Workload measurements and synchronisation into a distributed measurement laboratory

Andrea Condarcuri[1], Domenico Grimaldi[1], Antonino Panella[2]

[1)] Department of Electronics, Computer and System Sciences, University of Calabria, 87036 Rende –CS-
[2)] RoyalItalia s.r.l., 87036 Rende – CS - Italy

***Abstract*:** The Distributed Measurement Laboratory (DML) is an heterogeneous system and the workload evaluation is an important aspect to support the performance management strategies in the case the specific requirements must to be respected. The paper describes the method to evaluate the workload of the cooperating hardware equipments of the DML, in particular: (i) the connecting networks, and (ii) the PCs. In this manner useful indexes are achievable to monitor the functioning conditions of the whole DML. The proposed method is based on three different tools known and tested in literature. The first one concerns with the synchronization among the clocks of the different co-operating PCs. The second one concerns the One-Way Delay parameter measurements of the path from the source to the destination. The third one concerns the workload measurement of the CPU of each PC. The results of several experimental tests showing the performance and the accuracy of the method are shown.

Keywords: Delay measurement, Synchronisation, CPU occupancy, Distributed measurement.

I. Introduction

The Distributed Measurement Laboratory (DML) [1]-[4] operates on physical environments which can be distributed in local area and/or geographic wide area. It consists of a collection of distinct Distributed Measurement Systems (DMSs) [5]-[8] devoted to specialised measurement (sub) tasks, co-operating one with each other. Each DMS is responsible for several local measurements and includes various hardware platforms in order to perform different measurement tasks. The hardware includes: (i) server of local network, (ii) automatic measurement stations, and (iii) Local Area Networks (LANs). Normally, the automatic measurement station is equipped by Personal Computer (PC), standard interfaces (RS232, RS485, IEEE488 and wireless) and measurement instruments. The connection among the different DMSs can be obtained through the network bridge and/or the services of standard communications networks, including Internet in the case of geographic area network. Therefore, the DML is an heterogeneous system and the performance management strategies can be very complex in the case the specific requirements must be respected. Several important challenging requirements to the management have emerged in the contest of the expanding DMLs [5]-[10]. Indeed, there are trends towards (i) more efficient access to measurement information, (ii) measurement services that are customised according to individual needs, (iii) powerful tools for real time monitoring and measuring, and (iv) strong minimization of the communication network overhead. Examples of the actual requirements taken into account in the paper can be summarised as follows:

1. the correlation in the time domain of the measurements given by independent DMSs is imposed, the requirement is the DMS synchronisation or the detection of the time delay between the clock of different DMSs;
2. the Mobile Agent technology is adopted for measurement tasks, the requirement is the selection of the path from the source to the destination characterised by the shorter delay time of the packets. This parameter, along with the available bandwidth and packet loss rate, forms three important variables characterising the behaviour of the network traffic flow;
3. the management of the queue connection to the DML server is imposed for monitoring and measurement services, the requirement is the client selection by detecting the path characterised by the shorter delay time of the packets;
4. the distributed processing of the measurement results is adopted, the requirement is the selection of the processing resource characterised by the shorter workload among all available in the DML to permit the speed processing.

In the paper a method able to give adequate answer to all these requirements is proposed. The general architecture of the DML taking into account is shown in Fig.1. The basic idea is to distinguish in three different groups the tools to be used. The first one concerns with the synchronization among the clocks of the different co-operating PCs, as an example PC#1of DMS#1 and PC#m of DMS#n. In this manner the requirement 1) can be satisfied. The second one is correlated to the previous one and concerns with the One-Way Delay (OWD) parameter measurements of the path from the source to the destination. As an example, in Fig.1 the path can be constituted by (i) the connecting network#1, (ii) the connecting network#b, and (iii) one of the connections (α, β) into internet. In this manner the requirements 2) and 3) can be satisfied. The third one concerns with the workload measurement of each PC, as an example the PC#1 or the PC server#1 of the DMS#1. This is performed by measuring the CPU occupancy parameter in the fixed and well determined conditions, not influenced by the different clock frequencies of the CPUs. In this manner the requirements 4) can be satisfied. The clock synchronization, the OWD measurements and the CPU occupancy are based on three independent procedures known and tested in literature [11]-[16]. Both the OWD measurement and the CPU occupancy parameter are useful indexes of the workload evaluation of the DML.

In order to clearly describe the proposed method, in the following the description of the activities to evaluate the workload of the DML is given. Successively, the activities scheduled by the proposed method that must be executed by both the client and the server of the DML are highlighted. The factors motivating the choice of the procedures for (i) the clock synchronization, (ii) the

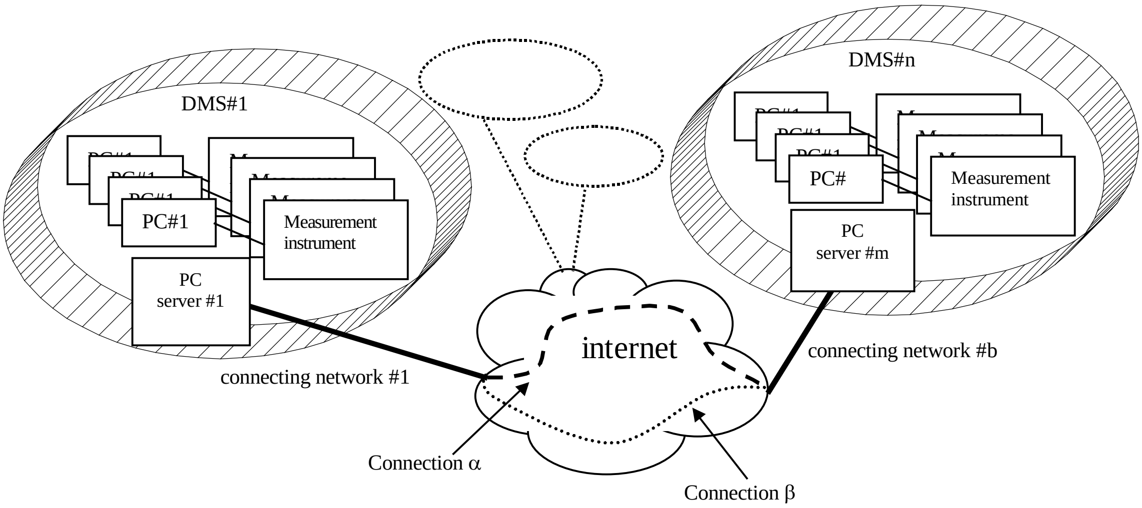

Figure 1 General architecture of the DML.

OWD measurements, and (iii) the CPU occupancy estimation and the modifications justifying their adaptation to be used into the DML are discussed, too. At the end, the results of several experimental tests showing the performance and the accuracy of the proposed method are described.

II. The method for the workload evaluation

The workload of the DML is determined by the workload on the cooperating hardware equipments, in particular (i) the connecting networks, (ii) the PCs configured as network server, and (iii) the PCs configured to perform measurement tasks. In the following, according to the method used, all the PCs involved in the workload evaluations are considered as server, without differentiation on the basis of their functionality into the DML. Therefore, the new scenario to be taken into account in the workload evaluation is constituted by (i) the PC server, (ii) the connecting network, and (iii) the PC client. Both the PC server and the connecting network are the cooperating hardware interested to the workload evaluation. The client PC is dynamically defined into the DML and executes the activities scheduled by the method. The monitoring activities of the proposed method are shown in Fig. 2a). The PC synchronization is performed if the OWD measurement needs. The other activities are performed in cooperation between the client and the server. In particular, according to the Fig. 2c), the PC client (i) generates the probe packets, (ii) monitors the received packets, (iii) evaluates the mean delay of the communication, (iv) determines the CPU occupancy, and computes the loss packets. According to the Fig. 2b), the PC server (i) receives the UDP packets, (ii) attaches the time stamps according to the delay evaluation of the communication network, (iii) executes the procedures according to the CPU workload evaluation, and (iii) send the UDP packets to the client.

III. Tools for the workload estimation

In the following the software tools are considered in order to perform the (i) clock synchronization of the PCs of the DML, (ii) OWD measurement, and (iii) CPU occupancy estimation of the PC.

A. Clock synchronization

In the DML, the clock synchronization is used to measure the delay of the communication between two PCs, as an example the PC#1 and PC#m of Fig.1. Among the various procedures, the Network Time Protocol (NTP), now established as an Internet standard protocol [11], is used to organize and maintain the clock synchronisation of the PC to the national time computer service. NTP is built on the Internet Protocol (IP) and User Datagram Protocol (UDP) [12], which provide a connectionless transport mechanism. Fundamental advantage of the NTP for application on DML is the fact that its protocol includes procedure to compensate for the effects of statistical delay variations encountered in wide-area networks and it is suitable for accurate and high resolution synchronisation throughout the Internet. The numerous experimental tests performed shown that, as a consequence of the clock drift of the PC, the synchronisation interval is an influencing factor of both the accuracy and the stability of the synchronization. Fig. 3 shows the trend of the synchronisation offset between two PCs, in the case the time interval is equal to ten minutes (Fig. 3a), and one minute (Fig. 3b), respectively. In this last case the synchronisation is more accurate.

B. One-way delay measurement

In the framework of the DML, the OWD measurement is used as easy parameter to evaluate the workload of the network connection under examination. The OWD can be evaluated once both the source PC and destination PC are synchronised by NTP. This coarse parameter furnish useful information for all type of network connection, Internet included [13], [14]. The

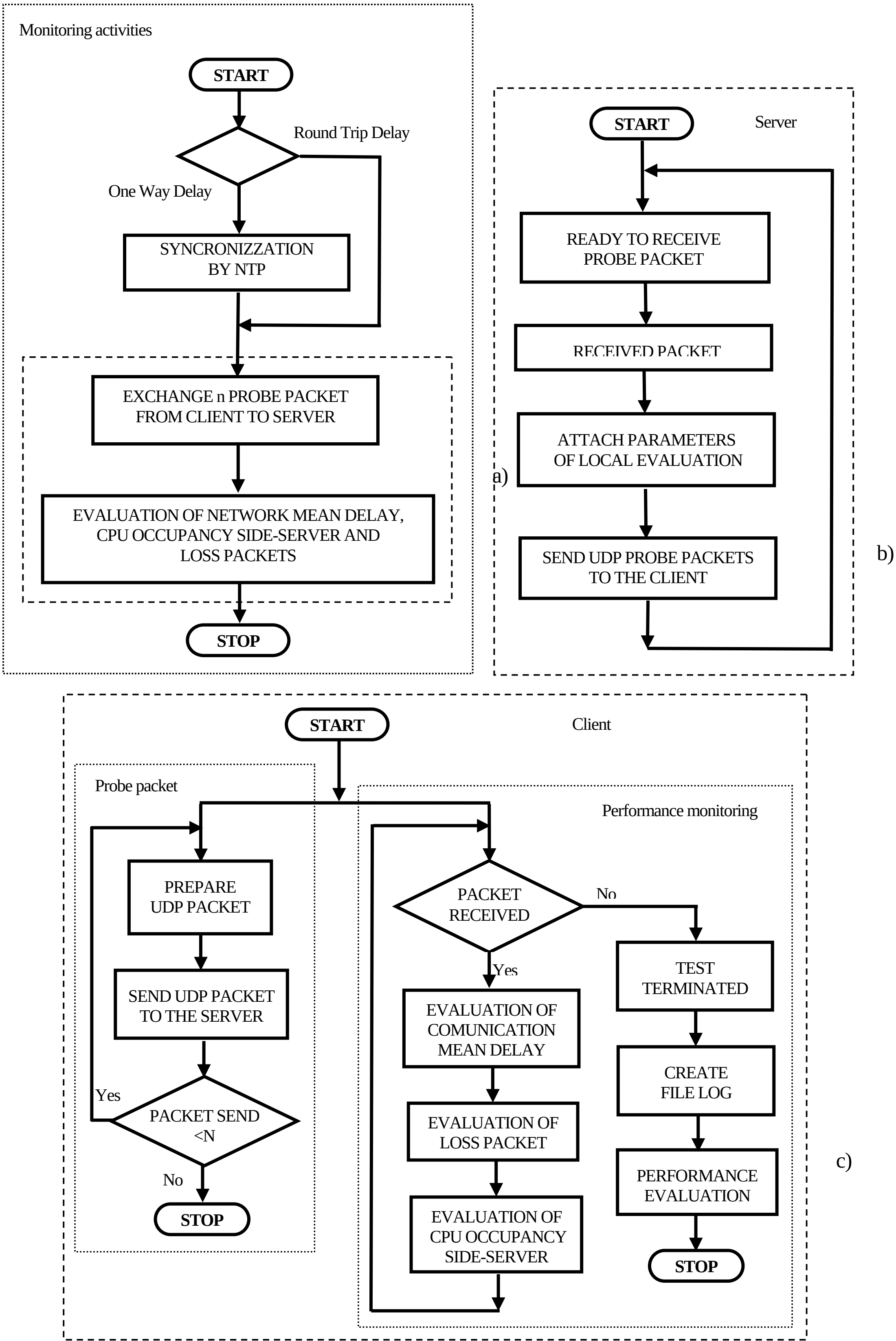

Figure 2 Flow diagram implementing the propose method.

measurements are realized through “active probing” [15] which injects measurement packets in the path to measure and observe the series of parameters of which the packets are spoiled. Each packet is characterised by the timestamp at the sender and at the receiver. The protocol used for connection is the UDP, the programming language for this purpose is JAVA [17]. The log file containing the timestamps of the packets is created when the measurements are terminated. The OWD parameter is evaluated by means of the occurrences of the delays detected by means of the timestamps. In similar way the Round Trip Delay (RTD) parameter can be evaluated. As an example, in Fig.1 the path can be constituted by (i) the connecting network#1, (ii) the

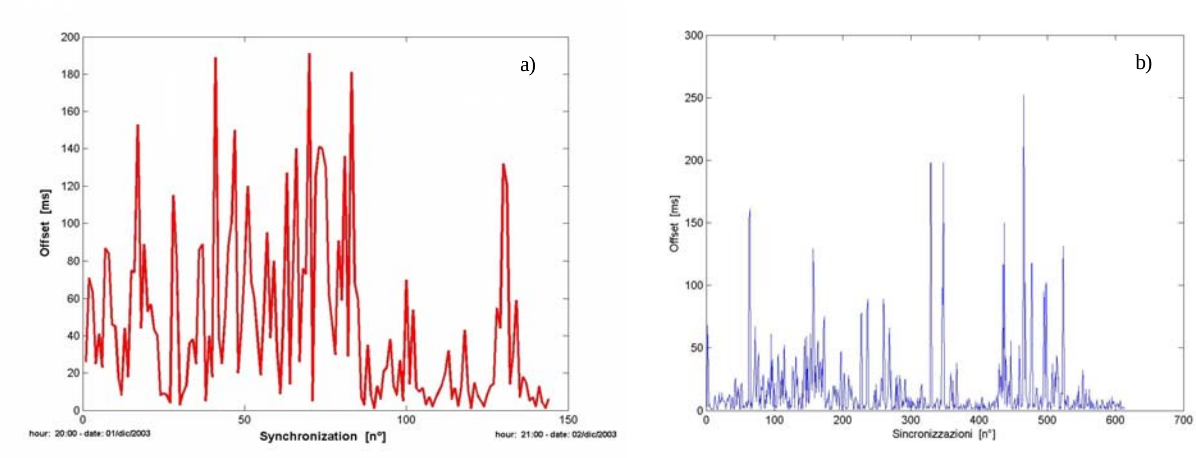

Figure 3 PC synchronization in presence of random perturbation a) every ten minutes, and b) every one minute.

connecting network#b, and (iii) the connections α into internet for the sent packet and the connection β for the received one. In this last case does not need the clock synchronisation.

C. CPU occupancy estimation

The workload of the PC can be evaluated by means of the CPU occupancy parameter. This parameter is defined on the basis of the delay time occurring to the system CPU to answer at the system call. Obviously, higher the occupancy of the system CPU, longer is the delay time of the answer. Low CPU occupancy parameter means that the PC can support other measurement tasks. Therefore, the estimation of the CPU occupancy parameter is based on the evaluation of the time response of the CPU by using some system library tied to the operating system. The technique is based on the intelligent probe packet sent by the PC client. As an example, in Fig.1 the PC#1 can be the object of the CPU occupancy evaluation and it must be configured as server. Consequently, another PC, as an example the PC#m, must be configured as client. Each probe packet operates according to the following steps: (i) n consecutive request to the system clock are sent, (ii) the first and the n-th answer containing the timestamps of the clock are taken into account and stored, and (iii) the CPU occupancy parameter is evaluated as difference between the two timestamps stored. In order to make the CPU occupancy parameter independent from the clock frequency of the system CPU, the final value is furnished as the percentage ratio between the time interval defined from the two timestamps stored and the maximum time delay of the CPU response. The reliable solution is accomplished by stepping outside Java and writing few C code lines able to be (i) used in the different operating systems [18] like Windows 2000/XP and Linux, and (ii) integrated with Java application [16], [17] via Java Native Interface.

IV. Experimental tests

A number of experimental tests were carried out on different operating conditions of both the connecting network and the PCs in order to assess the reliability and the effectiveness of the proposed method. The first ones are the OWD and the RTD measurements on two different Wide Area Networks, in the following denoted by WAN1 and WAN2. Both the two WANs have the same client node and different server node. Each WAN includes (i) optical fibre three, (ii) switched communication lines, (iii) routers, (iv) switches, and (iv) modem connection. After the clock synchronisation between the client and server by means of the NTP procedure, both the OWD and the RTD measurements are executed by sending 1000 probe packets. The network schemes to be taken into account in these measurements correspond to the two different paths that can be detected in the Fig.1. The first refers to the OWD measurement and it is constituted by (i) the connecting network#1, (ii) the connecting network#b, and (iii) the connections α into internet. The second refers to the RTD measurement and it includes the first one with added the connections β for the return back of the packets. By referring to the WAN1, Fig. 4a) and b) show the OWD measurement for each packet send from the client and received from the server and the corresponding occurrences of the time delays. Fig. 4c) shows the OWD measurement evaluated for each packet in the back direction, the packets are send from the server and received from the client. Fig. 4d) shows the corresponding occurrences of the time delays of the OWD measurements in the back direction. Fig. 4e) shows the RTD measurements. Fig. 4f) shows the corresponding occurrences of the time delays of the RTD measurements. By referring to the WAN2, Fig. 5a) and b) show the OWD measurement for each packet and the corresponding occurrences of the time delays. Fig. 5c) and d) show the RTD measurement for each packet and the corresponding occurrences of the time delays. It can be noted that the occurrence distributions for the WAN1 (Fig. 4) have the Gaussian shape, in the contrary for the WAN2 (Fig. 5) the distribution is far from the previous one. This is a consequence of the fact that the working conditions of the WAN2 is congested, if compared to the working conditions of the WAN1. The second ones are the CPU occupancy measurements on the PC configured as server. These measurements are not dependently from the connecting networks. The experimental tests were executed by sending 1000 probe packets at the time interval of 1s from the client to the server. Both the server and the client were equipped by platform Windows 2000. Fig. 6 shows the trend of the CPU occupancy parameter versus the successive probe packets in four different working conditions.

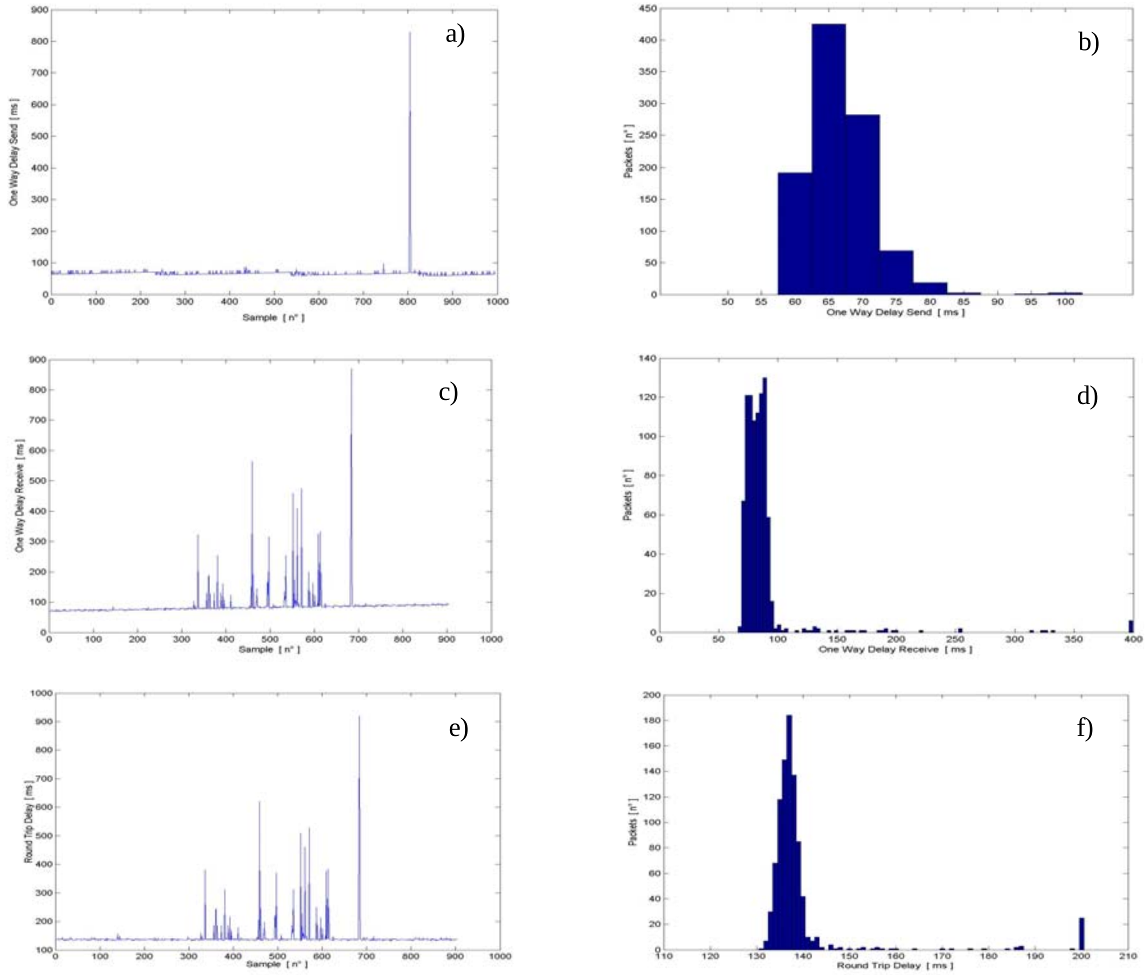

Figure 4 a), c), and e) OWD and RTD measurement executed on the WAN1 for each probe packet, and b), d), and f) the corresponding occurrences of the time delays.

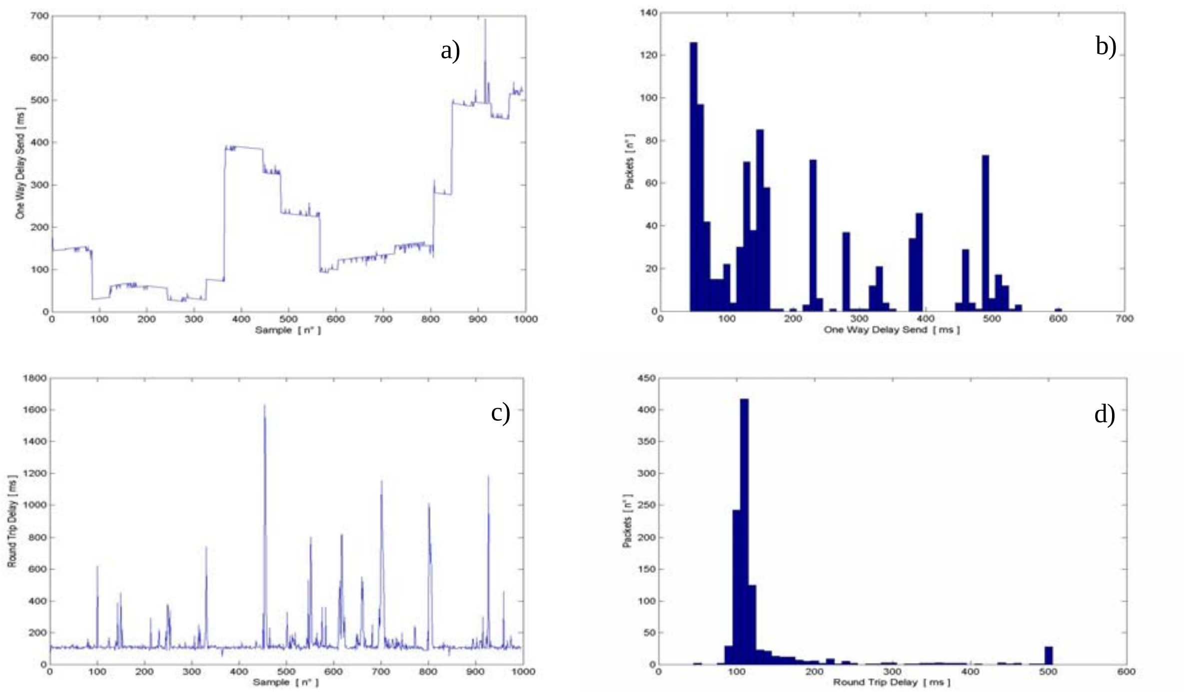

Figure 5 a), and c), OWD and RTD measurement executed on the WAN2 for each probe packet, and b), and d) the corresponding occurrences of the time delays.

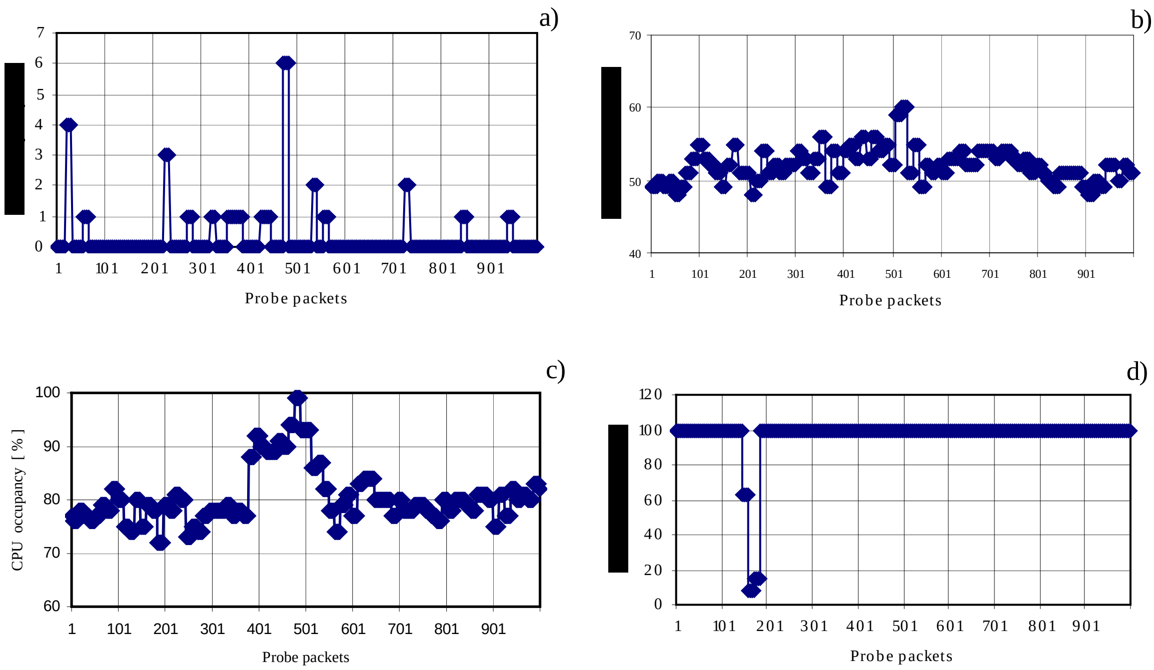

Figure 6 CPU occupancy measurement on 1000 probe packets: a) low occupancy, b) about 50% of occupancy, c) about 80% of occupancy, and d) full occupancy.

V: Conclusions

In the paper a method for evaluating the workload of the connecting networks and the PCs of the DMLs is presented.
The proposed method is based on three different tools known and tested in literature. The first one concerns the synchronization among the clocks of the different co-operating PCs. The second one concerns the One-Way Delay parameter measurement of the path from the source to the destination. The third one concerns the workload measurement of the CPU of each PC.
The experimental results given show the validity of the proposed method and make it useful in order to achieve indexes to monitor the functioning conditions of the whole DML.

Acknowledgements

The authors wish to thank Prof. Pasquale Daponte for his helpful suggestions and active interest during all the phases of the research.

References

[1] Grimaldi D., Nigro L., Pupo F., "Development issues of distributed measurement systems", *Proc. of Imeko TC-4, Budapest*, pp.111-114, September 1996.
[2] Arpaia P., Cennamo F., Daponte P., Savastano M., "A distributed laboratory based on object oriented systems", *Proc. of IMTC/96*, Brussels, 4-6 June, pp. 27-32, 1996.
[3] Arpaia P., Baccigaluppi A., Cennamo F., Daponte P., "A distributed measurement laboratory on geographic network", *Proc. of IMEKO TC-4, Budapest*, 10-12 September, pp.294-297, 1996.
[4] Arpaia P., Baccigaluppi A., Cennamo F., Daponte P,. "A distributed measurement laboratory on geographic network for remote test experiments", *Proc. of IMTC/98*, St. Paul, Minnesota (USA), May 18-21, pp.206-206A, 1998.
[5] Daponte P., Grimaldi D., Nigro L., Pupo F., "Distributed Measurement Systems: an Object-Oriented Architecture and a Case Study", *Computer Standards and Interfaces*, vol.18, No.5, pp. 383-395, June 1997.
[6] Grimaldi D. Nigro L., Pupo F., "Java Based Distributed Measurement Systems" *IEEE Trans. on Instrum. and Measur*, vol. 47, No. 1, pp.100-103, February 1998.
[7] Fortino G., Grimaldi D., Nigro L., "Multicast control of mobile measurement systems", *IEEE Trans. on Instrum. and Measur*, vol. 47, No. 5, pp.1149-1154, October 1998.
[8] Grimaldi D., Marinov M., "Distributed measurement systems", *Measurement*, 30, pp.279-287, 2001.
[9] Daponte P., Grimaldi D., Marinov M., "Real-time measurement and control of an industrial system over the Internet: implementation of a prototype for educational purposes**",** *IEEE Trans. on Instrumentation and Measurement*, August 2002.
[10] Fortino G., Grimaldi D., Nigro L., "An Agent based measurement laboratory over internet", *Proc. of IEEE Autotestcon'99*, S. Antonio, Texas, USA, Aug. 30–Sept. 2, 1999.
[11] Mills D. L.: "Internet Time synchronization: the Network Time Protocol[1,2,3]", Electrical Engineering Department University of Delaware, http://citeseer.nj.nec.com/mills91internet.html.

[12] RFC 768 – User Datagram Protocol - J. Postel ISI 28 August 1980 –http://www.freesoft.org/CIE/RFC/768/.
[13] Gurewitz O., Sidi M., " Estimating One-Way Delays from cyclic-path delay measurements", IEEE INFOCOM, pp.1038-1044, 2001.
[14] Hofmann U., Pfeiffenberger T., Hechenleitner B.: "One-Way-Delay Measurements with CM Toolset"–http://www3.fh-sbg.ac.at/~bhechenl/download/publications/ipccc2000/ipccc2000.pdf.
[15] Pàsztor A., Darryl Veitch: "High Precision Active Probing for Internet Measurement" – http://www.emulab.ee.mu.oz.au/~attila/pub/inet2001.pdf.
[16] Iarantsev A. "Performance meter and memory leaks detector", Dec 2001, http://www.codeproject.com/debug/compinfo.asp.
[17] *Sun's Core Java 2, Volume II Advanced Features, 5th Edition – Chapter 11 Native Methods*, Prentice Hall
[18] Bradford E.G.: *RunTime: High-performance programming techniques on Linux and Windows 2000* -Senior Programmer, IBM - 1 April 2001