

Estimation of Light Speed using PC/Windows Networked Computers*

R. C. WOODS

Department of Electrical and Computer Engineering and Microelectronics Research Center, 2128 Coover Hall, Iowa State University, Ames, Iowa 50011, USA. E-mail: cwoods@iastate.edu

A recent proposal for a classroom experiment to estimate the speed of light using a networked computer relies upon some special hardware and Linux machines able to measure the time-of-flight to the nearest microsecond. At first sight, ordinary PCs running Windows appear not to be capable of this application because their software time-of-flight measurement is limited to the nearest millisecond. This paper shows how an ordinary networked PC/Windows machine may be used for an equivalent time-of-flight rough estimation of the speed of light having literally zero cost if one networked computer is already available. This allows remote students to estimate the speed of light with only consumer-grade computers and no specialized hardware available. Feedback from students has been highly favorable.

INTRODUCTION

LEPAK AND CRESCIMANNO [1] recently described the use of the PING command in Linux to measure the speed of signals (c) in a cable using the time-of-flight for data packets between two computers linked by a coaxial local network. At first sight it appears difficult to achieve the same with computers running Windows since the corresponding PING command returns response times to the nearest millisecond rather than the nearest microsecond as given by Linux. The cable network used by Lepak and Crescimanno [1] introduced a delay of a few microseconds only and so it would be impossible to obtain meaningful results using the Windows PING command available at the DOS prompt.

While the technique of Lepak and Crescimanno [1] is well-suited to obtaining a reasonably accurate value for c using computers running Linux and some extra network hardware, as an initial class exercise in an optoelectronics course it was desirable for each student to be able to obtain worthwhile results with as little outlay as possible. In addition, some students in remote locations did not have access either to special hardware or to two computers running Linux. Most remote students have just one computer running Windows. Accordingly, the problem of estimating c using the Windows PING command was addressed.

METHOD

Rather than the ingenious method of Lepak and Crescimanno [1] in exploiting the noise in the

response-times, a straightforward time-of-flight measurement was sought.

Clearly, in order to use the time-of-flight method with response times of many milliseconds it is necessary to use much longer round-trips. Typically, nodes separated by many thousands of miles can be used. However, although PING reports the overall response time for the data acknowledgement, this time includes not only the time-of-flight of the signals but also the software processing time at each of many intermediate nodes along the route between the local and remote computers. Typically 16 or more nodes may be involved in a PING to a remote part of the globe. This introduces the problem of allowing for the added response time at each node. While it is easy to trace the route used in any particular PING and to ascertain how many node jumps were made, it is not so easy to find the response time at each node. Another potential problem involves knowing what medium is used for each leg of the journey; in principle, signals can be transmitted over successive legs involving twisted pairs, coaxial cables, fiber-optics, satellite links, or any combination of these.

If it is accepted that this method clearly cannot compete with others in terms of precision and accuracy, there remain techniques available to optimize the treatment of results obtained using Windows PING. Firstly, the velocity of signals in twisted pairs, coaxial cables, and fiber-optics will all be roughly the same at around 2×10^8 m/s. The value measured will be dominated by the largest hop. The velocity of signals through a satellite link will clearly be around 3×10^8 m/s but over a much longer distance to and from a geostationary satellite at altitude $\sim 35,786$ km. Therefore, even if not identified as such by the software, hops involving a satellite link may be readily identified.

* Accepted 11 February 2005.

To compensate for the software processing time at each node, the following method may be used. Each PING is modeled as taking the time-of-flight for the data packets plus a time proportional to the number of nodes used in the PING. The number of nodes used in each direction (n) can be found from the DOS command TRACERT used immediately after PING, so that the total number of nodes used is $2n-1$. In addition, the different time taken for processing at the source and target node (as opposed to those nodes merely passing the data packet along the route) may be modeled by a notional extra effective number of nodes N . So, for each measurement, the time reported by PING is:

$$t = (s/c) + (2n - 1 + N)\tau, \quad (1)$$

where s is the round-trip distance travelled, and τ is the average software processing time at each node. (1) may be written as:

$$s/(2n - 1 + N) = ct/(2n - 1 + N) - c\tau, \quad (2)$$

so that a graph of $s/(2n - 1 + N)$ vs. $t/(2n - 1 + N)$ for all results should ideally yield a straight line of gradient c and y -axis intercept $-c\tau$. The effective number of extra nodes N is unknown, but the correct value is that yielding the largest value for c . By choosing results corresponding to large numbers of nodes, both the effect of N and also mild variations in the response times between different nodes may be minimized. In practice, it turns out (see below) that the final estimated value of c is extremely insensitive to the value of N used, and that errors from other causes are much more significant.

The round-trip distance s strictly depends upon the route and is not necessarily the global great-circle overall distance. It is certainly simplest to use the global great-circle distance between source and destination, but this leads to large error bars in the data points. More enterprising students find the distance actually traversed by the data using the TRACERT listing; they are often surprised at how circuitous is the route actually taken. Good experimental technique requires that the largest possible range of times and distances should be used.

Best results are obtained by PINGing a chosen site multiple times and recording the lowest response time. There may also be some outlying results corresponding to the software processing time at one or more nodes being occasionally considerably greater than the average value. These outlying results will all fall below the ideal straight line given by Equation (2). The best value of c is clearly obtained using the greatest value of the gradient ignoring outlying points well below the best 'reasonable' main straight line.

RESULTS

This exercise has been set to two cohorts of students one year apart. For use as a class exercise

in optoelectronics, students were set the problem in the following form.

- Estimate the speed of light using a computer as follows. The DOS function PING on a *networked* PC sends out an interrogation message to a specified internet service provider node, which then replies to your own machine. Your machine then tells you how long this took. (To find out how to use PING in detail, type PING at a DOS prompt which will show you the required syntax and how to specify what location the computer is requested to PING. You may use instead any equivalent function on a non-PC/Windows networked machine.)
- Estimate the speed of data signals from the output given by PING and your knowledge of the distance the signals have traveled. Your submitted report should include printout(s) of all the output(s) from PING that you used, an explanation of how you calculated your results and what you did to ensure the best accuracy, together with your detailed calculations.
- What medium(s) do you think your value(s) correspond(s) to, and why? Hint: as this is a 'project' type exercise, albeit quite short, I have deliberately not told you all the details that you will need to discover for yourself to make this experiment as accurate as possible. I do NOT think this method will give a highly accurate result, but it is certainly possible to get within an order of magnitude.
- How accurate do you think your result is?

This was the first weekly exercise set in a senior and graduate course on optoelectronic devices and applications, after students had covered electromagnetic theory and had an appreciation of the propagation speed of electromagnetic waves through various dielectric media, but before any other wave phenomena had been covered.

Most students initially tried PINGing local machines until they realized that the distances involved were then too short to be useful. Student responses at that stage ranged from 'This exercise is impossible', through 'How do I know the location of a node?' (answer: always PING university sites) to trials of longer and longer hops. Most students soon realized that they had to use distances approaching the longest available globally, and the best were able to work out how to account for the node software delays. The exercise set above also requires an appreciation of error analysis and how to estimate the likely errors in a measurement that is certainly much less accurate than most experiments that the students will have attempted up to that time. Another problem encountered very recently is that many Internet sites refuse to acknowledge PING requests for security reasons. The sites that can be used in the exercise are limited to those that respond to PING. A web search will generally find quite quickly at least one site in a university city that will respond.

Table 1. Typical results when PINGing from Ames, Iowa

Site	Time (ms)	Number of nodes	Route	Round-trip distance (km)
Bath, UK	128	20	Kansas City-Indianapolis-London-Bristol	15 425
Berkeley, CA	63	15	Kansas City-Sunnyvale-LA	7544
New Delhi, India	366	19	San Francisco	31 644
Charlottesville, VA	50	15	Kansas City-Indianapolis-Chicago-NYC-DC	5944
Adelaide, Australia	212	16	Kansas City-Denver-Sydney-Melbourne	32 018
Sydney, Australia	193	14	Kansas City-Denver	29 234
Tokyo, Japan	163	22	Kansas City-Dallas-Bay area	23 594
Wien, Austria	143	14	Kansas City-Indianapolis-Chicago-NL-DE	17 914
Moscow, Russia	188	17	Kansas City-Indianapolis-Chicago-NL-UK-SE	21 972
Kiev, Ukraine	163	20	Kansas City-Indianapolis-Chicago-NL-DE-Wien	20 022
Minneapolis, MN	40	10	Kansas City	1986
Reykjavik, Iceland	198	19	UK-DK	21 142
Rio de Janeiro, Brazil	359	19	UK-FR	33 866
Guadalajara, Mexico	72	14	Houston-Mexico city	5958
Athens, Greece	176	16	DE	20 466

Some typical experimental results obtained by the present author are summarized in Table 1, and the corresponding graph is shown in Fig. 1. The two data points at the largest value of $t/(2n-1+N)$ are clearly outliers and correspond to PINGing Rio de Janeiro and New Delhi. These routes involve a satellite hop giving data incompatible with the other optical and electrical links. The route distance given in Table 1 is the sum of the hops on the earth's surface but the actual total traversed distance includes four hops between the earth and a satellite. To calculate c in cables and fiber-optic links, these two points were therefore discarded and the gradient calculated from the rest of the data. Using $N=0$ gives the value $c=1.57 \times 10^8$ m/s. Values of N as high as 100 reduce the computed value of c only to 1.49×10^8 m/s. The largest value of c is obtained using $N=-9.8$ as used in Fig. 1, giving

$c=1.60 \times 10^8$ m/s, so it was justifiable to assume that the estimated value of c is extremely insensitive to the value of N . This corresponds to quite a high value of refractive index for optical fibers, $\mu \approx 1.8$. (Simple evaluation of s/t for each data point, not allowing for node delays, gives an estimated average velocity of 1.46×10^8 m/s, so that ignoring the node delay clearly does not introduce enormous errors.) However, given the scatter in the data, the standard error will also be quite large, of the order of $\pm 15\%$. Remaining errors in the values of c and μ estimated are probably caused by the links between known nodes not necessarily being great circle routes, and also by the variation in the delays amongst individual nodes. From the y -axis intercept may be obtained the typical node delay $\tau=1.5$ ms, though this value is extremely sensitive to the value of N used and so is unreliable.

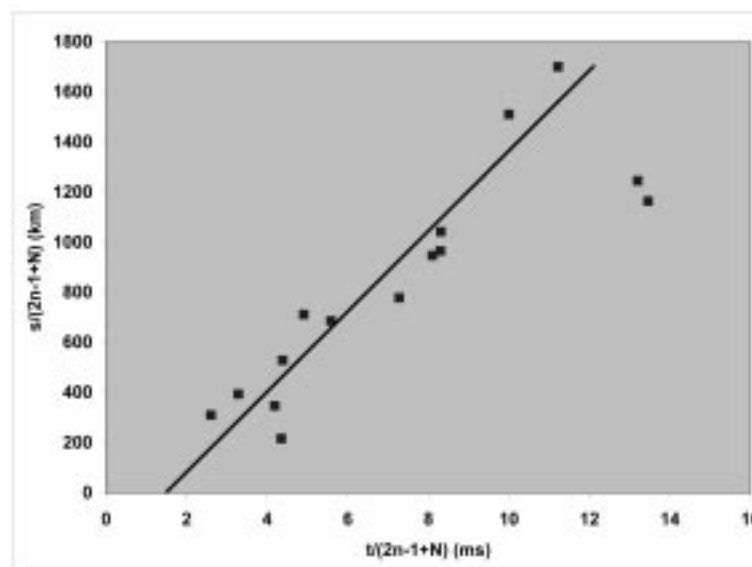


Fig. 1. Graph of $s/(2n-1+N)$ vs. $t/(2n-1+N)$ for all results shown in Table 1 (■), using $N=-9.8$. Line is best-fit to all results except the two outliers at the largest values of $t/(2n-1+N)$.

There was widespread appreciation amongst the students that they had been set a problem with a well-defined goal but without the straightjacketing of a conventional assigned problem from a text. They were also impressed by the fact that they could undertake in their own homes and with consumer-grade equipment an experiment that was quite challenging in most professional laboratories until just a few years ago. Many became immersed in the problem of how to account for the node delays and accordingly interest in the course was greatly increased.

CONCLUSIONS

A method of roughly estimating the speed of light is proposed requiring only a single networked computer. This method is so simple that, having zero cost assuming existing access to one PC/Windows computer, it can be used in almost any educational environment, and with varying degrees of sophistication regarding the allowance for node delays and error estimation. Students are forced to realize the value of estimation of errors and also that some experiments are valuable even though they do not yield very precise results.

REFERENCE

1. J. Lepak and M. Crescimanno, Speed of light measurement using PING, *Cornell arXiv:physics/0201053*, 2002 rev. 2004, p. 1.

R. C. Woods was born in Leicester, UK. He received the BA and MA degrees in Physics from New College, Oxford University, Oxford, UK, and worked in the Clarendon Laboratory, Oxford University, on magnetic resonance in rare earth metal alloys, for which he received the D.Phil. degree in 1980. Next followed a period as a Research Assistant in the Engineering Science Department, Oxford University, where he worked on surface-acoustic wave devices. From 1982 to 1983, he was Senior Scientist at Plessey Research (Caswell) Ltd., Towcester, UK, specializing in semiconductor lasers and LED systems. He was then a Lecturer (and Senior Lecturer since 1993) in Electronic and Electrical Engineering at the University of Sheffield, Sheffield, UK, where he was in charge of courses on Digital Systems and Semiconductor Devices for several years. He was appointed Professor of Electrical and Computer Engineering at Iowa State University, Ames, Iowa, in 2001, where his teaching responsibilities include a senior and graduate level course on optoelectronic devices and applications. He has research interests in novel solid-state devices, device modeling, and magnetic recording, and has authored more than 75 archival research publications in these fields. Prof. Woods is an invited member of the international awarding committee for Marie Curie Fellowships funded by the European Commission, and of several expert panels at the National Science Foundation, Washington DC. Prof. Woods is a member of ASEE.