35-3104

Performance Measurement of OSI Class 4 Transport Implementations

Kevin L. Mills
Jeff W. Gura
C. Michael Chernick

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Sciences and Technology
Systems and Network Architecture Division
Gaithersburg, MD 20899

January 1985

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NBSIR 85-3104

PERFORMANCE MEASUREMENT OF OSI CLASS 4 TRANSPORT IMPLEMENTATIONS

Kevin L. Mills
Jeff W. Gura
C. Michael Chernick

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Sciences and Technology
Systems and Network Architecture Division
Gaithersburg, MD 20899

January 1985

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

PERFORMANCE MEASUREMENT OF OSI CLASS 4 TRANSPORT IMPLEMENTATIONS

Kevin L. Mills
Jeff W. Gura
C. Michael Chernick

A measurement system to evaluate the performance of open system interconnection (OSI) transport protocol implementations is described. Several metrics are proposed to establish a quantitative characterization of layered protocol performance. Metrics specific to the OSI transport protocol are also proposed. The measurement system and metrics were applied to a multi-vendor National Computer Conference demonstration network and the results are reported.

I. Introduction

Over the past four years, the international standards community has begun to reach a consensus on protocols for the first four layers of the open system interconnection (OSI) reference model [1]. A key protocol within the OSI model is the end-to-end transport protocol [2]. This protocol has reached the status of a recommended international standard within the International Organization for Standardization (ISO) and is a 1984 CCITT recommendation. The National Bureau of Standards (NBS) has been an important contributor to the design, specification, and correctness testing of the most robust class of the OSI transport protocol, class 4. This paper reports the results of some additional work of the NBS concerning performance measurement of class 4 transport protocol implementations.

At the 1984 National Computer Conference (NCC), nine vendors demonstrated interoperability over an IEEE 802.3 (CSMA/CD) local area network using class 4 transport as the end-to-end protocol. The implementation strategies used ranged from integration of transport and CSMA/CD protocols on a single board to partitioning of the protocol layers across multiple computer systems. Another group of six computer vendors demonstrated interoperability over an IEEE 802.4 (token bus) local area network also using class 4 transport as the end-to-end protocol.

The NBS performed a major role in preparing these demonstrations by hosting a series of workshops detailing the specific features of the class 4 transport protocol to be used in the demonstrations and explaining NBS-developed test methods to ensure vendor interworking prior to

to the NCC [3-7]. In addition, the NBS provided a performance measurement node on the CSMA/CD network. This paper describes the structure of the NBS developed protocol performance measurement system (Section II); describes the measures that the system can make (Section III); and reports several results obtained through application of the measurement system during the pre-NCC vendor preparation period (Section IV).

II. Measurement System Structure

An important design requirement for the measurement system is to provide the ability to make performance measurements for a variety of operational modes, including: (1) real-time measurement experiments with off-line data analysis, (2) real-time measurement demonstrations with graphic displays, and (3) off-line analysis of data traffic collected in disk files. An additional design requirement, for the measurement system,is to work within a variety of environments, including: (1) as a node on a CSMA/CD network, (2) as a process on the NBS test center host, and (3) as an independent process within a time-sharing system.

These requirements led to a decision to structure the measurement system as three independent subsystems: (1) data collection, (2) measurement, and (3) analysis and display. The general relationships between these three subsystems are illustrated in Figure 1. For a number of reasons including portability, availability, and ease of development, the subsystems are implemented in the C language as three separate UNIX (TM AT&T

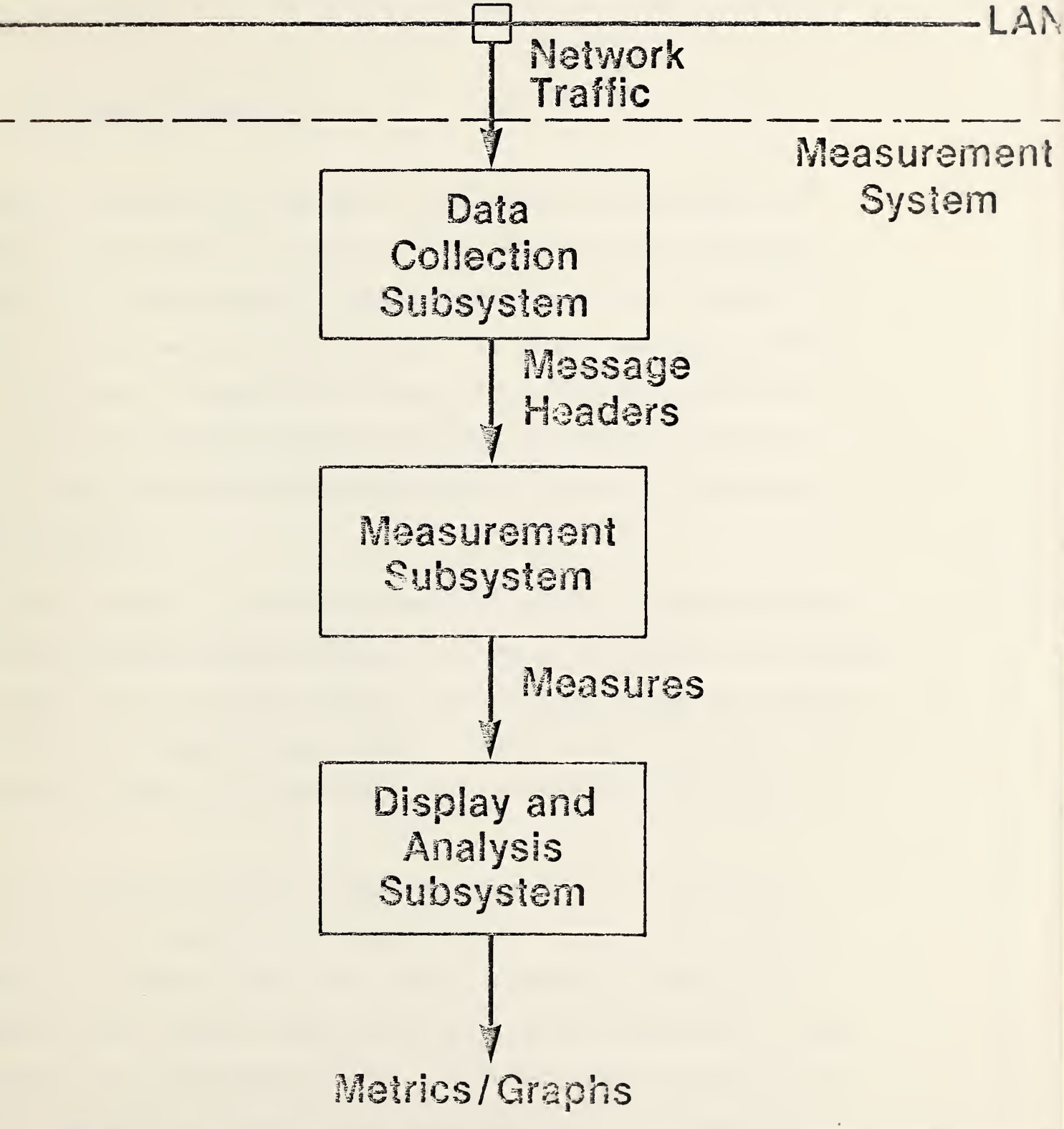
LAN
Network
Traffic
Measurement
System
Data
Collection
Subsystem
Message
Headers
Measurement
Subsystem
Measures
Display and
Analysis
Subsystem
Metrics/Graphs

FIGURE 1

Bell Laboratories) processes. The following paragraphs describe the measurement methodology used and role of each subsystem in implementing the methodology.

A. Measurement Methodology

Several approaches have been used to make communications protocol performance measurements. For example, at the NBS, a network measurement machine has been developed for measuring network service at the application level as seen by the user [10-12]. The network measurement machine is programmed to recognize user dialogue with a time-sharing system and all measures and metrics are user-oriented. This method is passive in that making the measures does not degrade the performance of the system being measured

A second approach is to measure protocol performance by instrumenting the transport service interface within a host computer system [13]. This permits accumulation of an accurate picture of the transport service experienced by an application program using a network. The measurement system need only recognize requests for transport service from an application program.

A third approach to protocol performance measurement is to internally instrument the communications programs within the hosts and/or switching nodes of a network [14-16]. This requires a detailed knowledge of all communications software involved and entails careful consideration of the possibility of measurement artifact. The measures that are obtained are very detailed but are specific to the particular software implementation strategy used.

A fourth approach to protocol performance measurement is to monitor peer to peer protocol exchanges on network links. This method has been proven to work in a commercial environment for making measures of response time, throughput, down time, line utilization, and traffic distribution [17]. Each line to be monitored is tapped so that traffic in both directions can be monitored without affecting the performance of the communicating systems. To use this method, the protocol being operated over the link must be recognized by the monitor.

The methodology adopted by the NBS and reported here is based upon passively monitoring peer to peer protocol exchanges on a LAN. Details of the protocol encoding recognized by the NBS protocol performance measurement system are provided below.

B. Encodings - Protocol Data Units

OSI packets are known as protocol data units (PDUs) consisting of a header and, optionally, data. For each layer of the OSI model above the physical layer, separate PDU types are defined. The NBS protocol measurement system recognizes the link layer (LPDUs), network layer (NPDUs), transport layer (TPDUs), and transport user. The transport user sends messages known as transport service data units (TSDUs). Each TSDU is composed of an ordered set of one or more TSDUs. The measurement system always recognizes TPDUs and TSDUs and may, depending upon specific configuration, recognize LPDUs and NPDUs. The most general form of PDU encoding recognized is shown in Figure 2.

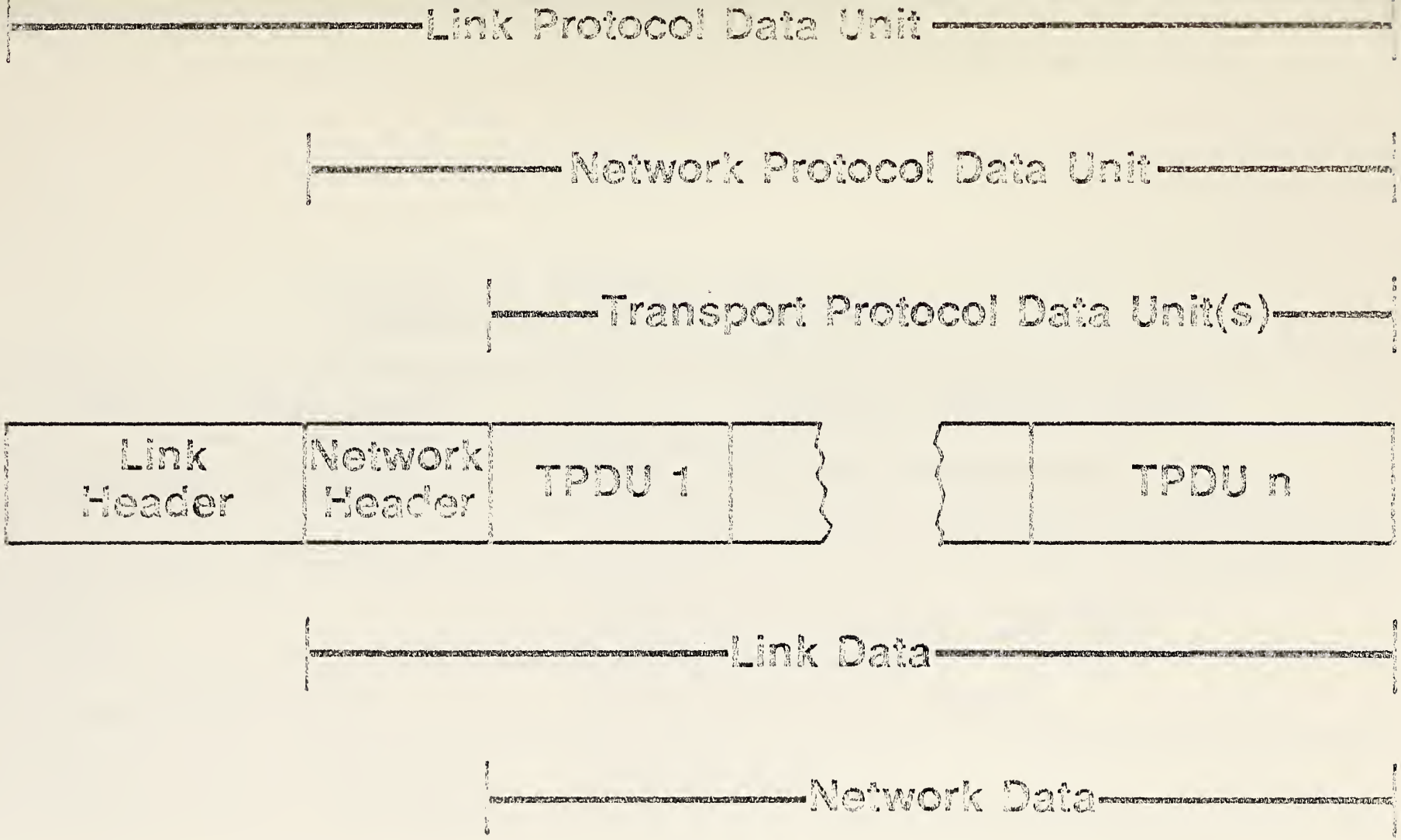

FIGURE 2

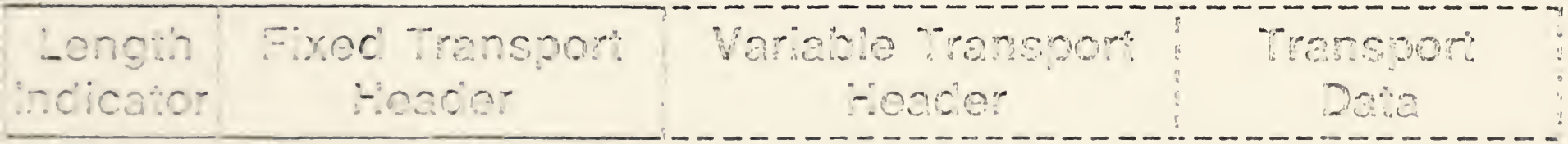

FIGURE 3

Table 1 TPDU Types

TPDUs with Data		TPDUs without Data	
CR	Connect Request	AK	Acknowledgement
CC	Connect Confirm	EA	Expedited Acknowledgement
DT	Data	DC	Disconnect Confirm
ED	Expedited Data	GR	Graceful Close
DR	Disconnect Request	ER	Error

On the demonstration LAN, a frame consists of one or more TPDUs nested within an NPDU itself nested within an LPDU. The LPDU consists of a link header and link data (i.e., NPDU plus TPDUs). The NPDU contains a network header and network data (i.e., TPDUs). Within the confines of the NCC demonstration, the link header had a fixed length of 17-bytes and the network header had a fixed length of 1-byte. The TPDUs used were of variable length and formatted as shown in Figure 3.

A TPDU is composed of a variable length header and, an optional, variable length data field. A TPDU header contains a length indicator, a fixed part, and a variable part. The length indicator describes the size of the TPDU header only and, thus, the length of the TPDU data must be inferred from information supplied by the network layer and from application of the transport protocol rules for concatenation of multiple TPDUs within a network protocol data unit.

The fixed part of the TPDU Header is present in every TPDU, but differs in size depending on TPDU type (see Table 1) and on negotiated options for a specific connection. For example, a DT TPDU always contains a sequence number, but the length of that field may be 1 or 4 bytes depending on the sequence number space agreed for the transport connection over which the DT TPDU flows. The size of the fixed part of a TPDU header is small, ranging from 4 to 9 bytes.

The variable part of the TPDU header is optional for each TPDU and varies in size depending upon the specific parameters encoded within it. Examples of parameters that may be carried in the variable part of a TPDU include a checksum, a subsequence number, connection establishment options and security parameters. In principle, the variable part of the TPDU header can be quite large because, for specific TPDU types, several options can be up to 255 bytes in length; however, for the NCC demonstration the field was used only for checksum, subsequence numbers, and flow control confirmation parameters ranging in size from 4 to 22 bytes.

The data portion of the TPDU is permitted only in CR, CC, DT, ED and DR TPDUs. For the demonstration, data was present only on DT and ED TPDUs. Data within an ED TPDU is limited to a maximum of 16 bytes. Data size within a DT TPDU is subject to negotiation on each transport connection and may range from 1 to 8192 bytes. For purposes of the NCC demonstration, actual data sizes were limited to 1496 bytes because no segmenting of NPDUs across multiple LPDUs was implemented and LPDUs were restricted to 1513 bytes.

The sections that follow describe the role of each subsystem in capturing and decoding the network traffic so that measurements may be made. The role of the subsystems in analysis is also described.

C. Data Collection

The data collection subsystem collects PDUs from the network, discards the TPDU data, time stamps each set of PDU headers, and provides a device-

like interface for use by the measurement subsystem to read PDU headers. Data collection on a local area network is illustrated in Figure 4. The node containing the data collection subsystem is operated in a mode such that all LPDUs may be read from the network. Using this technique, every LPDU between every host pair can be captured for measurement.

As illustrated in Figure 5, an alternate data collection mechanism is available for use when the measurement system is embedded as a process within the NBS test center host [8,9]. The exception generator process within the test center host is positioned as a data collection mechanism at the boundary between the transport and network processes. Every message (containing embedded TPDUs) crossing that interface can be captured by the exception generator. The exception generator logs these messages to a disk file and, optionally, can pass the messages directly to the measurement subsystem for real-time performance monitoring. This data collection mechanism is network independent so that measurement can be applied to the MILNET, the TELENET public data network, and IEEE 802.3 local networks.

Another data collection option is to capture on a disk file the messages and/or PDU headers. This information may be captured by the exception generator or by logging the output of the data collection subsystem of the performance measurement system to disk. This captured network traffic is then measured off-line as shown in Figure 6.

D. Measurement

The measurement subsystem: consumes the LPDU/NPDU/TPDU header; analyzes the protocol implications of each PDU; computes aggregate, host, and trans-

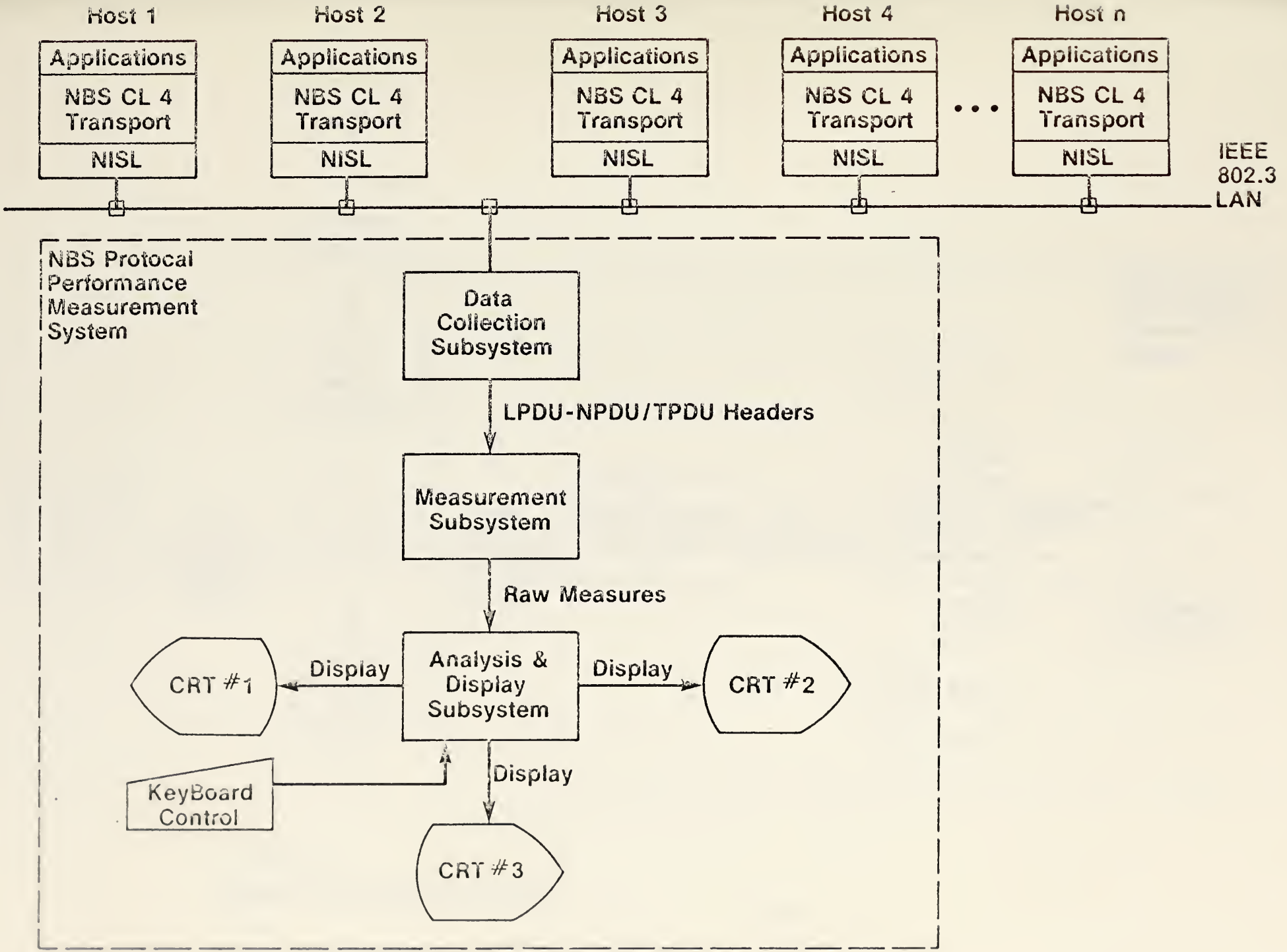
Host 1
Host 2
Host 3
Host 4
Host n
Applications
NBS CL 4 Transport
NISL
IEEE 802.3 LAN
NBS Protocal Performance Measurement System
Data Collection Subsystem
LPDU-NPDU/TPDU Headers
Measurement Subsystem
Raw Measures
Analysis & Display Subsystem
Display
CRT #1
CRT #2
CRT #3
KeyBoard Control

FIGURE 4

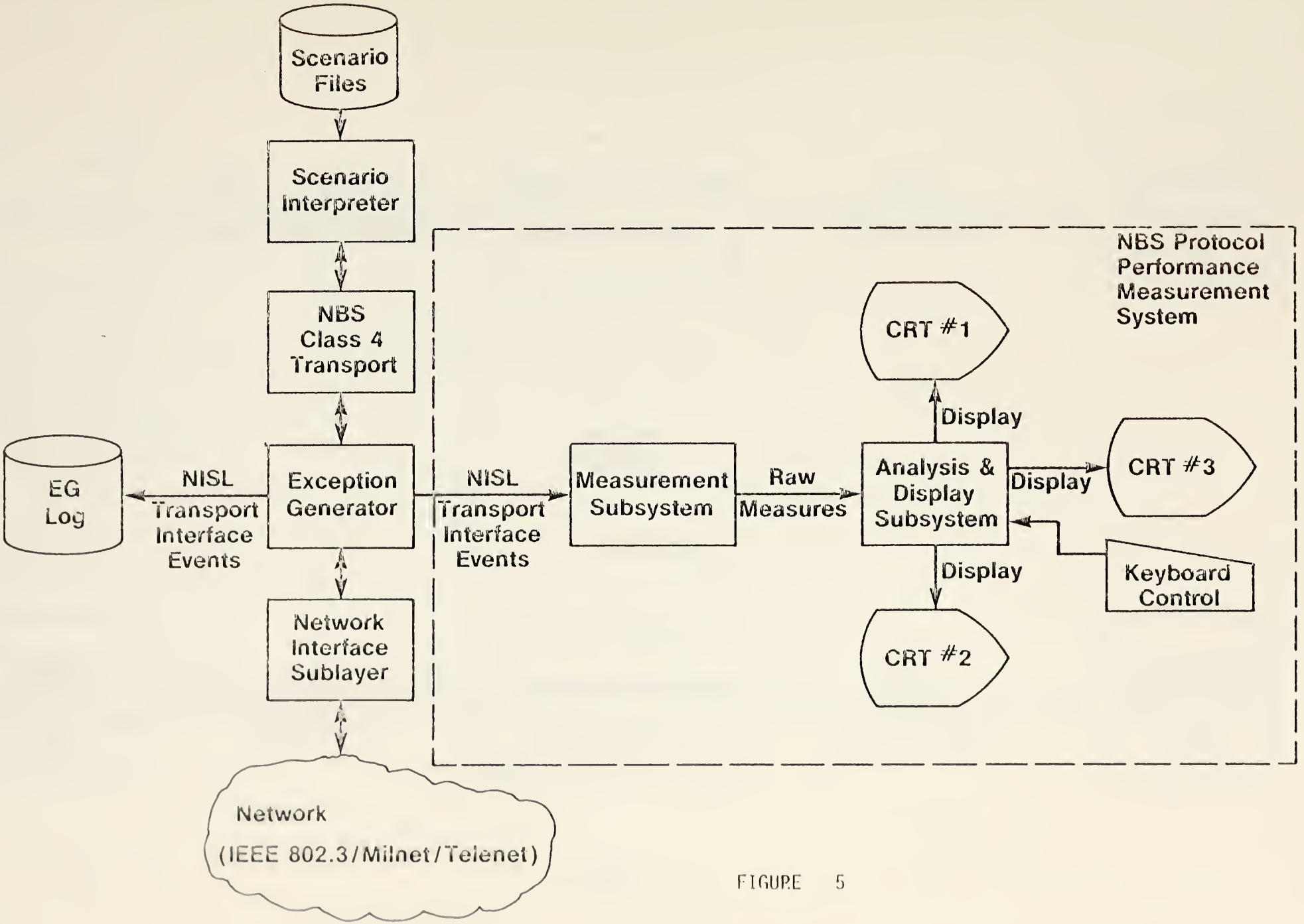
Scenario Files
Scenario Interpreter
NBS Class 4 Transport
Exception Generator
Network Interface Sublayer
Network (IEEE 802.3/Milnet/Telenet)
EG Log
NISL Transport Interface Events
NISL Transport Interface Events
Measurement Subsystem
Raw Measures
Analysis & Display Subsystem
NBS Protocol Performance Measurement System
CRT #1
CRT #2
CRT #3
Display
Display
Display
Keyboard Control

FIGURE 5

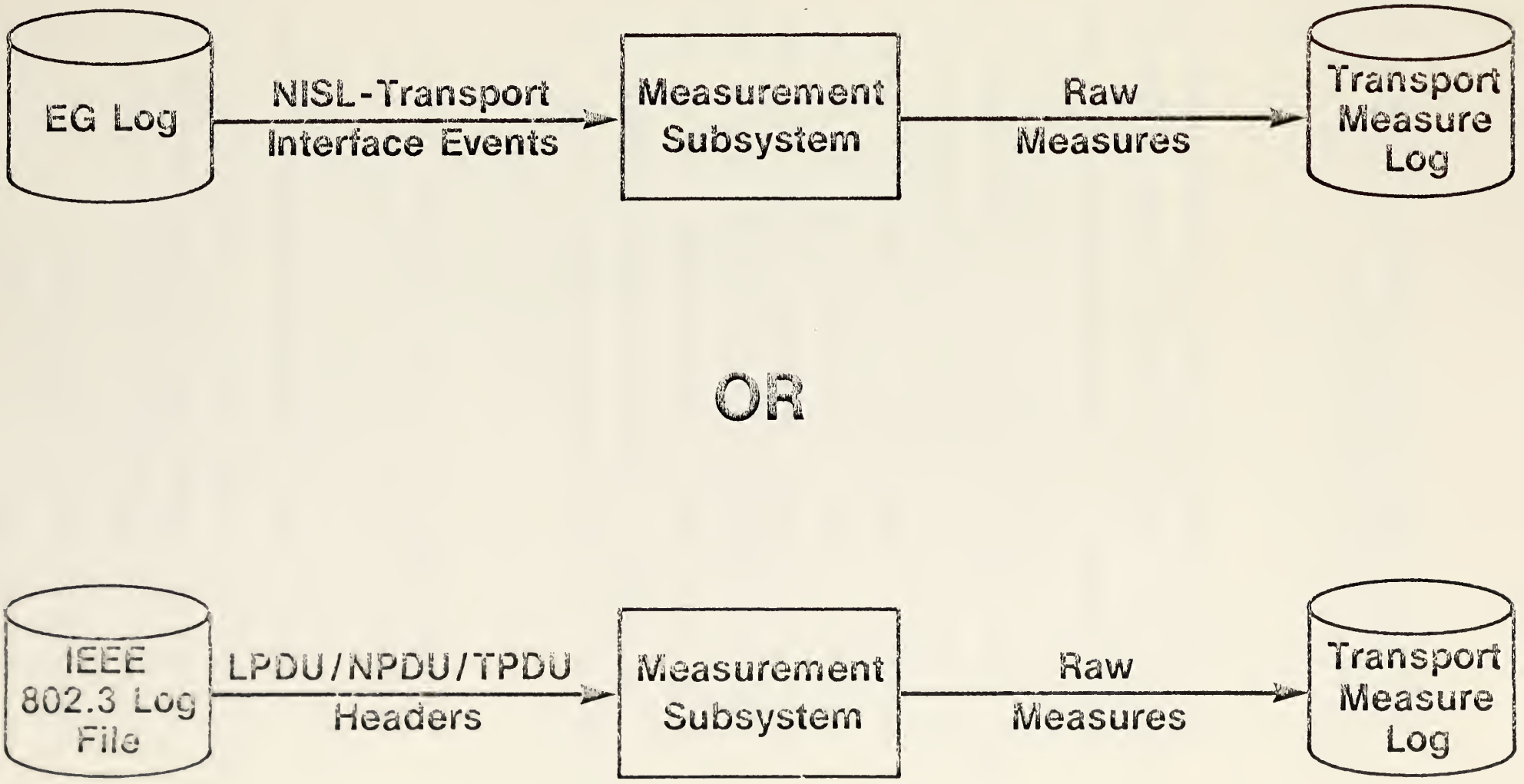
EG Log
NISL-Transport
Interface Events
Measurement
Subsystem
Raw
Measures
Transport
Measure
Log
OR
IEEE
802.3 Log
File
LPDU/NPDU/TPDU
Headers
Measurement
Subsystem
Raw
Measures
Transport
Measure
Log

FIGURE 6

port connection measures; and produces a periodic output stream of the computed measures. Details of the specific measures collected are given in Section III.

The measurement subsystem operates on the basis of separate measurement and reporting intervals. Measures are accumulated over a user defined measurement interval; then the accumulated measures are cleared and a new measurement interval is started. The user may also specify a separate and independent report interval. The separation of report intervals from measurement intervals permits one or more reports to be made during each measurement interval. This flexibility provides support for both performance experiments and real-time display updates. Typical performance experiments operate with identical measurement and report intervals equal to ten or fifteen minutes. Typical real-time display applications run with a measurement interval of two minutes and a report interval of ten seconds (i.e., 12 reports per measurement interval). The measurement system runs until stopped by manual intervention or until a given number of measurement intervals has been processed.

To simplify the internal measurement logic, the measurement system always collects every measure for which it is programmed as opposed to collecting a subset of measures under user instruction. However, if every measure were reported, the output data stream could easily exceed ten thousand bytes per report interval. This large output stream can slow the operation of the measurement subsystem, causing network traffic to be missed when running real-time experiments on high speed networks. To allow more flexibility to run real-time as well as off-line measurement experiments the measurement subsystem has been implemented so that

specific subsets of measures may be reported as requested by the user when the measurement subsystem is started.

E. Analysis and Display

The analysis and display subsystem consists of a number of independent programs that consume the stream of raw measures reported by the measurement subsystem and produce metrics and/or graphs. One display program simply decodes the raw measure stream and produces a human-readable report. This program is useful for debugging the measurement subsystem, for learning about the measurement subsystem output formats, and for obtaining ad hoc information to be used to design a specific analysis routine.

Two sets of programs produce multiple graphic displays in the configurations illustrated in Figures 4 and 5. One set of programs creates histograms and tables on character-oriented displays. These programs provide a low resolution bar graph output and were developed as part of a prototype measurement system. The second set of programs creates line graphs, 3-D histograms, block diagrams, and tables on high resolution color graphic displays. This set of programs was used at the NCC 1984 to demonstrate class 4 transport operation over a CSMA/CD network.

Another program produces a pair of matrices representing raw measures and the metrics computed from the measures. This program supports performance experiments and was used to produce the information presented in Section IV of this paper.

Since the analysis and display programs are independent of each other and operate from a well defined input stream, users can easily create new programs for specific experiments. However, if new measures are required, modifications must be made to the measurement subsystem.

III. Measures

The measurement subsystem collects a large number of measures, permitting many types of analyses to be conducted. The approach used to collect the measures is one of protocol monitoring. The measurement subsystem decodes every LPDU, NPDU, and TPDU captured by the data collection subsystem and monitors the progress of every transport connection on the network. Tracking transport connections enables accumulation of connection, host, and aggregate level measures. Several of the measures collected are described below.

A. Aggregate Level

Aggregate measures consist of the accumulation of link, network, and and transport measures for all hosts on the LAN. For the link, total LPDUs, total bytes in all LPDUs, and the analogous totals for the peak second are measured. For the network, total NPDUs and total bytes in all NPDUs are counted. The most detailed measures are for the transport layer.

Transport measurement is capable of discriminating information bytes from overhead bytes as well as data bytes from header bytes. The definition

of data and header bytes is taken directly from the format of a given TPDU. Overhead includes all header bytes plus all retransmitted data bytes. Information includes only the original transmission of data bytes.

The transport measures also contain a record of transport connection activity including the number of transport connection attempts, the number of successfully established connections, the number of connection refusals, and the number of connection negotiation failures. A number of measures are also provided by TPDU type (see Table 1) including count of TPDUs, count of retransmitted TPDUs, and count of information, overhead, data, and header bytes.

The measurement system is capable of recognizing the concatenation of DT TPDUs into larger messages known as transport service data units (TSDUs). The number of TSDUs and bytes in all TSDUs are counted. The measurement system also produces a histogram of TPDU and TSDU sizes. Ten size intervals are provided, and the bounds can be specified at program compilation time.

B. Host Level

Host measures consist of the accumulation of measures specific to each host on the network. Usually, a host contributes a portion of the total network traffic characterized by the aggregate measures. For a host, the measures are separated into a "transmitted" class and a "received" class. This allows an individual assessment of each host as a transport

sender and receiver. A performance analyst can work with the appropriate metrics for each host to determine the contribution of specific hosts to the aggregate performance.

Some measures not applicable at the aggregate level are available at the host level. For example, a counter is incremented each time a host sends into a closed flow control window. TPDUs sent into closed windows are usually discarded, therefore, frequent occurrence of this behavior involving a specific host or pair of hosts may indicate a poorly performing flow control strategy.

C. Connection Level

The measures collected at the connection level are fundamental because they allow the appropriate accumulation of the same measures at the host and aggregate levels. For example, if the measurement system did not track transport connections, it would be impossible to determine retransmissions and accumulate the total number of overhead bytes. In addition to being required in order to collect other measures, the connection level allows some measures that are not applicable at the aggregate and host levels. These additional measures are described further below.

For each transport connection, the current perceived state is always available along with a history of states. This allows determination of opening, established, closing, and closed connections. At a finer level of detail, connections closed abruptly can be distinquished from those closed gracefully and those closed as a result of a protocol violation.

Several time stamps are also recorded for each connection including: start, stop, and time of last activity. The period during which the flow control window is closed in each direction of flow is provided to aid a performance analyst in isolating an inadequate flow control strategy as a cause of poor throughput.

The remaining measures maintained for each transport connection are identical to those kept at the host level. Separate measures are collected for each direction of flow on every transport connection.

D. Limitations

The measurement subsystem as implemented has several limitations. First, every TPDU must be collected or the measures will be incomplete. Some TPDUs, however, are more critical than others. For example, if TPDUs associated with connection establishment are missed, the monitor will be unaware of the connection. In such a case, the remaining TPDUs on the connection are counted in a category of TPDUs that could not be assigned to a transport connection. If TPDUs associated with connection termination are missed, a measurement subsystem inactivity timer will eventually expire, thus terminating measurement on the connection. Experience with measurement under the loads generated by the NCC demonstration traffic uncovered no evidence of missed TPDUs.

The inability of the measurement subsystem to operate properly when packets are grossly misordered or when a transport implementation withholds

sending acknowledgements for extraordinarily long periods is a limitation that exists because a maximum duplication detection window of 32 TPDUs is implemented. This window size was selected because it was the largest that could be obtained while still enabling operations on the duplicate detection windows to be carried out in an expeditious manner through bit maps. During the demonstration this window size did not prove to be a limitation. Only a long propogation delay, high bandwidth environment is expected to present a problem (e.g., 1.544 Mbps or higher satellite channels).

A third limitation is the exclusion of delay measures for two reasons. First, the measurement system is a node on the network and thus not at either terminus of an end-to-end connection. This causes the measurement system to be unable to determine significant portions of the delay as seen by a transport service user. Several estimating techniques were developed to solve this problem, but the remaining difficulty of implementing a solution within the processing time and memory space budget of the measurement system could not be overcome during initial development.

Second, due to the possibility of misordering data and withholding of acknowledgements, maintaining a delay measure would require the measurement subsystem to buffer and reorder time stamped sequence numbers for both directions of flow for every transport connection. Buffering is required to allow matching of AK TPDUs to associated DT TPDUs and reordering is required to distinguish TSDUs. These operations are prohibitive

in terms of processing of time and memory space for the 100 to 150 connections expected.

IV. Application

As already noted, the performance measurement system described above was connected to a CSMA/CD network at the NCC 1984. The network was populated by computer products from the vendors shown in Table 2. Each vendor implemented the IEEE 802.3 standard at the link layer, a null network layer, ISO class 4 transport at the transport layer, and a subset of the planned ISO file transfer protocol at the application layer. No session or presentation layers were implemented.

The general concept of the NCC demonstration was user entry of a file at one computer system and transfer to another computer system for viewing. Additionally, a number of graphics files were located around the network for retrieval by users at graphics workstations. This configuration leads to a natural traffic pattern of bulk data transfer where several hosts are primary data sources while other hosts are primary data sinks. A few hosts served in an equal capacity as data sources and sinks.

While the NCC demonstration was running, the NBS protcol performance measurement system produced real-time color graphic displays. The displays showed the basic connectivity, the number of connections per host, the distribution of traffic by TPDU type, and the number of data octets transmitted by each host. In addition, during the demonstration,

Table 2 NCC 1984 CSMA/CD Demonstration

Participants

Advanced Computer Communications

Boeing Computer Services

Charles River Data Systems

Digital Equipment Corporation

Honeywell Information Systems

Hewlett-Packard Company

International Computers Limited

Intel Corporation

National Bureau of Standards

NCR Corporation

measures were collected to allow the generation of several metrics off-line. The metrics computed are described in the following paragraphs, and then specific values obtained during the pre-demonstration test period are presented and interpreted.

A. Metrics

Performance metrics provide a quantitative characterization of specific aspects of system performance [16, 18-25]. For the protocol performance experiments reported here, the metrics were derived from the available measures in one of two ways: (1) expressing some measures with respect to time or (2) expressing two measures as a ratio. Using these simple concepts, a large amount of information can be derived. The specific metrics used in these experiments are described below and a summary is shown in Table 3.

Throughput

Information throughput is the amount of user information transferred per unit of time. Two levels of throughput were computed: link information throughput and transport information throughput. In controlled experiments a system can be loaded as heavily as possible and the throughput observed can provide a measure of system capacity. For the purposes of this report, link information throughput (T_L) in bits per second, is defined as:

$$T_L = \frac{8N_b}{M} \qquad (1)$$

Table 3 Summary of Metrics

Metric	Definition
T_L	Link Information Throughput in Bits per Second
T_T	Transport Information Throughput in Bits per Second
U_C	Channel Utilization as a Fraction of Channel Capacity
E_C	Channel Efficiency as a Fraction of Channel Capacity
E_L	Link Protocol Efficiency as a Fraction of Perfect Efficiency
E_N	Network Protocol Efficiency as a Fraction of Perfect Efficiency
E_T	Transport Protocol Efficiency as a Fraction of Perfect Efficiency
C_E	Combined Protocol Efficiency as a Fraction of Perfect Efficiency
R_{OM}	Ratio of Original DTs to Total TPDUs
R_{OA}	Ratio of Original DTs to Original AKs
R_{RD}	Ratio of Retransmitted DTs to Total DTs
R_{RA}	Ratio of Retransmitted AKs to Total AKs
S_D	Average Data Field Length per DT

where N_b is the number of NPDU bytes transferred on the network during the measurement interval and M is the size of the measurement interval in seconds. Transport information throughput (T_T) is defined similarly:

$$T_T = \frac{8T_j}{M} \qquad (2)$$

where T_j is the number of TPDU information bytes transferred on the channel during a measurement interval and M is the size of the measurement interval in seconds.

Channel Utiliztion and Efficiency

A given channel is engineered to provide a specific information carrying capacity per unit time. For example, the demonstration IEEE 802.3 local area network provides a capacity of about 10 Mbps*. The ratio of the actual number of bits on the channel per unit time to the capacity is the channel utilization. Using measures from the measurement subsystem, channel utilization (U_c) as a fraction of capacity is computed as:

$$U_c = \frac{8L_b}{MR} \qquad (3)$$

where L_b is the number of LPDU bytes transferred on the channel during a measurement interval, M is the size of the measurement interval in seconds, and R is the capacity of the channel in bits per second.

*The actual capacity of a CSMA/CD network is a function of the traffic in a given second. This is true because of the required inter-frame time of 9.6 microseconds. Thus, 10 Mbps is an upper limit where the real rate observed will be lower but typically much less than one percent lower.

Channel efficiency is the ratio of user information bits on the network per unit time to the capacity of the channel. For the experiments under consideration the user is defined to be the transport service user. The computation of channel efficiency (E_c) as a fraction of capacity is:

$$E_c = \frac{8T_i}{MR} \qquad (4)$$

where T_i is the number of TPDU information bytes transferred on the channel during a measurement interval, M is the measurement interval size in seconds, and R is the capacity of the channel in bits per second.

Protocol Efficiency

Protocol efficiency is a measure of the ratio of information bits sent to total bits sent for a specific protocol layer or for all layers of protocol. The ideal protocol efficiency is one. Efficiency decreases as the value of the metric decreases. Several protocol efficiency metrics were used to evaluate the performance of the NCC demonstration system including: link, network, transport, and combined protocol efficiencies. Link protocol efficiency (E_L), as a fraction of perfect efficiency, is defined as:

$$E_L = \frac{N_b}{L_b} \qquad (5)$$

where N_b and L_b are the number of NPDU bytes and LPDU bytes, respectively, transferred on the channel during a measurement interval.

Network Protocol efficiency (E_N), as a fraction of perfect efficiency, is defined as:

$$E_N = \frac{T_b}{N_b} \tag{6}$$

where T_b and N_b are the number of TPDU bytes and NPDU bytes respectively, transferred on the channel during a measurement interval.

Transport protocol efficiency (E_T), as a fraction of perfect efficiency is defined as:

$$E_T = \frac{T_i}{T_b} \tag{7}$$

where T_i and T_b are the number of TPDU information bytes and TPDU bytes, respectively, transferred on the channel during a measurement interval.

Since protocol efficiency is the ratio of information bytes transmitted to total bytes transmitted, a layered set of protocols diminishes the protocol efficiency incrementally for each layer. This can be understood by considering a unit of information to be passed through a seven layer protocol system. At each layer, the original information has header bytes appended when being sent out of the system. The header bytes are removed as the original information passes through each layer coming into the system. Therefore, in general, combined protocol efficiency (C_E) as a fraction of perfect efficiency, in a layered protocol system is:

$$C_E = \frac{U_i}{S_b} \tag{8}$$

where U_i is the total number of bytes in a user message and S_b is the total number of bytes sent by the protocol system to transfer the user message.

In the three layer (Link, Network, Transport) model understood by the measurement system, the combined protocol effiency is:

$$C_E = \frac{T_i}{L_b} \tag{9}$$

where T_i is the number of TPDU information bytes sent on the channel and L_b is the number of LPDU bytes sent on the channel.

The concept that each protocol layer diminishes the combined protocol efficency is illustrated by an alternate method of computing combined protocol efficiency.

$$C_E = E_L \, E_N \, E_T \tag{10}$$

where E_L, E_N, and E_T are link, network, and transport protocol efficiencies, respectively. In general, for an N-layered protocol system:

$$C_E = \prod_{i=i}^{N} E_i \tag{11}$$

where N is the number of protocol layers in the system and E_i is the efficiency of the i^{th} layer.

Combined protocol efficiency can also be viewed as the ratio of channel capacity used for transferring user information to channel utilization. Thus, for the three layer model of the measurement system:

$$C_E = \frac{E_C}{U_C} \tag{12}$$

where E_C is channel efficiency and U_C is channel utilization.

Other Ratios

Several other ratios can be used as metrics for the evaluation of transport protocol implementation performance including: the ratio of original data transmissions to total transmissions (R_{OM}), the ratio of original data transmissions to original acknowledgements (R_{OA}), and the ratios of retransmissions to total transmissions for data (R_{RD}) and acknowledgements (R_{RA}).

R_{OM} and R_{OA} permit several inferences to be drawn. Normally, if R_{OA} is equal to or just below one, then a scheme of one acknowledgement per data message is being used, and data transfer has probably proceeded without interruption. As R_{OA} increases above one, then an acknowledgement withholding strategy is being used so that one acknowledgement covers multiple data messages. The ratio is computed as:

$$R_{OA} = \frac{(D_T - D_R)}{(A_T - A_R)} \qquad (13)$$

where D_T and A_T are the total number of DT and AK TPDUs, respectively, counted during a measurement interval and D_R and A_R are the total number of retransmitted DT and AK TPDUs, respectively, counted during the same measurement interval.

R_{OM} indicates the number of overhead messages required to transmit each data message on the network. This metric gives a measure of the efficiency of the protocol from the perspective of messages. The information is useful because each message transmitted requires some CPU processing time that is not a direct function of message size. As

R_{OM} approaches one the CPU message processing required to deliver the data approaches a minimum. The ratio is computed as:

$$R_{OM} = \frac{D_O}{M_T} \qquad (14)$$

where D_O is the count of original DT TPDUs sent on the channel during a measurement interval and M_T is the count of all TPDUs sent on the channel during the same interval. This formulation of the metric is acceptable if expedited data messages are ignored. No expedited data messages were used in the NCC demonstration.

The ratio of retransmitted data messages to total data messages (R_{RD}) should ideally be zero. As R_{RD} approaches one, data messages are being retransmitted at an unacceptably high rate and no service is provided to the user. Retransmission of data messages can have a number of causes including network errors, lost messages, and improperly tuned retransmission timers. The ratio is computed as:

$$R_{RD} = \frac{D_R}{D_T} \qquad (15)$$

where D_R is the count of retransmitted DT TPDUs sent on the channel during a measurement interval and D_T is the count of all DT TPDUs sent during the same measurement interval.

The ratio of retransmitted acknowledgements to total acknowledgements (R_{RA}) should ideally be zero assuming continuous full-duplex data flow on the transport connections. When data flows continuously in only one direction, R_{RA} increases because acknowledgements are sent in the direction of data flow at the rate of the window timer and these acknowledge-

ments will, in general, be duplicates. Other causes for a large R_{RA} include: open transport connections with no data flow, transport connections with bursty data flow, inappropriately short values for the window timer, and a receiver persisting in retransmitting acknowledgements for flow control confirmation following a closed window or credit reduction. The ratio is computed as:

$$R_{RA} = \frac{A_R}{A_T} \qquad (16)$$

where A_R is the count of retransmitted AK TPDUs sent on the channel during a measurement interval and A_T is the count of all AK TPDUs sent during the same interval.

Data Message Size

An important factor contributing to protocol efficiency is the size of the data portion of data messages. In an error free environment, the larger the size of the data messages sent the greater the protocol efficiency. When errors are introduced, an increasing message size provides better efficiency only to the point where the probability of an error within the block becomes so high that R_{RD} increases and, thus, protocol efficiency decreases. Because of the importance of block size, a metric specifying the average size of the data field in data messages (S_D) was computed as:

$$S_D = \frac{T_i}{(D_T - D_R)} \qquad (17)$$

where T_i is the count of TPDU information bytes sent during a measurement

interval and D_T and D_R are the count of total and retransmitted DT TPDUs, respectively, during the same interval.

The metrics described above are a subset of the metrics used within the NBS protocol performance laboratory for performance experiments and for simulation modeling involving the ISO class 4 transport protocol [26]. Several delay metrics are also used in the NBS protocol performance laboratory, but are not applicable to the experiments reported in this paper.

Results

The results described below were obtained by applying the measurement system to the IEEE 802.3 network installed at the NBS during preparations for the multi-vendor CSMA/CD demonstration at the NCC 1984. Nine vendors (see Table 2) had products on the network. Table 4 presents the measures obtained at the aggregate level over a two hour period. Table 5 presents the metrics that were computed from the measures in Table 4. The following paragraphs discuss the results.

Load and Throughput

As can easily be seen from the metrics for throughput, channel utilization, and channel efficiency, the 10 Mbps CSMA/CD channel was very lightly loaded at less than 1%. This result may seem somewhat surprising since up to nine hosts were involved in file transfers and as many as 54 transport connections were active over a fifteen minute period. The light load is

TABLE 4 RAW MEASURES COLLECTED JUNE 13, 1984 BETWEEN 5:16 p.m. and 7:16 p.m. (15 MINUTE INTERVALS)

MEASURES	INTERVALS 0	1	2	3	4	5	6	7	TOTALS
LPDUs	4524	4296	4258	5223	3169	302	579	785	23136
LPDU Octets	355779	294336	388972	418939	177910	16880	77207	132862	1862885
NPDUs	4524	4296	4258	5223	3169	302	579	785	23136
NPDU Octets	278871	221304	316586	330148	124037	11746	67364	119517	1469573
TPDU	4524	4296	4258	5223	3169	302	579	785	23136
TPDU Information Octets	184464	136103	197170	204145	61188	5570	55460	107648	951748
TPDU Overhead Octets	89883	80905	115158	120780	59680	5874	11325	11084	494689
TPDU Header Octets	53235	51850	54006	63729	59680	5874	11325	11084	310783
TPDU Data Octets	221112	165158	258322	261196	61188	5570	55460	107648	1135654
TPDU Total Octets	274347	217008	312328	324925	120868	11444	66785	118732	1446437
TSDUs	1847	1739	1483	1783	1027	70	111	100	8160
TSDU Octets	184464	136103	197170	203133	61188	5570	55460	107648	950736
DT TPDUs	1932	1791	1560	1840	1027	70	164	266	8650
DT Retransmission	52	49	77	71	0	0	0	0	249
AK TPDUs	2355	2296	2461	3144	2133	194	394	470	13447
AK Retransmission	114	123	145	116	70	77	116	125	886
Transport Connections	45	50	54	51	2	6	5	12	225

Table 5 Metrics Computed from Measures in Table 4

METRICS	0	1	2	3	4	5	6	7	TOTAL
T_L	2479	1967	2814	2935	1103	104	599	1062	1633
T_T	1640	1210	1753	1815	544	50	493	957	1058
U_C	.00031	.00026	.00035	.00037	.00016	.00002	.00007	.00012	.00021
E_C	.00016	.00012	.00018	.00018	.00005	.00001	.00005	.00010	.00011
E_L	.78	.75	.81	.79	.70	.70	.87	.90	.79
E_N	.98	.98	.98	.98	.98	.97	.99	.99	.98
E_T	.67	.63	.63	.63	.51	.49	.83	.91	.66
C_E	.52	.46	.50	.49	.35	.33	.72	.81	.51
R_{OM}	.42	.41	.35	.34	.33	.23	.28	.34	.36
R_{OA}	.84	.80	.64	.58	.49	.60	.60	.77	.67
R_{RD}	.03	.03	.05	.04	0	0	0	0	.03
R_{RA}	.05	.05	.06	.04	.03	.40	.29	.27	.07
S_D	100	78	126	115	60	80	338	405	113

explained by the fact that the average size of files transferred was about 4000 bytes. Many small files were used along with a few larger files, but a limit of 64K bytes per file was adopted by the demonstration participants. The preponderance of short files transferred during the demonstration limited the probability of long periods of simultaneous file transfers. Applications with multi-megabyte file transfers would present a different traffic pattern, raising the probability of long periods of simultaneous transfer and increasing the load on the network.

An important conclusion that can be drawn from the values for T_L, T_T, U_C, and U_E given in Table 5 is that the remaining metric values should be treated with caution due to the exceptionally light network loading. Additional experiments with much heavier loads are required to assess the affect of network load on the various metrics.

Protocol Efficiency

The understanding of network protocol efficiency is straightforward. Every link packet contained only a single byte of network header and, therefore, the efficiency of the network protocol was always quite high. The variability between .97 and .99 is a function of the distribution of sizes for TPDUs carried in the NPDUs. At the .97 value (interval 5), a large proportion (67%) of the traffic was AK TPDUs, which are small, thus the importance of DT TPDU size was diminished. At the .99 values (intervals 6 and 7), the average size of a DT TPDU (S_D) was relatively large.

The link protocol metric is also easy to explain. Each LPDU contains a fixed 17-byte header and, therefore, the efficiency of the link protocol is related directly to the distribution of sizes for TPDUs and to the number of TPDUs carried in the LPDUs. In the demonstration, TPDUs were mapped into LPDUs on a one-to-one basis; therefore, the size distribution of the TPDUs was the only factor affecting link protocol efficiency. In general, the larger the average DT TPDU size, the greater is the link protocol efficiency. This is offset by the special cases (e.g., interval 5) when the TPDU traffic is mostly composed of AK TPDUs.

Transport protocol efficiency is the most interesting of the efficiency metrics. Since the transport protocol efficiency measured on the demonstration network is always lower than efficiencies for the link and network protocols, the combined protocol efficiency is dominated by the transport protocol efficiency. The reasons that the transport protocol efficiency is always the lowest (in the context of this demonstration network) are traceable to the fact that transport is the first layer of protocol implementing a connection-oriented service that is providing end-to-end error detection and correction and explicit end-to-end flow control. Under circumstances of heavier loading, the medium access control sublayer of the link layer might experience collisions and retransmissions; however, in the demonstration network no evidence of collisions was found.

The header sizes for the transport protocol are, in general, smaller than the fixed link layer header; however, the transport layer was experiencing a retransmission rate for DT TPDUs of up to 5%. These retransmissions are direct overhead resulting in lower transport protocol efficiency.

For the demonstration network, the CSMA/CD channel was quite reliable and introduced no errors into LPDUs. The retransmission of DT TPDUs resulted from two other factors: (1) some of the LAN interface hardware would occasionally lose LPDUs and (2) some transport implementations would grant an amount of credit (i.e., permission to transmit) that exceeded the memory allocated to hold the received TPDUs. The former factor is caused by malfunctioning hardware that can be fixed. The latter factor is caused by inappropriate tuning of the credit granting mechanism for use in a local area network with short propagation delays.

A second trait of the transport protocol that contributes to lower efficiency is the use of messages that are completely overhead (e.g., AK TPDUs). This trait will be evident for any protocol that must provide a reliable, flow controlled service. This aspect of efficiency is demonstrated by R_{OM}. As R_{OM} becomes smaller, the transport protocol efficiency diminishes.

A final factor that affects the efficiency of the transport protocol is the average DT TPDU size. In the demonstration network, larger average DT TPDU sizes (S_D) increased the transport protocol efficiency (e.g., intervals 6 and 7).

Data and Acknowledgement Ratios

From the data and acknowledgement ratios presented in Table 5 one can infer the existence of certain inefficiencies within the transport

implementations on the network. The ratios can indicate such conditions as window timers set too low, poor flow control strategies, poor acknowledgement strategies, and transport connections open without data flow. The values for R_{RA}, R_{OA}, and R_{OM} in Table 5 are discussed below.

The class 4 transport protocol uses AK TPDUs for the following purposes: (1) acknowledgement of DT TPDUs, (2) flow control, (3) confirmation of flow control information, and (4) maintaining the open status of a quiescent transport connection. Due to the four part role of the AK TPDU, R_{OA} and R_{RA} are impossible to interpret meaningfully at the aggregate level and difficult to interpret at the host level. Several factors can cause changes in the R_{OA} and R_{RA} values and, at the aggregate level, these factors can tend to cancel one another. At the host level, given some knowledge of the transport implementations involved, more meaningful interpretations can be made. These points can be understood more clearly by considering each ratio in turn.

R_{RA}. To interpret R_{RA} the definition of a retransmitted acknowledgement must be understood. An original AK TPDU must either acknowledge new data, grant new permission to send data (i.e., credit), or reduce previously granted credit. Any other AK TPDU is a retransmission. Retransmitted AK TPDUs are caused by expiration of a window timer on a quiescent half of a transport connection, expiration of a flow control synchronization timer, or receipt of an AK TPDU requiring flow control confirmation. Therefore, a high value for R_{RA} can have several causes.

Since flow control confirmation rules were not used for the demonstration, the most likely cause is expiration of the window timer. Window timer expiration occurs only when AK TPDUs are not being sent to acknowledge data (i.e., few DT TPDUs are being received). The rate of window timer expiration is a function of the window timer period and the duration of periods during which no DT TPDUs are received. Thus, R_{RA} is normally higher for a data source in a uni-directional file transfer. A good example of this phenomenon is shown for Host A in Table 6 (R_{RA} = .41).

The low R_{RA} values for source Hosts E and I in Table 6 can only be explained by a low rate of window timer expiration. This can be attributed to a long window timer period or to the fact that the file transfers involving Hosts E and I were of very short duration. The same is true for R_{RA} for Host C in Table 7.

R_{OA}. To interpret R_{OA} recall the definition of an original AK TPDU from the previous discussion. Since explicit credit reduction was not used during the NCC demonstration, original AK TPDUs were transmitted only to acknowledge new data and to grant new credit. Normally new credit is granted in the same AK TPDU that acknowledges new data and R_{OA} tends toward one (assuming no withholding of acknowledgements). However, an implementation of taut flow control does not fit the normal pattern.

For example, consider a transport implementation that grants permission to send one DT TPDU, acknowledges the DT TPDU with an AK TPDU that gives no permission to send (i.e., closes the window), and then, after some

time, issues a new AK TPDU that gives permission to send another DT TPDU. Thus, two original AK TPDUs are sent for every original DT TPDU and R_{OA} tends toward .50. This situation is illustrated in Table 7 where two hosts from a single computer vendor are transferring files using taut flow control. Enough original AK TPDUs are transferred to prevent expiration of the window timer and, thus, R_{RA} is low.

Another source of change in R_{OA} is the existence of host computers acting as data sources on the network. This is illustrated by Host A in Table 6 where R_{OA} is 8.73. The cause of this high R_{OA} is that Host A is a source for many DT TPDUs, but sends only enough original AK TPDUs to acknowledge the file transfer protocol commands from the data sink. The remainder of the AK TPDUs sent by Host A are retransmissions stimulated by expiration of the window timer.

R_{OM}. The ratio R_{OM} provides an indication of the CPU processing burden required to transfer data messages. The burden is represented by the total number of protocol messages required to transfer a set of protocol data messages. Table 5 provides values for R_{OM} over a two hour period. During the periods of highest R_{OM}, six overhead messages are required for every four data messages. During the periods of lowest R_{OM}, three overhead messages are required for each data message. For the entire two hour period, two overhead messages were required for each data message.

Table 6 and 7 provide R_{OM} values for individual hosts during intervals 2 and 4 respectively. Two patterns of R_{OM} are evident. For five of the

Table 6 Host Metrics - Interval 2

Host	R_{OM}	R_{OA}	R_{RA}	Mode
A	.33	8.73	.41	Source
E	.39	.90	.03	Source
F	.33	.03	.02	Sink
G	.40	.62	.00	Sink
I	.40	.90	.03	Source

Table 7 Host Metrics - Interval 4

Host	R_{OM}	R_{OA}	R_{RA}	Mode
C	.39	.51	.03	Source
D	.39	.49	.04	Sink

hosts (E, G, I, C, and D), six overhead messages are required for every four data messages. For Hosts A and F, two overhead messages are required for every data message.

Average Data Block Size

The average data block size experienced over the two hour measurement period varied between 60 bytes and 405 bytes as shown in Table 5. Generally, the larger the DT TPDU data size, the greater the transport protocol efficiency experienced. The DT TPDU data size variability observed was due to the different implementation strategies adopted by the vendors involved and differences in the format of data transferred around the demonstration network (e.g., ASCII display files vs. graphics output files).

V. Conclusions

Several conclusions can be drawn from the experiences and results reported in this paper. First, many useful performance metrics at three levels (link, network, and transport) of the OSI reference model can be obtained without measurement artifact via a centralized monitoring device on a local area network. The metrics accessible include throughput, channel utilization and efficiency, protocol bandwidth efficiency, various message ratios, and average message size. Delay metrics cannot be easily calculated using the centralized monitoring approach without making certain limiting assumptons or relying upon estimating techniques.

A second conclusion resulting from the work reported here is that protocol bandwidth efficiency in a layered protocol system is diminished by each protocol layer. In fact, the total protocol efficiency of the layered system is the product of the protocol efficiency of each protocol layer. This is a remarkable conclusion that has serious implications for the design and implementation of layered systems of protocols.

A third conclusion is that the transport protocol message ratio metrics described in this paper are of limited use when computed at the aggregate level. At the host and connection levels, the inability of the measurement subsystem to distinguish between the four roles of the AK TPDU within the OSI class 4 transport protocol limits the interpretations that can be made from the ratio metrics unless the performance analyst has certain a priori knowledge about the characteristics of the transport implementations using the network.

Finally, the OSI class 4 transport protocol can be implemented for efficient operation over a local network. This conclusion is supported by the protocol efficiency metrics reported. It appears that reasonable efficiency was achieved given that many of the implementations were prototypes. The results also demonstrate several means available for improving the efficiency of the implementations.

VI. Acknowledgements

The prototype measurement system architecture was established by Robert Rosenthal and Kevin Mills. The data collection subsystem was designed

and implemented by Robert Toense. The measurement subsystem was designed by Kevin Mills and implemented by Kevin Mills and Jeff Gura. Porting of the measurement subsystem to the NCC demonstration hardware was accomplished by Michael Chernick. The prototype analysis and display subsystem was designed and implemented by Daniel Stokesberry, Timothy Gardner, and Patrick Johnstone. The analysis and display subsystem used at the NCC demonstration was designed and implemented by Daniel Stokesberry and Timothy Gardner. Several other analysis and display programs were designed and implemented by Kevin Mills and Jeff Gura.

References

[1] John D. Day and Hubert Zimmermann, "The OSI Reference Model," Proceedings of the IEEE, pp. 1334-1340, December 1983.

[2] Keith G. Knightson, "The Transport Layer Standardization," Proceedings of the IEEE, pp. 1394-1396, December 1983.

[3] Proceedings of the First LAN-Transport Workshop, Report No. NBSIR 83-2673, February 1983.

[4] Proceedings of the Second LAN-Transport Workshop, Report No. NBSIR 83-2717, May 1983.

[5] Proceedings of the Third LAN-Transport Workshop, Report No. NBSIR 83-2757, July 1983.

[6] Proceedings of the Fourth LAN-Transport Workshop, Report No. NBSIR 83-2796, October 1983.

[7] Proceedings of the Fifth LAN-Transport Workshop, Report No. 84-2855, March 1984.

[8] Kevin Mills, "Testing OSI Protocols: NBS Advances the State-of-the-Art," Data Communications, March 1984.

[9] R. Jerry Linn and J. Stephen Nightingale, "Testing OSI Protocols at the National Bureau of Standards," Proceedings of the IEEE, pp. 1431-1434, December 1983.

[10] Marshall D. Abrams, et al., Measurement of Computer Communications Networks, Department of Commerce, July 1976.

[11] Marshall D. Abrams, et al., "The Network Measurement System," IEEE Transactions on Communications, 1976.

[12] R. Rosenthal, et al., The Network Measurement Machine -- A Data Collection Device for Measuring the Performance and Utilization of Computer Networks, NBS Technical Note 912, April 1976.

[13] D.R. Wortendyke, et al., User-Oriented Performance Measurements on the ARPANET, NTIA Report 82-112, November 1982.

[14] H. Opderbeck and L. Kleinrock, "The Influence of Control Procedures on the Performance of Packet-Switch Networks," Proceedings of the National Telecommunications Conference, 1974.

[15] S.S. Poh, Comparison of PWIN and ARPANET Performance Measurement Capabilities, Mitre Corporation, February 1975.

[16] C.J. Bennett and A.J. Hinchley, "Measurements of the Transmission Control Protocol," Computer Network Protocols, University of Liege, 1978.

[17] MS-109 User Documentation, Volume 2, Tesdata Systems Corporation, March 1980.

[18] Wayne H. McCoy, et al., "Assessing the Performance of High-level Computer Network Protocols," Proccedings of INWG/NPL Workshop Protocol Testing - Towards Proof?, National Physical Laboratory, Teddington, England, 27-29 May, 1981.

[19] Kevin L. Mills, "Performance Measurement Problems in a Packet-Switch Network," Proceedings of CMG XII Conference, December 1981.

[20] P.J. Lloyd and R.H. Cole, "A Comparative Study of Protocol Performance on the Universe and SATNET Satellite System," Satellite and Computer Communications, North-Holland, 1983.

[21] K.S. Raghunathan, et al., "Relationship Between Performance Parameters for Transport and Network Service," ACM SIGCOMM 83 Symposium Proceedings, March 1983.

[22] Wushow Chou, "Analysis of Data/Computer Networks," Computer Communications, Prentice-Hall, 1983.

[23] Mischa Schwartz, Computer-Communication Network Design and Analysis, Prentice-Hall, 1977.

[24] Fouad A. Tobagi, et al., "Modeling and Measurement Techniques in Packet Communications Networks", Proceedings of IEEE, November 1978.

[25] Wushow Chou, "Performance Metrics In The Transport Layer," unpublished NBS Report, October 1982.

[26] Marnie Wheatly and Richard Colella, Joint COMSAT/NBS Experiment Plan, unpublished NBS Report, June 1983.

Figure List

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 85-3104	2. Performing Organ. Report No.	3. Publication Date February 1985

4. TITLE AND SUBTITLE

Performance Measurement of OSI Class 4 Transport Implementations

5. AUTHOR(S)

K.L. Mills, J.W. Gura, C.M. Chernick

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A measurement system to evaluate the performance of open system interconnection (OSI) transport protocol implementations is described. Several metrics are proposed to establish a quantitative characterization of layered protocol performance. Metrics specific to the OSI transport protocol are also proposed. The measurement system and metrics are applied to a multi-vendor National Computer Conference demonstration network and the results are reported.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Computer Networks; Multi-vendor Networks; NCC Demonstration; OSI Transport; Performance Measurement System; Protocol Performance

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

52

15. Price

$10.00