IMTC 2005 – Instrumentation and Measurement
Technology Conference
Ottawa, Canada, 17-19 May 2005

Measurement of Processing and Queuing Delays Introduced by a Software Router in a Single-Hop Network

Leopoldo Angrisani[1], Lorenzo Peluso[2], Annarita Tedesco[2], Giorgio Ventre[1]
[1]Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, via Claudio 21, 80125 Napoli, Italia.
Phone:+39 0817683170, Fax: +39 081 2396897, E-mail: angrisan@unina.it, giorgio@unina.it.
[2]Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale (C.R.I.A.I.),
Piazzale E. Fermi 1, Loc. Porto del Granatello, 80055 Portici (NA), Italia.
Phone:+39 0817766905, Fax:+39 0817760583, E-mail: l.peluso@criai.it, a.tedesco@criai.it.

Abstract *– Measurement of main contributions of single-hop delay peculiar to a software router is here dealt with. A new method is proposed, capable of distinguishing the time interval during which a generic packet stays in either input or output queue (queuing delay) of the router under analysis and that characterizing the effective routing process (processing delay) the packet undergoes. Thanks to proper measurement probes, i.e. kernel-layer functions, the method makes the occurring time of events of interest available at application-layer, thus giving the possibility of separately evaluating the aforementioned delays and, ultimately, pursuing a deeper insight of the considered router.*

After brief remarks concerning various delays a packet experiences when passing through a generic router, the measurement principle underling the method is presented in detail. Particular emphasis is put on its capability of locally monitoring the transit of each packet from the input to the output port of a software router along with main features and implementation issues of the proposed measurement probes. Results obtained in many experiments carried out on an ad-hoc test-bed in different operating conditions are then given in order to highlight method's reliability and effectiveness.

Keywords *– Computer networks test and measurement, Router characterization, Delay measurement, Router queuing delay, Router processing delay, Quality of service*

I. INTRODUCTION

The rapid evolution of applications and services offered in computer networks is supporting the definition of newer and newer requirements of Quality of Service (QoS, degree of user satisfaction), and, consequently, the development of more and more sophisticated strategies to grant the desired QoS level. Moreover, the market keeps on dictating optimal use of network resources, so networks have to be designed in such a way as to meet traffic demand and optimize performance at the same time [1],[2]. It is therefore necessary to point out key metrics capable of differentiating services, and to choose or define appropriate methods for their measurement in order to optimally dimension, design and plan a network [3]. These metrics, such as delay, jitter, loss-rate, available bandwidth, become the central meaning of QoS, in the sense that their value establish or characterize the service requirements the network should satisfy. Although the relevance of a given metric and related measurement method depends on the specific application, delay and available bandwidth are almost always of strong interest because of their key role in prediction and optimization of network end-to-end transport performance; this is particularly true whenever real time applications are involved [4].

The paper pays attention to single-hop delay, which accounts for the time a data packet takes to pass through a single router. Single-hop delay consists of three basic contributions: transmission delay, processing delay, and queuing delay. Transmission delay refers to the time amount the router takes both to acquire an entire packet from the input link and to place the same packet on the output link, processing delay is related to the time the router needs to determine the appropriate output port and forward the packet to it, and queuing delay occurs when there is contention at input and/or output ports of the router. While transmission delay is a function of the capacity of the input and output links as well as packet size (systematic contribution), processing and queuing delays prove fundamental to quantitatively describe the performance of the router and suitably model its behavior [5]-[8].

Measuring processing and queuing delays separately is not an easy task, especially in the presence of a hardware router. Only their sum, generally referred to as router transit time, can agilely be measured; two monitoring systems, located respectively at the input and output port of the router, can be adopted to the purpose [5],[6]. However, the random behavior of this sum, mainly due to the additional and unpredictable operations (checksum calculation, packet transfer from the input to output port) the router performs during packet processing, makes it difficult to extract from measurement results the two different delay contributions and, consequently, grant deep router characterization. Suitable techniques and procedures, capable of giving an effective answer to the highlighted challenge, are therefore claimed for by researchers, designers as well as service providers [1],[2],[8].

At this concern, an interesting proposal has recently appeared [5],[6]. Moving from the measurement of single-hop delay characterizing a hardware router, it succeeds in separating the two aforementioned contributions through the exploitation of reasonable hypotheses concerning processing delay and input and output queue model. No experimental evidence is, however, given for the formulated hypotheses to be maintained.

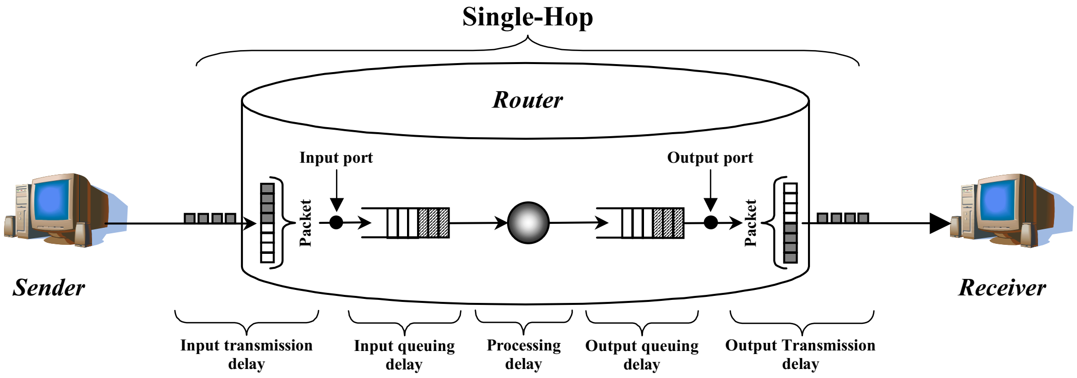

Fig.1. Main contributions of single-hop delay.

With reference to a software router, the authors propose a new method for separately measuring processing and queuing delays. Kernel code of the operating system installed on the router is properly modified in order to introduce specific software probes. The probes give the opportunity of pointing out the time instants in correspondence of which each packet (i) arrives at the input port of the router, (ii) is going to be processed (leaves the input queue), (iii) enters the output queue (the processing stage is over), and (iv) is going to be delivered on the output link. The time interval during which the packet stays in either input or output queue can be set apart from the effective processing delay the packet undergoes inside the router, thus proceeding to a satisfying router characterization. It is also possible to experimentally assess the rightness of the hypotheses formulated in [5],[6], and gain a deeper insight into router behavior for an accurate model to be achieved.

II. SINGLE-HOP DELAY

Single-hop delay accounts for the time a data packet takes to pass through a single router, i.e. the time amount a packet spends inside the router [5]-[8]. For a generic packet n, let we denote its arrival time at the input port of the router as $t_{in}(n)$, and its departure time from the output port as $t_{out}(n)$. Single-hop delay, $d(n)$, is thus equal to:

$$d(n) = t_{out}(n) - t_{in}(n) \quad . \qquad (1)$$

As sketched in Fig.1 with special regard to a tier-1 network [9], single-hop delay enlists three key contributions: transmission delay, processing delay, and queuing delay. Transmission delay, $d_t(n)$, refers to the time amount the router takes both to acquire an entire packet from the input link and to place the same packet on the output link. It depends on the capacity of input, C_{in}, and output, C_{out}, links as well as packet size, $l(n)$. The value of $d_t(n)$ can be attained through the following relation:

$$d_t(n) = d_{tin}(n) + d_{tout}(n) \quad ; \qquad (2)$$

where $d_{tin}(n)$, input transmission delay, and $d_{tout}(n)$, output transmission delay, are given respectively by

$$d_{tin}(n) = \frac{l(n)}{C_{in}} \quad \text{and} \quad d_{tout}(n) = \frac{l(n)}{C_{out}} \quad . \qquad (3)$$

Processing delay is the time the router needs to carry out the so-called routing process, i.e. to (i) examine packet header, (ii) find out packet route, and (iii) forward the packet to the appropriate output port. To fulfill the task, a proper comparison of packet header destination address to routing table entries is necessary.

Queuing delay occurs when there is contention at input and/or output ports of the router. It includes both the time interval a packet, which has already reached the input port of the router, wait before going through the routing process and the time amount the packet, already processed by router, spends before leaving the output port. Queuing delay depends on traffic load both along input and output link; it can thus vary over time.

III. PROPOSED METHOD

As stated above, the authors are going to propose a new method allowing the characterization of a software router in terms of processing and queuing delays.

The method aims at modifying the source code of the operating system running on the router under analysis in order to add proper measurement probes, which give the possibility of distinguishing the processing stage from queuing stage each packet experiences inside the router. It is suggested to act directly at kernel-layer rather than creating an ad-hoc measurement tool at application-layer based on some system calls.

Two reasons justify the choice. The first reason concerns management of network events by an operating system. Kernel-layer assures immediate management of network events through hardware and software interrupts; processes at application-layer are, instead, executed in deferring mode, i.e. active processes are scheduled by the kernel only if no interrupt has to be served. The second reason refers to running mode of an operating system. Kernel-layer considers interrupt service routines as atomic operations; no interruption can occur before their completion. At application layer, instead, processes are always managed in time-sharing

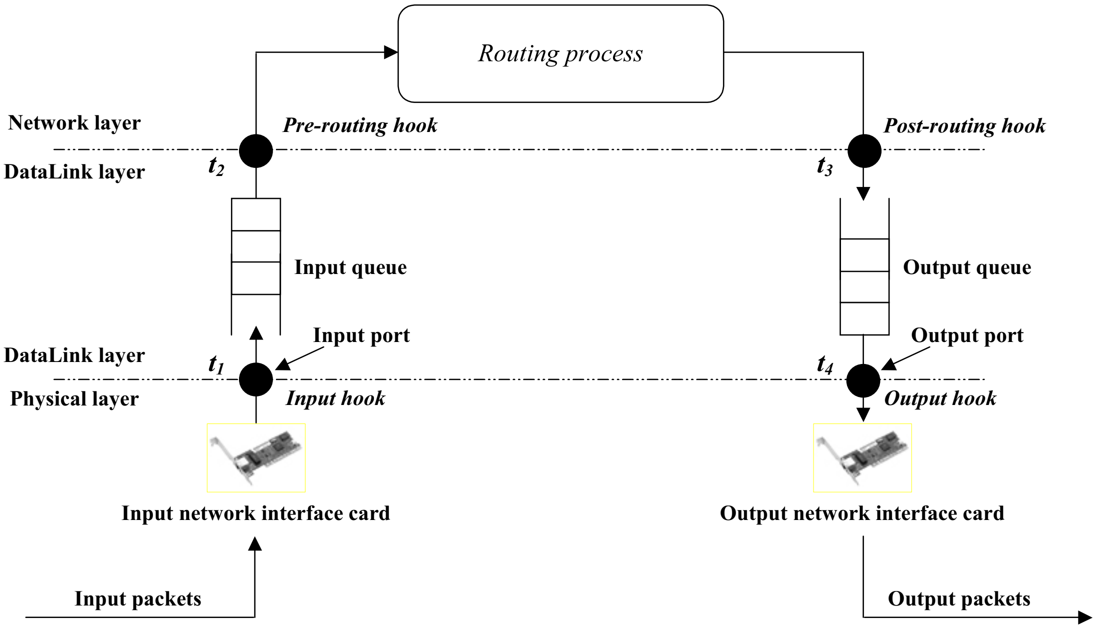

Fig.2. Location of the four measurement hooks with reference to a tier-1 network and a standard ISO-OSI protocol stack.

mode, and no function can have higher priority than that associated with kernel operations.

Hence, the measurement probes are realized by way of hooks, i.e. kernel-layer functions that make the occurring time of a specific event available at application-layer. For the sake of clarity, the location of the four measurement hooks proposed by the authors is sketched in Fig.2 with reference to a tier-1 network and a standard ISO-OSI protocol stack [9].

In particular:

- Input Hook operates at DataLink layer, and allows the pointing out of the time instant, t_1 (input time instant), in correspondence of which a data packet, already gained by the input network interface card, passes through the input port of the router. The size of the input queue is observed;
- Pre-Routing Hook refers to Network layer, and measures the time instant, t_2 (pre-routing time instant), when the router begins the decisional process in order to forward the input packet to the appropriate output port. The beginning of the processing stage of each packet is monitored;
- Post-Routing Hook, at Network layer, is capable of singling out the time of entry, t_3 (post-routing time instant), of a packet into the output queue for being transmitted. The end of the processing stage of each packet is detected;
- Output Hook, at DataLink layer, identifies the time instant, t_4, when a packet is delivered to the output network interface card. The time interval during which each frame stays in the output queue of the router is assessed.

More specifically, because data packets are gathered by the DataLink layer of the operating system, the Input Hook should be added in the kernel module implementing the driver of the input network interface card. Network layer manages the routing stage, the forwarding process of which prevails over the other ones for the tier-1 network in Fig.2; packets wait in the input queue for the routing module to be ready to process them, and are pushed into the output queue after being forwarded. Pre-Routing Hook and Post-Routing Hook should thus be included into the kernel module associated with Internet Protocol (IP). The last hook (Output Hook) concerns the transit of packets through the output port of the router in order to reach the physical link. Being this task managed by the DataLink layer of the operating system, the Output Hook should be introduced into the kernel module implementing the driver of the output network interface card.

Thanks to the proposed hooks, the input queuing delay, $d_{iq}(n)$,

$$d_{iq}(n) = t_2(n) - t_1(n) \quad , \tag{4}$$

processing delay, $d_p(n)$,

$$d_p(n) = t_3(n) - t_2(n) \quad , \tag{5}$$

and output queuing delay, $d_{oq}(n)$,

$$d_{oq}(n) = t_4(n) - t_3(n) \quad , \tag{6}$$

can be measured for each packet during a suitable time interval. Besides separating the fundamental contributions of the router transit time, measurement results provided by the hooks can allow the estimation of the probability density function (pdf) characterizing the aforementioned delays, d_{iq}, d_p, and d_{oq}.

Moreover a proper analysis of the number of packets that go through the input port of the router during each pre-routing inter-arrival, i.e. the interval, $\Delta_{pr}(n)$, between the pre-routing time instants related to two consecutive packets, n and n-1, accessing the routing stage,

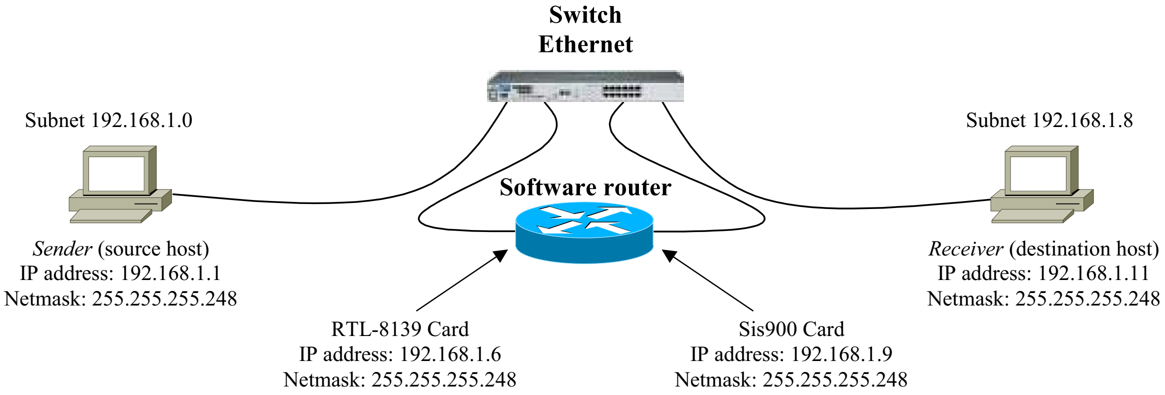

Fig.3. Test-bed adopted in the experiments.

$$\Delta_{pr}(n) = t_2(n) - t_2(n-1) \quad , \qquad (7)$$

gives the opportunity of characterizing the size of the input queue over time. In a similar manner, a proper analysis of the number of packets that come out of the routing stage during each output inter-arrival, i.e. the time, $\Delta_o(n)$, elapsing between the output time instants related to two consecutive packets, n and n-1, going through the output port of the router,

$$\Delta_o(n) = t_4(n) - t_4(n-1) \quad , \qquad (8)$$

allows the assessment of the evolution of the size of the output queue versus time.

Finally, from the analysis of input and output queuing delays, it is possible to assess the occurrence of time intervals during which the router does not process any packet even though (i) new packets leave the input queue, (ii) output queue is empty, and (iii) output link is not busy. These occurrences are generally referred to as coffee breaks, and the duration of the associated time intervals could affect the size of the input queue.

IV. EXPERIMENTS

A. Adopted test-bed

A simple test-bed, roughly sketched in Fig.3, has been set up. It consists of two end-points, *Sender* (source host) and *Receiver* (destination host), and one intermediate software router, all realized by way of identical personal computers featuring an Intel Celeron™ processor with 2.4 GHz clock frequency and 256 Mbyte RAM (Read Only Memory). The two end-points are connected to the router through an Ethernet switch, *D-Link DES-3226S*^{TM}, which supports both Ethernet (10 Mbit/s, half-duplex, and 20 Mbit/s, full-duplex) and Fast Ethernet (100 Mbit/s, half-duplex, and 200 Mbit/s, full-duplex) transmission mode. Type 5 UTP cables are exploited, the nominal capacity of which is 100 Mbit/s. Both end-points are equipped with *SiS900*^{TM} network interface card by *SiS*; as for the software router, the input network interface card is by *Realtek*, namely *RTL-8139*^{TM}; at the output, another *SiS900*^{TM} card by *SiS* is installed. The chosen network topology (end-to-end type) makes both input and output queue comply with a typical FIFO (First In First Out) model.

With no loss of generality, the router has been equipped with Fedora 2 Linux operating system (Linux OS), which provides a practical mechanism to configure and activate its kernel forwarding module in a very simple way. Moreover, the open-source license of the Linux OS allows the extension of its functionalities by directly modifying kernel source code [10].

According to what stated in the previous section, Linux kernel has been modified in such a way as to introduce, at DataLink layer, two hooks that get the time instants related to the transit of each packet respectively through the input and output port of the router. Pre-routing and post-routing time instants have, instead, been gained through the exploitation of *netfilter* infrastructure, already available in the same kernel at Network layer [11]. All code utilized can be downloaded from the web site given in [12].

Moreover, the simple end-to-end topology adopted has made packet identification unnecessary at each hook. The same is not true if a more complex router (e.g. characterized by more than one input and/or more than one output) is involved. In this case, a specific strategy for pursuing the overall path of each packet through the router has to be enlisted.

B. Test traffic generation

D-ITG (Distributed Internet Traffic Generator) software tool has been exploited for test traffic generation [13]. It is a platform capable of producing IPv4/IPv6 (Internet Protocol versions 4 and 6) traffic, peculiar to network, transport, and application layers [14], according to appropriate stochastic processes for inter-departure time and packet size random variables; several statistical distributions, such as Constant, Uniform, Exponential, Pareto, Cauchy, Normal, Poisson, and Gamma, are available [15]. A large variety of protocols, such as TCP (Transport Control Protocol), UDP (User Datagram Protocol), and ICMP (Internet Control Message Protocol) is supported, and it is also possible to set the TOS (Type of Service) and TTL (Time to Live*)* IP header fields [9]. Further details can be found in [13].

D-ITG software tool has been installed in *sender* mode on the source host and *receiver* mode on the destination host.

C. Test traffic features

Two sets of experiments have been carried out. In the first one, the *Sender* has been configured in such a way as to generate IP test traffic characterized by constant packet size and transmission rate, the values of which could suitably be adjusted. In the second set, the source host has produced IP test packets having variable size and constant rate. UDP has been used as transport layer protocol.

For each operative condition (i.e. a given value or statistical distribution of packet size and associated rate), one hundred tests have been carried out. In all tests, measurement time has been as long as to allow about 150,000 packets to be analyzed.

- *Constant size*

Different sizes of packet payload at transport layer have been considered: 20, 550, and 1400 bytes. The three values represent the most frequent payload sizes in Internet traffic, as described in [5]. With regard to transmission rate, the choice has fallen on lossless throughput, i.e. the maximum rate at which the count of packets transmitted by the source host equal the count of packets received by the destination host (no transmitted packet is dropped); critical conditions have thus been induced. Lossless throughput has experimentally been evaluated, and it has resulted equal to 17500 packet/s (~ 8.7 Mbit/s) for 20 bytes payload size, 19000 packet/s (~ 90 Mbit/s) for 550 bytes payload size, and 8000 packet/s (~ 92 Mbit/s) for 1400 bytes payload size.

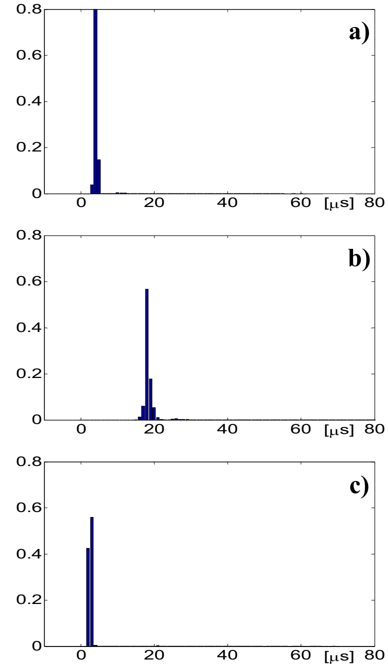

Fig.4. Relative frequency histogram related to the measures of a) input queuing, b) processing, and c) output queuing delay attained in the presence of packets having a payload size equal to 20 bytes.

- *Variable size*

Some tests characterized by random packet payload size and constant transmission rate have been carried out. According to what described in [16] with reference to VBR (Variable Bit Rate) video traffic, Gaussian distribution with mean and standard deviation equal respectively to 600 bytes and 200 bytes, and Pareto distribution with shape and scale parameters equal respectively to 12 and 550 have been considered. A transmission rate of 18000 packet/s (~ 92.5 Mbit/s, on average) has been imposed.

D. Results

Fig.4, Fig.5, Fig.6, and Fig.7 account for the results attained in the first set of experiments. Fig.4, Fig.5, and Fig.6 show the relative frequency histogram related to the measures of d_{iq}, d_p, and d_{oq} attained in the presence of packets having a payload size equal respectively to 20, 550, and 1400 bytes; the time resolution is 1 μs. For the sake of brevity, Fig.7 illustrates the relative frequency histogram related to the measures of input queue size, output queue size, and coffee-break inter-occurrence (i.e. the number of packets after the transmission of which a new coffee break is likely to occur) attained from the analysis only of packets the payload size of which is equal to 550 bytes. Roughly similar outcomes have been experimented for the other two payload sizes.

With regard to the second set of experiments, only Fig.8 is given. It, in particular, depicts the relative frequency histogram concerning the measures of the three aforementioned delays attained with Gaussian distributed packet payload size.

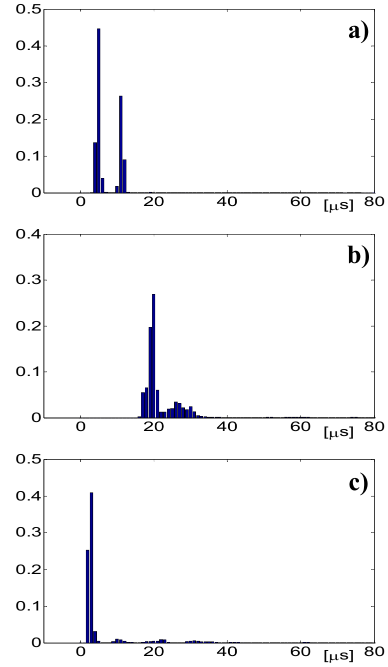

Fig.5. Relative frequency histogram related to the measures of a) input queuing, b) processing, and c) output queuing delay attained in the presence of packets having a payload size equal to 550 bytes.

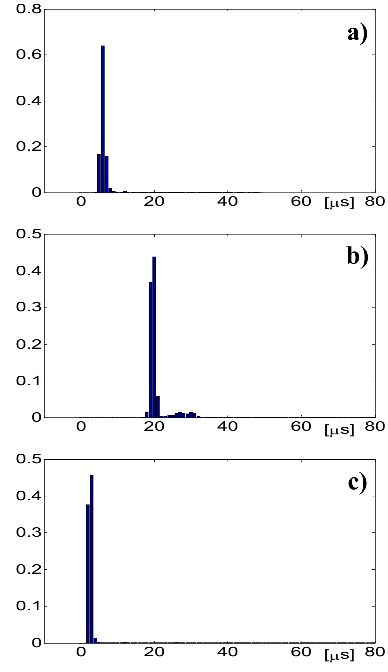

Fig.6. Relative frequency histogram related to the measures of a) input queuing, b) processing, and c) output queuing delay attained in the presence of packets having a payload size equal to 1400 bytes.

The time resolution is again 1 μs.

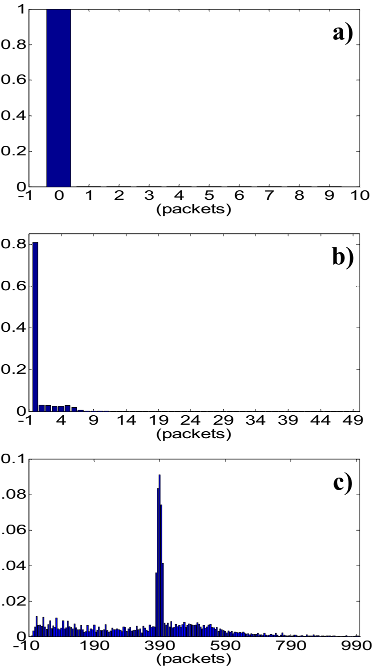

Fig.7. Relative frequency histogram related to the measures of a) input queue size, b) output queue size, c) coffee-break inter-occurrence attained in the presence of packets having a payload size equal to 550 bytes.

From the obtained results the following considerations can be drawn.

- Contrarily to what reported in [5],[6], the processing delay is not constant, and it cannot be evaluated by simply considering the minimum router transit time measured. Several peaks can, in fact, be noticed in the histograms given in Fig.4b, Fig.5b, and Fig.6b.
- Concerning d_{iq} and d_{oq}, the related histograms highlight, as expected for the simple test-bed adopted, limited queuing phenomena. This trend is confirmed by the results given in Fig.7a and Fig.7b.
- Certain regularity in coffee-break inter-occurrence has been experienced. As shown in Fig.7c, a new coffee break is, in fact, very likely to occur after about 390 packets have left the input queue to access the processing stage.
- Processing and queuing delays seem to be dependent only on the mean value adopted for packet payload size. The specific type of statistical distribution does not seem to affect measurement results. Comparison of the histograms in Fig.8 to those in Fig.5 clarifies the assumption.

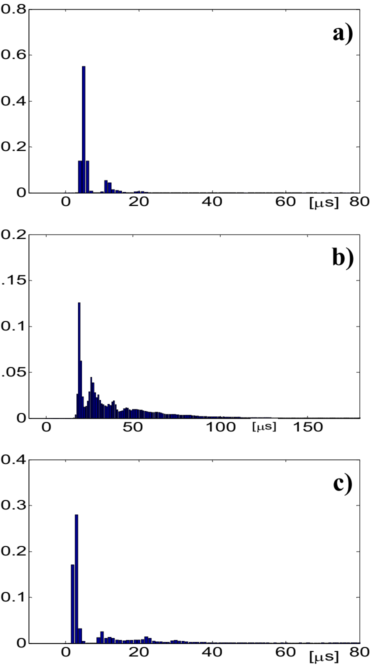

Fig.8. Relative frequency histogram related to the measures of a) input queuing, b) processing, and c) output queuing delay attained in the presence of packets having a Gaussian distributed payload size.

V. CONCLUSIONS

The possibility of characterizing a software router in depth through the separate examination of processing and queuing delays it introduces has been investigated. A new method has been presented, which is capable of singling out the significant time instants peculiar to the transit of each packet through the router under analysis; suitable measurement probes, realized by way of hooks inserted into the kernel of the operating system running on the router, have, in particular, been suggested.

Many experiments conducted on a simple test-bed have highlighted the efficacy and helpfulness of the method. Moreover, the good concurrence of the results attained in tests characterized by similar traffic conditions, in terms of packet payload size and transmission rate, has also proved its reliability.

Ongoing research activity is mainly oriented to assess the performance of the method in the presence both of competitive traffic, in the same test-bed, and a more complex network topology.

REFERENCES

[1] N.Hu, L.Li, Z.M.Mao, P.Steenkiste, J.Wang, “Locating Internet bottlenecks: algorithms, measurements, and implications,” in Proc. of *SIGCOMM'04,* Aug. 30–Sept. 3, 2004, Portland, USA, pp.41-54.

[2] J.M.Matta, A.Takeshita, “End-to-end voice over IP quality of service estimation through router queuing delay monitoring,” in Proc. of *IEEE GLOBECOM'02*, November 17-21, 2002, vol.3, pp.2458-2462.

[3] L.Angrisani, A.Baccigalupi, G.D’Angiolo, “A frame-level measurement apparatus for performance testing of ATM equipment,” *IEEE Trans. on Instr. and Meas.*, vol.52, No.1, pp.20-26, Feb. 2003.

[4] S.Wang, D.Xuan, R.Bettati, W.Zhao, “Providing absolute differentiated services for real-time applications in static-priority scheduling networks,” *IEEE/ACM Transactions on Networking*, vol.12, No.2, pp.326-339, April 2004.

[5] K.Papagiannaki, S.Moon, C.Fraleigh, P.Thiran, C.Diot, “Measurement and analysis of single-hop delay on an IP backbone network,” *IEEE Journal on Selected Areas in Communications*, vol.21, No.6, pp.908-921, August 2003.

[6] N.Hohn, D.Veitch, K.Papagiannaki, C.Diot, “Bridging router performance and queuing theory,” in Proc. of *SIGMETRICS/ Performance’04,* June 12–16, 2004, New York, NY, USA, pp.355-366.

[7] X.Qie, A.Bavier, L.Peterson, S.Karlin, “Scheduling computations on a software-based router,” in Proc. of *ACM SIGMETRICS 2001,* June 2001, Cambridge, MA. USA, pp.13-24.

[8] P.Zerfos, G.Zhong, J.Cheng, H.Luo, S.Lu, J.Jia-Ru Li, “DIRAC: a software-based wireless router system,” in Proc. of *IEEE MobiCom’03*, September 14–19, 2003, San Diego, California, USA, pp.230-244.

[9] A.S. Tanenbaum, *Computer Networks*, Prentice Hall, 2002.

[10] B.Henderson, “Linux loadable kernel module HOWTO”, The Linux Documentation Project, July 2001 (http://tldp.org/HOWTO/Module-HOWTO.)

[11] Netfilter Hacking HowTo, http://www.netfilter.org/documentation.

[12] http://www.criai.it/projects/measurement/package.tar.gz.

[13] S.Avallone, D.Emma, A.Pescapè, G.Ventre, “A distributed multiplatform architecture for traffic generation,” in Proc. of *Int. Symp. on Performance Evaluation of Computer and Telecommunication Systems*, July 25-29, 2004, San Jose, California (USA), pp.659-670.

[14] S.Deering, R.Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460, December 1998.

[15] D.Zwillinger, S.Kokoska, *Standard Probability and Statistics Tables and Formulae*, Chapman & Hall/CRC, 2000.

[16] M.W.Garrett, W.Willinger, “Analysis, modeling and generation of self-similar VBR video traffic,” in *ACM SIGCOMM Computer Communication Review*, vol.24, No.4, pp.269-280, October 1994.