Measurement of Processing and Queuing Delays Introduced by an Open-Source Router in a Single-Hop Network

Leopoldo Angrisani, Giorgio Ventre, *Associate Member, IEEE*, Lorenzo Peluso, and Annarita Tedesco

***Abstract*—Measurement of the main contributions to the single-hop delay introduced by an open-source router is dealt with. A new method is proposed, which is 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 the time interval 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 the 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 underlying 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 an open-source router along with main features and implementation issues of the proposed measurement probes. Results obtained in many experiments carried out on a suitable test bed in different operating conditions are then given in order to highlight the method's reliability and effectiveness.**

***Index Terms*—Computer networks test and measurement, delay measurement, quality of service (QoS), router characterization, router processing delay, router queuing delay.**

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].

Manuscript received June 15, 2005; revised April 11, 2006.

L. Angrisani and G. Ventre are with the Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, 80125 Napoli, Italy (e-mail: angrisan@unina.it).

L. Peluso is with the Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale (CRIAI), 80055 Portici (NA), Italy (e-mail: l.peluso@criai.it).

A. Tedesco is with the Ansaldo Signal S.p.A., 80147 Napoli, Italy (e-mail: tedesco.annarita@asf.ansaldo.it).

Digital Object Identifier 10.1109/TIM.2006.876542

These metrics, such as delay, jitter, loss rate, and available bandwidth, become the central meaning of QoS, in the sense that their values 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 the single-hop delay, which accounts for the time a data packet takes to pass through a single router. The single-hop delay consists of three basic contributions: transmission delay, processing delay, and queuing delay. Transmission delay refers to the amount of time 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, which are 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 the single-hop delay characterizing a hardware router, it succeeds in separating the two aforementioned contributions through the exploitation of reasonable hypotheses concerning processing

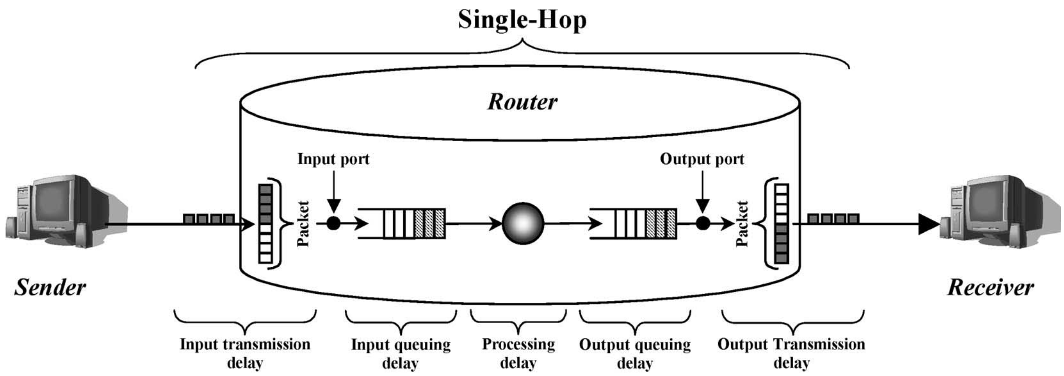


Fig. 1. Main contributions of the single-hop delay.

delay and input and output queue model. No experimental evidence is, however, given for the formulated hypotheses to be maintained.

With reference to an open-source 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 1) arrives at any input port of the router; 2) is going to be processed (leaves the input queue); 3) enters any output queue (the processing stage is over); and 4) 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

The 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 us denote its arrival time at the router as $t_{\mathrm{in}}(n)$ and its departure time from the router as $t_{\mathrm{out}}(n)$. The single-hop delay $d(n)$ is thus equal to

$$d(n) = t_{\mathrm{out}}(n) - t_{\mathrm{in}}(n). \tag{1}$$

As sketched in Fig. 1, with special regard to a tier-1 network [9], the single-hop delay enlists three key contributions: transmission delay, processing delay, and queuing delay. Transmission delay $d_{\mathrm{t}}(n)$ 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 $d_{\mathrm{t}}(n)$ can be attained through the following relation:

$$d_{\mathrm{t}}(n) = d_{\mathrm{tin}}(n) + d_{\mathrm{tout}}(n) \tag{2}$$

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

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

Processing delay is the time the router needs to carry out the so-called routing process, i.e., to 1) examine packet header; 2) find out packet route; and 3) 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, waits before going through the routing process and the time amount the packet already processed by the 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 an open-source 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 mode, and no function can have higher priority than that associated with kernel operations.

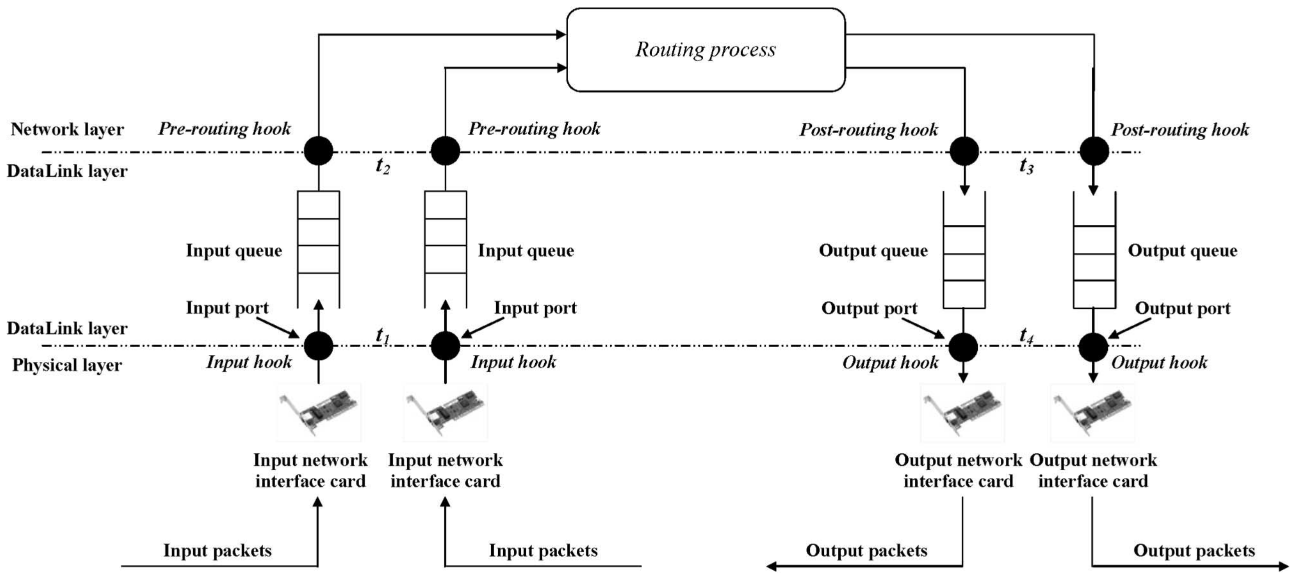


Fig. 2. Location of the four measurement hooks with reference to a router having two inputs and two outputs and a standard ISO-OSI protocol stack.

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 both to a router having two inputs and two outputs and a standard International Organization for Standardization (ISO)-open systems interconnection (OSI) protocol stack [9].

In particular, we have the following.

1) The input hook operates at the DataLink layer and allows the pointing out of the time instant t_1 (input time instant) in correspondence to which a data packet already gained by any input network interface card passes through the input port of the router. The size of any input queue is observed.
2) The prerouting hook refers to the Network layer and measures the time instant t_2 (prerouting 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.
3) The postrouting hook, at the Network layer, is capable of singling out the time of entry t_3 (postrouting time instant) of a packet into any output queue to be transmitted. The end of the processing stage of each packet is detected.
4) The output hook, at the DataLink layer, identifies the time instant t_4 in correspondence to which a packet is delivered to the output network interface card. The time interval during which each packet stays in any 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 drivers of input network interface cards. Network layer manages the routing stage; packets wait in the input queue for the routing module to be ready to process them and are pushed into the output queue after their route has been found out. The prerouting hook and postrouting hooks should thus be included in 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. Because this task is managed by the DataLink layer of the operating system, the output hook should be introduced into the kernel module implementing the drivers of output network interface cards. In addition, a specific strategy for pursuing the overall path of each packet through the router has to be enlisted (packet matching). The hash function, already used in [5], can prove helpful to the purpose. Thanks to the proposed hooks, input queuing delay $d_{\mathrm{iq}}(n)$

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

processing delay $d_{\mathrm{p}}(n)$

$$d_{\mathrm{p}}(n) = t_3(n) - t_2(n) \tag{5}$$

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

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

can be measured for each packet during a suitable time interval. Besides separating the fundamental contributions to 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 prerouting interarrival, i.e., the interval $\Delta_{\mathrm{pr}}(n)$ between the prerouting time instants related to two consecutive packets n and $n-1$ accessing the routing stage

$$\Delta_{\mathrm{pr}}(n) = t_2(n) - t_2(n-1) \tag{7}$$

gives the opportunity of characterizing the size of input queues over time. In a similar manner, a proper analysis of the number

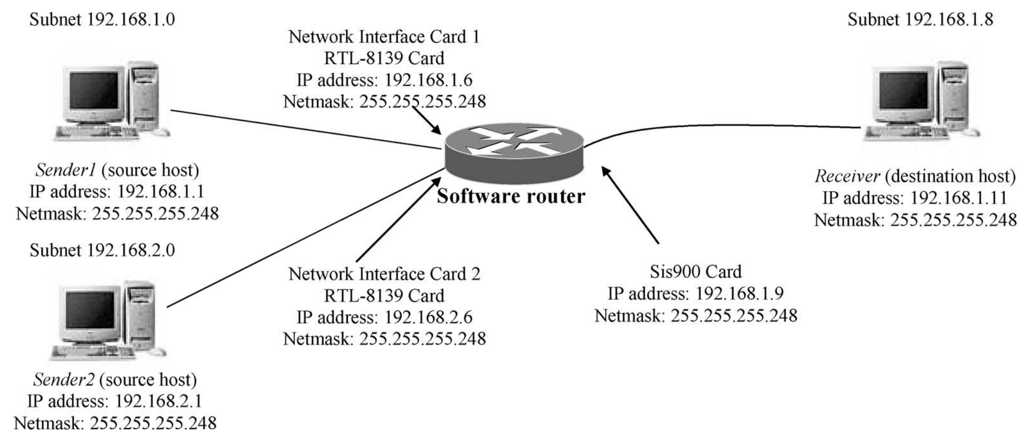


Fig. 3. Test bed adopted in the experiments.

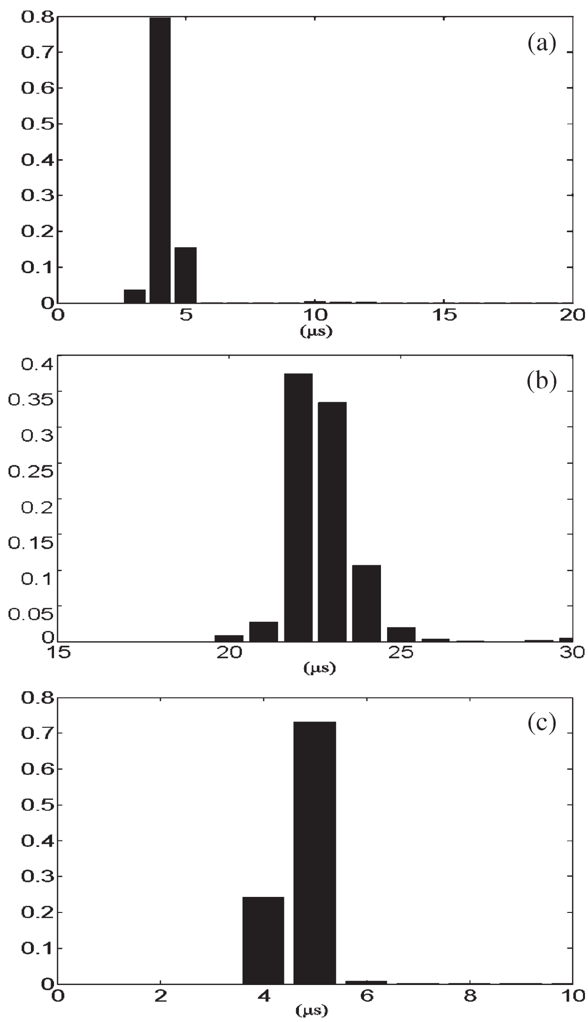


Fig. 4. Relative frequency histogram of the measures of (a) input queuing, (b) processing, and (c) output queuing delay attained in the presence of packets having a payload size equal to 20 B.

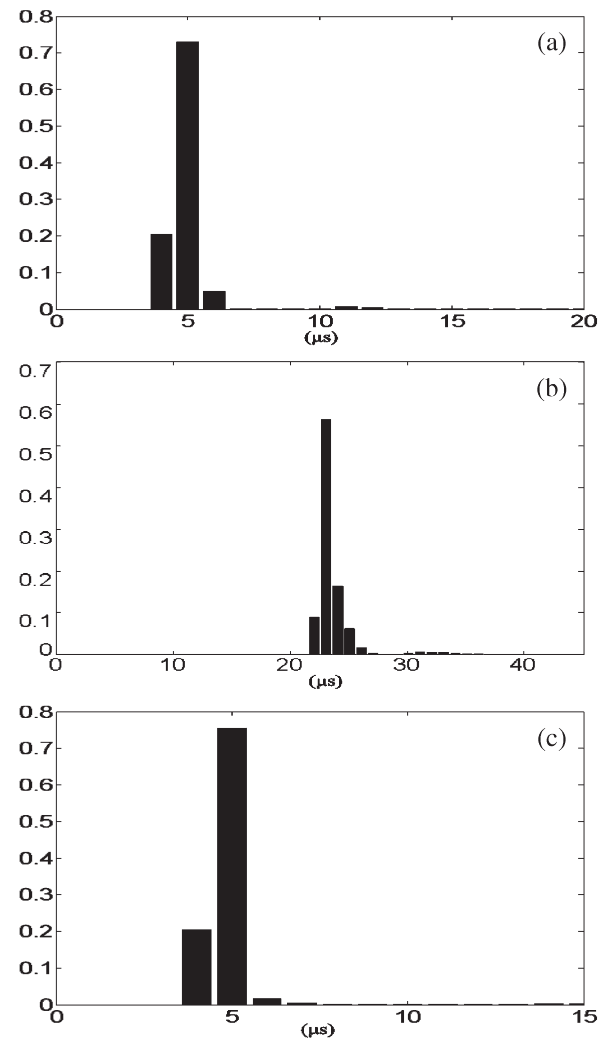


Fig. 5. Relative frequency histogram of the measures of (a) input queuing, (b) processing, and (c) output queuing delays attained in the presence of packets having a payload size equal to 550 B.

of packets that come out of the routing stage during each output interarrival, 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

$$\Delta_o(n) = t_4(n) - t_4(n-1) \tag{8}$$

allows the assessment of the evolution of the size of output queues 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 output queues are empty and output link is not busy.

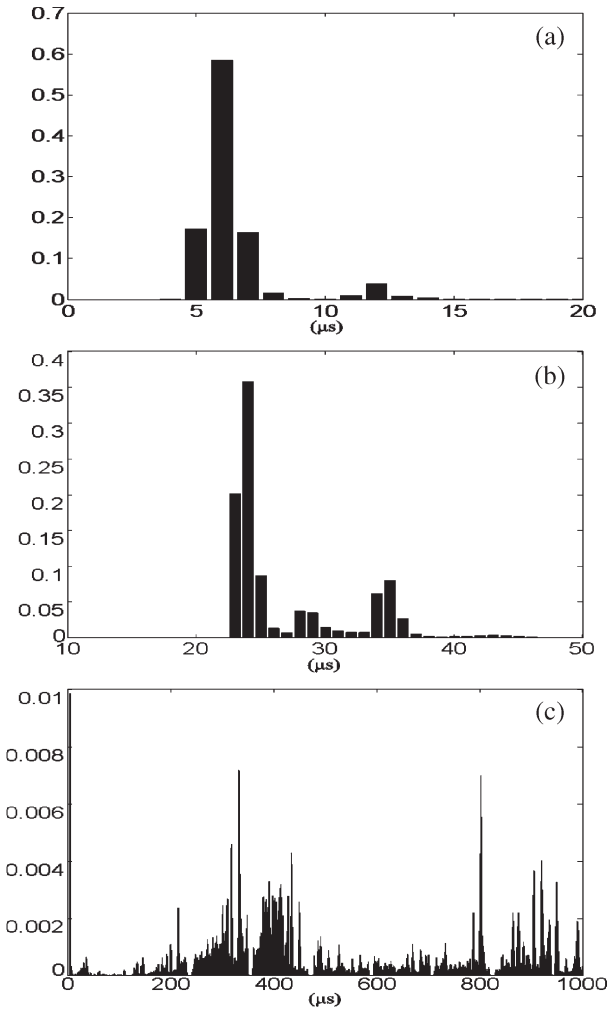


Fig. 6. Relative frequency histogram of the measures of (a) input queuing, (b) processing, and (c) output queuing delay attained in the presence of packets having a payload size equal to 1400 B.

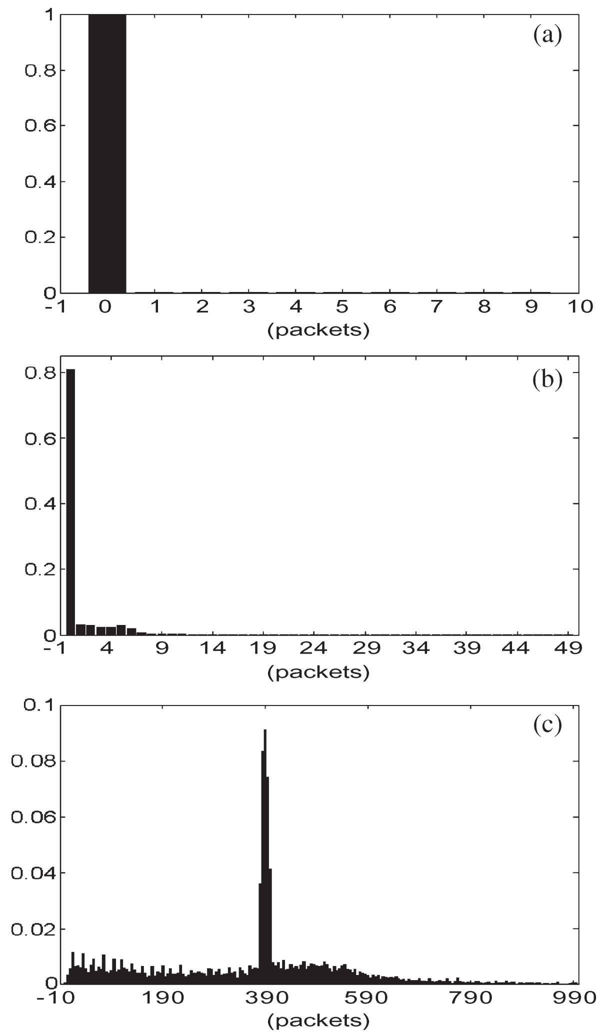


Fig. 7. Relative frequency histogram of (a) input queue sizes, (b) output queue sizes, and (c) coffee-break interoccurrences noticed in the presence of packets having a payload size equal to 550 B.

These occurrences are generally referred to as coffee breaks, and the duration of associated time intervals could affect the size of input queues.

IV. Experiments

A. Adopted Test Bed

A suitable test bed, roughly sketched in Fig. 3, has been set up. It consists of three endpoints [Sender1 (main traffic source host), Sender2 (competitive traffic source host) and Receiver (destination host)], and one intermediate open-source router, all realized by way of identical personal computers featuring an Intel Celeron processor with 2.4-GHz clock frequency and 256-MB random access memory (RAM). The three endpoints are connected to the router through type 5 unshielded twisted pair (UTP) cables, the nominal capacity of which is 100 Mb/s, and they are equipped with SiS900 network interface card by SiS. As for the router, the two input network interface cards are by Realtek, namely RTL-8139; at the output, another SiS900 card by SiS is installed. The chosen network topology makes input and output queues comply with a typical first in, first out (FIFO) 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, three hooks (two input hooks and one output hook) that get the time instants related to the transit of any packet through the input and output ports of the router. Prerouting and postrouting time instants have, instead, been gained through the exploitation of the netfilter infrastructure, that is already available in the same kernel at the Network layer [11]. All code utilized can be downloaded from the website given in [12].

B. Test Traffic Generation

The distributed Internet traffic generator (D-ITG) software tool has been exploited for test-traffic generation [13]. It is a platform capable of producing Internet Protocol versions 4

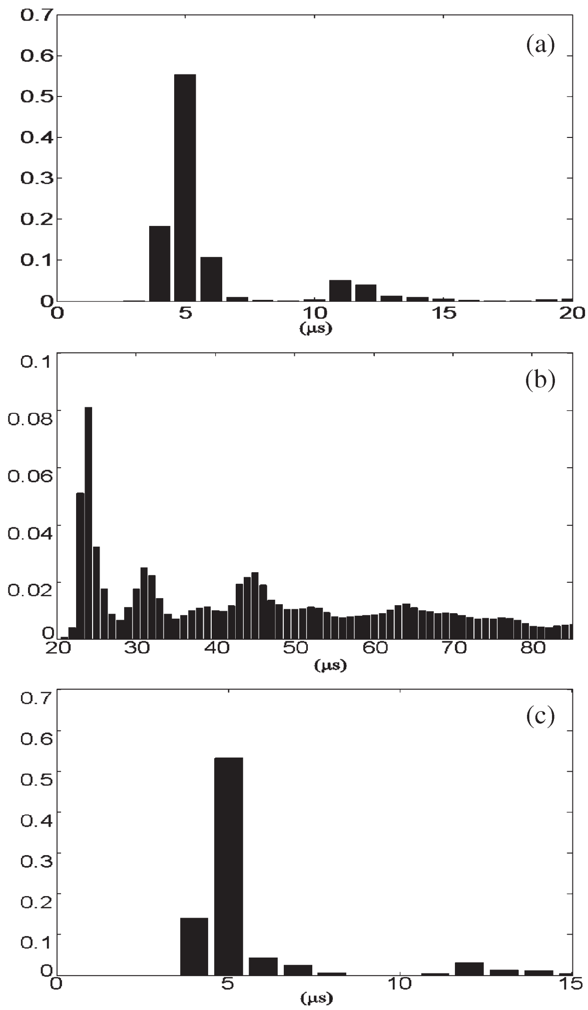


Fig. 8. Relative frequency histogram of the measures of (a) input queuing, (b) processing, and (c) output queuing delay attained in the presence of packets having a Gaussian-distributed payload size.

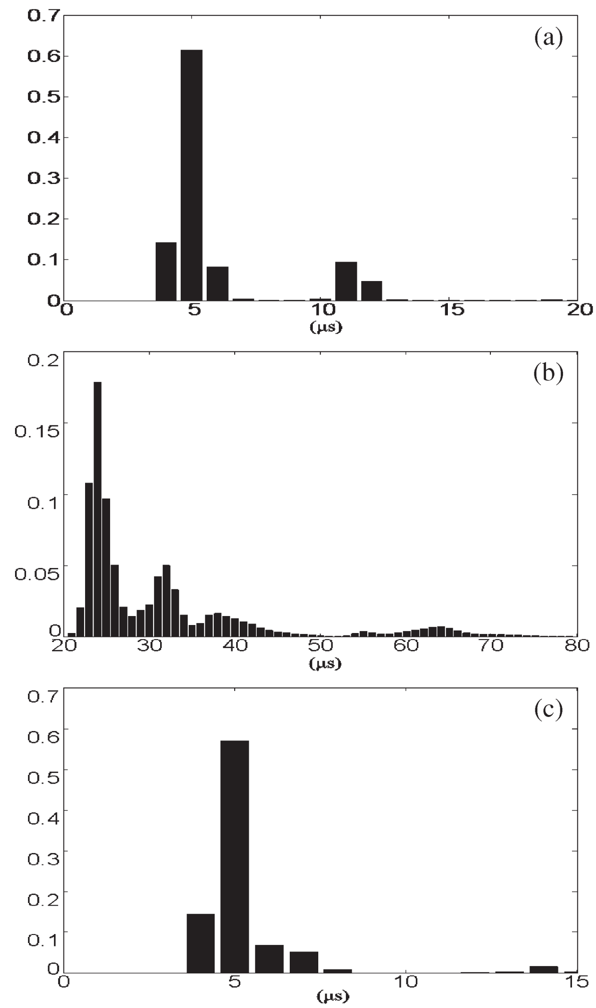


Fig. 9. Relative frequency histogram of the measures of (a) input queuing, (b) processing, and (c) output queuing delay attained in the presence of packets having Pareto-distributed interdeparture time.

and 6 (IPv4/IPv6) traffic that is peculiar to Network, Transport, and Application layers [14], according to appropriate stochastic processes for interdeparture 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 Transport Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) is supported, and it is also possible to set the type of service (TOS) and time to live (TTL) IP header fields [9]. Further details can be found in [13].

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

C. Test Traffic Features

Three types of IP test traffic, which in the following are referred to as Type1, Type2, and Type3, have been considered.

Type1 traffic has been characterized by constant packet size and transmission rate, the values of which could suitably be adjusted. Three different sizes of packet payload at transport layer have, in particular, been considered: 20, 550, and 1400 B. The three values represent the most frequent payload sizes in Internet traffic, as described in [5].

Type2 traffic has allowed for test packets having variable size and constant rate. According to what described in [16] with reference to variable bit rate (VBR) video traffic, packet payload sizes distributed according both to a Gaussian pdf, with mean (μ) and standard deviation (σ) equal, respectively, to 600 and 200 B, and a Pareto pdf, with shape α and scale k parameters equal, respectively, to 12 and 550, have been taken into account.

In Type3 traffic, IP packets exhibiting constant size and variable interdeparture time have been enlisted. For the sake of completeness, the same three sizes used in Type1 traffic have been considered. For each of them, two different distributions (Poisson and Pareto), have been associated to interdeparture time.

With regard to transmission rate, lossless throughput, i.e., the maximum rate at which the count of packets transmitted by the source hosts equals the count of packets received by the destination host (no transmitted packet is dropped) has always been pursued; critical conditions have thus been induced. Moreover, UDP has been used as transport layer protocol.

TABLE I
TESTS IN THE PRESENCE OF COMPETITIVE TRAFFIC: OPERATIVE CONDITIONS CONSIDERED IN THE FIRST SET OF EXPERIMENTS

	Sender2 packet payload size (bytes)		20	550	1400
Sender1 packet payload size (bytes)	*Sender1* transmission rate (packet/s)		*Sender2* transmission rate (packet/s)		
20	first scenario	11200 (~ 5.6 Mbit/s)	8000 (~ 4.0 Mbit/s)	7000 (~ 33.1 Mbit/s)	6500 (~ 74.9 Mbit/s)
	second scenario	8000 (~ 4.0 Mbit/s)	10000 (~ 5.0 Mbit/s)	8000 (~ 37.9 Mbit/s)	8000 (~ 92.1 Mbit/s)
550	first scenario	9800 (~ 46.4 Mbit/s)	6000 (~ 3.0 Mbit/s)	6000 (~ 28.4 Mbit/s)	4000 (~ 46.0 Mbit/s)
	second scenario	7000 (~ 33.1 Mbit/s)	10000 (~ 5.0 Mbit/s)	9000 (~ 42.6 Mbit/s)	5500 (~ 63.3 Mbit/s)
1400	first scenario	5950 (~ 68.5 Mbit/s)	10000 (~ 5.0 Mbit/s)	6000 (~ 28.4 Mbit/s)	2500 (~ 28.8 Mbit/s)
	second scenario	4250 (~ 48.9 Mbit/s)	9000 (~ 4.5 Mbit/s)	9000 (~ 42.6 Mbit/s)	4250 (~ 49.0 Mbit/s)

TABLE II
TESTS IN THE PRESENCE OF COMPETITIVE TRAFFIC: OPERATIVE CONDITIONS CONSIDERED IN THE SECOND SET OF EXPERIMENTS

	Sender2 packet payload size distribution		Gaussian μ = 600 bytes s = 200 bytes	Pareto a =12 k =550
Sender1 packet payload size (bytes)	*Sender1* transmission rate (packet/s)		*Sender2* average transmission rate (packet/s)	
20	first scenario	11200 (~ 5.6 Mbit/s)	7000 (~ 36.0 Mbit/s)	7000 (~ 33.1 Mbit/s)
	second scenario	8000 (~ 4.0 Mbit/s)	8000 (~ 4.1 Mbit/s)	8000 (~ 39.0 Mbit/s)
550	first scenario	9800 (~ 46.4 Mbit/s)	6000 (~ 31.0 Mbit/s)	6000 (~ 28.4 Mbit/s)
	second scenario	7000 (~ 33.1 Mbit/s)	9000 (~ 46.2 Mbit/s)	9000 (~ 42.6 Mbit/s)
1400	first scenario	5950 (~ 68.5 Mbit/s)	6000 (~ 31.0 Mbit/s)	6000 (~ 28.4 Mbit/s)
	second scenario	4250 (~ 48.9 Mbit/s)	9000 (~ 46.2 Mbit/s)	9000 (~ 42.6 Mbit/s)

D. Results Without Competitive Traffic

Three sets of experiments have been conducted without competitive traffic; Sender2 has remained inactive. In the first set, Type1 traffic has been produced by Sender1, and the imposed lossless throughput, which has been experimentally established, has been equal to 16 000 packet/s ($\sim$ 8.0 Mb/s) for 20-B payload size, 14 000 packet/s ($\sim$ 66 Mb/s) for 550-B payload size, and 8500 packet/s ($\sim$ 98 Mb/s) for 1400-B payload size. The second set has enlisted Type2 traffic with a lossless throughput of 16 000 packet/s ($\sim$ 82.2 Mb/s, on average). In the third set, the following choices have granted lossless throughput conditions for Type3 traffic. Concerning Poisson distribution, the average number of packets per second has been set equal to 16 000, 14 000, and 8500 for a payload size, respectively, of 20, 550, and 1400 B. As for Pareto distribution, the shape parameter α has been fixed equal to 1.5, while the scale parameter k has assumed the values of 0.03, 0.03, and 0.01 for a payload size, respectively, of 20, 550, and 1400 B.

For each operative condition (i.e., a given value or statistical distribution of packet size and associated rate), 100 tests have

TABLE III
TESTS IN THE PRESENCE OF COMPETITIVE TRAFFIC: OPERATIVE CONDITIONS CONSIDERED IN THE THIRD SET OF EXPERIMENTS

	Sender2 packet inter-departure time distribution	Pareto or Poisson (20 bytes payload size)	Pareto or Poisson (550 bytes payload size)	Pareto or Poisson (1400 bytes payload size)
Sender1 packet payload size (bytes)	*Sender1* transmission rate (packet/s)	*Sender2* transmission rate (packet/s)		
20	11200 (~ 5.6 Mbit/s)	8000 (~ 4.0 Mbit/s)	7000 (~ 33.1 Mbit/s)	6500 (~ 74.9 Mbit/s)
550	9800 (~ 46.4 Mbit/s)	6000 (~ 3.0 Mbit/s)	6000 (~ 28.4 Mbit/s)	4000 (~ 46.0 Mbit/s)
1400	5950 (~ 68.5 Mbit/s)	10000 (~ 5.0 Mbit/s)	6000 (~ 28.4 Mbit/s)	250 (~ 2.9 Mbit/s)

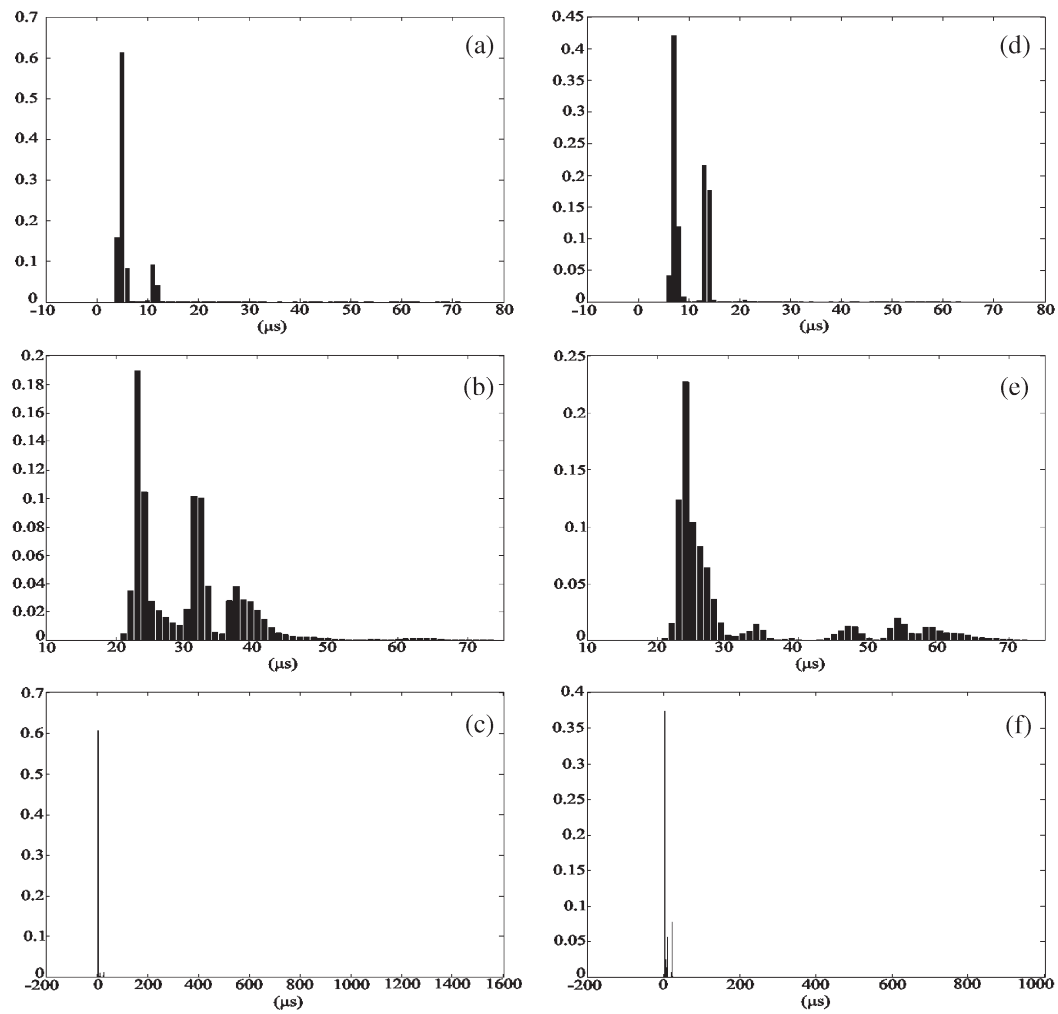


Fig. 10. Relative frequency histogram of the measures of (a) and (d) input queuing, (b) and (e) processing, and (c) and (f) output queuing delay associated with packets coming, respectively, from Sender1 and Sender2. Only Type1 traffic (550-B payload size) is involved, and the transmission rate of Sender1 and Sender2 is equal, respectively, to 9800 packet/s ($\sim$ 46.4 Mb/s) and 6000 packet/s ($\sim$ 28.4 Mb/s).

been carried out. In all tests, measurement time has been as long as to allow about 150 000 packets to be analyzed. Moreover, the simple end-to-end topology adopted has made packet matching unnecessary.

Figs. 4–7 account for the results attained in the first set of experiments. Figs. 4–6 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,

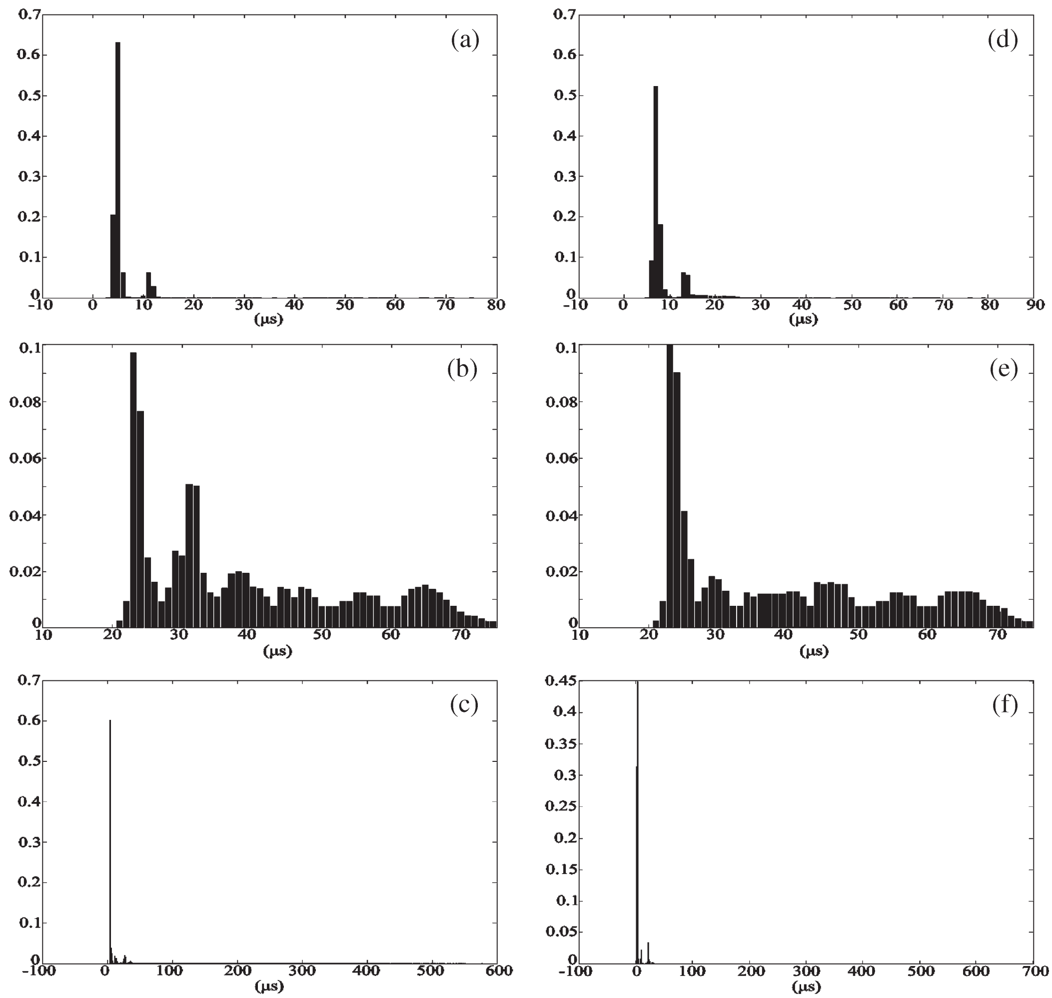


Fig. 11. Relative frequency histogram of the measures of (a) and (d) input queuing, (b) and (e) processing, and (c) and (f) output queuing delay associated with packets coming, respectively, from Sender1 and Sender2. Sender1 transmits packets with a fixed (550 B) payload size and at a rate equal to 7000 packet/s ($\sim$ 33.1 Mb/s), while Sender2 delivers packets characterized by a Gaussian-distributed payload size and a transmission rate of 9000 packet/s ($\sim$ 42.6 Mb/s).

to 20, 550, and 1400 B; 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 interoccurrence (i.e., the number of packets after the transmission of which a new coffee break is likely to occur) related to packets, the payload size of which is equal to 550 B. Roughly similar outcomes have been experimented for the other two payload sizes.

With regard to the other set of experiments, only Figs. 8 and 9 are given. They, in particular, depict the relative frequency histogram concerning the measures of the three aforementioned delays attained in the presence of packets characterized, respectively, by Gaussian-distributed payload size and constant rate, and constant size (550 B) and Pareto-distributed interdeparture time ($\alpha = 1.5$, $k = 0.02$). Time resolution is again 1 μs.

With regard to the first set of experiments, the following comments can be given.

1) Contrarily to what has been reported in [5] and [6], processing delay is not constant, and it cannot be evaluated by simply considering the minimum router transit time measured. In spite of the simple test bed adopted, several peaks can, in fact, be noticed in the histograms given in Fig. 4(b), Fig. 5(b), and Fig. 6(b).
2) Concerning d_{iq} and d_{oq}, the related histograms highlight, as expected, limited queuing phenomena. This trend is confirmed by the results given in Fig. 7(a) and (b).
3) Certain regularity in coffee-break interoccurrence has been experienced. As shown in Fig. 7(c), a new coffee break is, in fact, very likely to occur after about 390 packets have left the input queue to access processing stage. Further comments arise if the results achieved in the second and third set of experiments are taken into account.
4) More frequent measures of processing and queuing delays seem to be dependent only on the mean value adopted for packet payload size; the specific type of adopted statistical distributions do not seem to affect the position of main histogram peaks. Comparison of the results in Figs. 8 and 9 to those in Fig. 5 clarifies the assumption.
5) Paying attention to average delay value, significant differences have, instead, been noticed in the various measurement configurations. Certain influence of the adopted statistical distributions can thus be inferred.

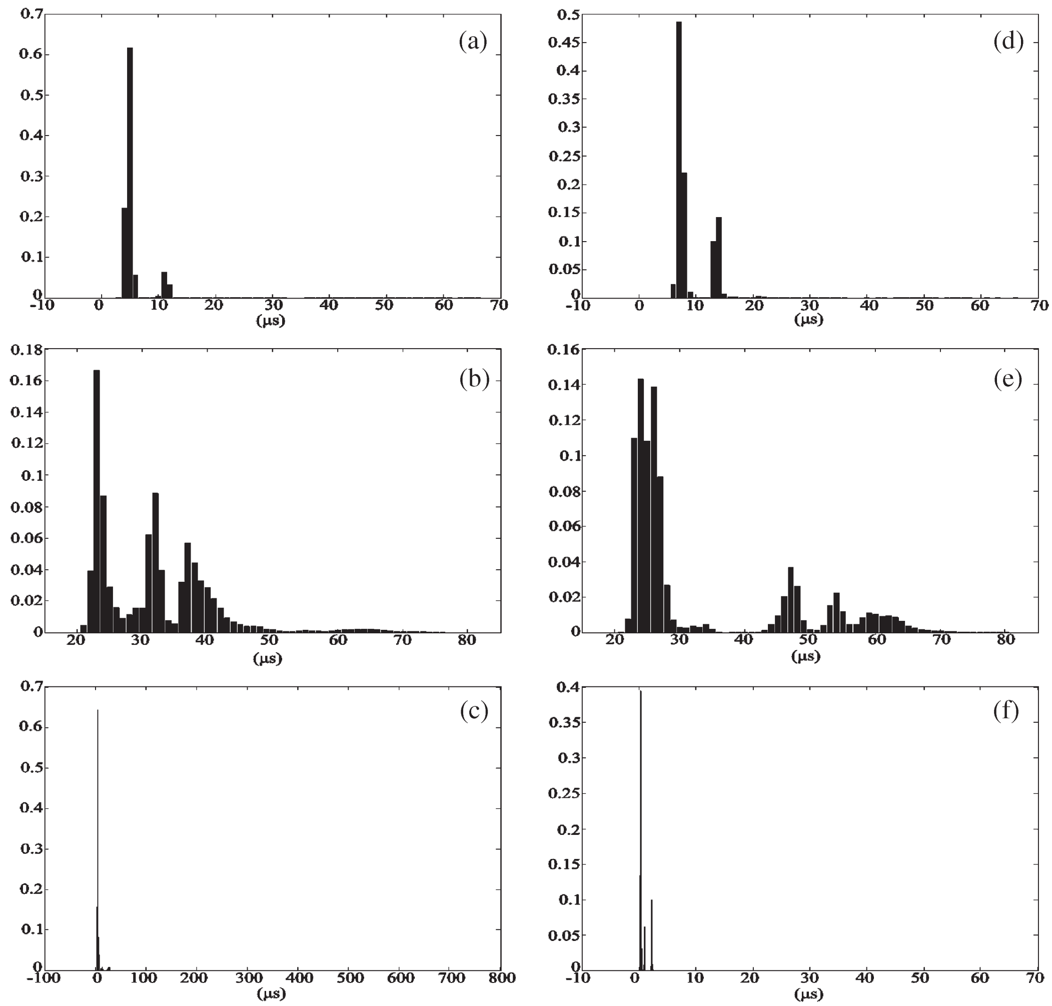


Fig. 12. Relative frequency histogram of the measures of (a) and (d) input queuing, (b) and (e) processing, and (c) and (f) output queuing delay associated with packets coming, respectively, from Sender1 and Sender2. Sender1 transmits packets with a fixed (550 B) payload size and at a rate equal to 9800 packet/s ($\sim$ 46.4 Mb/s), while Sender2 delivers packets with the same payload size and a Pareto-distributed interdeparture time (transmission rate equal about to 6000 packet/s).

E. Results in the Presence of Competitive Traffic

Three sets of experiments have been carried out also in the presence of competitive traffic; both source hosts have been active. Sender1 has always produced Type1 traffic, while Sender2 has been mandated to provide either Type1 (first set), Type2 (second set), or Type3 (third set) traffic. For the first and second set of experiments, two different scenarios have been considered, depending on transmission rates imposed on the source hosts. In the first (second) scenario, Sender1 has exhibited a transmission rate equal to 70% (50%) of the lossless throughput assessed without competitive traffic, while the transmission rate of Sender2 has been regulated in such a way as to get to the current lossless throughput, experimentally established. Only the first scenario has characterized the third set of experiments. Details concerning each operative condition (i.e., a given value or statistical distribution of packet size and associated rate both for Sender1 and Sender2) are given in Tables I–III.

For each operative condition, 100 tests have been carried out; in all tests, measurement time has been as long as to allow about 150 000 packets to be analyzed. Moreover, the adopted topology has claimed for packet matching. To this aim, the TOS field peculiar to IP protocol header has been exploited. The two source hosts have been configured in such way as to generate traffic characterized by a definite value (4 for Sender1 and 5 for Sender2) in its TOS field.

For the sake of brevity, only some of the results are shown in Figs. 10–12. Each of them gives the relative frequency histogram of the measures of d_{iq}, d_{p}, and d_{oq} associated both with packets transmitted by Sender1 and those provided by Sender2; time resolution is 1 μs. Fig. 10 concerns the first set of experiments. It describes the outcomes of the operating condition in which all packets are characterized by a payload size of 550 B and the transmission rate of Sender1 and Sender2 is equal, respectively, to 9800 packet/s ($\sim$ 46.4 Mb/s) and 6000 packet/s ($\sim$ 28.4 Mb/s). Fig. 11 refers to the second set of experiments. It accounts for the operative condition in which Sender1 transmits packets with a fixed (550 B) payload size and at a rate equal to 7000 packet/s ($\sim$ 33.1 Mb/s), while Sender2 delivers packets characterized by a Gaussian-distributed payload size and a transmission rate of 9000 packet/s ($\sim$ 42.6 Mb/s). Fig. 12 is related to the third set of experiments.

It goes over the operative condition in which Sender1 transmits packets with a fixed (550 B) payload size and at a rate equal to 9800 packet/s ($\sim$ 46.4 Mb/s), while Sender2 delivers packets with the same payload size and a Pareto-distributed interdeparture time (transmission rate equal about to 6000 packet/s).

From the analysis of all results, significant considerations can be drawn, which highlight the deep insight into router behavior granted by the method.

1) Despite roughly the same amount of traffic at the input port of the router, the single-hop delay has on average been greater than that experienced without competitive traffic; a mutual interference among packets coming from the two different hosts has taken place.
2) In each operative condition, more frequent measures of processing and queuing delays associated with the traffic delivered either by Sender1 or Sender2 have been close to those attained when the same type of traffic was the only one present.
3) As in the absence of competitive traffic, the histograms related to $d_{\rm iq}$ and $d_{\rm oq}$ highlight limited queuing phenomena.
4) With regard to the processing delay $d_{\rm p}$, all histograms emphasize an increased average and dispersion of the measured values with respect to those attained without competitive traffic, especially in the third set of experiments. Two main reasons justify this outcome. The first concerns the longer time taken by the router to carry out its routing process; a slightly more complicated routing table has to be checked with two different source IP addresses being involved. The second reason refers to the greater number of coffee-break occurrences; the kernel is mandated to a more intense scheduling activity for the management of two input network interface cards.

V. Conclusion

The possibility of characterizing an open-source 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 suitable test bed, arranged by the authors and including two source hosts, one destination host, and one intermediate router, 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. The application of the method is, however, precluded in those cases in which the kernel source code is either not available or not modifiable due to hardware and/or software protection limitations.

Ongoing research activity is mainly oriented to assess the performance of the method in a more complex network topology and to consider a tradeoff approach, which enlists measurement probes at the application layer, to enlarge its range of applicability.

References

[1] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating Internet bottlenecks: Algorithms, measurements, and implications," in *Proc. SIGCOMM*, Portland, OR, Aug. 30–Sep. 3, 2004, pp. 41–54.

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

[3] L. Angrisani, A. Baccigalupi, and G. D'Angiolo, "A frame-level measurement apparatus for performance testing of ATM equipment," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 1, pp. 20–26, Feb. 2003.

[4] S. Wang, D. Xuan, R. Bettati, and W. Zhao, "Providing absolute differentiated services for real-time applications in static-priority scheduling networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 326–339, Apr. 2004.

[5] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 908–921, Aug. 2003.

[6] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot, "Bridging router performance and queuing theory," in *Proc. SIGMETRICS/Perform.*, New York, Jun. 12–16, 2004, pp. 355–366.

[7] X. Qie, A. Bavier, L. Peterson, and S. Karlin, "Scheduling computations on a software-based router," in *Proc. ACM SIGMETRICS*, Cambridge, MA, Jun. 2001, pp. 13–24.

[8] P. Zerfos, G. Zhong, J. Cheng, H. Luo, S. Lu, and J. J.-R. Li, "DIRAC: A software-based wireless router system," in *Proc. IEEE MobiCom*, San Diego, CA, Sep. 14–19, 2003, pp. 230–244.

[9] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[10] B. Henderson. (Jul. 2001). "Linux loadable kernel module HOWTO," *The Linux Documentation Project*. [Online]. Available: http://tldp.org/HOWTO/Module-HOWTO

[11] Version 1.13, R. Russell, *Linux Networking-concepts HOWTO*, Jul. 29, 2001. [Online]. Available: http://www.netfilter.org/documentation

[12] Measurement tool kernel Implementation. [Online]. Available: http://www.criai.it/projects/measurement/package.tar.gz

[13] S. Avallone, D. Emma, A. Pescapè, and G. Ventre, "A distributed multiplatform architecture for traffic generation," in *Proc. Int. Symp. Perform. Eval. Computer Telecommun. Syst.*, San Jose, CA, Jul. 25–29, 2004, pp. 659–670.

[14] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, Dec. 1998, RFC 2460.

[15] D. Zwillinger and S. Kokoska, *Standard Probability and Statistics Tables and Formulae*. Boca Raton, FL: CRC, 2000.

[16] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," *ACM SIGCOMM Computer Communication Rev.*, vol. 24, no. 4, pp. 269–280, Oct. 1994.

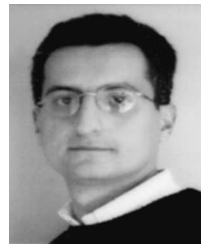
Leopoldo Angrisani was born in Nocera Superiore, SA, Italy, on April 16, 1969. He received the M.S. degree (*cum laude*) in electronic engineering from the University of Salerno, Salerno, Italy, in 1993 and the Ph.D. degree in electrical engineering from the University of Napoli Federico II, Napoli, Italy, in 1997.

Since 2002, he has been Associate Professor of electrical and electronic measurements, Department of Computer Science and Control Systems, University of Napoli Federico II. He is currently involved in research into new digital signal-processing methods for assessing the performance of wireless or wired digital communication systems and apparatuses, testing and troubleshooting communication networks, detecting, measuring, and classifying nonstationary signals, and evaluating uncertainty in indirect measurements.

Giorgio Ventre (M'05–A'05) received the Laurea degree in electronic engineering and the Ph.D. degree in computer engineering from the University of Napoli Federico II, Naples, Italy, in 1985 and 1989, respectively.

He is a Professor of computer networks in the Department of Computer Engineering and Systems, University of Napoli Federico II, Napoli, Italy, where he is leader of the Computers for Interaction and Communications (COMICS) team, which is a research initiative in the areas of networking and multimedia communications. After working at ITEM [the first research laboratory of the Italian University Consortium for Informatics (CINI)], he is now President and CEO of Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale (CRIAI), which is a research company active in the areas of information technologies. As leader of the networking research group at University of Napoli Federico II, he was Principal Investigator for a number of national and international research projects. Currently, he is involved with the Emerging Technologies in Networking (E-NEXT) Network of Excellence of the VI Framework Programme of the European Union, where he is leading the activities of the Working Group (WG) on Traffic Engineering and Monitoring. He has coauthored more than 150 publications.

Mr. Ventre is member of the IEEE Computer Society and of the Association for Computing Machinery (ACM).



Lorenzo Peluso was born in Napoli (NA), Italy, on August 31, 1974. He received the M.S. degree in telecommunication engineering from the University of Napoli Federico II in 2001.

Since 2004, he has been working at the Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale (CRIAI), Portici, Italy, in the research group on computer network design and testing. His current research interests include new measurement procedures for communication networks, new infrastructure for distributed network security, context-aware services, and adaptive multimedia systems.

Annarita Tedesco was born in Pompei (NA), Italy, on August 8, 1973. She received the M.S. degree in telecommunication engineering from the University of Napoli Federico II, Napoli, Italy, in 2002.

From 2003 to 2005, she worked at the Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale (CRIAI), Portici, Italy, in the research group on computer network design and testing. Since 2005, she has been working at Ansaldo Signal S.p.A., Napoli, in the engineering division. Her research activity is mainly focused on new digital signal-processing methods for testing digital wireless equipment and new measurement procedures for designing and testing communication networks.