OPIC
OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA

CIPO
CANADIAN INTELLECTUAL PROPERTY OFFICE

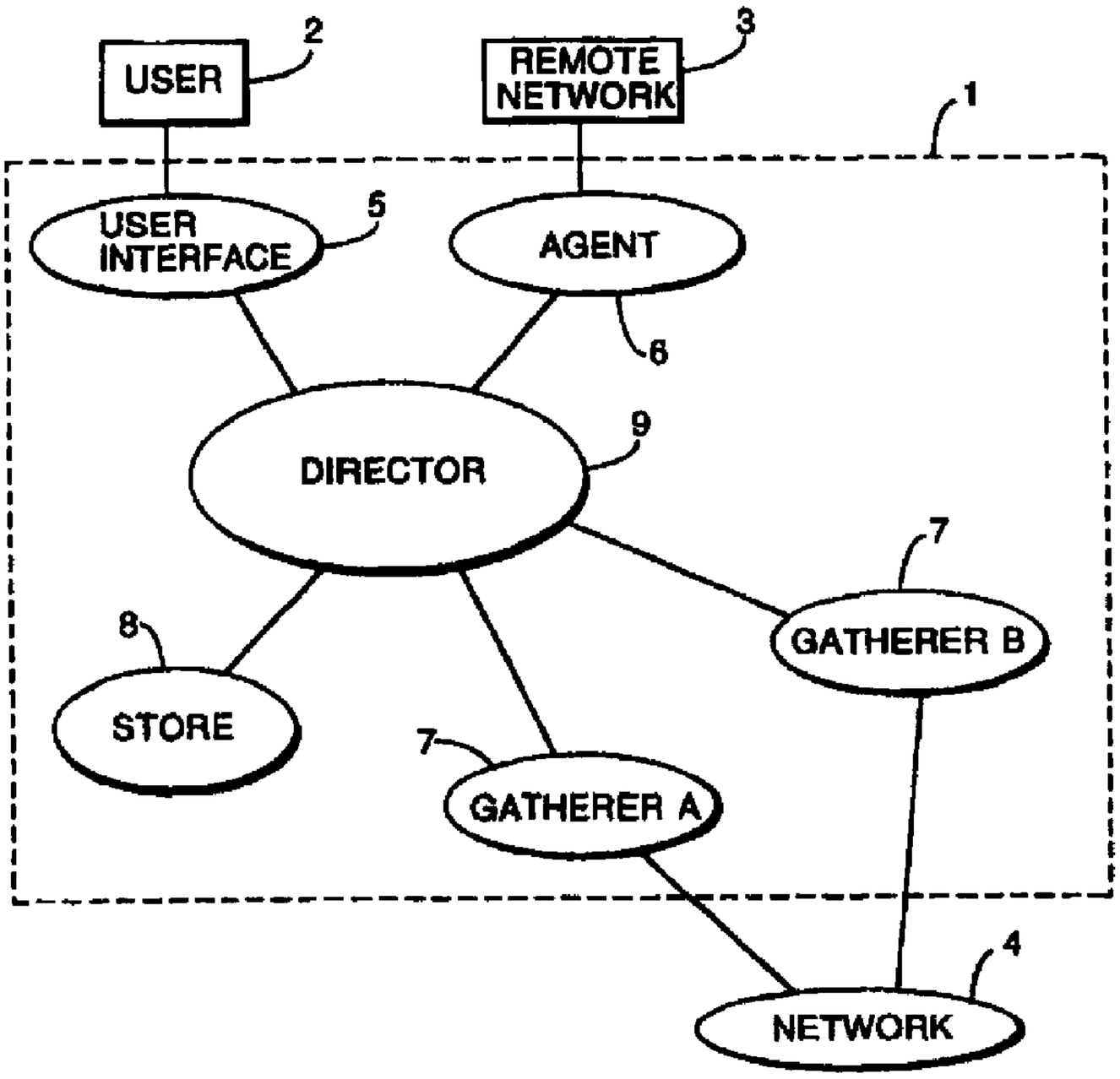
(12) (19) (CA) **Demande-Application**

(21) (A1) **2,285,585**
(86) 1998/04/15
(87) 1998/10/22

(72) ADAMS, JOHN LEONARD, GB
(72) SPENCER, TIMOTHY JOHN, GB
(72) COOPER, NICHOLAS JEREMY PAUL, GB
(72) PHILLIPS, IAIN WARWICK, GB
(72) PARISH, DAVID JOHN, GB
(71) BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY, GB

(51) Int.Cl.[6] H04Q 11/04
(30) 1997/04/16 (97302619.8) EP

(54) **ESSAI DE RESEAU**
(54) **NETWORK TESTING**



(57) L'invention concerne un système d'essai et de surveillance d'un réseau dans lequel des paquets d'essai sont envoyés entre des stations d'essai reliées à un réseau, afin de déterminer les caractéristiques de fonctionnement du réseau. Les résultats de l'essai sont

(57) A network testing and monitoring system in which test packets are sent between testing stations connected to a network, to determine the performance characteristics of the network. The results of the test are analysed and may be used to automatically control the

OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(21) (A1) **2,285,585**
(86) 1998/04/15
(87) 1998/10/22

analysés et peuvent être utilisés pour commander automatiquement une nouvelle génération de paquets d'essai dans le réseau afin de localiser et d'isoler des défaillances du réseau et obtenir d'autres informations sur les caractéristiques du réseau. Un système de communication d'incidents fonctionne avec les résultats de l'essai sur une base continue pour déterminer si des effets significatifs en termes de fonctionnement du réseau se produisent. Si de tels événements se produisent, ils sont communiqués à l'exploitant de réseau comme incidents de réseau.

further generation of test packets across the network to locate and isolate network failures and to obtain further information about the network characteristics. An incident reporting system operates on the test results on a continuing basis to determine whether any effects which are significant in terms of network operation are occurring. If such events do occur, they are reported to the network operator as network incidents.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) **International Patent Classification** 6 : **H04Q 11/04** — **A1**

(11) **International Publication Number:** **WO 98/47308**

(43) **International Publication Date:** 22 October 1998 (22.10.98)

(21) **International Application Number:** PCT/GB98/01091

(22) **International Filing Date:** 15 April 1998 (15.04.98)

(30) **Priority Data:**
97302619.8 16 April 1997 (16.04.97) EP
(34) Countries for which the regional or international application was filed: GB et al.

(71) **Applicant** *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

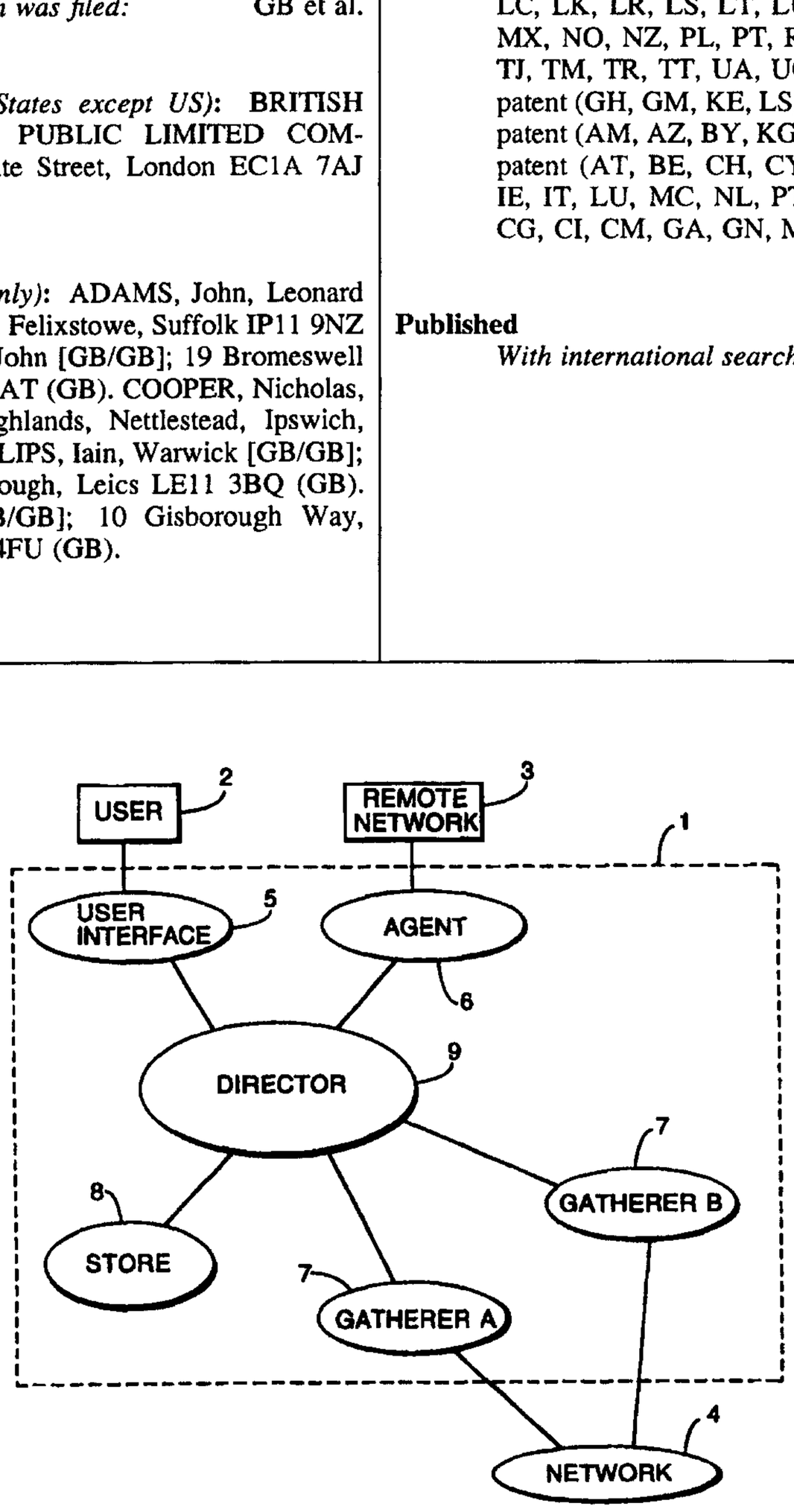
(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: ADAMS, John, Leonard [GB/GB]; 24 Keswick Close, Felixstowe, Suffolk IP11 9NZ (GB). SPENCER, Timothy, John [GB/GB]; 19 Bromeswell Road, Ipswich, Suffolk IP4 3AT (GB). COOPER, Nicholas, Jeremy, Paul [GB/GB]; Highlands, Nettlestead, Ipswich, Suffolk IP8 4QS (GB). PHILLIPS, Iain, Warwick [GB/GB]; 15 Curzon Street, Loughborough, Leics LE11 3BQ (GB). PARISH, David, John [GB/GB]; 10 Gisborough Way, Loughborough, Leics LE11 4FU (GB).

(74) **Agent:** EVERSHED, Michael; BT Group Legal Services, Intellectual Property Dept., Holborn Centre, 8th floor, 120 Holborn, London EC1N 2TE (GB).

(81) **Designated States:** AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) **Title:** NETWORK TESTING



(57) **Abstract**

A network testing and monitoring system in which test packets are sent between testing stations connected to a network, to determine the performance characteristics of the network. The results of the test are analysed and may be used to automatically control the further generation of test packets across the network to locate and isolate network failures and to obtain further information about the network characteristics. An incident reporting system operates on the test results on a continuing basis to determine whether any effects which are significant in terms of network operation are occurring. If such events do occur, they are reported to the network operator as network incidents.

Network Testing

Field of the Invention

This invention relates to the testing of a communications network, particularly but not exclusively to a testing and monitoring apparatus for measuring and managing the performance of a communications network.

Background

It is known to test the characteristics of communications networks by performing user initiated tests, in which a stream of test messages, such as data packets in a packet based network, is transmitted across a network between test stations. By recording the transmission and receipt times of each packet, various network performance characteristics may be determined, such as the packet delay over a particular route, or the proportion of packets that become corrupted or lost.

For example, a commercially available system such as the Alcatel 8640 Broadband Test System is intended, inter alia, for testing transmission characteristics of communication paths through Asynchronous Transfer Mode (ATM) communications networks.

The Alcatel 8640 system enables a user to select a test to be performed on the network and returns performance data to the user to be analysed off-line. The user may then initiate further tests if required.

In a complex network, off-line analysis of the results of a series of user-initiated tests is a time-consuming and difficult process, relying on the user to initiate appropriate further tests based on analysis of previous test results. The delays inherent in this process may prejudice the location of certain types of error, for example, intermittent network faults.

A device and method for measuring the performance characteristics of a communication path in an ATM network is disclosed in EP-A-0 528 075. The disclosed device includes a test packet generator for generating test packets and transmitting them through a communication path in a switching network, and a packet analyser for receiving the test packets from the network and measuring the performance characteristics of the path. To ensure that the existing traffic on the network is not disturbed by the measurements, the test packets are modified live traffic-carrying packets in which the communication data has been replaced by performance measurement data.

A fault diagnosis scheme that can continually monitor an ATM connection length in a system is disclosed by Itoh & Miyaho: "Function Test Methods using Test Cells for ATM Switching System", Communication-Gateway to Globalisation. Proceedings of the Conference on Communications, Seattle, June 18-22, 1995 and Volume 2, 18 June 1995, Institute of Electrical and Electronics Engineers, pages 982-987. This describes a connectivity testing mechanism whereby test cell fold-back at different points in the network is used to identify the fault location.

An object-oriented system for supervising and detecting faults in a complex system such as a telecommunications network is disclosed in WO95/28047. This is based on a chain of fault detection elements located at particular points in the system so as to enable each of them to detect a particular fault.

Summary of the Invention

It has been recognised that the results of a given network test may reveal that a further test or series of tests is desirable, and that the testing apparatus may be arranged to automatically initiate those further tests, where, for example, the results of previous tests reveal that insufficient data is currently available to answer a query from a user, or where, during the testing cycle, more specific information

about a particular network condition, such as an unusually high network loading, is required.

A given test may, for example, consist of the background monitoring of one or more transmission characteristics of a signal path in a communications network. Such a test may continue for long periods of time, even semi-permanently, and entail the transmission over the path of a stream of test packets at a relatively low rate. The results of such a test may be analysed on a continuous basis to search for anomalies, for example packet delays or packet loss exceeding a pre-determined threshold value. On the occurrence of such an anomaly, a new test pattern is automatically initiated, which may entail transmission of test packets over the path at a higher rate for a short period. In turn, analysis of the results of the new test pattern may trigger the automatic initiation of another new test or a new instance of a previous test.

The present invention provides apparatus for testing a signal path in a communications network, comprising means operative to generate test signals to be transmitted over the path, means operative to analyse test signals received from the path so as to determine a characteristic of the path,
means responsive to the path characteristic to determine if additional testing in the network is required, and means operative to automatically initiate said additional testing in the event that it is determined to be required.

The nature of the failure in, for example, a signal path in a communications network, may be a total loss of service, increasing packet delay due to increased user load on the network, or some other measurable event. The apparatus according to the invention has particular advantages where the nature of the failure is not immediately apparent and where the emergence of a particular trend may be used to initiate a more detailed investigation automatically, so that a likely source of failure may be isolated before it occurs, and so that the performance

characteristics of a signal path, including its loss and delay characteristics, may be more accurately determined.

The apparatus according to the invention may allow the setting of user defined parameters, such as threshold levels, which may be compared with the determined network characteristics to automatically initiate further testing. For example, when a threshold specifying a maximum network loading is exceeded, the apparatus may initiate further testing to determine the cause of the loading, or to investigate the current loading on an alternative network path.

The apparatus according to the invention may also allow the user to request information about a network characteristic. In this case, there may be insufficient data available to determine that characteristic to within the parameters set by the user. Automatic testing of the network may then be initiated to generate further data to allow that characteristic to be sufficiently determined.

Further, the apparatus may include means operative to automatically switch between a number of signal paths, when the analysing means indicates that a particular fault has occurred or desired threshold has been exceeded.

In accordance with the invention there is also provided a communications network, comprising at least one signal path, means operative to generate test signals to be transmitted over the path, means operative to analyse test signals received from the path so as to determine a characteristic of the path, means operative to determine if additional testing in the network is required in dependence on the determined path characteristic, and means operative to automatically initiate said additional testing in the event that it is determined to be required.

In accordance with the invention there is further provided a method of testing and monitoring a signal path in a communications network, comprising generating test signals, transmitting the test signals over the path, receiving the test signals from

the path, analysing the received test signals to determine a characteristic of the signal path, determining if additional testing in the network is required in dependence on the determined path characteristic, and

automatically initiating said additional testing in the event that it is determined to be required.

In accordance with a further aspect of the invention, there is provided automated incident reporting apparatus for reporting network incidents in a communications network, based on a predetermined plurality of performance parameters for said network, comprising means operative to systematically compare first and second network performance parameters selected from said plurality of parameters, and means responsive to said comparison to determine whether said first and second parameters are equivalent in accordance with predetermined criteria, and in the event that said parameters are not equivalent, to report the non-equivalence as a network incident.

There is also provided network testing apparatus comprising means operative to systematically determine first performance parameters relating to a performance characteristic of a communications path in a communications network, means operative to compare said first parameters with second network performance parameters, and means responsive to said comparison to determine whether said parameters are equivalent in accordance with predetermined criteria, and in the event that said parameters are not equivalent, to report the non-equivalence as a network incident.

In accordance with the invention, there is further provided a method of automated incident reporting in a communications network, based on a predetermined plurality of performance parameters for said network, comprising systematically comparing first and second network performance parameters to determine whether said parameters fall within a predetermined relationship, and in the event that said

parameters do not fall within said relationship, reporting this event as a network incident.

The first and second network performance parameters may respectively represent the same parameter measured at different respective times. Alternatively, the second parameter may represent a predetermined network performance.

The ability to perform automated systematic testing of a network allows the correlation of network events to determine underlying patterns, as opposed to having to consider each network event in isolation.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a schematic top level diagram showing the inputs to and outputs from the testing and monitoring apparatus according to the present invention;

Figure 2 is a schematic object based diagram of the apparatus of Figure 1;

Figure 3 is a schematic diagram showing the general form of a TicketedJob object;

Figure 4 shows the TicketedJob object for a Retrieve to Store operation;

Figures 5a to 5f show a series of TicketedJobs illustrating the implementation of the adaptive nature of the apparatus of Figure 1;

Figures 6a to 6f show the TicketedJobs illustrating the implementation of a further type of adaptive behaviour;

Figure 7 is a schematic diagram showing a network configuration based on an SMDS network incorporating the apparatus of Figure 1;

Figure 8 is a schematic block diagram of a timing card used in the implementation of the Gatherers shown in Figure 7;

Figure 9 shows a network management system according to a further aspect of the invention including an automated incident reporting apparatus;

Figure 10 shows schematically a table of performance data on which the automatic incident reporting apparatus of Figure 9 can operate;

Figure 11 is a schematic diagram illustrating the automated incident reporting apparatus of Figure 9; and

Figure 12 illustrates the interaction between the testing and monitoring apparatus and the automatic incident reporting apparatus shown in Figure 9.

Detailed Description

Referring to Figure 1, a testing and monitoring apparatus 1 accepts requests from a human user 2 and/or through a remote network 3 to perform monitoring and testing functions on a test network 4. The results of the tests may be returned to the user 2 and/or the remote network 3.

The apparatus 1 will be described based on object oriented principles in terms of a number of executable objects. The physical implementation of the object oriented scheme will be described below with reference to Figure 7.

Referring to Figure 2, the apparatus 1 comprises a User Interface object 5 to interface with the user 2, and an Agent object 6 to interface with the remote network 3. An interface to the test network 4 is provided by one or more Gatherer objects 7 which are responsible for the gathering of test information from the network 4. A Store object 8 holds and processes the gathered information. There may be a plurality of Store objects within the apparatus 1. A Director object 9 is a continually running object which provides central control of the apparatus 1.

The User Interface object 5 provides an interface for a human operator to control and view the system. It implements text and graphic based applications to display information held in the Store 8 to the user 2. It also supports forms to be filled in by the user to query the Store and to enable input of commands to initiate testing of the network 4.

The Agent 6 is an automated interface between the Director 9 and any other network management object, for example, a remote network 3. The Agent 6 is able

to schedule tasks to be executed by the Director 9 and passes the results back to other objects. For example, following a request from the remote network 3, the Agent 6 may schedule tasks to notify the remote network 3 when traffic levels on the network 4 reach a particular threshold. There may be a number of Agent objects 6 for each Director object 9 in the apparatus 1.

The operation of the apparatus 1 is based around the execution of individual tasks, in the form of objects known as Jobs. On receipt of a command from the user 2 or a request from the remote network 3, the User Interface 5 or the Agent 6, respectively, creates a TicketedJob object 10, shown schematically in Figure 3, which is immediately submitted to the Director 9 for subsequent execution. The TicketedJob object 10 comprises a Ticket object 11 and a Job object 12. The Ticket 11 contains as a sole parameter the time at which the Job is to be executed by the Director 9 (Execution Time), which may be an absolute time or a time relative to another event.

A Job 12 represents an action within the system, and has a set of parameters, which depend largely on its type, for example, the name of the Gatherer 7 which is to initiate a network test, together with a communications path to be tested in the network 4 and the time at which the test should be executed on the Gatherer object, which may or may not be the same as the execution time of the Job controlling the test.

The Director 9 maintains an interface between the User Interface 5 and currently running Jobs. Once the TicketedJob 10 has been created, the User Interface 5 submits it to the Director 9, and the Director adds it to a Job list. At this stage, the Director 9 creates a JobRunner object to supervise the execution of the Job, which includes maintaining a Job status. The status may be one of:

a) Waiting: the Job's execution time has not yet arrived;

b) Running: the Job is currently running, and may be waiting for results from a Gatherer or Store object; or

c) Finished: the Job has finished.

If the Job is in a Waiting state, then the JobRunner will wait until the Ticket (execution) time, when it will run the Job.

The Job runs in its own thread on the Director and calls methods in Gatherer and Store objects to do the required processing. Threads and methods are well-known concepts in object-oriented programming.

The Store 8 holds information about the measurements made on the network 4. The Store will provide answers to queries from the Director 9. Queries are created by a Job running on the Director and passed to the Store as objects. The Store responds with a Result, which may be a single object or a stream of objects to be processed by the Job originating the Query.

Each Gatherer object 7 comprises a number of controlling methods including;

Start gathering

Stop gathering

Retrieve gathered information

Maintain Gatherer

Provide Status

In the case of Jobs which initiate testing on a Gatherer 7, once the test has been initiated, the initiating Job finishes and is removed from the Job list. The test is then run under the control of the appropriate Gatherer 7.

A Job running on the Director 9 initiates a network test by passing a StartTest signal to the appropriate Gatherer 7, for example, Gatherer A, with a number of parameters including the test identifier (test_id), number of packets, packet size

and packet interval, and test route, or destination Gatherer, for example, Gatherer B.

Gatherer A transmits a packet over the network and continues to transmit packets as specified in the test parameters. The time at which it transmits each packet is recorded by Gatherer A in a log file. At predetermined intervals set by a scheduled RETRIEVE Job, such as once per day, the log file is sent to the Director 9 which sends it on to the Store 8. The Director 9 may be arranged to automatically retrieve log files as soon as a particular test has finished.

Gatherer B receives test signals from the network 4 and logs the receive time of every test signal which it receives. The log file is also sent to the Store 8 via another RETRIEVE Job running on the Director 9.

The Maintain Gatherer controlling methods allow the user some control over the operation of the Gatherers 7 in the event of a system failure.

The time at which a test is initiated on the Gatherer 7 may or may not be the same as the execution time of the Job, since it may be desirable for the Job to begin execution on the Director 9 at a separate time to its invocation of methods in Gatherers 7. For example, the Director may initiate testing depending on its knowledge of the Gatherer's current workload.

Jobs may be scheduled to repeat after a particular event or time. For example, a weekly report Job may be submitted to run every week but only after the Job to process and retrieve the measurements from the Gatherer 7.

Jobs with status Waiting may be dependent on another Job finishing first. Such Jobs will not be run until a second Job provides the Waiting Job with a start time. For example, referring to Figure 4, to query the Store after measurements have been retrieved from Gatherer 7 by a RETRIEVE Job arbitrarily referred to as

RETRIEVE_150. the User Interface schedules a further Job 12, arbitrarily referred to as QUERY_150. in which the Ticket time is specified to be "after RETRIEVE_150".

The status of each Job is shown below:

Identifier	RETRIEVE_150	QUERY_150
Status $t_{initial}$	Running	Waiting
$t \uparrow$	Finished	Running
t_{final}	Finished	Finished

When RETRIEVE_150 has finished, its status changes to Finished and the status of QUERY_150 is changed to Running. This Job queries the Store 8 and return the results to the user.

The adaptive nature of this embodiment of the invention may be seen by reference to the example illustrated in Figure 5. The user may require that the network 4 is monitored for anomalous behaviour. for example, to ensure that the total message delay remains below a threshold value supplied by the user. In the event that the delay exceeds this threshold value, the user may require a detailed breakdown of the location and cause of this fault.

The user enters these requirements via the user interface.

The apparatus schedules Jobs to query the Store 8 and compare measurements made by previously run tests with the user defined threshold value. The results of this comparison may cause the Director to automatically initiate further tests to perform more intensive testing of the network if the threshold value has been exceeded.

This example assumes that a background test pattern is already running, arbitrarily referred to as test_id_10, in which, for example, a continuous stream of test packets is transmitted across a network path at a relatively low rate, for example, 1 packet per minute. It also assumes the existence of a RETRIEVE Job, arbitrarily referred to as RETRIEVE_20, which retrieves the log files from Gatherer A and Gatherer B to the Store at midnight every day. Fig 5(a) illustrates the TicketedJob for RETRIEVE_20.

The User Interface 5 schedules a single monitoring Job to perform the comparison with the user-defined threshold value. Figure 5(b) illustrates the TicketedJob object for the new scheduled Job, arbitrarily referred to as MON_1. It is arranged to run after RETRIEVE_20 has finished and therefore after the measurements produced by test_id_10 have been sent to the Store. This TicketedJob is passed to the Director 9. The function of MON_1 is to determine whether the average packet delay exceeds the predetermined threshold.

After RETRIEVE_20 has finished, the status of MON_1 is changed to Running. This Job creates a Query which is passed to the Store object to request it to calculate the average packet delay, for example on a per day basis. The Store object 8 contains a method to calculate average delay based on the difference between the transmit time and receive time for each test packet. The result of this calculation is returned to the Director 9 where the running Job MON_1 performs the comparison. If the comparison indicates that the threshold value has been exceeded, the Director 9 may then automatically schedule further Jobs to perform more intensive testing, rather than, or as well as, notifying the user.

For example, the next Job, MON_2, shown in Figure 5(c), may be scheduled to run immediately and further query the Store 8 to determine the time periods in which the delay occurred, for example by looking at average delay on a per hour, rather than a per day, basis. If MON_2 identifies that the exception occurred in a certain time period, it may then initiate three further Jobs.

The first of these further Jobs, arbitrarily referred to as TEST_200, is set to run immediately but to initiate a further test in the relevant time period on the following day, in which the rate at which data packets are sent over the network path is increased to one per second. This test is arbitrarily referred to as test_id_223. For example, MON_2 determines that the time of interest is 6pm to 8pm on the following day. The TicketedJob for TEST_200 is shown in Figure 5(d).

MON_2 may also schedule a further RETRIEVE Job, arbitrarily referred to as RETRIEVE_200, to execute after the test initiated by TEST_200 will have finished, as shown in Figure 5(e).

Finally, MON_2 may schedule a QUERY job to return the results to the user, scheduled to execute after the RETRIEVE Job has finished, as shown in Figure 5(f).

As an alternative, where a Gatherer 7 is located at an intermediate point in the network path, the Director 9 may schedule a new test with an altered route to determine the location of the fault.

A further example of a different type of adaptive behaviour is where the user requests information which requires a sufficient number of data points to be available to enable a meaningful calculation. For example, a user may request information as to whether a service level agreement is being met on a particular route to a given level of confidence. The User Interface 5 again schedules a Job to interrogate the Store 8, as shown in Figure 6(a). In this case, however, the Store may not contain sufficient information to answer the request. The Director 9 therefore automatically initiates three further Jobs, the TicketedJobs for which are shown in Figures 6(b) to (d). The first Job is to run immediately and to create an appropriate test pattern to generate the required information, the test arbitrarily

referred to as test_id_500, as shown in Figure 6(b). The Director 9 then automatically schedules retrieval of the log files from the Gatherers to the Store after the completion of test_id_500, as shown in Figure 6(c). The third Job is to wait until the second Job is complete and then to query the Store 8 and return the results to the user, as shown in Figure 6(d).

The nature of the faults that may be detected is not limited to increased packet delay, but could be total loss of service or some other measurable event. For each time period, for example one hour of a week, a Job may be submitted to process and calculate some measurement information, such as the average delay of the slowest 5% of packets. By storing these values and comparing over time, any changes may be noted. Such changes include:

a significant step change in delay;

a step change in delay that bursts for a short period of time;

a gradual change in delay over time;

a change in the delay of a proportion of packets only;

repeated individual packets with significantly greater delay than the average;

an asymmetric difference in delay, where the one-way delay in one direction is significantly different to that in the reverse direction;

a duplicated packet, including the asymmetric and burst cases; and

a large number of lost packets.

By scheduling a series of tests covering important areas of the network topology, and building an analysis system based on the test results and that topology, any detected failure in one path can automatically initiate further investigation of intermediate paths. When the apparatus has isolated the problem area topologically, a message can be sent to the operator to suggest reconfiguration of the network around the failure. Alternatively, an automatic reconfiguration could be implemented to redirect traffic around the failure point, while that is undergoing repairs.

The detailed structure and operation of this embodiment will be described with reference to Figure 7, in which the network under test is a Switched Multimegabit Data Service (SMDS) network 4. All of the objects referred to above are implemented using the JAVA object-oriented programming language. Where objects are implemented on computers which are connected only through a network, they are known as remote objects. Such remote objects may be accessed using a standard JAVA technique known as Remote Method Invocation (RMI). This technique enables objects to be passed to one another across networks.

The Director 9 and Store 8 objects run on a Sun Microsystems Workstation 30 which is connected to the network 4 via a router 31, for example the Cisco 2500, using dedicated Ethernet connections 32. The User Interface object 5 is implemented on a standard IBM compatible PC 33 also connected to the Sun Workstation 30 via an Ethernet link 32. The Gatherer objects 7, Gatherer A and Gatherer B, are implemented on monitor stations 34 and 35 respectively and comprise standard PCs fitted with proprietary timing cards. The timing cards are required as a standard PC is unable to provide the required timing accuracy and synchronisation. The Agent 6 is also implemented on a standard PC 36. The monitor stations 34, 35 and Agent station 36, are connected to the network 4 via dedicated Ethernet links 37 through routers 31.

Referring to Figure 8, the timing card comprises a Motorola Global Positioning System (GPS) receiver 40, an Intel 8751 microprocessor 41 and a 28 bit wide 10MHz counter 42. The GPS receiver 40 is programmed to provide time of day and day of year information, together with a 1Hz pulse. GPS receivers are synchronised to generate this pulse with a claimed accuracy of under one microsecond anywhere in the world. This is referred to as the 1 pulse per second option (1PPS). The counter 42 is reset on arrival of the 1PPS signal and then allowed to run for one second. In addition, the total count is recorded when the counter is reset giving the full count. The microprocessor 41 decodes the time and date from the GPS receiver 40 and encodes this into a 32 bit quantity. The PC

interface 43 has latches 44, 45 and 46 which can at any time latch the counter and date/time information and then retrieve the full count, the date/time information and the latched, or partial, count.

When each data or test packet is sent by the transmitting monitor station 34, a time stamp is inserted in the information field of the packet, which consists of the contents of the monitor station's clocked counter 42 at the instant that the test packet is sent.

When the test packet is received at the receiving monitor station 35, the time of receipt is recorded by the monitor station 35.

An operator may access the system by, for example, starting a local JAVA application on the user terminal. This will result in a window appearing on the display of that terminal with a list of known Directors 9. Clicking on a Director name allows that Director to be selected. Further windows will show the Jobs list and list the Stores 8 and Gatherers 7 that are controlled by the selected Directors 9. Clicking on a Job will display a dialog showing summary information about that Job and allow for the Job and its status, as well as its execute time, to be altered or deleted.

Clicking on Store or Gatherer objects will show a dialog allowing maintenance of those objects.

Data produced by the testing and monitoring apparatus described above may be used to perform automated incident reporting, whereby notable changes, whether positive or negative, in the performance of the network, referred to herein as incidents, can be notified to a network operator. Figure 9 shows a network management system according to the invention which includes both the testing and monitoring apparatus and an automated incident reporting apparatus according to a further aspect of the invention. The testing and monitoring apparatus 50 tests a

network 4 and data from the testing is available to the automated incident reporting apparatus 51, which can itself control the testing apparatus 50 if it requires further test data via a link 52. Network incidents produced by the reporting apparatus 51 are notified to a network operator 53.

For example, from the test data produced by the testing and monitoring apparatus, the automated incident reporting apparatus according to the invention can produce network performance data in the form of a series of performance parameters P_x for different time periods over various network paths, as represented in Figure 10 in the form of a table.

Referring to Figure 10, P_1 may, for example, represent a particular level of packet loss over time period T3 over network path NP5 and P_2 may represent the level of packet loss over time period T4 over the same network path, where T3 and T4 are consecutive time intervals over which systematic testing occurs, for example, periods of a week. A comparison of P_1 and P_2 produces a measure of the increase or decrease in packet loss over path NP5 from one week to the next. This measure may or may not represent a network incident in accordance with predetermined threshold levels. For example, if the measure of packet loss is below a predetermined acceptable difference between P_1 and P_2, then no network incident is reported to the network operator. If, on the other hand, it has been specified that any deterioration in packet loss should be treated as a network incident, for example because it almost invariably has a deleterious effect on network performance, then any such deterioration from P_1 to P_2 is reported to the network operator as a network incident.

Alternatively, rather than making a comparison with performance in the previous time interval, a series of parameters P_x may be compared with a pre-defined or desired path behaviour P_D, and predetermined differences from such behaviour notified as network incidents. For example, assuming that P_D defines a maximum acceptable packet loss of 2 packets/hour, then a network incident is reported if the

packet loss level represented by one member of the series P_x, for example P_2, exceeds this limit.

As well as, or as an alternative to, reporting the incident, the automated incident reporting apparatus can be configured to instruct the testing and monitoring apparatus to automatically initiate further network tests in response to the incident. For example, in the case of excessive packet loss, such further testing can involve scheduling more intensive testing on a daily basis to determine the times at which greatest packet loss occurs.

The comparison between network performance parameters is facilitated by defining a set of performance classes and using, for example, statistical techniques to determine class equivalence. For example, P_1 is defined as a performance class by reference to a set of statistical parameters together with a set of values for those parameters. There are a large number of possible types of class depending on the combination of statistical parameters chosen. For example, for a particular type of class the set of parameters may be the median delay of all packets carried over a particular communications path during a specified interval, the standard deviation of the delay and the period of any repeating delay spikes, which represent abnormally long packet delays over short periods of time. A set of values for the median delay, standard deviation and spike period is then determined from the data gathered for period T3 to fully define the class P_1. Similarly, a set of values for the same parameters and so the same type of class, but defining instead the performance class P_2, is determined from the data for period T4.

Having defined classes P_1 and P_2, a statistical confidence test is then applied to determine whether P_2 falls within the same class as P_1. If there is a statistically significant difference, then P_2 does not fall within the same class as P_1 and this is notified to the network operator as an incident. For example, a threshold which may be applied is that the median delay for class P_2 must not exceed the median delay for class P_1 by more than 2%. If that threshold is exceeded, then a network

incident is reported. Appropriate statistical tests can be applied to each of the parameters within each class and to combinations of those parameters, depending on the particular aspect of network performance which is being considered. For example, in a subsequent time interval T5 in which the same type of class P_3 is defined, the threshold may be that the median delay must not exceed the median delay for class P_2 by more than 2% and further that it must not exceed the median delay for class P_1 by more than 3%. By constructing a chain of classes based on such tests, a gradual increase in delay over a period of time will be detected and reported as a network incident.

Further, other statistical parameters can be used to define different types of class to focus on different performance characteristics. For example, a class may be defined to include the mean of 95% of the fastest packets and the mean of 5% of the slowest packets when examining delay related performance.

Adaptive behaviour can also be incorporated in the automated incident reporting apparatus. For example, if over a period of weeks, the set of classes P_2 .. P_n always fall within the same class as P_1, then this network reporting requirement can be terminated and reporting based on a different set of classes or time periods initiated.

The automated incident reporting apparatus according to the invention can be used with any system which is capable of providing test data related to network performance characteristics. Referring to Figure 11, the automated incident reporting apparatus 51 comprises a performance summariser 60 which receives low level testing information 61 from a testing system 62, for example, the testing and monitoring apparatus. The resulting performance parameters are stored in a chronological performance database 63. A comparator 64 retrieves the performance parameters from the database 63 and performs the appropriate comparisons, for example, on a class basis using the statistical techniques described above. The comparator 64 can also perform the comparisons with

baseline information 65 which is provided by, for example, a network operator. The baseline information 65 includes information such as desired network performance, thresholds for triggering additional testing and so on. The comparator 64 produces network incident information which can be stored in an incident database 66. The comparator 64 can also automatically initiate further testing on the testing system 62, as indicated by the link 67. As described above for the testing and monitoring apparatus, the functionality of the automated incident reporting apparatus 51 can be implemented using an object-oriented approach in the JAVA language, for example on a Sun Workstation.

The interaction between the automated incident reporting apparatus and the testing and monitoring apparatus at an object-oriented level is described in detail below.

Referring to Figure 12, the testing and monitoring apparatus according to the invention includes a first Director 9, Gatherers 7 and a plurality of Store objects 8, and carries out the background testing required to produce the data to be used by the automated incident reporting apparatus 51 on a continuing basis. The automated incident reporting apparatus includes a second Director 70 which interrogates the Store objects 8. The respective Directors 9, 70 of the testing and monitoring apparatus and the automated incident reporting apparatus have a respective link 71, 72 to respective network operators 73, 74. There is a further link 75 between the first and second Directors 9, 70 and a link 76 between respective network operators 73, 74.

In the event of an incident report being made to the automated incident reporting operator 74, he can initiate further testing as required by a request to the testing and monitoring operator 73 over the link 76. Alternatively, the second Director 70 can, via the link 75, automatically make a request to the first Director 9 to carry out the required further testing without further reference to either operator 73, 74. The second Director 70 can contain all the functionality required to perform the incident reporting functions described above.

Although embodiments of the invention are conveniently implemented in the JAVA computer language, other languages may be used for implementation, including non-object-oriented languages. The invention may also be implemented partly or completely in hardware.

Further, although an implementation was described for an SMDS network, the invention could be implemented with any other type of network, including an ATM network, for example, by using an ATM interface card in the PCs and Director Workstation 30 rather than Ethernet links. The User Interface PC 33 and Director Workstation 30 may also be connected over the Internet.

Claims

1. Apparatus for testing a communication path in a communications network, comprising:

means operative to generate test signals to be transmitted over said communication path;

means operative to determine a characteristic of said communication path by analysing test signals received from said communication path;

means operative to determine whether additional testing in the network is required in dependence upon said path characteristic determination; and

means operative to automatically initiate said additional testing in the event that additional testing is determined to be required.

2. Apparatus according to claim 1, wherein the additional testing comprises the generation of additional test signals.

3. Apparatus according to claim 1 or 2, wherein additional testing is required in the event that the test signals do not determine the path characteristic to a predetermined level of confidence.

4. Apparatus according to claim 1 or 2, wherein additional testing is required in the event that more information is required about a path characteristic.

5. Apparatus according to any one of the preceding claims, wherein the additional testing determination means includes means operative to compare the determined path characteristic with predetermined parameters for said characteristic.

6. Apparatus according to any preceding claim, wherein the test signals comprise a test signal pattern.

7. Apparatus according to claim 6, wherein the pattern repeats with a predetermined period.

8. Apparatus according to claim 6 or 7, wherein the additional testing comprises modification of the test pattern.

9. Apparatus according to claim 8, wherein modification of the test pattern comprises changing the period of the test pattern.

10. Apparatus according to any preceding claim, further including means operative to record the time at which each test signal is launched onto and/or received from the signal path.

11. Apparatus according to any preceding claim, wherein the test signal comprises a data packet.

12. Apparatus according to any one of the preceding claims, wherein the network comprises a plurality of communication paths, including means operative to switch between said communication paths in dependence on the path characteristic.

13. A communications network comprising:

at least one communication path;

means operative to generate test signals to be transmitted over the communication path;

means operative to determine a characteristic of the path by analysing test signals received from the path;

means operative to determine if additional testing in the network is required in dependence upon the determined path characteristic; and

means operative to automatically initiate said additional testing in the event that additional testing is determined to be required.

14. A method of testing and monitoring a communication path in a communications network, said method comprising:

generating test signals;

transmitting the test signals over the communication path;

receiving the test signals from the communication path;

analysing the received test signals to determine a characteristic of the communication path;

determining if additional testing in the network is required in dependence on the determined path characteristic; and

automatically initiating said additional testing in the event that additional testing is determined to be required.

15. A method according to claim 14, including analysing the received test signals by examining the relationship between the transmitted and received signals.

16. A method according to claim 14 or 15, wherein the network comprises a plurality of communication paths and wherein the path being tested comprises a first communication path, said method including:

initiating additional testing on at least one of the plurality of signal paths other than the first communication path, so as to determine a path characteristic of said first communication path.

17. A method of operating a communications network comprising a plurality of communication paths, said method comprising:

a testing and monitoring method according to any one of claims 14 to 16; and

switching between said communication paths in dependence on a path characteristic determined in accordance with said method.

18. A method according to claim 17, including switching to another communication path when the characteristic of the path being tested indicates that no service is available on the tested path.

19. Network testing apparatus comprising:

means operative to systematically determine first performance parameters relating to a performance characteristic of a communications path in a communications network;

means operative to compare said first parameters with second network performance parameters; and

means responsive to said comparison to determine whether said parameters are equivalent in accordance with predetermined criteria, and in the event that said parameters are not equivalent, to report the non-equivalence as a network incident.

20. Apparatus according to claim 19, wherein the first and second network performance parameters represent the same parameter measured at different respective times.

21. Apparatus according to claim 19, wherein said second parameters represent a predetermined network performance.

22. Apparatus according to any one of claims 19 to 21, further comprising means responsive to a network incident to automatically initiate further testing.

23. Apparatus according to any one of claims 19 to 22, wherein said predetermined criteria specify that said parameters are equivalent when they differ by no more than a predetermined margin.

24. A method of automated incident reporting in a communications network, based on a predetermined plurality of performance parameters for said network, comprising:

systematically comparing first and second network performance parameters to determine whether said parameters fall within a predetermined relationship, and in the event that said parameters do not fall within said relationship, reporting this event as a network incident.

25. An apparatus for testing and monitoring a signal path, comprising:

means operative to generate test signals to be transmitted over the path;

means operative to receive the test signals from the path;

means operative to analyse the relationship between the transmitted and received signals to determine characteristics of the signal path;

means operative to automatically control the test signal generating means depending on the output of the analysing means; and

means operative to determine if additional signals need to be generated in order to determine sufficiently a characteristic of the signal path.

AMENDED SHEET

Fig.1.

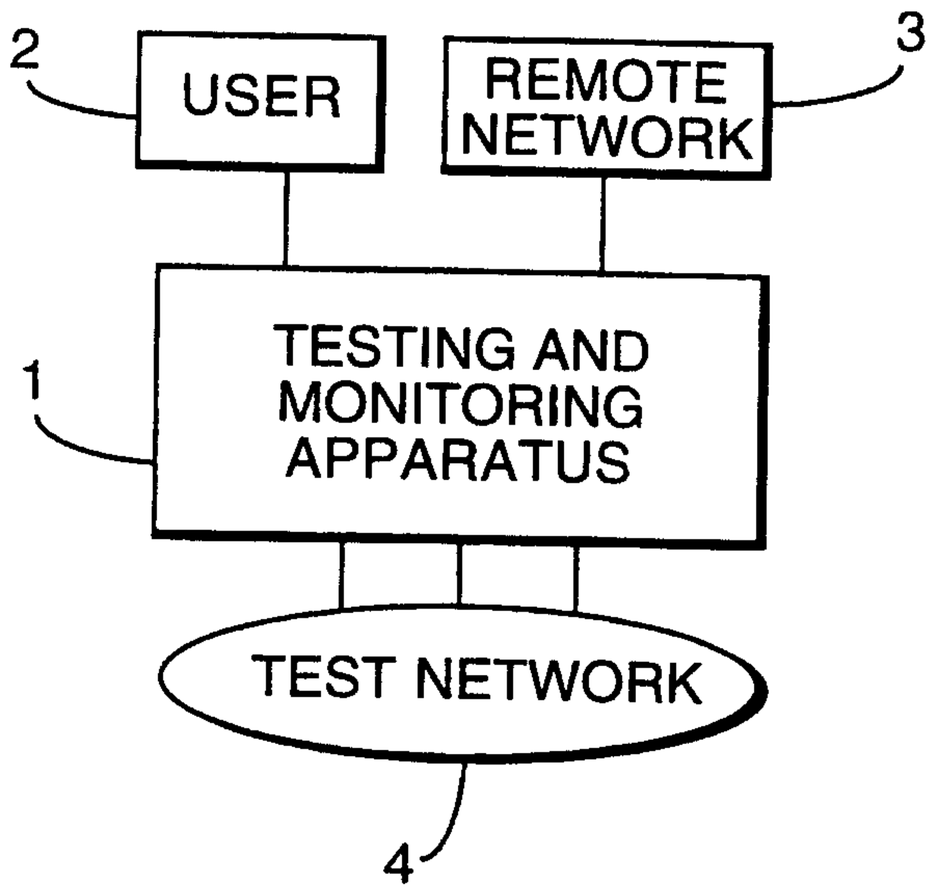

2
USER
REMOTE NETWORK
3
TESTING AND MONITORING APPARATUS
1
TEST NETWORK
4

Fig.2.

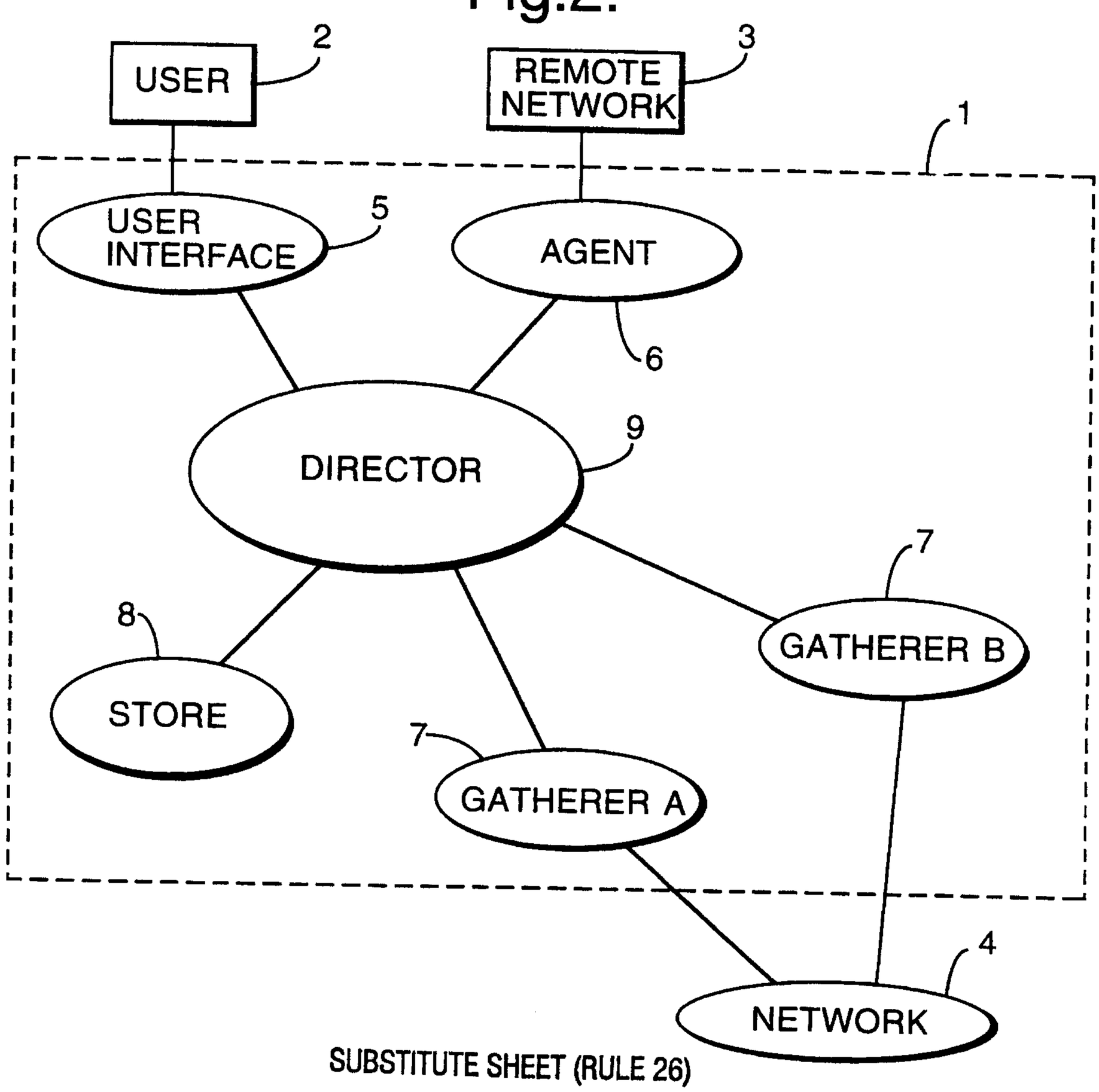

USER
2
REMOTE NETWORK
3
1
USER INTERFACE
5
AGENT
6
DIRECTOR
9
7
8
STORE
7
GATHERER A
GATHERER B
4
NETWORK

Fig.3.

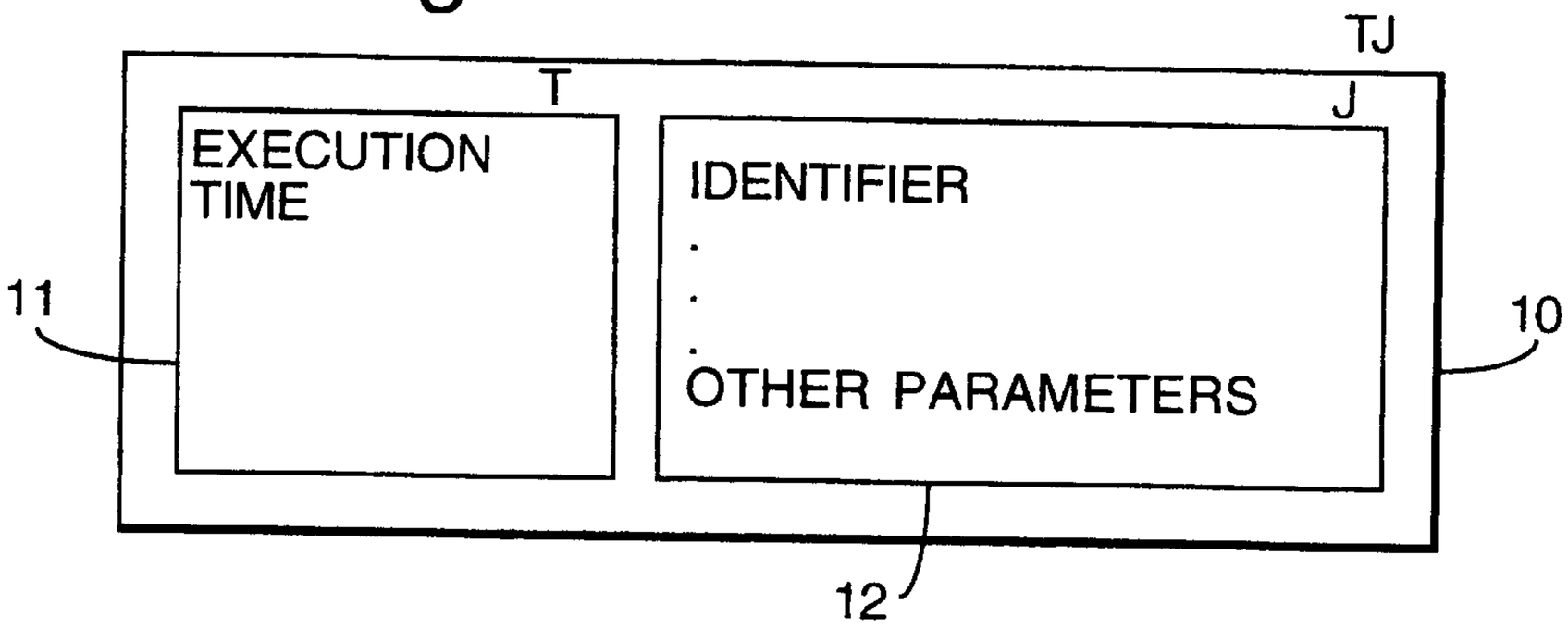


Fig.4.

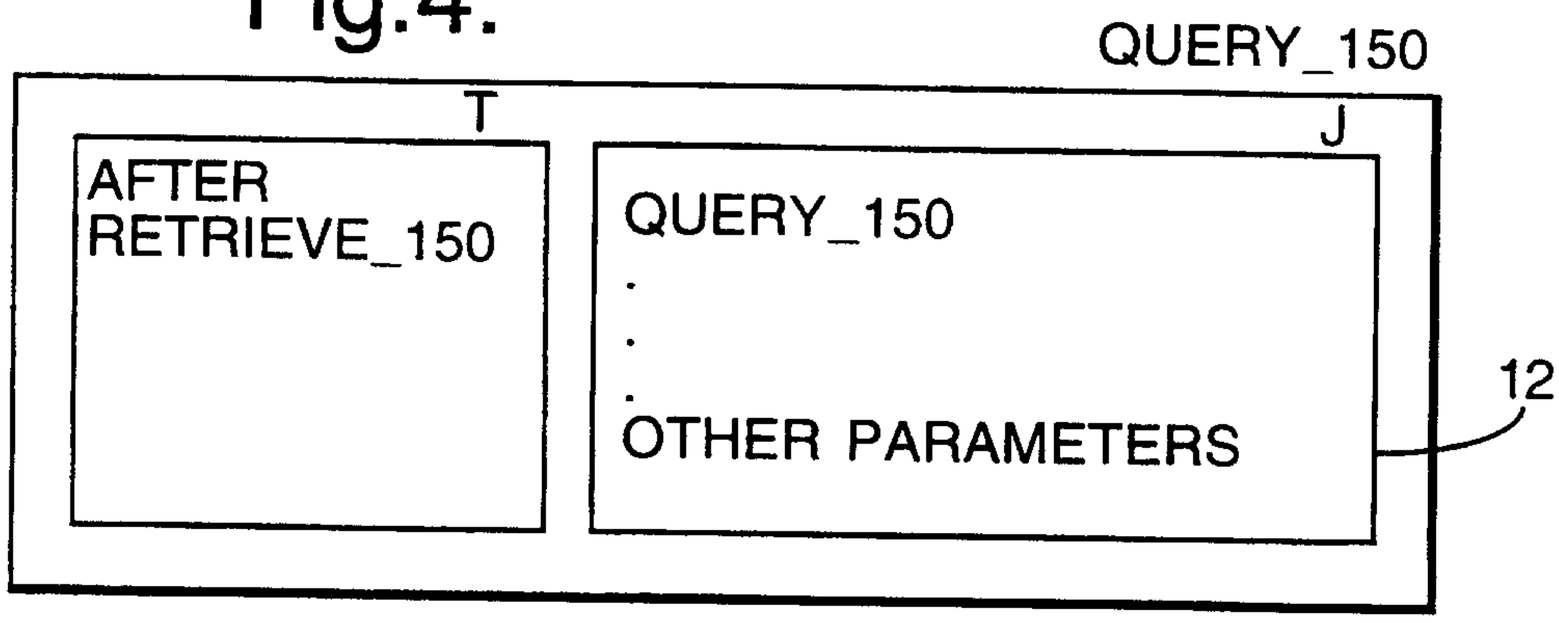


Fig.5(a).

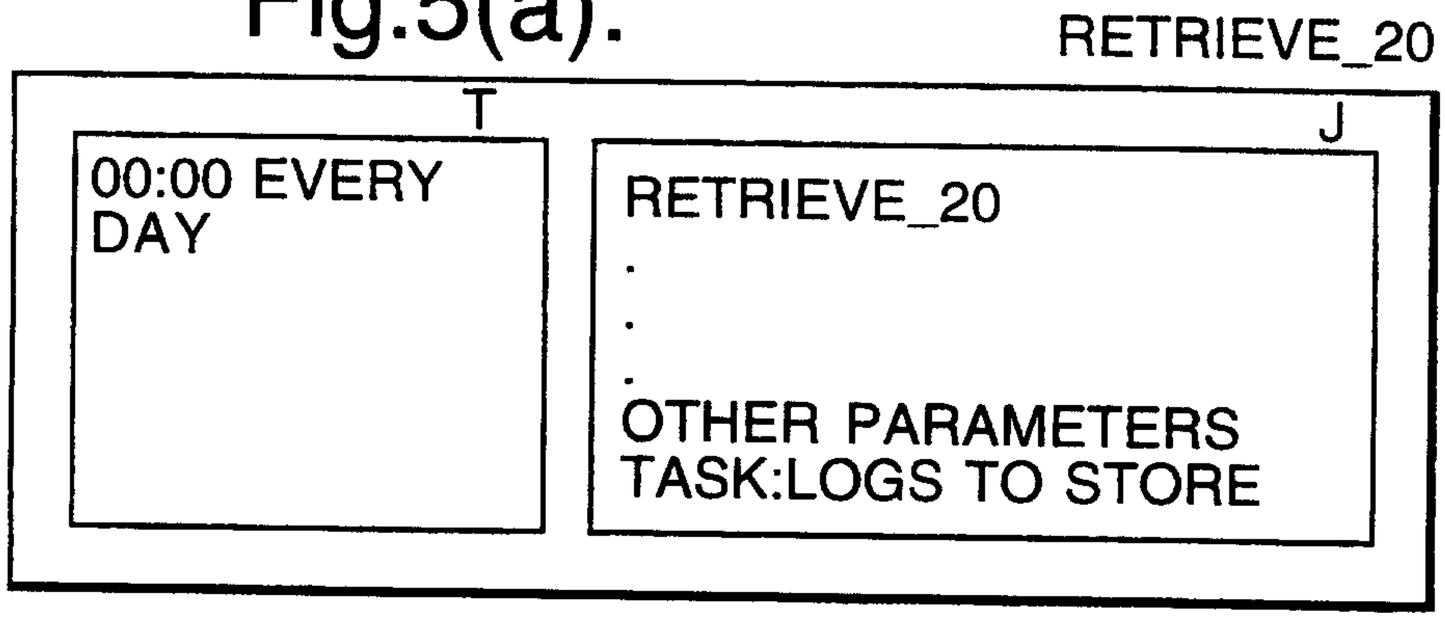


Fig.(5b).

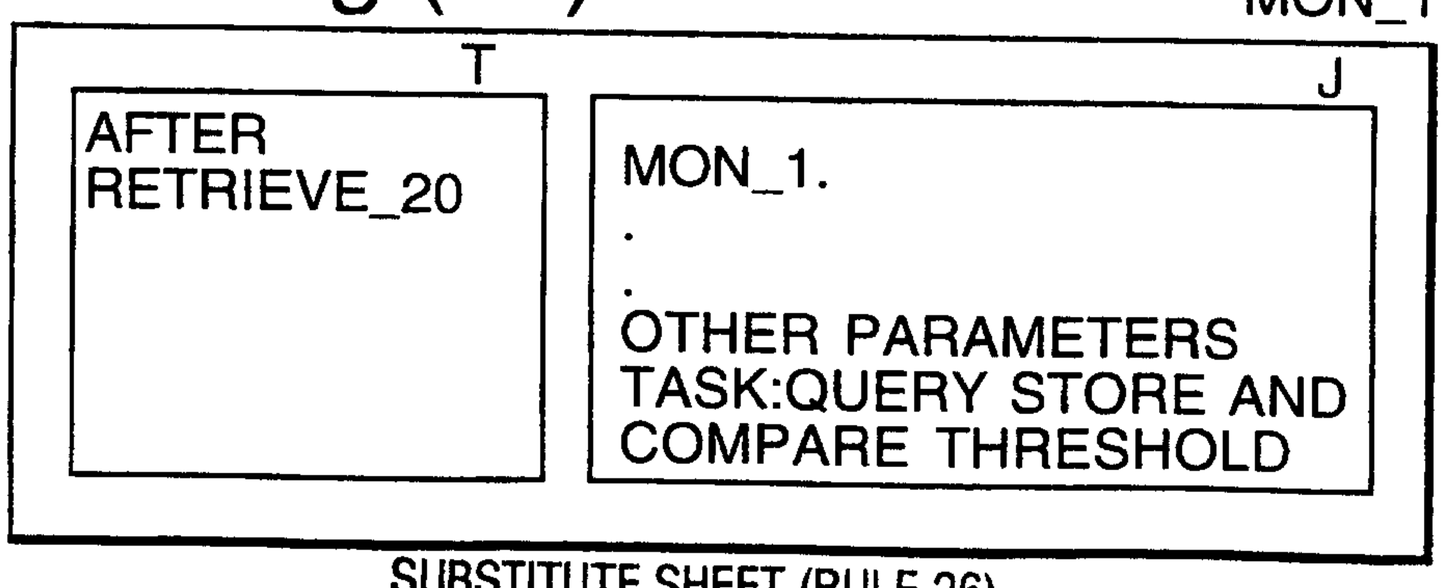


Fig.5(c).

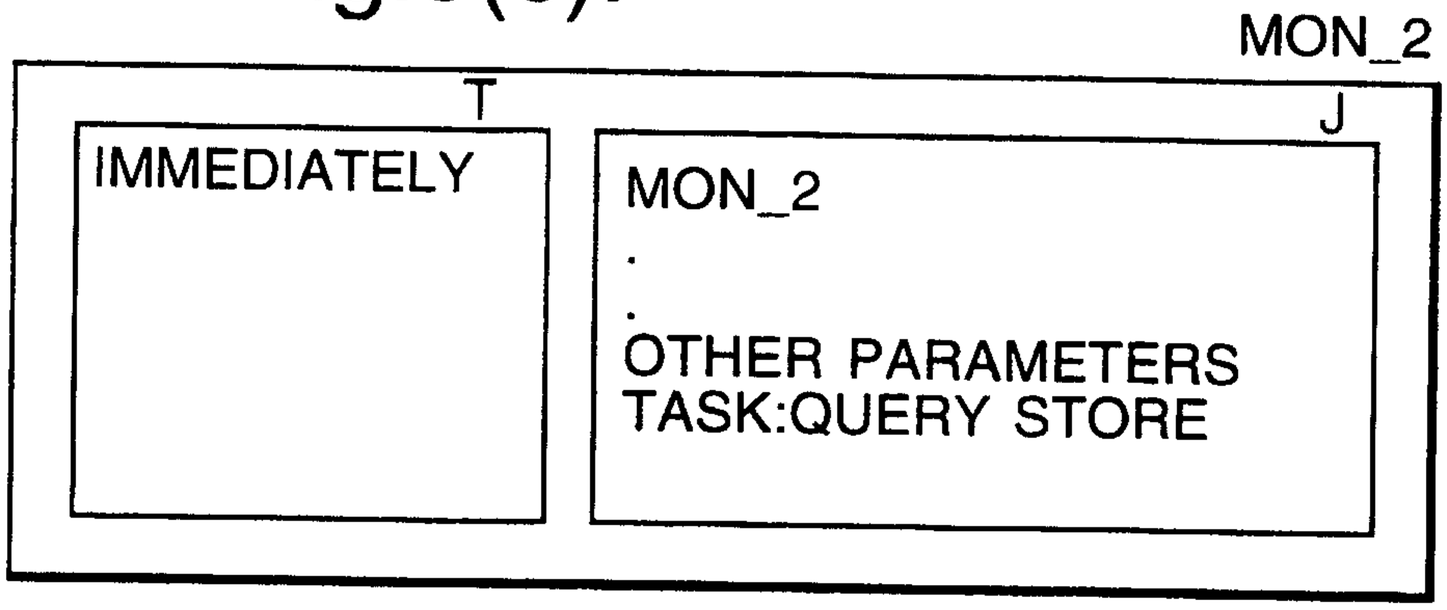


Fig.5(d).

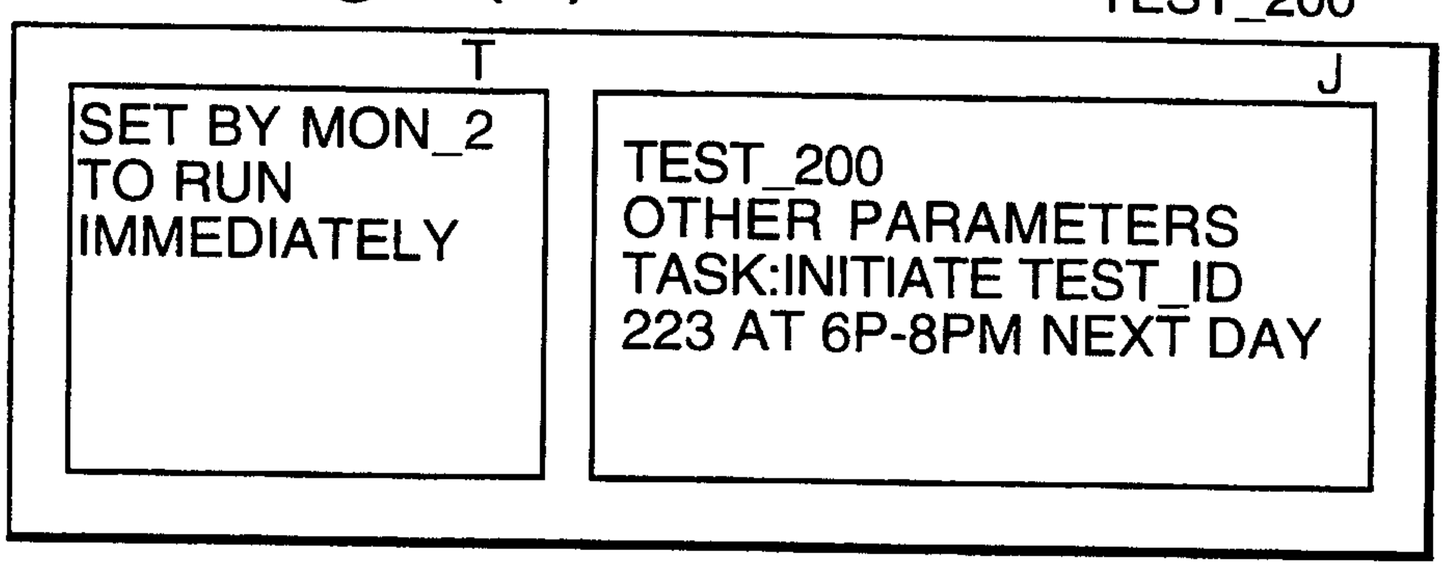


Fig.5(e).

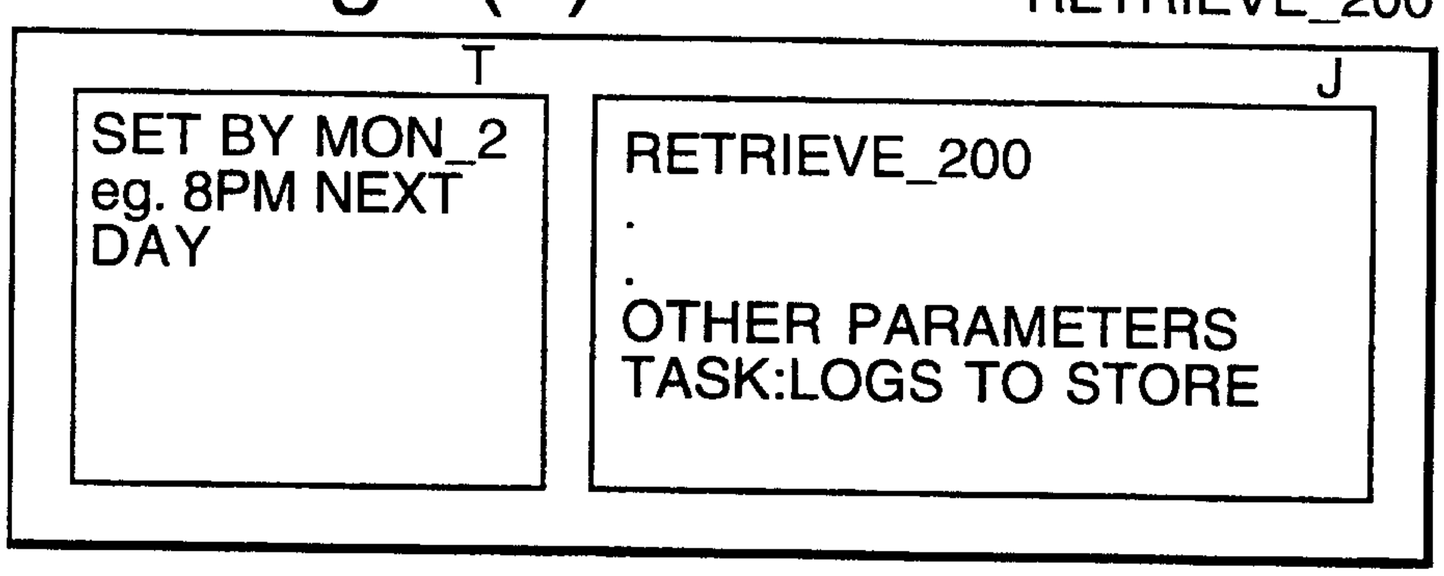


Fig.5(f).

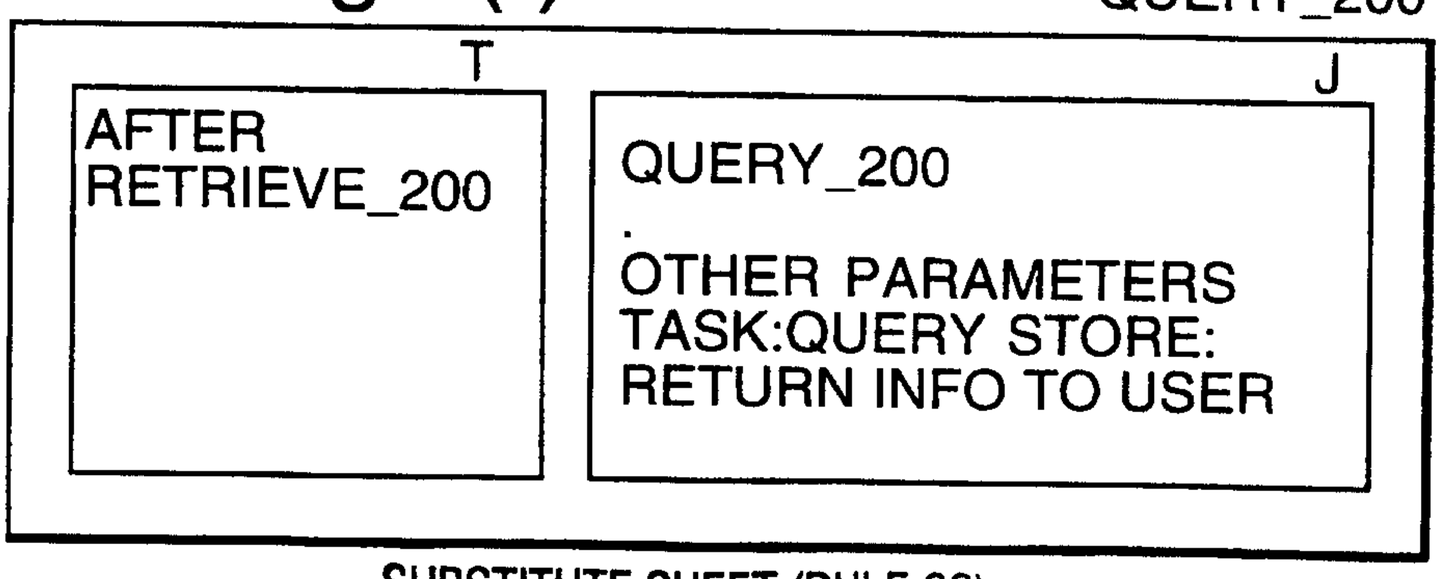


Fig.6(a).

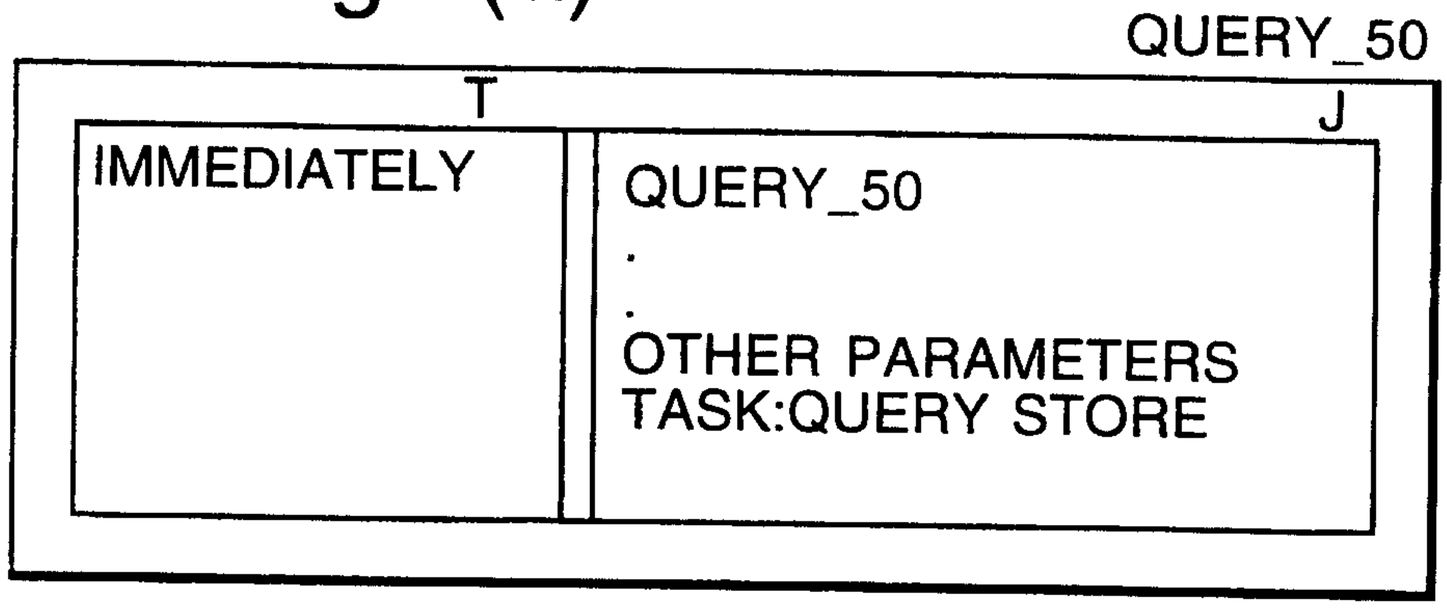


Fig.6(b).

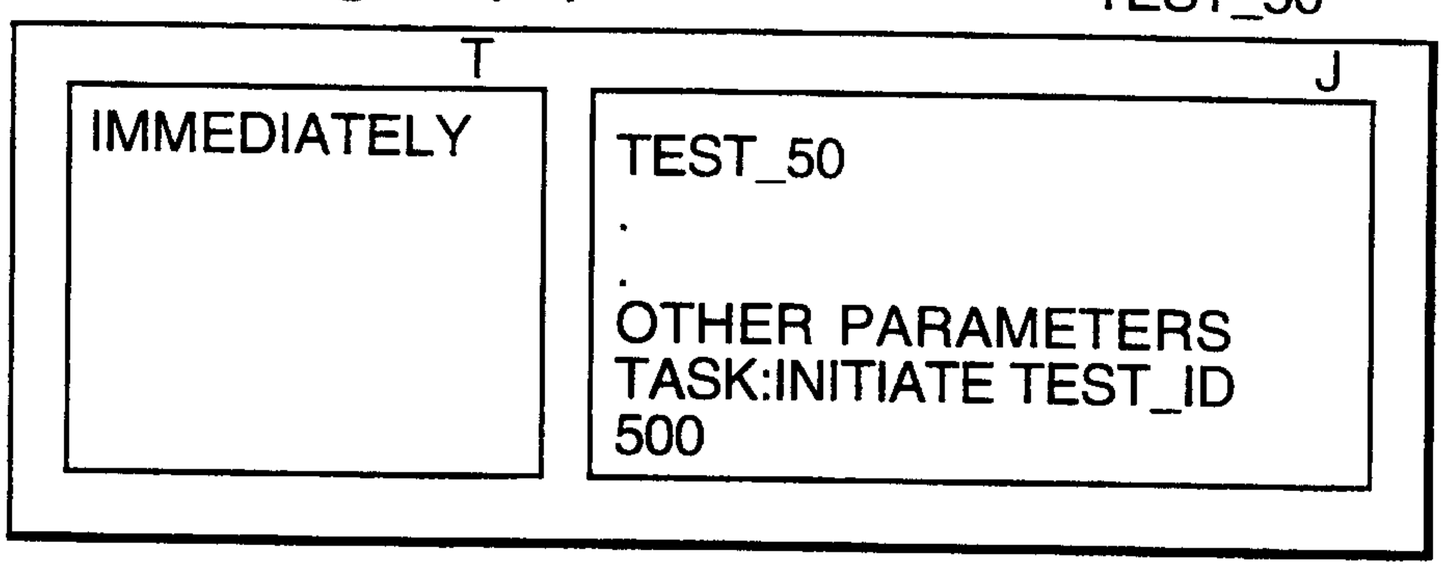


Fig.6(c).

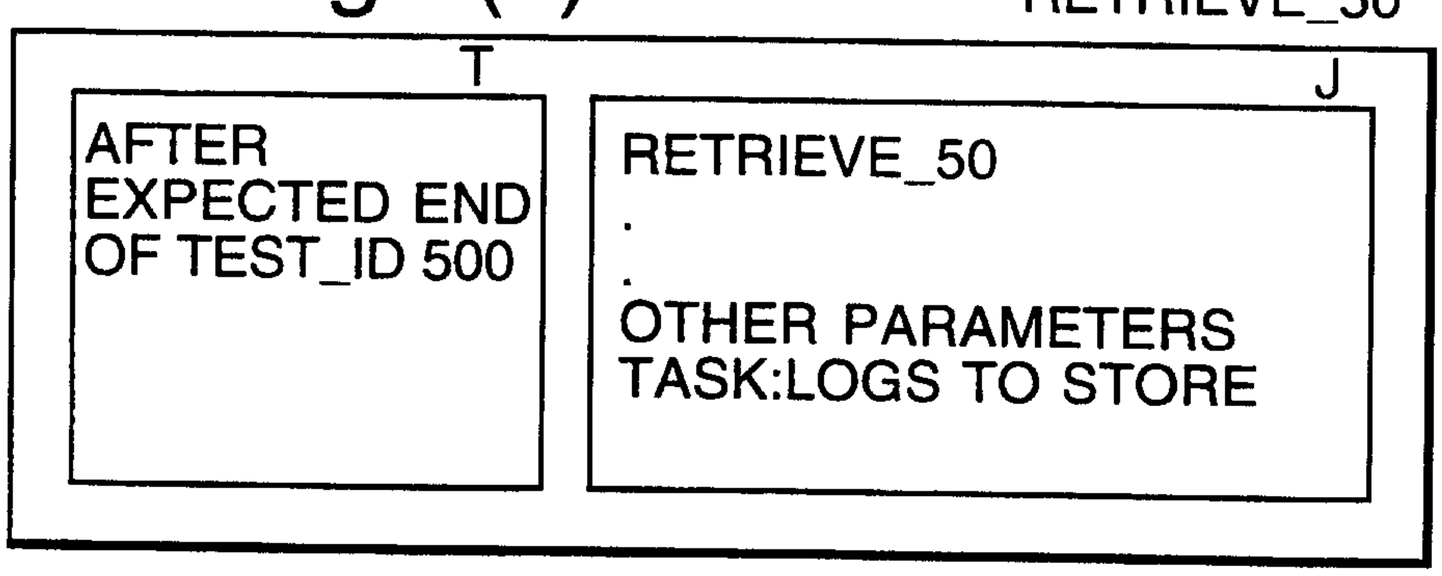


Fig.6(d).

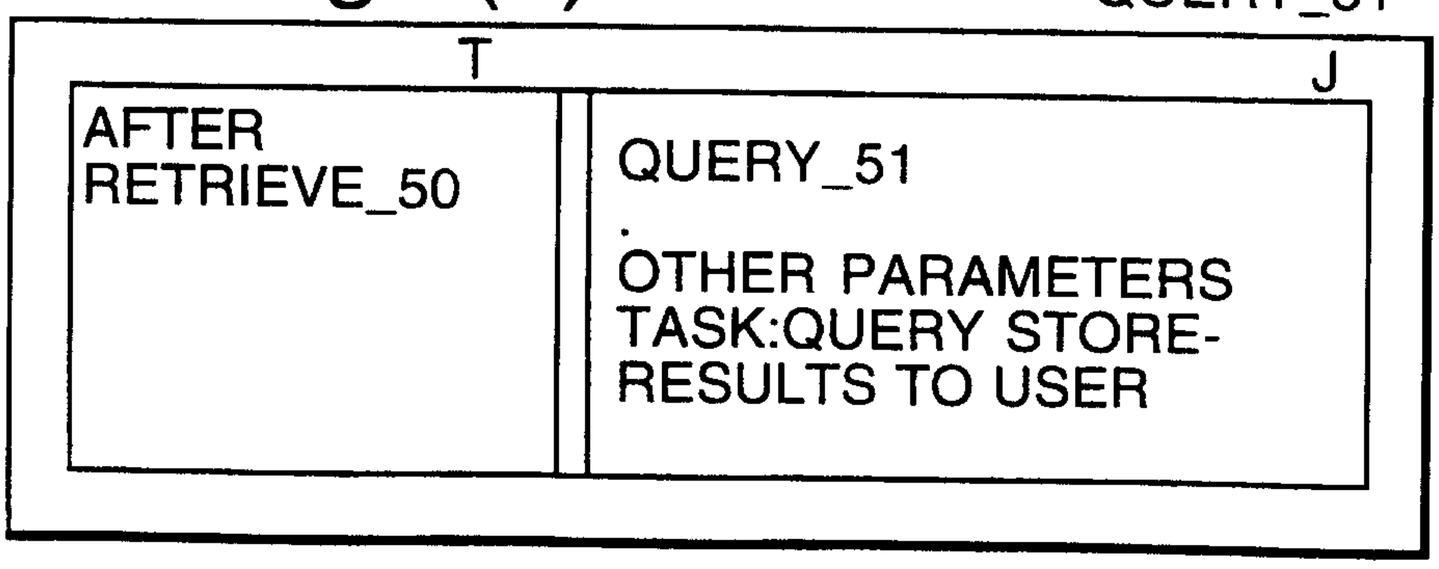


Fig.7.

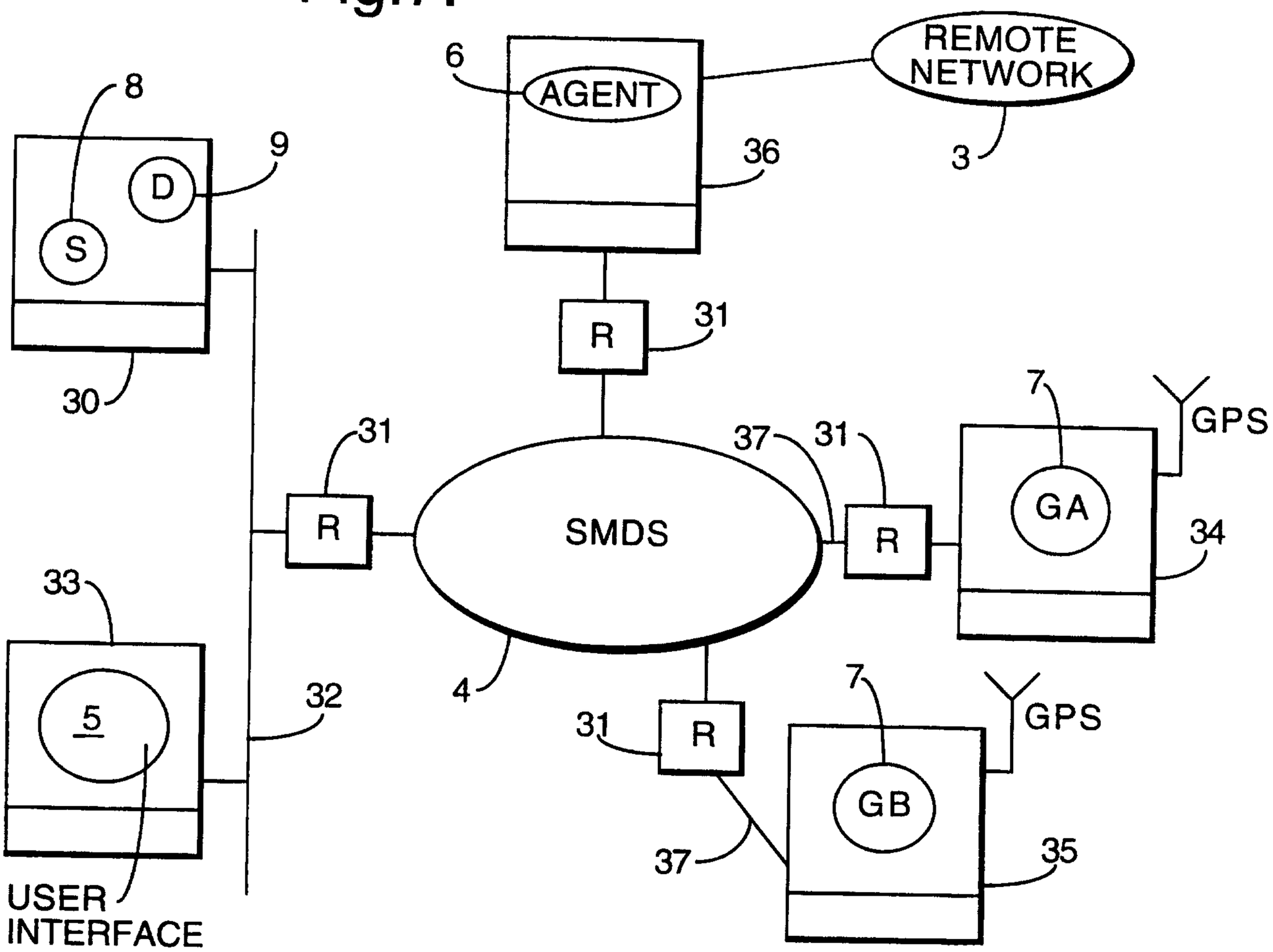

8
9
D
S
30
6
AGENT
REMOTE NETWORK
36
3
31
R
7
GPS
37
31
GA
34
31
R
SMDS
R
33
5
32
4
7
31
R
GPS
GB
37
35
USER INTERFACE

Fig.8.

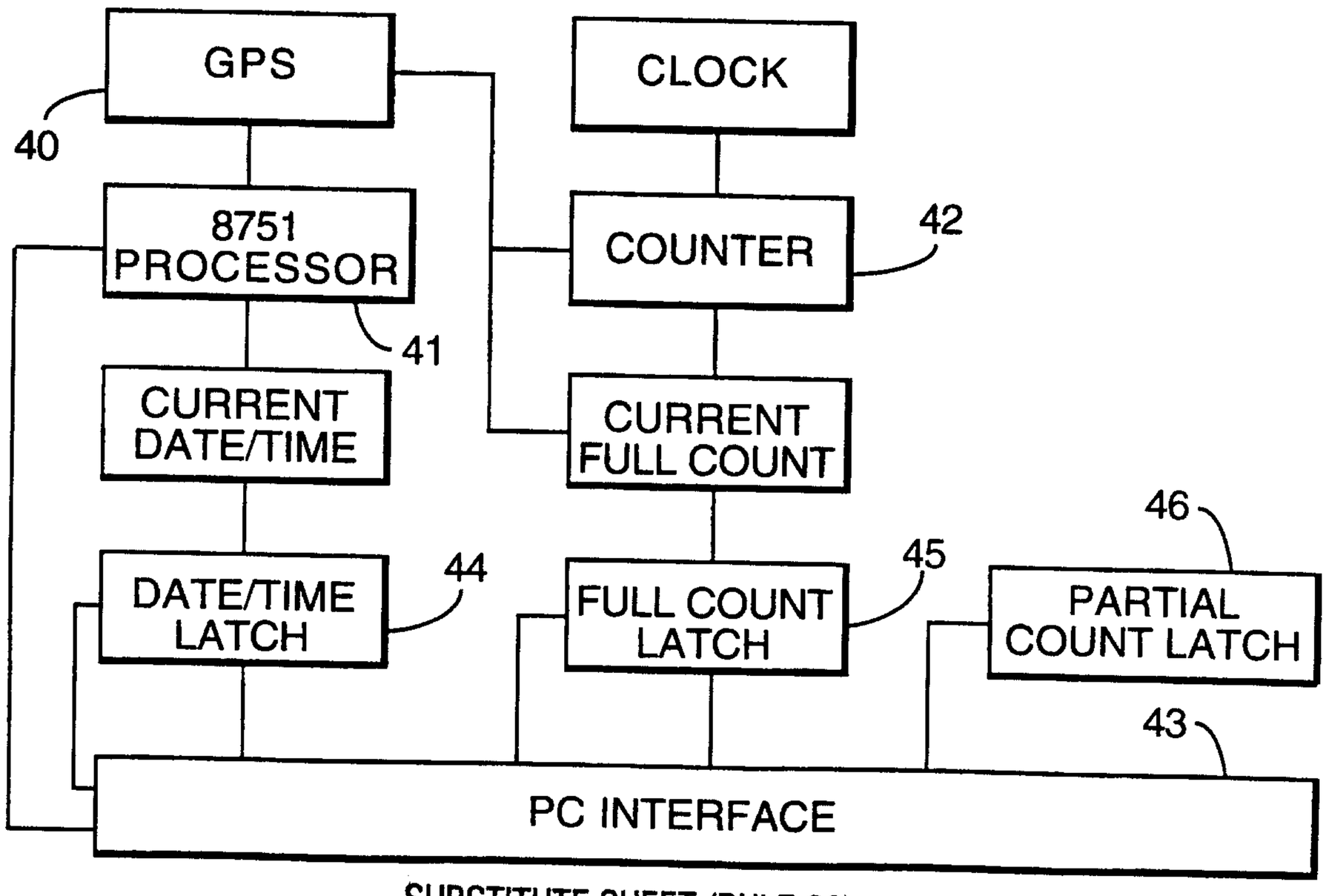

GPS
40
CLOCK
8751 PROCESSOR
41
COUNTER
42
CURRENT DATE/TIME
CURRENT FULL COUNT
46
DATE/TIME LATCH
44
FULL COUNT LATCH
45
PARTIAL COUNT LATCH
43
PC INTERFACE

Fig.9.

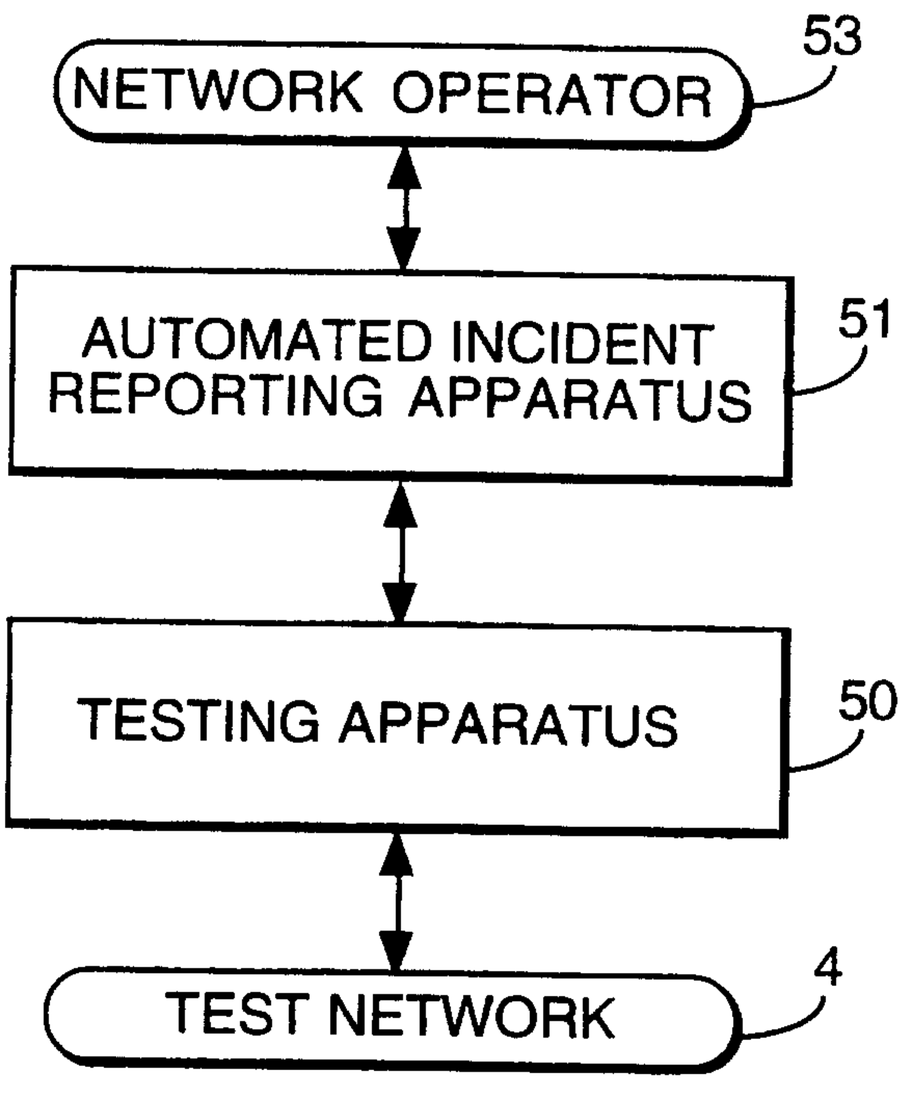


Fig.10.

NETWORK PATH / TIME PERIOD	..	NP4	NP5	..
..				
T3			P_1	
T4			P_2	
T5			P_3	

Fig.11.

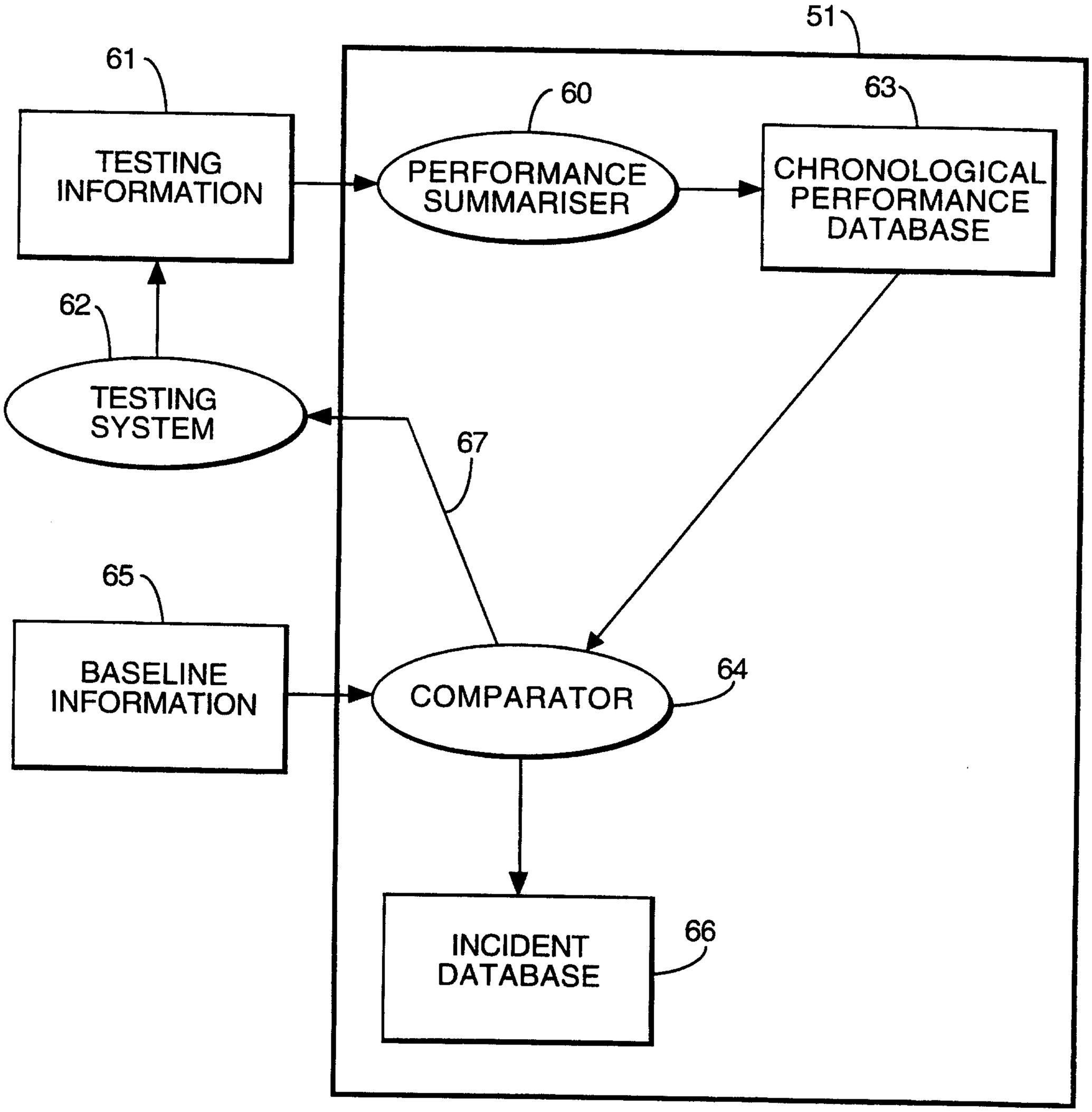
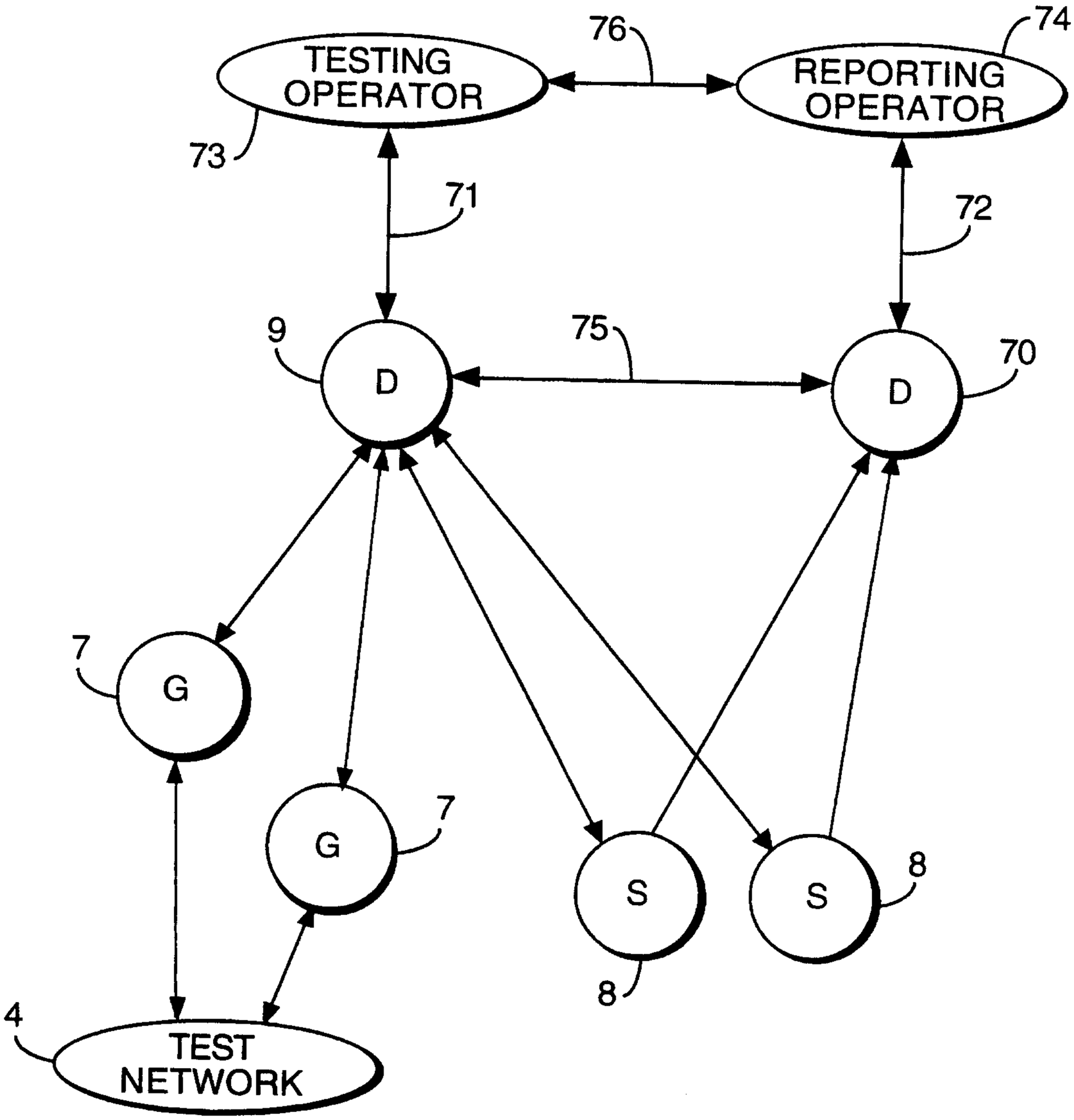

51
61
TESTING INFORMATION
60
PERFORMANCE SUMMARISER
63
CHRONOLOGICAL PERFORMANCE DATABASE
62
TESTING SYSTEM
67
65
BASELINE INFORMATION
COMPARATOR
64
INCIDENT DATABASE
66

Fig.12.


TESTING OPERATOR
REPORTING OPERATOR
76
74
73
71
72
9
D
75
D
70
7
G
G
7
S
S
8
8
4
TEST NETWORK