

TK 155.252

KFKI-1980-130

D. AMBRÓZY
A. SZABÓ
K. TARNAY

A PROGRAM PACKAGE ARCHITECTURE
FOR COMPUTER NETWORK MEASUREMENT

Hungarian Academy of Sciences

**CENTRAL
RESEARCH
INSTITUTE FOR
PHYSICS**

BUDAPEST

10-1-17

THE UNIVERSITY OF MICHIGAN
LIBRARY

UNIVERSITY OF MICHIGAN

UNIVERSITY OF MICHIGAN
LIBRARY

A PROGRAM PACKAGE ARCHITECTURE
FOR COMPUTER NETWORK MEASUREMENT

D. Ambrózy, A. Szabó, K. Tarnay

Central Research Institute for Physics
H-1525 Budapest 114, P.O.B. 49, Hungary

ABSTRACT

This paper deals with general principles of computer network measurement with the scheme and architecture of a monitor and with some already implemented modules:

1. The RTS/BSC module provides an interface to a BSC oriented telecommunication line/network.
2. The BSCMON modul is responsible for basic measuring capabilities.
3. The simulator executes simulation experiments.

АННОТАЦИЯ

После ознакомления с принципами измерений сети ЭВМ в статье описывается архитектура и блок-схема монитора и представляются уже внедренные модули:

1. Модуль RTS/BSC обеспечивает интерфейс для BSC-ориентированных телекоммуникационных линий/сетей.
2. Модуль BSCMON обеспечивает основные функции измерения.
3. Симулятор осуществляет эксперименты моделирования.

KIVONAT

A számítógéphálózatok mérési elveinek ismertetése után a cikk ismerteti egy monitor architekturáját és blokkvázlatát, majd bemutatja a már implementált modulokat:

1. Az RTS/BSC modul BSC orientált távközlési vonalakhoz/hálózatokhoz interfacet biztosít.
2. A BSCMON modul az alapvető mérési funkciókat látja el.
3. A szimulátor végzi a szimulációs kísérleteket.

1. INTRODUCTION TO COMPUTER NETWORK MEASUREMENTS

1.1 Requirements

Nowadays computer networks are spreading all over the world and therefore their measurement came into prominence. In the Central Research Institute for Physics in Hungary a program package for computer network measurement is under development. We want to speak on its architecture and about some modules.

Which kind of requirements can be raised in connection with the monitor?

- a hierarchial architecture
- no interference with the operation of the computer network
- measurement of optional network parameters
- well separable modules
- expandable in a simple way
- easily applicable to different computer networks

1.2 The hierarchy of measurements

The hierarchy of our program package contains three levels:

- diagnostics
- performance measurement
- analytic measurement

The purpose of diagnostic measurements is the control of correct operation of computer networks, the detection and diagnostics of errors. The operation of computer network is characterized by events of different types. The main events can be classified into two groups:

- current network configuration
- current operational status

The performance of a computer network is determined by the quality and quantity of services for a given workload. Some of the parameters - to be measured - are for instance:

- time (time necessary to set up and disconnect a connection, average delay time by several routing policies, time duration for using the resources etc.)

- throughput (number of transmitted and processed messages, parameter: packets per message, number of transmitted bits etc.)
- capacity of the node stores (main and background stores)
- queues (number of messages, packets, bits in different waiting queues)

The highest hierarchical level is the analytical measurement: the analysis of normal operation under dynamically changing conditions. The analysis is an effective tool for understanding processes in computer communication, their impact on the system performance and for proving relationships between network parameters.

2. THE DRAFT OF THE PROGRAM PACKAGE

2.1 The scheme of the program package

The task of the monitor program package is the measurement of deterministic and stochastic parameters of computer network. The monitor (Fig. 1) as a host is connected to a single node of the network. The connection is implemented through a communications subsystem which forwards the signals from the control unit and, optionally, the artificial workload to the network. The signals - to be measured - get to a sampler, the frequency and the time- or event-controlled start of sampling are regulated by the control unit. The sampled signals are the inputs of the measuring subsystem. The components of the measuring subsystem depend on the measuring methods and on the parameters to be measured. After the required computing the evaluation routine keeps a diary on the important results, gives an alarm if it is necessary and collects information for further processing. The workload generator generates different artificial workloads for the network. This workload is optional, we can measure the network also under real normal operational conditions. Fig. 2. shows the scheme of workload generator, it contains four separable modules:

- computer network simulator
- message mix
- operational conditions
- protocol generator

2.2 The components of the measuring subsystem

The parameters of the computer network are deterministic or stochastic. The first decision of the measuring subsystem, is to be seen in Fig. 3. After a second and a third decision, if the parameter is deterministic we can perform the measurements as follows:

- an event detector routine
- a counter routine
- a time measurement routine

If the parameter is stochastic, two decisions are possible. It has to be decided whether or not it is

- a density function

if not, whether or not it is

- an autocorrelation function?

We have three possibilities depending on the answer to the previous questions:

- a histogram routine in the first case
- an autocorrelation routine in the second case
- a crosscorrelation routine in the third case

The output from the measurement subsystem is the input to the evaluation unit.

2.3 Sample modules and their relationships

Figure 4. shows the architecture of some implemented modules. On telecommunication level 1 and 2 two modules are implemented: the basic telecommunications subsystem RTS/BSC (see Chapter 3) which is the measured and the basic measurement module BSCMON (see Chapter 4) which is the measuring part, on level 3 and 4 the network simulator is already implemented and can use the RTS/BSC on line (see Chapter 6). The evaluation of the measurement results is described in Chapter 5.

3. RTS/BSC - THE TELECOMMUNICATIONS SUBSYSTEM

The RTS/BSC package is an implementation of IBM's standard Binary Synchronous Communications (BSC) telecommunication protocol into the OS/8, RTS/8 and COS 300 operating system of Digital Equipment's PDP-8 small computer.

3.1 Telecommunication protocol level 1 - physical line handling

The package implements a synchronous procedure on physical line handling level with bit and character synchronisation which is re-established for each transmission. The package uses half-duplex or full-duplex line/modem over privately owned, leased or switched lines. The package drives a synchronous interface and modem which conform to CCITT V. 24 recommendations. Line speed from 600 to 9600 baud allowed.

3.2 Telecommunication protocol level 2 - line protocol

RTS/BSC implements the point-to-point or multipoint operation procedure. In multipoint environment the small computer can be used on a multidrop line as a control or as a tributary station of a star network. The control station can handle up to 32, the tributary station up to 8 logical addresses. Multiple stations can be simulated with multiple modems, interfaces and copies of the RTS/BSC on a single computer with stations working independently or connected together to simulate point-to-point connection. The algorithm can handle ASCII, EBCDIC or transparent (any) data format. The implemented algorithm enables the computer to be connected to:

- IBM or SIEMENS mainframe using BSC (or MSV1, MSV2) telecommunication procedure;
- to an IBM 2780, IBM 3780, IBM 2770, IBM 3270 telecommunication terminal or a SIEMENS Transdata 840X terminal;
- to a minicomputer using an IBM/BSC telecommunication terminal emulator software package, i.e. PDP-11 with any operating system, PDP-10 etc.;
- to a communication network supporting IBM 2780.

3.3 Telecommunication capabilities on level 3 and 4 - user level

On the user level the RTS/BSC package can be used to add three different features to PDP-8's, OS/8, RTS/8 and to COS 300 operating systems:

3.3.1 Capabilities in the OS/8 or COS 300 operating system

The package can be used to add telecommunication capability like a new standard peripheral device to the operating system. Telecommunication in the OS/8 looks like a new character oriented input/output enabled multiunit (on a multipoint line) device which can be used from any standard OS/8 program or high-level language like:

- OS/8 FORTRAN
- OS/8 BASIC
- OS/8 FOCAL
- OS/8 MINI-COBOL

In the COS 300 commercial operating system the telecommunication lines are implemented as logical units with sequential access. Several logical units can be used from the high-level language of the COS 300, DIBOL, or can be attached to the Data Entry Package of the COS 300. Telecommunication can be performed simultaneously from DIBOL and from some of the foreground displays. The telecommunication link can be used from the utilities of the operating systems like PIP (Peripheral Interchange Program) or in program packages running under the batch monitor of the operating systems.

The capability to handle the telecommunication link from high-level languages makes it possible to easily implement several high-level telecommunication protocol functions and line-monitoring experiments.

Measurements:

- assembling of telecommunication test packages consisting of messages of different length or with special content. Transmitting and receiving of these packages;
- request for local measurement results from central site, assembling of measurement results into special messages and forwarding them to a control centre, receiving of such packages summarizing and analyzing them on central site, performance of network measurements;
- performance of long-term unattended measurement cycles or special measurements under operator's supervision.

Implementation of level 3 and 4 telecommunication protocols:

- implementation of messages which control the program and operating system running on the destination machine. Transmitting of batch control files;
- building of logical lines over several nodes with fix routing;
- implementation of a routing algorithm;
- implementation of packet switching.

Implementation of high-level functions of telecommunication terminals:

The different features of different types of telecommunications terminals can be easily emulated with high-level language programs. Special capabilities such as data handling, handling of special control characters, data compression/expansion, handling/emulating of special local peripherals such as displays, cassettes and performance of background computing can be easily implemented.

3.3.2 Using the telecommunications subsystem in the RTS/8 real-time operating system

The package can also be used in the stand-alone RTS/8. In the RTS/8 the package looks like a standard device handler and its message syntax is compatible to the device driver modules of the standard mass-storage devices. The package enables the user to use the telecommunications capability while running real-time tasks dealing with laboratory measurements or data acquisition. The RTS/BSC can be used to connect several PDP-8's and to distribute the real-time work among them.

3.3.3 Using RTS/BSC as a stand-alone package in OS/8 or COS 300

A stand-alone RTS/BSC package enables the user to emulate on his small computer an IBM 2780 or IBM 3780 or SIEMENS Transdata 840X terminal. Use of the stand-alone package under the BATCH monitor, makes it possible to add telecommunication, to any job stream running on PDP-8.

4. BSCMON - THE MEASURING MONITOR PROGRAM FOR RTS/BSC

4.1 The monitor characteristics

Level 1

The program monitors the direction of the telecommunication line, the line turn commands, the line turn completion and the line turn time-outs (for half-duplex modems) as well as the events of sending or receiving of a character or input character time-outs.

Level 2

BSCMON monitors the major state of algorithm (i.e. send or receive), the bit and character synchronisation phases, the characters sent or received and distinguished the control characters from text characters. It traces the telecommunication algorithm on both sides of line: control phase sequences, bid retries, unsuccessful inquiries;

sending/receiving of data blocks, parity errors (ASCII data) block check errors, positive or negative block acknowledgements and successful or unsuccessful retransmissions. It shows finally also the algorithm error recovery sequences like acknowledge sequence errors and local or remote abort sequences as well.

Level 3

This level appears in form of external events in the monitoring listing. External events are inquiries for logical connections.

Establishing of logical links for send or receive with local logical line number and local user of the link. Arrival of user data request with buffer location and length. Finally the irregular abort or the regular closing of a logical link.

4.2 The method of monitoring

Monitoring is accomplished by several assembler level routines inserted into the level 1 and level 2 assembler code of RTS/BSC. These routines collect the monitored events which are represented by special codes and put these unique codes into a core resident alternating buffer.

An external RTS/8 task (the BSR task) writes one half of the buffer on the mass storage device while the other half is filled with control codes. The file on the mass storage device is used as an endless ring buffer, therefore the last N events are stored in the disk buffer, where N (the length of the disk buffer) is determined by an assembler time parameter. At the end of the telecommunication activities a special end-of-buffer marker is written into the file to indicate the beginning/end of the ring buffer.

4.3 Evaluation of RTS/BSC activities

Evaluation occurs off-line because of the speed of the telecommunication line. A BASIC program converts the data in the control buffer and interprets them, currently a line printer listing is produced which contains the last 2 K events. Also some formatting is done to make listing easily readable:

- consecutive "SYN" characters are compressed
- the listing is broken up on each line turn and on each end of a telecommunication session
- the control characters are commented
- the events are explained.

4.4 Further possibilities

Instead of the listing it is also possible to produce a file containing the compressed information about the telecommunication activities available for any OS/8 FORTRAN or BASIC program for further statistical or analytical evaluation.

5. TELECOMMUNICATION AND NETWORK MEASUREMENTS

For the different measuring experiments the RTS/BSC package can be regarded as the measured and BSCMON as the measuring software, the packages enable us to implement several measuring experiments where the listed parameters can be modified.

5.1 Level 1

- speed of telecommunication line from 600 to 9600 baud
- half-duplex or full-duplex modem/line
- bit synchronisation enable/disable
- number of synchron characters sent/to be received
- time interval between character synchronisation sequences

5.2 Level 2

- package length
- number of packages in a message
- message length
- content of the package (special characters, long sequence of special characters)
- delay between send/receive of packages

5.3 Level 3

- using of multiple logical lines
- different loadings of simultaneously used logical lines
- different length of messages for simultaneously used logical lines
- abort of selected logical lines

5.4 Loading of the computer

- modifying of the HW throughput of the computer
- DMA (cycle stealing) loading

- number of external interrupts and software interrupts overhead
- high priority task level load
- background computing load

6. NETWORK SIMULATOR WITH VARIABLE PARAMETERS

6.1 Our objectives

The simulator has two purposes:

- observation of the behaviour of computer networks
- testing of prospective nodes or hosts under a wide variety of shifting conditions.

This means in the first case to establish the problems of the model to be measured and in the second case to devise a measuring system (containing the simulator as signal generator and test load).

We attacked the problem with a simple, highly modular simulator. It is possible to insert a measuring module into this simulator or to insert the simulator itself in a measuring system or to let them work concurrently in a multiprogram environment.

Its chief part is a driver program which can model any traffic network (net of roads, railway, telephone, neural structures, any formation on which the load forms a stochastic process).

The simulated network is produced as a series of discrete states. The transitions are observed periodically - the time-unit of the simulator is accordingly a period, composed of 100 tacts.

The driver program accepts some series of signals on which it performs the acts corresponding to forwarding these series. The accepted signals may be given

- by the random number generator of the driver
- by the experimenter (via the teletype console)
- by a host or a node of a real or simulated network (via the appropriate interface).

The present assembly interprets these series as message traveling on a computer network. Accordingly, the meaning of the signals are:

1. identifier of the source machine
2. identifier of the destination machine
3. identifier of the node where the message actually resides
4. data describing the structure and the length of the message
5. the age of the message (i.e the number of tacts elapsed since its entry into the traffic).

(The last is of course zero at the time of acceptance and is incremented by the simulator during the message's active life.)

As the messages proceed, the state of the simulated system changes; the driver observes and reports the relevant events.

6.2 The driver program

The driver is responsible for

1. the timing of message departure
2. the timing of pass-overs of messages
3. the timing of the break downs of the network's various entities
4. the activating of the segments which carry out the above actions
5. initiating a report on the age and actual residence of each message traveling on the network after every period
6. initiating a snapshot on the state of the network as a whole after a given integer number of periods.

In order to fulfill its duties, the driver program requires the following data:

1. the number of nodes of the simulated network
2. a seed for the random number generator
3. the number of periods after which it has to give the snapshot

4. the functional parameters:

- the average number of departing messages per period
- the average number of line breakdowns per period
- the fraction of faulty deliveries (the pass-over of which has to be repeated)
- the data concerning the structure of the messages (maximal number of packets per messages - if zero, then variable length is supposed).

These data are indispensable for the proper working of the simulator. Additionally the experimenter may assign two adjacent nodes which are to be disconnected during the first period after their designation. He may suppress the period-reports too.

The simulator request the above data from the experimenter or the testrun operator as follows:

- all of them during the initiating dialogue with the frame
- 2 or 3 after the specified number of periods (this number is given in the 3rd part of the dialogue).

The seed controls the simulation process. If it is positive, the whole dialogue takes place, if zero, the already given parameters prevail, if negative, the simulation process terminates itself. In the latter two cases the 4-th part of the dialogue is skipped, but the snapshot is duly given.

The entries of the messages and the line breakdowns form Poisson processes with stationary increments, with a., resp. b. as parameters for the underlying distributions. The distribution of the faulty (and repeated) passovers is uniform, that of the length of the messages is exponential.

The first question of the dialogue has some far reaching implications.

6.3 The exchangeable units

If the simulated network is ring structured, or serially structured, then the number of nodes is quite enough for the program.

However, if the pattern is more complicated a wide variety of

forwarding and routing techniques are possible and it can become desirable to test and analyse any of them. This happens by means of different interchangeable forwarding routines inserted into the frame and called by it whenever necessary.

The driver has no knowledge of the technicalities of the forwarding processes, neither of their strategies, nor of the topology and the technical data of the network.

The forwarding routines may need various (and rather different) information on the network. In this case it is the INPUT program-segment which calls for the network description and puts in into the prearranged tables. The INPUT is the first exchangeable segment; it has to be written according to the needs of the updating and routing strategies.

After the intake of the description through the INPUT segment

- and of the parameters via the 2.-4. parts of the dialogue
- the steps initiated by the driver are executed by the following exchangeable segments
- the segment UPDATE enlists the generated or accepted message into the queue of the first station of its patch; this segment arranges the pass-over of the message to the succeeding stations of its propagation - the pass-over process simulates the procedures of the prevailing protocols in a simple way and updates the tables of the program after every event
 - UPDATE is called only by the driverprogram -
- the segment SHUTDOWN actualizes the line breakdowns between one or several pairs of nodes determined either by the experimenter or by the random-generator or both; when a node loses in this way its last living line, the segment notifies the simulator with the registration of its shutdown in the appropriate tables (tables used by UPDATE e.g.)
 - SHUTDOWN is called only by the driverprogram -
- the segment ROUTING (if there exists any routing) handles the tables of the path-generation, it assigns the next station or the whole to the messages

- ROUTING may be called by UPDATE and SHUTDOWN both but the driverprogram never knows when.

The exchangeable segments are chosen when loading the simulator. They determine fully the mode of the simulation.

The driver activates the snapshot-segment called REPORT. This works on the tables of the whole program, generates its reports and delivers them either on the lineprinter or by spooling on the disk.

7. CONCLUSIONS

As the development of remote data processing systems came to prominent importance, our attention turned to the monitoring of small- and medium-scale computer networks. We worked out a monitor architecture and some basic modules. The highly modular architecture of the described network measuring package enable us to build up a very flexible measuring package being able to execute a wide variety of network measuring experiments.

LITERATURE

- [1] L. Kleinroch: Queueing Systems Vol. II. pp. 422-515
John Wiley & Sons, New York, 1976.
- [2] G.J. Nutt: Computer System Monitors
Computer 1975. nov. pp. 51-61
- [3] L. Svobodova: Computer Performance Measurement and Evaluation
Methods: Analysis and Applications
Stanford University, Technical Report No. 72
- [4] BSC General Information
IBM GA 27-3004
- [5] IBM 3780 Data Communication Terminal
IBM GA 27-3063
- [6] RTS/8 Users Manual
DEC-08-ORTMA-B-D Digital Equipment Corporation, 1975.

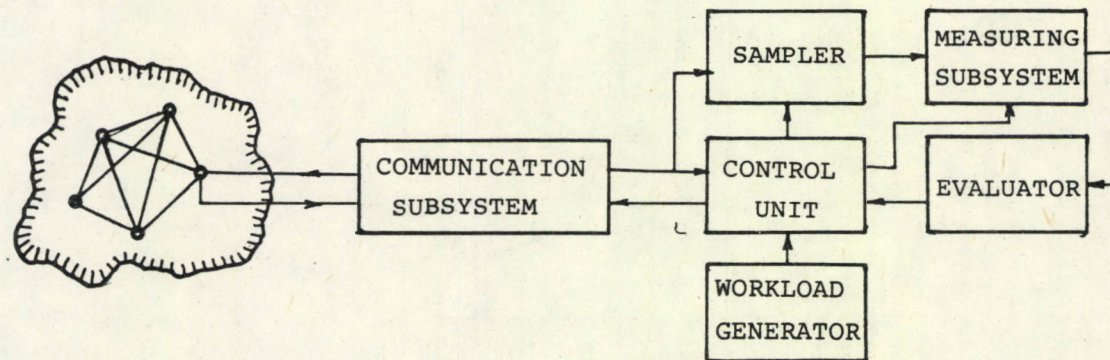


Fig.1. The scheme of monitor

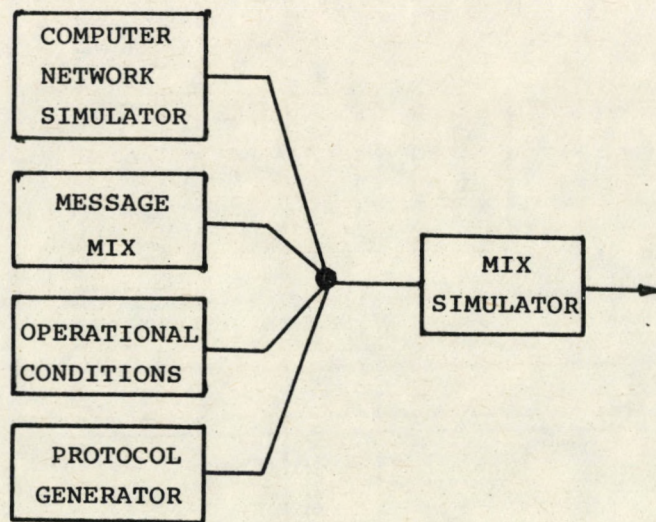


Fig.2. The workload generator

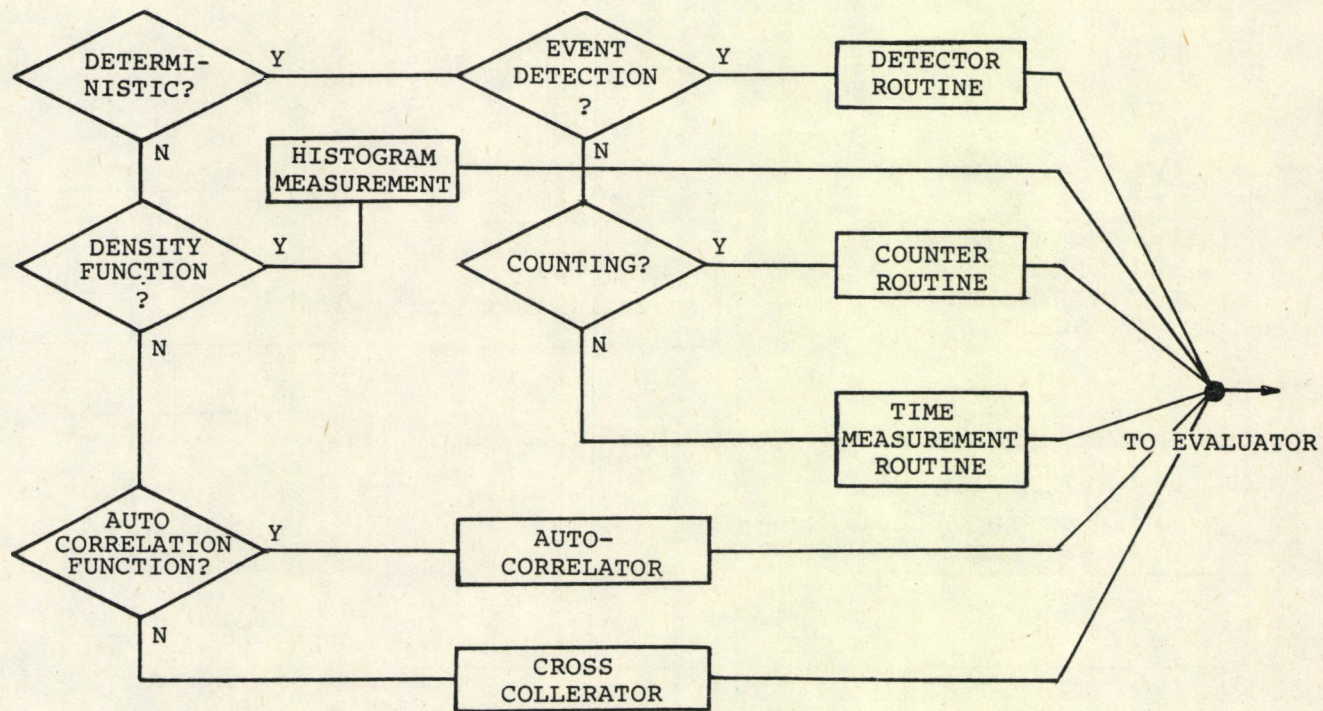


Fig.3 The scheme of measurement subsystem

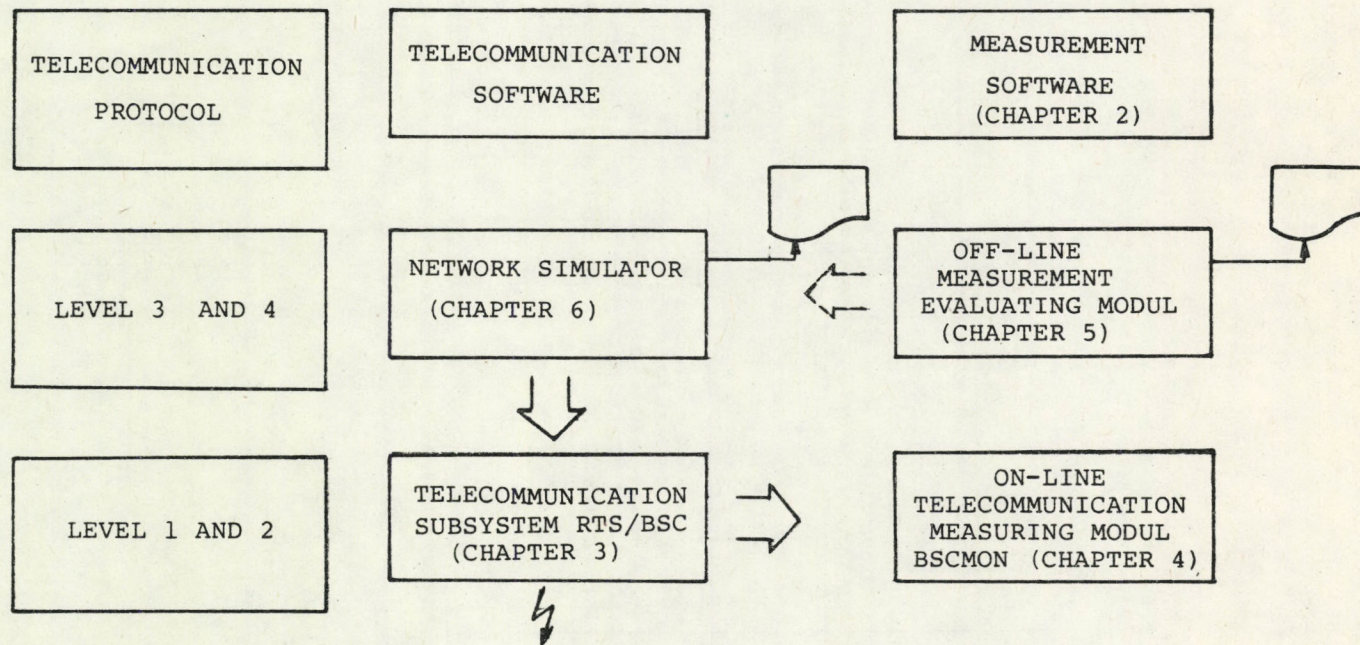


Fig. 4. The network measuring package

THE UNIVERSITY OF CHICAGO
LIBRARY
540 EAST 57TH STREET
CHICAGO, ILL. 60637
TEL: 773-936-3000
WWW.CHICAGO.EDU

67.114



Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Sándory Mihály
Szakmai lektor: Csákány Antal
Nyelvi lektor: Ambrózy Denise
Példányszám: 345 Törzsszám: 80-772
Készült a KFKI sokszorosító üzemében
Felelős vezető: Nagy Károly
Budapest, 1980. december hó