Managing session performance using the NetView Performance Monitor

by L. Temoshenko

Managing the performance of network devices and their interaction with host applications is a complex task that entails the collection and reduction of information related to the underlying session. Depending on the specific management task at hand, differing types, correlations, and formattings of session performance measurements are required. The NetView™ Performance Monitor has a flexible set of facilities that can be used to provide the information needed to manage session performance. Its facilities to collect, correlate, and present session performance measurement are discussed in relation to typical network management tasks.

Networks containing thousands of terminals connected to host applications through ever-evolving physical devices and mediums are common in today's information processing environment. Throughout each day any given terminal in the network may be used to access one or more applications in order to perform some unit of work. After the connection between the terminal and the application has been established, a session is formed, and the pair are free to exchange requests and responses.

As the reliability of the physical connections between a terminal and an application continues to improve, greater emphasis is being placed on the performance of sessions over these physical connections. Session performance is often characterized by such things as the frequency, volume, and duration of exchange needed to accomplish the unit of work. Optimal session performance may be projected on the basis of these session characteristics, a given path through the network, and known processing requirements of the host application. These projections may not be reflective of the actual session performance at any given time for a variety of reasons. For example, current conditions in the network may not be conducive to optimal network flows. Likewise the host application may not be processing requests at anticipated levels.

With the increasing complexity of the underlying sessions, paths through the network, and host applications, network management products have become essential in providing the information needed to manage session performance. This performance information is crucial not only in managing the current environment but also in planning the future environment.

The IBM NetView* Performance Monitor (NPM) is a network management product designed to provide the essential information needed to manage network and session performance. NPM runs as

a Virtual Telecommunications Access Method (VTAM*) application in the Multiple Virtual Storage (MVS) and virtual machine (VM) operating system environments. It gathers performance information from real-time and historical sources. Once gathered, NPM provides analysis through on-line panels, a graphic subsystem, and batch reporting.

Although NPM has both a network and session performance component, this paper limits its discussion to the NPM session performance component. The remainder of this paper has been arranged to assist the reader in gaining an understanding of how the facilities of NPM can provide the information needed to manage session performance. Following this introduction are sections on session performance measurements, collection options, correlation schemes, and presentation formats.

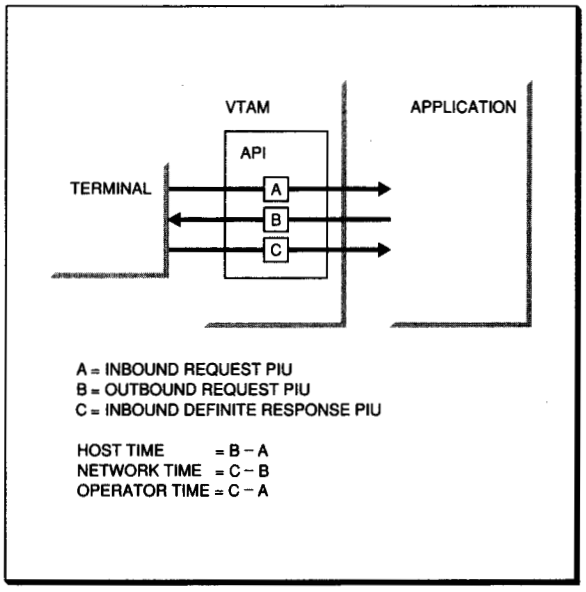
Session performance measurements

NPM has the ability to measure the performance of standard Systems Network Architecture (SNA) logical unit type 0, 2, and 6.2 sessions between a host application and a terminal in terms of transaction counts, transit times, and volume measurements. These measurements are based on the actual path information units (PIU) that pass through the VTAM application program interface (API) for a given logical unit pair. Individually or in combination, these measurements are indicative of the underlying characteristics and throughput of the session during a given period of time.

The session between a host application and a terminal is composed of SNA physical and logical units. SNA users are associated with a physical unit (PU) and communicate with each other over a logical unit (LU) session. The SNA architecture currently defines PU types 1 through 5 and LU types 0 through 7, with LU type 5 being undefined. Terminals are a PU type 2.0 or 2.1 peripheral node, depending on whether the device is a simple 3270-type display or an intelligent workstation. A host computer is a PU type 5 subarea node.

The interaction between a terminal and host application employs either an LU type 0, LU type 2, or LU type 6.2 session. Although LU type 0 sessions are free to use SNA formats and protocols in any desired way, actual implementations involve an application program in a PU type 5 node communicating with a PU type 2.0 or 2.1 peripheral node. LU type 2 sessions are defined for use by an application program running in a PU type 5 subarea node to communicate with a PU type 2.0 or 2.1 peripheral node in an interactive environment. LU type 6.2 sessions are defined for use by application programs to communicate with other application programs.

Figure 1 Transactions, transit times, and volume measurements are collected at the VTAM API



NPM can collect performance measurements for SNA LU type 0, 2, and and 6.2 sessions between an application program that runs in a PU type 5 subarea node and its partner terminal or application program that runs in either a PU type 2.0 or 2.1 peripheral node. Currently NPM cannot collect performance measurements for communications that pass directly between application programs running in PU type 2.1 peripheral nodes without the assistance of a PU type 5 subarea node.

A transaction can be viewed as a series of PIUs to accomplish a unit of work. Figure 1 illustrates a simple interactive session between a terminal and a host application. In this example, a transaction would begin with a new, first-in-chain, inbound request PIU to the application. After the application program processed the request, it would send one or more outbound request PIUs containing the customer data. If the application or the NPM op-

erator, through the dynamic definite response capability of NPM, requested a definite response, the underlying hardware would in turn send a definite response PIU back to the application program indicating receipt of the customer data. Regardless of whether or not the definite response PIU is a part of the exchange between the terminal and the host application, a separate transaction would be processed for each new inbound request PIU.

Within NPM terminology, the overall transit time for a transaction to complete is referred to as operator time. The NPM session collection routines calculate operator time by summing the host time component and network time component for a transaction. Host time in turn refers to the time spent in the application during a transaction. It is calculated by subtracting the time of the first inbound PIU from the time of the first subsequent outbound PIU. If the application or the NPM operator, through the dynamic definite response capability of NPM, requested a definite response, the network time is calculated by subtracting the time of the outbound PIU from the time of the subsequent inbound definite response PIU.

Application programs may run in definite or exception response mode. When the definite response PIU is a part of the exchange between the application program and the terminal, the application is said to be running in definite response mode. The application program does have the option of being notified only when there is a problem in the receipt of the outbound request PIUs. When this protocol is used, the application program is said to be running in exception response mode. The determination as to which response mode will be used for the session is performed by the terminal and application during a negotiation of parameters at session establishment. From a practical standpoint, the response mode of the application program is usually documented by the authors of the application program or will be indicated by a zero response count from NPM.

In situations where the network time and true operator time of the session are required for exception response applications, NPM provides a dynamic definite response capability. The feature simulates a definite response environment between the terminal and application in an effort to collect the network time. This facility does, however, have its drawbacks and should be used with discretion. Application programs normally implement the exception response protocol in an effort to improve response time and throughput. The improvements are made possible by eliminating the response PIU from the session flow and freeing the application from waiting for a response prior to continuing its processing. Simulating a definite response environment for these types of applications may affect the program's attempt to improve its performance or possibly create protocol problems in the network. Another alternative for collecting response times on exception response applications may be to force the application to run in definite response mode by altering the VTAM LOGMODE entry for the terminal. If the entry is defined to indicate that the terminal is unable to run in exception response mode, the application will be forced to run in definite response mode. This approach would solve the potential protocol problems in the network, but not the possible application performance concerns.

The collection of performance measurements is controlled by the NPM operator.

Volume measurements are the number of bytes contained in the inbound and outbound PIUs. This measurement is further subdivided into the number of system and user data bytes for both inbound and outbound PIUs. System data are defined as the PIUs that are involved in session control on behalf of the user and that may be sent by VTAM, the application, or hardware microcode. User data are defined as the PIUs involved in processing the customer's actual request and include the PIUs sent by either LU partner.

The collection of performance measurements is controlled by the NPM operator. The NPM operator designates the resource and collection options through session collection commands that are issued automatically during NPM initialization or manually through an NPM on-line panel. Once collection is started on a resource, the data are gathered from VTAM, processed by the NPM session

collection routines, and written to files for later analysis.

The collection of performance measurements may be started on a resource automatically or manually. By coding a SESSCOLL command in the NPM command data set, session collection will automatically be started during NPM initialization. If additional collections or option changes are needed, they may be made manually through the on-line facility of NPM. The facility can be accessed directly from the local NPM or indirectly from a remote NPM running elsewhere in the network. Also available is a console command interface that permits session collection commands to be issued from the system console. This provides an additional manual means of starting collection when used directly and automation possibilities when used through products such as NetView.

NPM provides a function that permits the dynamic addition, replacement, and deletion of network control program (NCP) resource definitions to NPM. This function enhances session collection on terminals associated with a given NCP by eliminating the manual and operational effort associated with redefining an NCP to NPM. The function is implemented through an NPM EXEC command, which can be issued automatically or manually. By coding an NPM EXEC command in the NPM command data set, the command will be automatically issued during NPM initialization. The console command interface also supports the NPM EXEC command, thus permitting it to be issued directly by a console operator or indirectly through products such as NetView. Additionally if session collection was active for resources on an NCP that is to be replaced, NPM will automatically restart collection on those resources.

Once collection has been started, the performance measurements are gathered at the VTAM API, processed in accordance with the collection options, and written to files for use during later analysis. Records regarding the terminal are referred to as detail records. A detail record is written for each application with which the terminal has been in session during a given time period. Records regarding the application are referred to as summary records. They summarize all terminal sessions for the application in a given time interval.

Beginning with NPM 1.5 and VTAM 3.4.1, NPM will utilize a new methodology to gather session performance measurements. From a functional standpoint, this solution is primarily an internal replacement of the NPM session collection command processing and data gathering routines. These new routines are very efficient. Early performance runs have indicated the new internal processing reduces CPU time, reduces common storage area usage, and favorably influences the overall network throughput when compared to previous releases of NPM.

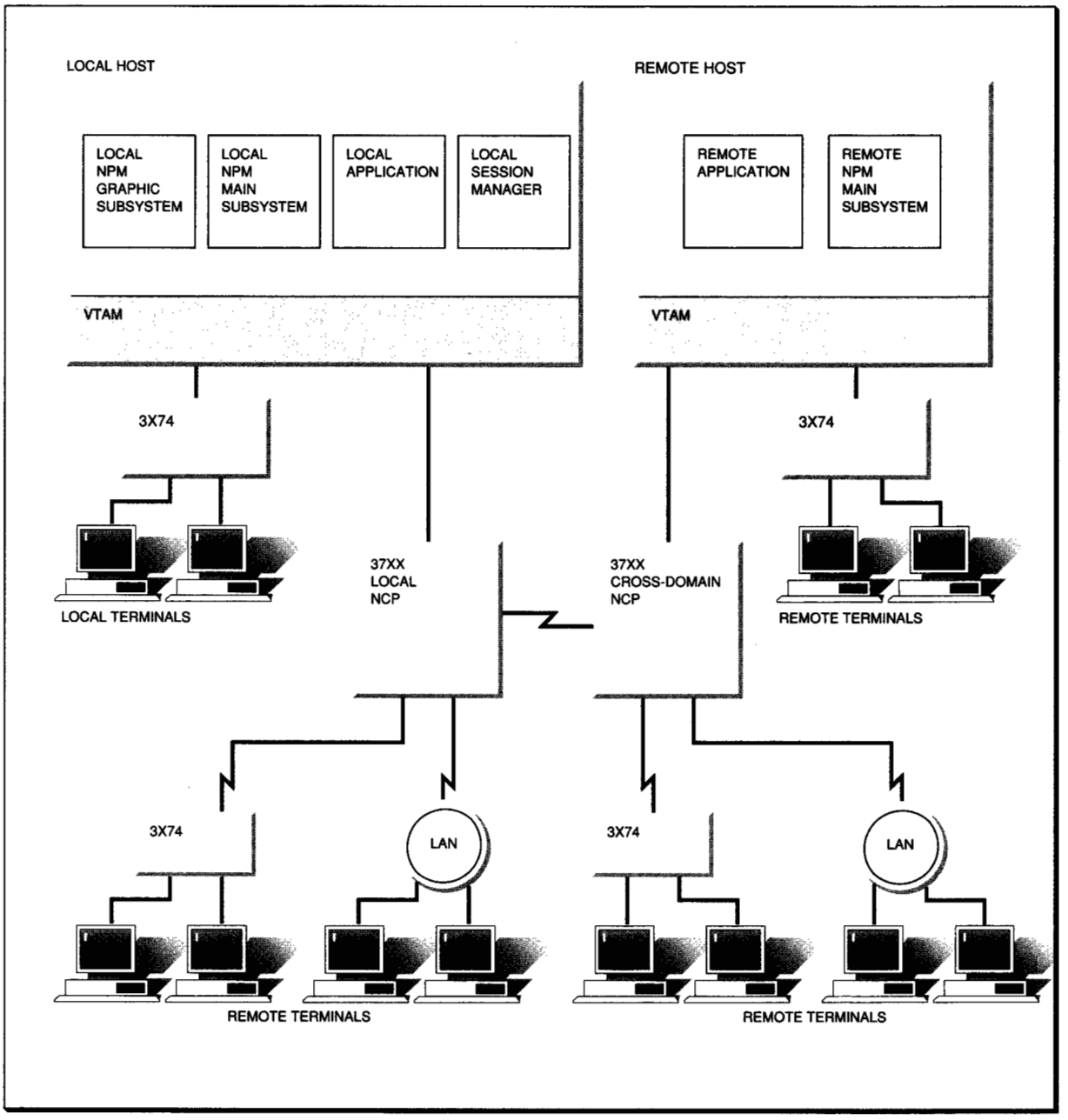
The new methodology of NPM resolves the usability, compatibility, and serviceability problems associated with its previous use of the Buffer Trace in VTAM to gather session statistics. The usability problem was a result of mutually exclusive use of the Buffer Trace. Since NPM was using the Buffer Trace in VTAM to gather session performance measurements, customers were unable to conveniently use it to gather diagnostic traces in relation to problems associated with hardware and software. The compatibility problem ensued from customers matching specific releases of NPM with specific releases of VTAM. Previously, NPM was internally sensitive to the release level of VTAM and would not function with releases it did not recognize. The serviceability problem centered on common processing by NPM and VTAM. Often a problem in one product could manifest itself as a problem in the other product.

NPM will utilize this new methodology to control and gather session statistics for all releases of VTAM that support the data collection enhancement. NPM will continue to use its present Buffer Trace approach for releases of VTAM that do not support the enhancement.

Collection options

Identifying the appropriate resource and associated collection options for the actual network management task at hand is crucial. Typical network management tasks associated with session performance include problem determination, new product evaluation, service-level monitoring, and capacity planning. Each task requires varying types and amounts of session data from differing combinations of physically and logically related sessions. These requirements should be used in the development of a session collection strategy

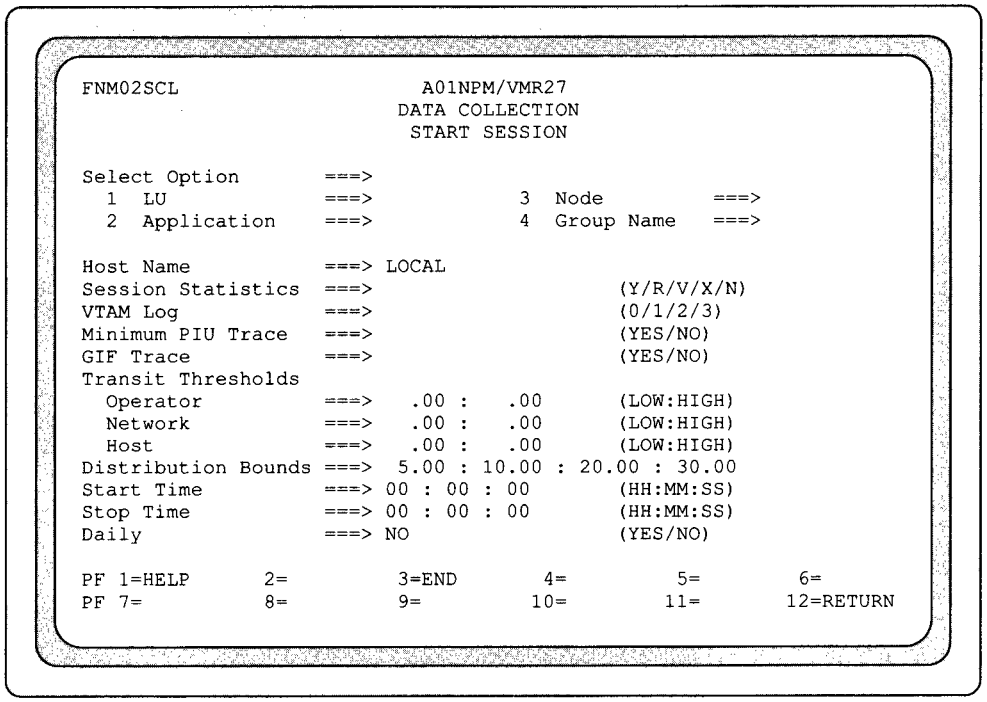
Figure 2 The session collection environment



to collect the specific information needed for later analysis and reporting.

NPM can collect session data on a variety of resources, as illustrated in Figure 2. The specific resource may be an individual entity, a group of physically related entities, or a group of logically related entities. An individual entity may be a terminal or an application. A group of physically related entities may be all the terminals associated with an NCP. A group of logically related entities may be a group of customer-defined ter-

Figure 3 The start session collection panel



```
FNM02SCL                    A01NPM/VMR27
                          DATA COLLECTION
                           START SESSION

Select Option          ===>
  1  LU                ===>            3  Node          ===>
  2  Application       ===>            4  Group Name    ===>

Host Name              ===> LOCAL
Session Statistics     ===>                     (Y/R/V/X/N)
VTAM Log               ===>                     (0/1/2/3)
Minimum PIU Trace      ===>                     (YES/NO)
GIF Trace              ===>                     (YES/NO)
Transit Thresholds
  Operator             ===>   .00 :   .00       (LOW:HIGH)
  Network              ===>   .00 :   .00       (LOW:HIGH)
  Host                 ===>   .00 :   .00       (LOW:HIGH)
Distribution Bounds    ===>  5.00 : 10.00 : 20.00 : 30.00
Start Time             ===> 00 : 00 : 00        (HH:MM:SS)
Stop Time              ===> 00 : 00 : 00        (HH:MM:SS)
Daily                  ===> NO                  (YES/NO)

PF 1=HELP      2=          3=END       4=          5=         6=
PF 7=          8=          9=          10=         11=        12=RETURN
```

minals and applications. As a general rule of thumb, NPM can collect performance measurements on any resource known to VTAM.

Session data will be collected when a selected resource is involved in a session regardless of whether collection is active for the partner resource. For example, if collection is started on an individual terminal, session statistics will be collected for all application sessions in which the selected terminal is involved. Likewise if collection is started on an application, session statistics will be collected for all terminal and application sessions in which the selected application is involved.

As illustrated in Figure 3, the basic resource types are an LU, an application, a node, and a group. Each resource type along with a possible reason for selection is now described.

An LU selection provides the finest collection granularity possible and represents a single terminal-to-application session. The LU may be a local terminal to either the local host or a remote host. It may also be a remote terminal attached to a local NCP or to a remote NCP. Selection of a specific LU would be typical in a specific problem determination or a customer help desk situation. After determining the LU name of the problem terminal, session performance measurements would be started to gather the information needed to resolve the specific problem.

The application selection provides a comprehensive measurement of all terminals that go into session with an application. Since NPM gathers performance data from the local VTAM, the application must reside on the same host as does NPM and VTAM. However, there is one exception. If a cross-domain application is accessed through NetView Access Services or an equivalent session manager that resides on the local host with NPM and VTAM, NPM is able to provide performance measurements through a feature called Session Manager Support. This feature permits an end-to-end view of a terminal that accesses local or cross-domain applications through a session manager. Selection of an application is especially applicable to service-level monitoring of

production applications, to new product evaluations, and to capacity planning. Although the application selection would probably not be the first collection started in a problem determination situation, it may be the logical second step.

Node selection provides a comprehensive measurement of all terminals associated with a given node. A node is typically an NCP and may be either a local NCP or a remote NCP. Since a given NCP may have thousands of terminals associated with it, this option should be used with discretion. If performance measurements are actually desired for every terminal associated with a specific NCP, this selection would be a quick way to start the collections. If performance measurements on a scale that is this large were not intended, NPM would probably fill its data sets in short order for no specific purpose. A more common use of the node selection is in conjunction with the LU selection. By specifying an LU name with a node name, the association between a local terminal and host can be made. This selection would be applicable to capacity planning and global performance. However, depending on the makeup of the network, this selection may also provide a convenient means to gather service-level information for a particular group or site.

Group selection provides the ability to provide performance measurements for a user-defined logical group of terminals. The members of the group may or may not be physically associated with any given application or node. This selection allows one to create representative samples of LUs for any given situation. The flexibility provided by this option makes it powerful and useful in a variety of management tasks. Selection of a group is especially applicable to conservatively provide service-level and capacity planning information. It can also be used for problem determination by providing a quick and easy way to create a sample of logically related LUs.

A wide variety of collection options are available for all the resource types. These options control the type of processing NPM will perform for a session with a given resource.

The host name option provides the ability to route a start command to any NPM in the network. This option can be left to default to the local NPM or changed to start a collection on any remote NPM. This option is particularly useful in starting collections on remote applications. By indicating the remote host name, the command can be routed to the remote NPM for execution without logging off the current NPM and logging on to another NPM.

The session statistics option controls the type of session performance measurements that NPM will collect. In situations where transaction counts, transit times, and volume measurements are desired, a "Y" or an "R" would be specified depending on the response mode of the application. The "Y" would be used for definite response applications, whereas the "R" would be used for exception response applications. For most situations, one of these indicators would be chosen. In situations where only volume measurements are desired, a "V" would be specified. The volume only indication is typically used in pure throughput types of analysis where one is evaluating the routing or cost effectiveness capabilities of hardware. The "X" indicator would cause NPM to exclude this resource from collection. This indicator is typically used to exclude devices, such as printers, when collection is started on an application. The "N" indicator would cause NPM to also exclude this resource from collection, but allow PIU tracing. This indicator would be used in a diagnostic situation where a full PIU trace of a session is wanted, but the performance measurements are not.

The VTAM log, minimum PIU trace, and generalized trace facility (GTF) trace options control the writing of PIUs for diagnostic purposes. The VTAM log option determines whether inbound, outbound, inbound and outbound, or no PIUs should be written to the NPM VTAM log data set. For performance management activities, the option would be set to not collect PIUs. The minimum PIU trace option indicates whether a complete or partial copy of the PIU should be written to the NPM VTAM log. It should be noted that NPM has the ability to trace the full PIU as opposed to other tracing facilities in which there are size limitations. After the PIUs are traced, they can be formatted and printed with an NPM batch utility program. This utility program, FNMTAP, can provide transit time summary reports by transaction or interval and can format TH, TH4, RH, RU, 3270 RU orders and data, and NPALU RU data fields. This ability can be invaluable in diagnostic situations. In general though, these options should only be specified in problem determination situations.

Figure 4 The resource summary panel

```
FNM03SMN                    A01NPM/VMR27
                            DATA ANALYSIS
          NPM SESSION ANALYSIS SUMMARY - APPLICATION


Command ===>
HOST        = LOCAL           RECORDS =          2      VR/TP =  /
APPLICATION= TSO01               LINE =                    PU =
GROUP       =                    NODE =                    LU =
Requested from   91/09/13  07:20:00  to  91/09/13  24:00:00
Actual    from   91/09/13  07:23:47  to  91/09/13  07:24:49
                          TOTAL       RESPONSE
Transactions                 24             24
                         OPERATOR           HOST           NETWORK
Average Transit               .20            .20               .00
Maximum Transit              2.62           2.61               .01
                        PIU COUNT       AVG SIZE       TOTAL BYTES
User Data In                 26              9               236
User Data Out                27            300              8101
System Data In               26              3                78
System Data Out               0              0                 0

PF 1=HELP      2=LIST     3=END      4=         5=        6=
   7=          8=         9=        10=        11=       12=RETURN
```

These data are not used by NPM for performance monitoring.

The transit thresholds and distribution bounds options control the grouping of transactions. High and low thresholds can be set for operator, network, and host times. These defined thresholds are later used to determine how the actual transit times compare with these predefined thresholds. Four distribution bounds can be set for operator time. The bounds are later used to group and compare actual operator times. These two options are useful for service-level monitoring, capacity planning, and problem determination activities. By setting the thresholds and distribution in accordance with site-specific service-level agreements, past results, or anticipated results, one can later analyze how the actual times compare to these specified limits.

The start time, stop time, and daily options control when collection will be active. If left to the default, collection will begin immediately and continue until NPM is stopped. However, in situations where performance measurements are only required during specific times, these options can provide a means to automatically start and stop collection at those times. These options are particularly useful in managing production environments where specific performance measurements are needed for a given period of time each day.

Correlation schemes

The session performance measurements can be analyzed from either an application or a terminal viewpoint using a variety of correlation schemes. Each correlation scheme represents a specific association and provides an extensive list of qualifiers to create global, physical, logical, and specific samples for analysis. The basic correlation schemes include:

- Resource-summary-by-time
- Transit-time-by-time
- Volume-by-time
- Transit-time-by-volume
- Transit-time-exceeding-threshold-by-time

Figure 5 The transit-time-by-time panel

```
FNM03TI1                    A01NPM/VMR27
                            DATA ANALYSIS
                       TRANSIT TIME/INTERVAL

Command ===>
HOST = LOCAL      LU = A01A704
          OPERATOR  HOST/NET  TOTAL/RESP       FROM/TO        SESSION MGR
APPL       TRANSIT  TRANSIT      TRANS      DATE      TIME   USERID/SLU
NPM01          .08       .03         24  91/09/13 07:19:53    OPER1
                         .05         24           07:21:59

TSO01          .35       .20         11  91/09/13 07:23:29      LEO
(RELAY)                  .15         11           07:23:47      EMSYAT01

EMSYAS01       .21       .20          1  91/09/13 07:23:05      LEO
(CONTROL)                .01          1           07:23:48

TSO01          .50       .20         13  91/09/13 07:17:49      USER1
                         .30         13           07:24:49



PF 1=HELP       2=          3=END        4=          5=VOLUME  6=DISTRIB
PF 7=BACKWARD   8=FORWARD   9=SUMMARY   10=TOP      11=BOTTOM 12=RETURN
```

The resource-summary-by-time correlation provides a complete summary of all performance measurements within a specified time. Resource summary correlations are available for either an application or a terminal. Figure 4 illustrates an application summary. The display represents a composite of all terminals that were in session with this application during the specified period of time. It is actually a display of the application summary records that were produced by totaling and averaging all of the individual sessions. A similar summary could be produced for a given LU or terminal. However, in this case the display would represent all application sessions that the terminal had in a specified period of time. A terminal summary is produced by totaling and averaging all of the individual detail records. Each detail record in turn represents a specific terminal-to-application session.

Summary correlations contain transaction counts, transit times, and volume measurements. The total transaction count is indicative of the work completed during this period of time. Although it is not known whether or not the customer wished to complete more transactions, it does provide a measurement of the actual amount of work completed. The total response count indicates whether or not the application is running in exception or definite response mode. If the response count is zero, the network times are also zero, and the operator time is only representative of the host time. The average operator time represents the total average amount of time taken to complete one transaction. This number will be the sum of the associated average network and host time. These averages can provide a quick indication of whether or not the resource is within its anticipated levels. The average host and network time provide a breakdown of the time that the transaction spent in the application as opposed to that spent in the network. It must be understood that these are averages, and the number may be skewed by several extremely high values. For this reason the maximum operator, host, and network times are also displayed.

The PIU count, average size, and total bytes represent the volume of network traffic that was generated to complete the total number of transactions. These numbers can be used to assess the

composition of the interaction between the terminal and the application. A definite response application uses at least three PIUs per transaction as opposed to at least two PIUs for exception response applications. The number above this amount will typically be found in the user data-out category. Applications typically can be classified by the volumes they generate. In general, the volume characteristics of an application affect its transit times.

The transit-time-by-time correlation provides the detailed view of transit times and transaction counts within a specified period of time. This correlation is available for either an application or a terminal. Figure 5 illustrates a transit time correlation display for a specific terminal. A similar correlation could be produced for an application. The panel is a display of all LU detail records associated with the terminal in a specified period of time. Each detail record represents a specific session during a collection interval. The length of a collection interval is customer-defined. Short intervals produce more records and finer granularity than long intervals. Short intervals are better-suited to problem determination situations. In contrast, long intervals are better-suited to service-level monitoring and capacity planning.

The transit-time-by-time correlation contains transit times and transaction counts. Additionally, in the terminal correlation display, session type and user identifiers are provided. The operator, host, and network times represent averages for the total number of transactions in a given collection interval. The total transaction count represents the actual number of transactions completed in the collection interval. The total responses indicate whether or not the session was run in definite or exception response. As in the resource summary correlation, if the response count is zero, the network time will be zero, and the operator time will only represent host time. For sessions created through NetView Access Services or an equivalent session manager, the session type and user identifier will also be shown. A session type of CONTROL indicates a session between an end terminal and the session manager. A session type of RELAY indicates a session from an end terminal to an application that is passing or relaying through a session manager. In addition to displaying the user's identifier, relay session entries also display the session manager's SLU name that was used to establish the session with the end application.

The transit-time-by-time correlation can be used to identify trends, peak usage periods, and transaction rates of applications. Over a period of days, weeks, or months, trends will become evident. Peak usage in terms of time of day and time of year can be identified. Likewise transaction rates can be monitored and tracked.

The volume-by-time correlation provides a detailed view of the type and amount of PIUs that were generated in a given period of time. Like the summary and transit time correlation, this one is also available for either an application or a terminal. The correlation is made from the same LU detail records that the transit time correlation used. The correlation represents the volume details that are associated with the transit time details. In situations where only volume statistics are needed, a medium-to-long interval will provide sufficient detail, save processing time, and require less disk space.

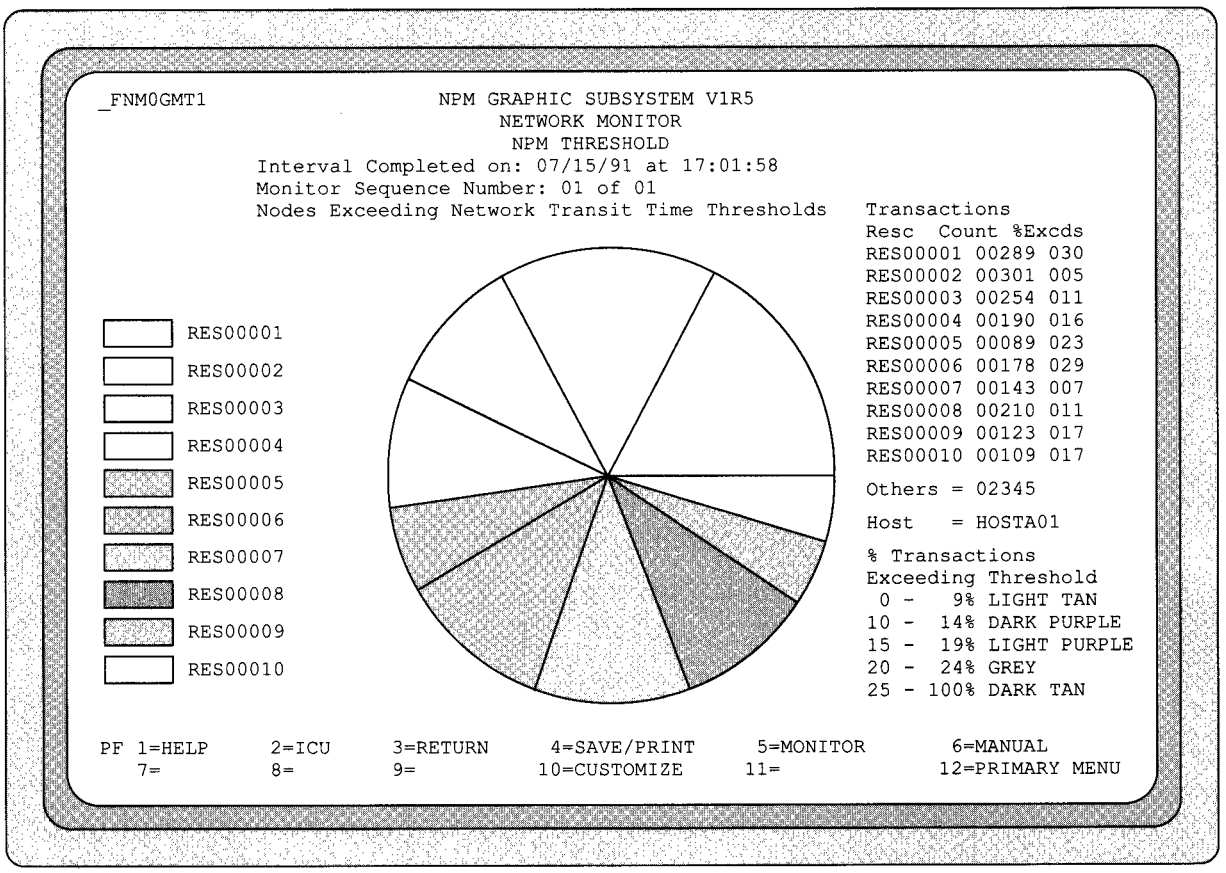
The volume-by-time correlation is particularly suited to capacity planning and route selection activities. The details from this correlation may indicate a bottleneck in a specific path through the network. They may also indicate whether or not the underlying hardware is capable of providing sufficient throughput for the application.

The transit-time-by-volume correlation provides a representation of how transit time is affected by volume for a specified period of time. As in the previous correlations, this one is also available from either an application or a terminal viewpoint. This correlation combines the information from the transit time and volume correlation to create another way of viewing the performance measurements.

The transit-time-by-volume correlation can provide a variety of subcorrelations. This correlation permits designation of the specific component of both transit time and volume. Transit time can be represented by the operator, host, or network component. Similarly, volume can be represented by PIUs, bytes, or transactions. Based on the specific components selected, the same basic correlation can be represented from slightly different perspectives.

The transit-time-by-volume correlation can be used to determine specific relationships and their

Figure 6 Graph of network transit time exceeding thresholds



Nodes exceeding operator transit time graph (FNM0GMT1)

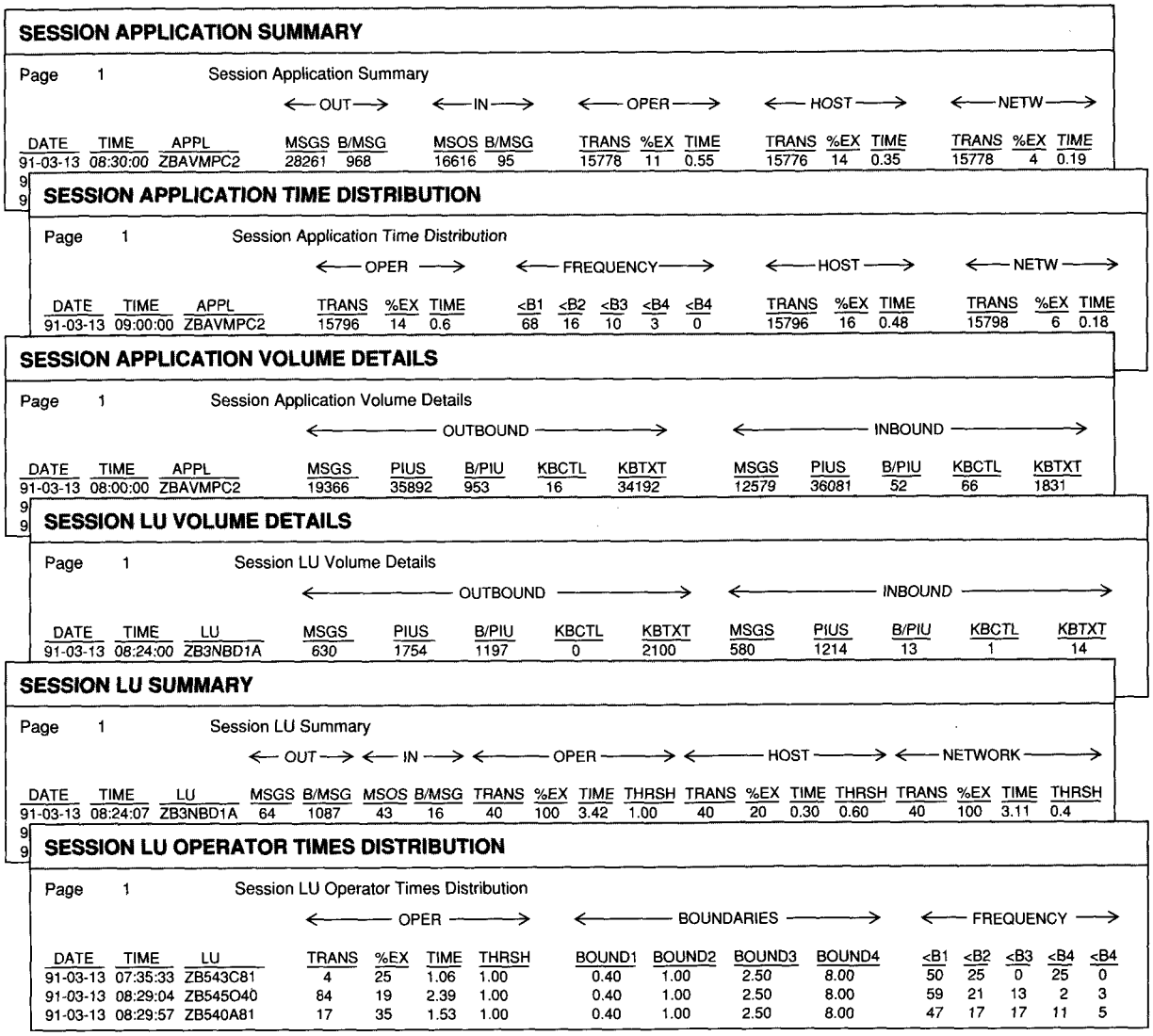
characteristics. All things being equal, one would expect volume to negatively affect transit time. However, in any given environment, all things may not be equal, and some interesting relationships may be uncovered. For example, if the volume does not seem to affect transit time, this condition would indicate that the application or the network path, or both together, can handle extremely large volumes without a deterioration of transit time.

The transit-time-exceeding-threshold-by-time correlation provides a detailed view of the percent of transactions that exceed their high threshold. Figure 6 illustrates a graph of network time exceeding the thresholds. A similar graph could be made for operator and host time. This correlation permits multiple resources to be graphed together. The graph contains the resource name, transaction count, and percent of transactions exceeding their transit time threshold. The information for this graph can be gathered as it is currently being collected or from previously collected data.

The transit-time-exceeding-threshold-by-time correlation is especially well-suited for monitoring active resources. By setting the data source to the currently active resources, the graph will be continually updated and change color, depending on the percent of transactions that exceed their transit time. This facility may serve to inform help desk personnel of a problem before their customers call to report it.

The correlation schemes have been implemented in a flexible manner. Through the use of qualifiers,

Figure 7 Batch reports



SESSION APPLICATION SUMMARY

Page 1 Session Application Summary

			← OUT →		← IN →		← OPER →			← HOST →			← NETW →		
DATE	TIME	APPL	MSGS	B/MSG	MSOS	B/MSG	TRANS	%EX	TIME	TRANS	%EX	TIME	TRANS	%EX	TIME
91-03-13	08:30:00	ZBAVMPC2	28261	968	16616	95	15778	11	0.55	15776	14	0.35	15778	4	0.19

SESSION APPLICATION TIME DISTRIBUTION

Page 1 Session Application Time Distribution

			← OPER →			← FREQUENCY →					← HOST →			← NETW →		
DATE	TIME	APPL	TRANS	%EX	TIME	<B1	<B2	<B3	<B4	<B4	TRANS	%EX	TIME	TRANS	%EX	TIME
91-03-13	09:00:00	ZBAVMPC2	15796	14	0.6	68	16	10	3	0	15796	16	0.48	15798	6	0.18

SESSION APPLICATION VOLUME DETAILS

Page 1 Session Application Volume Details

			← OUTBOUND →					← INBOUND →				
DATE	TIME	APPL	MSGS	PIUS	B/PIU	KBCTL	KBTXT	MSGS	PIUS	B/PIU	KBCTL	KBTXT
91-03-13	08:00:00	ZBAVMPC2	19366	35892	953	16	34192	12579	36081	52	66	1831

SESSION LU VOLUME DETAILS

Page 1 Session LU Volume Details

			← OUTBOUND →					← INBOUND →				
DATE	TIME	LU	MSGS	PIUS	B/PIU	KBCTL	KBTXT	MSGS	PIUS	B/PIU	KBCTL	KBTXT
91-03-13	08:24:00	ZB3NBD1A	630	1754	1197	0	2100	580	1214	13	1	14

SESSION LU SUMMARY

Page 1 Session LU Summary

			← OUT →		← IN →		← OPER →				← HOST →				← NETWORK →			
DATE	TIME	LU	MSGS	B/MSG	MSOS	B/MSG	TRANS	%EX	TIME	THRSH	TRANS	%EX	TIME	THRSH	TRANS	%EX	TIME	THRSH
91-03-13	08:24:07	ZB3NBD1A	64	1087	43	16	40	100	3.42	1.00	40	20	0.30	0.60	40	100	3.11	0.4

SESSION LU OPERATOR TIMES DISTRIBUTION

Page 1 Session LU Operator Times Distribution

			← OPER →				← BOUNDARIES →				← FREQUENCY →				
DATE	TIME	LU	TRANS	%EX	TIME	THRSH	BOUND1	BOUND2	BOUND3	BOUND4	<B1	<B2	<B3	<B4	<B4
91-03-13	07:35:33	ZB543C81	4	25	1.06	1.00	0.40	1.00	2.50	8.00	50	25	0	25	0
91-03-13	08:29:04	ZB545O40	84	19	2.39	1.00	0.40	1.00	2.50	8.00	59	21	13	2	3
91-03-13	08:29:57	ZB540A81	17	35	1.53	1.00	0.40	1.00	2.50	8.00	47	17	17	11	5

the correlation sample can be altered to include or exclude specific records. These qualifiers consist of date, time, application, line, physical unit, virtual route, and transmission priority.

Presentation formats

NPM can present the correlation schemes through on-line statistical panels, on-line graphical panels, and batch reports. Each environment has unique characteristics and processing options.

The NPM on-line system provides a menu-driven set of on-line session collection and analysis panels. These panels permit performance measurements to be started, stopped, altered, and correlated. From the session analysis panels, correlation schemes and their corresponding options can be selected. The data are retrieved from Virtual Storage Access Method (VSAM) files and displayed in standard tabular formats. These files typically contain the most recently collected session performance measurements but may in fact be from any point in

time. Although screen prints of the analysis panels can be made, the on-line system does not have a formal reporting facility.

The versatility of the NPM on-line system makes it well-suited for problem determination, service-level monitoring, and new product evaluation. If historical files are also defined to the NPM on-line system, these data can be correlated and analyzed for capacity planning purposes.

The NPM graphics subsystem provides a menu-driven set of on-line analysis panels. These panels permit performance measurements to be correlated, formatted, and printed in a variety of graphical representations. The graphics subsystem can access currently collected data as well as previously collected data. It can provide dynamically updated displays as well as static displays. The graphics subsystem, however, cannot start, alter, or stop the collection or performance measurements.

The correlation flexibility of the graphics subsystem makes it well-suited to capacity planning and service-level monitoring. When used in conjunction with the NPM on-line system, it provides a convenient way to monitor transit times.

FNMREPRT is a batch reporting utility. As illustrated in Figure 7, this utility is able to produce a variety of standard reports on session performance statistics. These standard reports may be customized. The batch utility has no interfaces or dependencies on the NPM on-line system. This independence permits reports on session performance statistics to be produced without having to dedicate terminals and operators for the duration of the task.

The batch reporting function analyzes previously collected session performance statistics. The session statistics collected from the NPM on-line system can be archived and later used as input to the batch reporting utility.

A variety of session performance statistics report formats are available. These reports provide session performance statistics from both an application and LU perspective. They further provide reporting capabilities at the detail and summary level.

All reports have customization options. Formatting options include modifications to the title, page size, line size, and actual report. Resources can be included and excluded using standard Boolean expressions.

Summary

The NetView Performance Monitor has a flexible set of facilities that can be used to provide the information needed to manage session performance. These facilities provide an extensive means to collect, correlate, and present session performance measurements.

Session performance measurements consist of transaction counts, transit times, and volume measurements. Automatic and manual options are provided to selectively start, stop, and alter the collection of session performance measurements. When needed, these same facilities can also record the actual PIU flow between session partners.

A variety of correlation schemes are available for both applications and terminals. The basic correlation schemes include: resource-summary-by-time, transit-time-by-time, volume-by-time, transit-time-by-load, and transit-time-exceeding-threshold-by-time.

The correlation schemes can be presented through on-line statistical panels, on-line graphical panels, and batch reports. These flexible facilities can provide the information needed to resolve problems, evaluate new products, monitor service-level agreements, and plan for the future.

*Trademark or registered trademark of International Business Machines Corporation.

General references

IBM Programming Announcement, 291–295 (June 6, 1991); available through IBM branch offices.

NetView Performance Monitor at a Glance, SH20-6359, IBM Corporation; available through IBM branch offices.

NetView Performance Monitor Graphic Subsystem Manual, SH20-6362, IBM Corporation; available through IBM branch offices.

NetView Performance Monitor Installation and Customization Manual, SH20-6361, IBM Corporation; available through IBM branch offices.

NetView Performance Monitor Operation Manual, SH20-6360, IBM Corporation; available through IBM branch offices.

NetView Performance Monitor Reports and Record Formats Manual, SH31-6147, IBM Corporation; available through IBM branch offices.

NetView Performance Monitor Version 1 Release 4.1 and Release 5 New Functions, GG24-3681, IBM Corporation; available through IBM branch offices.

System Network Architecture Formats Manual, GA27-3136, IBM Corporation; available through IBM branch offices.

VTAM Programming Manual, SC31-6409, IBM Corporation; available through IBM branch offices.

Accepted for publication December 11, 1991.

Leo Temoshenko *IBM Networking Systems, 4205 S. Miami Boulevard, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Temoshenko is currently a staff programmer at the IBM Networking Systems Programming Laboratory. Previously he was a member of the NPM development team from 1989 to 1991. During that time he received an IBM Outstanding Technical Achievement Award for his work on NPM Release 4, an Invention Achievement Award for his work on NPM Release 5, and several patents related to session performance management. Prior to joining IBM in 1989, he was employed as an application and system programmer. Mr. Temoshenko holds a B.S. in education from Slippery Rock University and an M.S. in information science from the University of Pittsburgh.

Reprint Order No. G321-5474.