

## MEASUREMENT OF ONE-WAY DELAYS IN IP NETWORKS

N. I. Vinogradov, E. S. Sagatov, and A. M. Sukhov

UDC 004.738.5

*The NetTestBox combined hardware–software complex, which supports measurement of IP network performance metrics, is created. The metrics are intended for estimation of the quality of Internet connections. The basic aspects of the implementation of measurements of one-way network delays in the complex are described. The selection of a network time protocol server for NetTestBox is evaluated. Results of measurements of the duration of the main components of a network delay are presented.*

**Keywords:** performance metrics, one-way network delay, network time protocol server, exact time server, timing accuracy.

Telecommunications is an important constituent element of technological progress while at the same time network technologies are continuously developing. The aggregate traffic continues to grow at exponential rates, a circumstance that requires constant modernization of the equipment and the introduction of new data transmission technologies.

A universal tool for evaluating the quality of data transmission is needed to enable monitoring of the state of a network and determination of the sources of problems. A family of IETF (Internet Engineering Task Force) standards for the description of the quality of Internet connections was created in the late 1990s. These standards are combined under the general designation of IPPS (IP Performance Metrics) [1]. The basic performance metrics relate to delays of packets (one- or two-way delays), network jitter (a variation of a delay), losses of packets, and the available capacity of a channel [2]. However, there are no simple and measurement tools available today that could be used to determine the magnitude of one-way delays and the available capacity of distant network nodes. Losses of packets and network jitter are key parameters in multimedia applications related to user communication in a network. Such applications include Internet telephony and video conferencing [3].

The main difficulty related to measurement of the available capacity of a network channel is that IP networks are based on the principle of packet switching. Different types of traffic compete between each other in common communication channels and estimation of the parameters of the connections related to allocation of a band represents a rather complicated problem. One-way delays of packets (OWD), one of the key performance metrics of IP networks, is described in [4]. According to the standard [5], this metric is decisive for real-time applications and interactive applications as well as management systems that utilize the global network. One-way delays are also used to identify asymmetric IP channels, in which the path of a network packet differs between the forward and reverse directions. Applications such as Internet agents [6], medical networks, and tactile Internet [7] have appeared and are being actively developed today.

The main difficulty in measurements of one-way delays lies in the need for synchronization of systems time in remote devices. Systems of global positioning and the software of network time protocols (exact time servers) are used for such synchronization. Below, we present a survey of one-way delay measurement systems.

The objective of the present study is to present an analysis of the process of measuring a one-way network delays in the NetTest Box combined hardware–software complex [8]. The complex was developed at the Samara National Research University and began to successfully function in 2015. Four networks of the complex are now functioning, three in Russia (Samara, Togliatti, Rostov-on-Don) and one in the United States (Columbia, Missouri).

**Survey of existing measurement methods.** The European Regional Internet Recorder RIPE NCC [9], One-Way ping (OWAMP) [10], and the Russian development of NetTestBox [8] are the principal projects involved in measurement of one-way network delays. In 2001, RIPE NCC created the special service Test Traffic Measurement (RIPE TTM) for monitoring of one-way delays. By means of the service, it is possible to measure three of the four performance metrics of IP networks: one-way network delay, a variation in delay (jitter), and losses of packets between measurement nodes on the Internet. The experimental data are stored and processed statistically.

In terms of hardware, RIPE TTM was implemented in the following way. A server with special board and external GPS antenna for stable reception of signals from satellites were located at the ultimate user. The precision of measurements of one-way network delays was achieved by attaching a precise time mark to the packet. For this purpose, a rectangular *PPS* (pulse per second) signal with 1 sec period is generated at the output of the GPS receiver once the geographic location is found. The signal front was synchronized with UTC time (Coordinated Universal Time) to within  $\pm 100$  nsec. Mismatch of the systems time of two measurement nodes amounts to  $\pm 500$  nsec [11]. The service has not become widespread due to the high cost of the installation and maintenance and, therefore, the project was curtailed in 2014.

The OWAMP program is provided by a group of Internet2 developers and is standardized by IETF in the form of a document [10]. There are now two software implementations of the protocol, the first proposed and developed by the Internet2 developer firm [10] and the J-OWAMP project implemented in the Java programming language by a group of specialists from the Institute of Telecommunications of Aveiro University (Portugal). The J-OWAMP project is a protocol that supports measurement of one-way network delays with the use of time synchronization of measurement nodes by means of the *NTP* network time protocol.

In 2005, the J-OWAMP developers conducted several experiments designed to measure one-way network delays between Aveiro (Portugal) and Salzburg (Austria). Experimental data that confirmed that J-OWAMP and OWAMP can be used in a unified measurement infrastructure for the purpose of acquiring exact data on one-way network delays between nodes are presented in [12].

**NetTestBox combined hardware-software complex.** The NetTestBox complex constitutes a compact, high-performance single-board computer with the specialized client server software *owping* and multisystem GLONASS/BeiDou/GPS receiver, which assigns a synchronization timing mark to the modern *Chrony NTP* server [8]. The four main network performance metrics may be measured by the complex, which is also able to store statistics and transmit the statistics to a remote server for further processing of the data. The hardware component of the complex is based on a Raspberry single-board computer PI Model B with ARM architecture and the UBX-M8030 navigation receiver from the firm of U-blox (Switzerland). The overall dimensions of the measurement node amount to  $90 \times 70 \times 25$  mm. A 15-W, 5 V dc stabilized source serves as the power source for the computer and the peripheral devices connected to it. An adapted version of the Linux operating system, called Raspbian with a complete set of software for support of the hardware devices of the computer is has been started on the computer. Exchange with the storage server and processing of the experimental data is realized by means of a Fast Ethernet connection to the Internet.

Precise synchronization of systems time is supported by the *Chrony NTP* server, which processes the *PPS* signal from the navigation receiver as well as data from several Internet network time protocol servers. The *owping* client server program was developed to enable measurements of the network metrics. The client part of the program, which is launched on each measurement node, transmits measurement packets to the server part, receives response packets, establishes the exact time of their transmission and reception, performs primary processing of the obtained data, and transmits the data to a unified storage server. The server part of the program responds to newly arriving packets with queries, adding the fixed reception time of the query and the transmission time of the response to the response packet.

**Nature of network delays.** Different classification schemes have been proposed for use in the analysis of network delays. There are universal approaches that are recognized by all researchers, one of which consists in the analysis of the nature of network delays. Two components are identified in this approach, a physical component and a telecommunications component. The physical delay  $D_{ph}$  is the time it takes an electromagnetic wave containing information about a digital code to propagate through a telecommunication channel. The telecommunication component  $D_{tel}$  contains all the delays related to the creation and processing of the packet. Queuing theory serves as the theoretical basis for the description of this component.

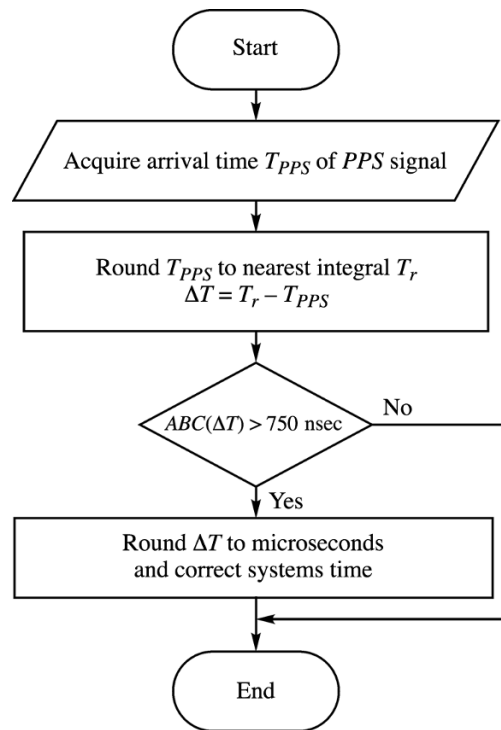


Fig. 1. Block diagram of synchronization algorithm: *PPS*, time synchronization signal;  $T_{PPS}$ , acquisition time of *PPS* signal;  $T_r$ , integral value of arrival time of *PPS* signal;  $ABC(\Delta T)$ , absolute error of synchronization of systems time.

From the point of view of mathematics, a delay may also be divided into two components. The first component is a constant quantity determined by the length of the minimal delay  $D_{\min}$  and the second component a variable quantity which is random in nature. The physical component of a delay is bounded above by the minimum delay time,  $D_{ph} \leq D_{\min}$  [13].

In the analysis of data by RIPE TTM, it was proposed that the following delays should be extracted from the telecommunication component [8, 10]:

*processing*  $D_p$ , or the time needed to create a packet with data, prepare the packet for transmission, and process the packet at an assignment point for extraction of data;

*transmission*  $D_t$ , or the time it takes to transmit a packet from the first to the last bit in a telecommunication channel. This quantity depends on the width of the communication channel; and

*queuing*  $D_q$ , or the time expended by the packet in waiting and processing in the network devices. This delay is described by queuing theory.

**Method of time synchronization.** Let us consider the main stages in the measurement of one-way network delays and methods of measuring and synchronizing time.

In the NetTestBox device, acquisition of a synchronizing time *PPS* signal occurs through a *GPIO* (general-purpose input-output) processor interrupt. Such an interrupt has a high priority, consequently, the remaining executable problems will be halted by the kernel of the operating system and processing of a newly arriving *PPS* signal by the kernel will commence once an interrupt is obtained. In the device, the *PPS-gpio* module of the kernel of the Linux operating system is responsible for processing this interrupt. The systems time is controlled by the kernel of the operating system based on the processor tick counting or data of the hardware timer. The NetTestBox equipment possesses a built-in hardware timer that generates readings every 52 nsec.

An *NTP* server, which corrects the systems time of the operating system, is used to synchronize and increase the precision of the time count. Upon the arrival of the *PPS* signal, the *NTP* server compares its data with the systems time and

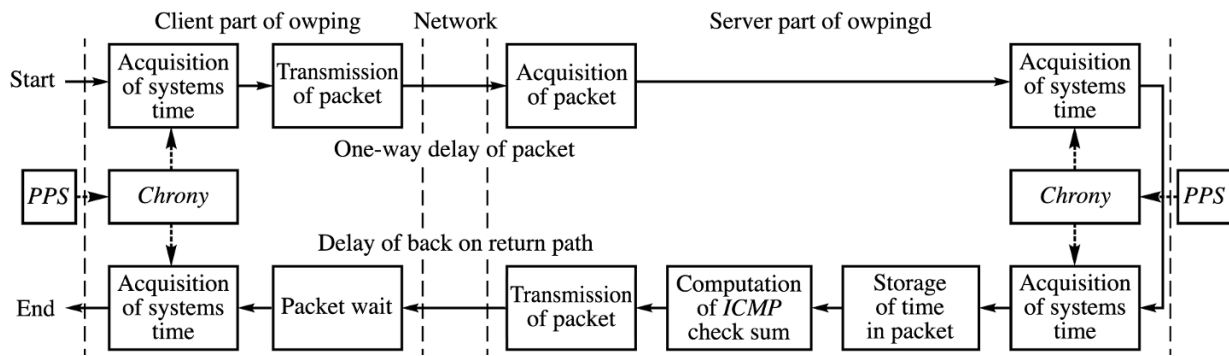


Fig. 2. Implementation of method of measurement of delays of packets by means of NetTestBox: *PPS*, time synchronization signal; *Chrony*, NTP server; *owping*, client component of software of software of NetTestBox node; *owpingd*, server component of software of NetTestBox node; *ICMP*, packet of protocol of inter-network control messages.

establishes the difference with the nearest integral value in seconds. The server first synchronizes the systems time in the global network with at least four network time protocol servers. Therefore, data on seconds are considered precise and only the fractional part is subjected to correction.

This type of *NTP* server instantaneously varies the systems time only where there is significant divergence in the time scales (more than 1000 sec), and in other cases gradation synchronization is used. Special correction functions are provided in the Linux operating system for automatic synchronization. The functions accelerate or slow down the course of systems time and the algorithm used in this method functions with microseconds.

Let us determine the error in synchronization of the receiver's systems time, using different methods of synchronization for purposes of comparison, such as *ntpd* server, *Chrony* server, the built-in utility *PPS synch*, and one-time synchronization without subsequent correction. In all of these methods, a *PPS* signal is used as the reference network time protocol source. The error in the synchronization of the systems time is found by establishing the systems time at the moment of arrival of the *PPS* signal. The *PPS* utility, which functions according to the algorithm presented in Fig. 1, was developed for use in verification of the precision of time synchronization by the *ntpd* and *Chrony* servers.

The *PPS-sync* utility establishes the arrival time  $T_{PPS}$  of the processor interrupt from the *PPS* signal to within nanoseconds. The systems time  $T_r$  is rounded to the nearest integral second, which makes it possible to calculate the correction  $\Delta T = T_r - T_{PPS}$ . If the absolute value of the correction  $\Delta T$  is greater than 750 nsec,  $\Delta T$  is rounded to within microseconds and the systems time is corrected to this value. In the interval from  $-750$  to  $750$  nsec, the time is not corrected.

When configuring the software, it was established that the readings of the *PPS-gpio* module are not received by the NetTestBox equipment as reliable information. In tests conducted on a computer, the *PPS* signal was received through an RS-232 interface, while the *ntpd* server exhibited worse synchronization precision characteristics than did the *Chrony* server. Similar results were confirmed in [14]. Correction by means of the *ntpd* server was not considered in the study.

Both methods of correction successfully handled the problem, but without correction time will increase by roughly 0.5 msec every 40 min. Consequently, the systems time must necessarily be corrected in order to measure one-way delays.

The standard deviation for the systems time in the *Chrony* server amounts to 0.584  $\mu$ sec, and in the newly developed utility 0.852  $\mu$ sec. Since a better result in time synchronization is obtained with the use of the *Chrony* server network time protocol, further studies will be performed with the user of this protocol.

**Basic stages of the method used in measurements of a one-way delay.** Let us now describe the basic stages in the method used in measurements of one-way network delays in IP networks. A design for implementation of this method is presented in Fig. 2. Using a *PPS* signal, the *Chrony* server synchronizes the systems time of the Linux operating system at two nodes of NetTestBox, at one of which the client application *owping* is launched, and at the other the server application *owpingd*. From Fig. 2, it follows that the determination of the length of a one-way delay comprises several stages between

TABLE 1. Data on Minimum Delay in a Local Network

Dimension of packet, byte	Type of packet	Minimum delay, $\mu\text{sec}$	
		Client-server	Server-client
64	ICMP	175	151
64	UDP	175	151
1518	ICMP	387	365
1518	UDP	390	364

measurements of the systems time at the two nodes. A delay in the transit of the packet from the client to the server consists of the acquisition of the systems time, procedures involved in packet transmission, and transfer of the packet through the network and its acquisition at the server.

The creation of the packet at the client node and computation of the check sum of the transmitted ICMP packet occur in the program *owping* until establishment of the systems time, that is, it does not have any effect on the delay of the packet. On the server side, the acquired packet is processed and its membership to the *owping* program is verified along with other service parameters. Immediately after the last bit is received, the *owpingd* program establishes the time at the server. Next, a response packet is then created and this program stores the established time. This stage does not affect the measured delay. On the next stage, the packet transmission time is established and its value is written to the packet. In order to transmit the ICMP packet, the *owpingd* program computes the final check sum. To transmit the UDP packet, the kernel of the operating system computes the check sum. The packet is then sent, after which it is transferred through the network from the server back to the client.

The client node awaits the arrival of the response packet from the server. The *owping* program stores the arrival time of the last bit. Copying of the packet from the kernel region into the region of the *owping* program, processing of the packet, computation and processing of the results, and representation of the results does not add to the delay and has no effect whatsoever on the delay. In this case, a wait procedure must be used, since any possible loss of the packet must be tracked. If the packet does not return in a course of a second (by default), it is considered lost.

One of the principal steps in the analysis of data on delays is to segregate its different components and, mainly, the physical component  $D_{ph}$  and the telecommunications  $D_{tel}$  component. Moreover, the construction of the telecommunications component is quite complex and its analysis is of no lesser interest for gaining an understanding of the nature of delay. An experiment in which two NetTestBox devices were located in the same local network was performed in order to minimize the contribution of the physical component. During the experiment, the Fast Ethernet ports were connected directly by means of a cross-overcable without any intermediate network devices. The results of the measurements are presented in Table 1.

In this case, the physical delay is negligible, consequently, three basic components remain:

$$D = D_l + D_t + D_p,$$

where  $D_l$  is latency.

By latency we will understand the delay prior to the start of transmission of the data, i.e., the time required to establish a connection. For Fast Ethernet, it may be estimated as  $D_l = 50\text{--}80 \mu\text{sec}$ , depending on the conditions under which the tests are conducted and the type of equipment [15].

A delay in transmission is identical to the time it takes to load the packet into the network interface. For a small packet, this time is negligibly low. The effective band of a Fast Ethernet network varies from  $B_1 = 6 \cdot 10^7$  bits/sec for packets of dimension  $W_1 = 64$  bytes to  $B_2 = 9 \cdot 10^7$  bits/sec for packets of maximally permissible dimension  $W_2 = 1518$  bytes, which amounts to  $D_t = W_2/B_2 \approx 135 \mu\text{sec}$ , where  $W_2$  is the dimension of a packet expressed in bits.

The time  $D_r = 100\text{--}180 \mu\text{sec}$  is consumed in processing the signal in transmission and reception of the packet. With an increase in the distance between the nodes and the appearance of intermediate network devices, the physical delay  $D_{ph}$  and the delay in the queue  $D_q$  become substantial.

The next feature of the measurement results that have to be discussed is the fact that the delay along the client–server segment is always greater than in the opposite direction. This is valid not only for an experiment on a local network, but also for any pair of measurement nodes that are active in an experiment.

Processes involved in transmission of packets from a client origin to a server application and back are entirely identical, except for the elimination of the last stage. The procedure involved in acquisition of a packet by server and client applications differ with respect to transmission of data from the core of the operating system in the *owping* program. The length of this process amounts to 20–25  $\mu\text{sec}$  (cf. Table 1). Consequently, a delay along the packet’s return path must be shorter by this length of time.

**Conclusion.** The present experiment has demonstrated that the *Chrony NTP* server assures maximum precision of 0.6  $\mu\text{sec}$ , provided that synchronization is performed by means of *PPS* signals generated by a receiver of the global positioning system. This degree of precision is sufficient for measurement of one-way network delays in real-world IP networks. The use of highly stable clocking sources and high-frequency hardware timers are necessary to achieve further increases in the precision of time synchronization, and this entails additional financial costs and falls outside the scope of the present article.

By means of an experiment with a direct connection of the nodes, it became possible to reduce the physical component of the delay  $D_{ph}$  down to zero and demonstrated that measurements of one-way delays take at least 150  $\mu\text{sec}$ . This length of time is determined by the limited total capacity of a network channel in the Fast Ethernet technology, the latency of the network connection, and the procedures involved in processing the packets as they are transmitted and acquired by the measurement nodes.

The present study was carried out with the support of the Ministry of Education and Science of Russia within the scope of the project part of a State Assignment for 2017–2019 (2.974.2017/PCh) along with a grant from the Russian Foundation for Basic Research (No. 16-07-00218a).

## REFERENCES

1. J. Mahdavi and V. Paxson, “IPPM metrics for measuring connectivity, RFC 2678,” [www.rfc-editor.org/info/rfc2678](http://www.rfc-editor.org/info/rfc2678), acc. 06.20.2016, DOI 10.17487/RFC2678.
2. M. Jain and C. Dovrolis, “End-to-end available bandwidth: Measurement methodology dynamics and relation with TCP throughput,” *Proc. ACM SIGCOMM Internet Measur. Workshop*, **32**, No. 4, 295–308 (2002).
3. P. Calyam, M. Sridharan, W. Mandrawa, and P. Schopis, “Performance measurement and analysis of H.323 traffic,” in: *Proc. 5th Int. Workshop on Passive and Active Network Measurement, PAM 2004* (2004), pp. 137–146.
4. G. Almes, S. Kalidindi, M. Zekauskas, and A. Morton, “STD 81, RFC 7679: A one-way delay metric for IPPM,” [www.rfc-editor.org/info/ref7679](http://www.rfc-editor.org/info/ref7679), acc. 06.0.2016, DOI 10.17487/RFC7679.
5. Rec ITU-T Y.1541, *Network Performance Objectives for IP-Based Services*, ITU Standard (2003).
6. I. S. Kargin, “Analysis of random-access wireless multi-step network with low packet delivery time,” *Information Technologies and System 2015: Proc. 39th Interdiscipl. School-Conf. of Kharkevich IPPI RAN*, Sochi (2015), pp. 659–670.
7. A. E. Kucheryavyi, M. A. Makolkina, and R. V. Kirichek, “Tactile Internet. Communications network with ultra-low delays,” *Elektrosvyaz*, No. 1, 44–46 (2016).
8. N. I. Vinogradov and E. S. Sagatov, “A technique for the measurement of one-way network delays with GLONASS synchronization,” *Telekommunikatsii*, No. 4, 16–23 (2016).
9. C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, et al., “Analysis of end-to-end delay measurements in Internet,” in: *Proc. 3rd Int. Workshop on Passive and Active Network Measurement, PAM 2002* (2002), pp. 1–8.
10. S. Shalunov, B. Teitelbaum, A. Karp, et al., “RFC 4656: A one-way active measurement protocol (owamp),” [www.rfc-editor.org/info/ref4656](http://www.rfc-editor.org/info/ref4656), acc. 06.20.2016, DOI 10.17487/RFC4656.
11. S. Donnelly, I. Graham, and R. Wilhelm, “Passive calibration of an active measurement system,” in: *Proc. 2nd Int. Workshop on Passive and Active Network Measurement, PAM 2001* (2001), pp. 1–8.

12. H. Veiga, R. Valadas, P. Salvador, et al., "OWAMP performance and interoperability tests," in: *4th Int. Workshop on Internet Performance, Simulation, Monitoring and Measurement, IPS-MoMe 2006*, Salzburg, Austria (2006).
13. L. Carbone, F. Coccetti, P. Dini, et al., "The spectrum of Internet performance," in: *Proc. 4th Int. Workshop on Passive and Active Network Measurement, PAM 2003* (2003), pp. 75–88.
14. F. Mora, "Bring an atomic clock to your home with chrony," *Linux J.*, No. 101, 3 (2002).
15. D. A. Kuz'min, "A cluster approach for high-performance computations," *Komp. Modelir. Nove Tekhnol.*, No. 7, 7–15 (2003).