "Telnet Protocol Specification", RFC 854, May 1983.

[17] Postel, J., and Reynolds, J., "Telnet Timing Mark Option", RFC 860, May 1983.

[18] Rekhter, J., "Telnet 3270 Regime Option", RFC 1041, January 1988.

[19] Kelly, B., "TN3270 Enhancements", RFC 2355, June 1998.

[20] White, K. and Moore, R., "Base Definitions of Managed Objects for TN3270E Using SMIv2", Internet-Draft Work in progress, April 1998.

[21] IBM, International Technical Support Centers, "Response Time Data Gathering", GG24-3212-01, November 1990.

[22] Hovey, R., and S. Bradner, "The Organizations Involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.

[23] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.0 Authors' Addresses

Kenneth D. White
Dept. BRQA/Bldg. 501/G114
IBM Corporation
P.O.Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709, USA
E-mail: kennethw@vnet.ibm.com

Robert Moore
Dept. BRQA/Bldg. 501/G114
IBM Corporation
P.O.Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709, USA
Phone: +1-919-254-7507
E-mail: remoore@us.ibm.com

11.0 Full Copyright Statement