Session ENG 206-117

Virtual Instrument and Object Oriented Programming Modules to Enhance Computer Networking Course

Nikunja Swain, Ph.D., PE
South Carolina State University
swain@scsu.edu

Raghu Korrapati, Ph.D.
Walden University
rkorrapa@waldenu.edu

Abstract

Today's test environment is more challenging than ever as pressure on increasing quality and meeting time-to-market continue to increase. Because of these recent advances in computer and other technologies, it is becoming evident that a sound foundation in computer networking is critical for success in many kinds of computer-based work and universities must produce graduates with solid foundation in computers and their applications to solve real world problems.

One way to achieve this is through interactive learning and teaching through the use of software packages like LabVIEW (Virtual Instruments) and Java/Visual Basic (Object Oriented Programming)**.** This will also allow the student to have greater interaction with the subject matter and improve his/her skills in the use of number of applied engineering software packages.

This paper will discuss design and development of interactive instructional modules for Number Systems, Internet Protocol Addresses, Address Classes, Subnet Masks, and Subnetting using LabVIEW and Visual Basic/Java.

I. Introduction

The students' over reliance upon formulas and routine use of technique in problem solving too often lead to poor performance in advanced courses and a high attrition rate in the engineering, technology, and science programs. The students' lack of comprehension of mathematical concepts results in time wastage during laboratory experiments, misinterpretations of lab data and underachievement in standardized science and engineering tests that stress the fundamentals. This problem can be effectively addressed by improving the student's conceptual understanding and comprehension of the topics covered in introductory science and technology courses. One way to achieve this is through interactive learning and teaching and upgrading the

existing laboratories with modern equipment. This will require increased funding and resources. But in recent years there is a decrease in resource allocation making it increasingly difficult to modernize the laboratories to provide adequate levels of laboratory and course work and universities are under pressure to look for alternative cost effective methods.
One way to achieve this is through interactive learning and teaching through the use of software packages like LabVIEW (Virtual Instruments) and Java/Visual Basic (Object Oriented Programming).

LabVIEW and Java/Visual Basic are currently used in a number of engineering schools and industries for simulation and analysis. By introducing virtual instrumentation (LabVIEW) and object oriented programming (Java/Visual Basic) to the existing laboratory facilities and course(s) the students can be well trained with the latest design techniques and computer aided instrumentation, design and process control used throughout industry. This will also allow the students greater interaction with the subject matter and improve his/her skills in the use of number of applied engineering software packages.

LabVIEW is based on graphical programming and easy to use. It is an interactive problem-solving environment, where the students can analyze, visualize, and document real-world science and engineering problems. A number of books [1, 2, 3, 4] are available covering the use and application of LabVIEW. LabVIEW programs are called Virtual Instruments, or VIs for short, LabVIEW is different from text-based programming languages(such as Fortran, C++, Java and Visual Basic) in that LabVIEW uses a graphical programming languages, known as the G programming language, to create programs relying on graphics symbols to describe programming actions. LabVIEW also provides an extensive library of virtual instruments and functions to help in programming. It also contains application specific libraries for data acquisition, GPIB and serial instrument control, data analysis, and file input/output with conventional program debugging tools that can be used to set breakpoints, single-step through the program, and animate the execution so that the flow of data can be observed.

At South Carolina State University (SCSU), the PC and VI based system concept is used to design and develop a laboratory called Computer Based Virtual Engineering Laboratory (CBVEL). The CBVEL consists of IBM compatible computers with appropriate software and hardware from National Instruments (NI), Simulation software such as C++, Visual Basic, Java, PSPICE, etc., and is connected to College of Science, Mathematics, Engineering and Technology (CSMET) network and existing equipment. Virtual Instrument (VI) and Object Oriented Programming modules for different courses and research areas are currently developed and used to teach various courses. Examples of some of these VIs are Circuit Analysis, Electronics, Communications, Digital Signal processing, Networking, and Digital Filters [5, 6, 7]. This paper discusses design and development of some of the modules such as Internet Protocol Addresses, Address Classes, Subnet Masks, and Sub-netting.

This paper is arranged as follows: section II presents theoretical discussions on Internet Protocol (IP) Addressing, IP Classes, Subnet Mask and Subnets, section III presents the LabVIEW VIs and Visual Basic instructional modules, section IV presents the conclusion and section V presents the references.

II. Internet Protocol (IP) [8, 9, 10]

The Internet Protocol (IP) mainly specifies that every computer or device (called a host) on the network must have its own unique address. This address is otherwise called as the “IP Address”. Currently the IP version 4 (IPv4) is used for networking, although a newer version of IP (IPv6) currently being tested and implemented in few organizations. This newer version is designed to enhance the capabilities of the existing IPv4. This paper will focus on IPv4.

The IP address is a 32-bit integer. Usually this number is represented as a group of four 8-bit numbers separated by periods. This representation is known as “Dotted Decimal Notation”. Each IP address is divided into a prefix and a suffix as shown in Figure 1. This is known as two part IP addressing. The Prefix identifies network to which computer is attached and the Suffix identifies computer within that network. This type of addressing format makes routing efficient.

Network	Host

Figure1 – Two part IP Addressing

Network and host numbers

Every network in a TCP/IP internet is assigned a unique *network number* and each host on a specific network is assigned a *host number* or *host address* that is unique *within that network.* Host's IP address is the combination of the network number (prefix) and host address (suffix).

Properties of IP addresses

Network numbers are unique. Host addresses may be reused on different networks; combination of network number prefix and host address suffix will be unique. Assignment of network numbers must be coordinated globally; assignment of host addresses can be managed locally.

Designing the format of IP addresses

As mentioned earlier, IP designers chose 32-bit addresses. They allocate some bits for prefix, some for suffix. The allocation of large prefix and small suffix results in many networks, few hosts per network and the allocation of small prefix and large suffix results in few networks, many hosts per network. The IP designers need to allow for both large and small networks because of variety of technologies.

Classes of addresses

Designers use multiple address formats that allow both large and small prefixes. Each format is called an address class and the class of an address is identified by first four bits of the IP address as shown in Figure 2 and Table 1.

<table>
<tr><td>Bits</td><td>0</td><td>1</td><td>2</td><td>3</td><td>7</td><td>8</td><td>16</td><td>24 31</td></tr>
<tr><td>Class A</td><td>0</td><td colspan="4">Prefix</td><td colspan="3">Suffix</td></tr>
<tr><td>Class B</td><td>1</td><td>0</td><td colspan="4">Prefix</td><td colspan="2">Suffix</td></tr>
<tr><td>Class C</td><td>1</td><td>1</td><td>0</td><td colspan="4">Prefix</td><td>Suffix</td></tr>
<tr><td>Class D</td><td>1</td><td>1</td><td>1</td><td>0</td><td colspan="4">Multicast Address</td></tr>
<tr><td>Class E</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="4">Reserved for Future Use</td></tr>
</table>

Figure 2 – IP Class from Binary bit pattern

Table 1 – Range of values and IP Classes

Class	Range of Values
A	0 – 127
B	128 – 191
C	192 – 223
D	224 – 239
E	240 – 255

Using IP address classes

Class A, B and C are primary classes. They are used for ordinary host addressing. Class D is used for multicast, a limited form of broadcast in which Internet hosts join a multicast group. The Packets are delivered to all members of group. The Routers manage delivery of single packet from source to all members of multicast group. This is also used for *mbone* (multicast backbone). Class E is reserved for future use.

Networks and hosts in each class

The number of networks and hosts per network are summarized in Table 2.

Table 2 – Summary of IP Classes

Network Class	Network bits	Number of Networks	Host bits	Number of hosts/Network
A	8	$(2^8 – 2)$	24	$(2^{24} – 2)$
B	16	$(2^{16} – 2)$	16	$(2^{16} – 2)$
C	24	$(2^{24} – 2)$	8	$(2^8 – 2)$
D	Multicasting			
E	Reserved for future use			

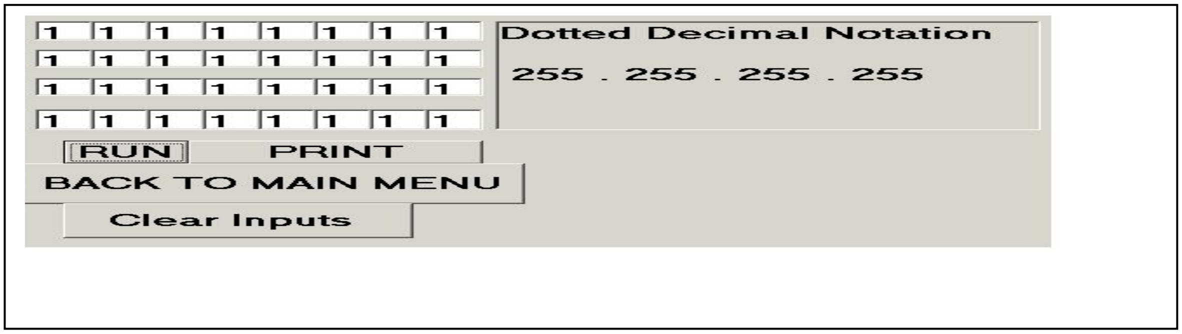
Subnets and Subnet Masks

The two-phase assignment of Internet addresses was created for administrative convenience. Although central Internet registrars were willing to keep track of several thousand networks, they were not willing to take on the task of assigning addresses to billions of individual host computers. Organizations with Class A and Class B networks had a similar problem. They faced the need to assign individual Internet addresses to thousands or millions of individual hosts within the organization. Again, this would have been a network nightmare. As a result, organizations wanted a three-phase process for assigning Internet address to an individual host. First, a central authority would assign a block of addresses to an individual organization. This would be done in the traditional way by assigning a network part. However, the organization would give smaller blocks of address to suborganizations. It would then make each small organization responsible for assigning the Internet addresses within its blocks. This would require a three-part address. The network part would remain the same. However, the local part would be subdivided into a subnet part and a host part. The division of the local part into subnet part and host part is achieved through the use of another 32 bit number called subnet mask. The Internet Registrar provides each network a second number to supplement Internet addresses. This is the subnet mask. It begins with a block of ones and ends with a block of zeros.

III. Examples of few Visual Basic LabVIEW VI Modules

A. Visual Basic Module for Dotted Decimal and IP Classes without Subnet

This is a menu driven module with menu options such as *About, Dotted Decimal Notation, IP Classes without Subnet*, *Print*, and *Quit* menus. The *About* menu provides a brief description about the program. The *Dotted Decimal Notation* menu converts the binary IP address to equivalent Dotted Decimal Notation. The user provides the inputs (0 or 1) and clicks on the RUN button. The display is the equivalent Dotted Decimal Notation. Execution of the *IP Class without Subnet* menu provides the user with IP class, network and host bits, number of networks and number of hosts/network information. Execution of the *Print* menu prints the inputs and result and execution of the *Quit* menu terminates program execution. Figure 2 shows Dotted Decimal Notation and Figure 3 indicates network information with a subnet.

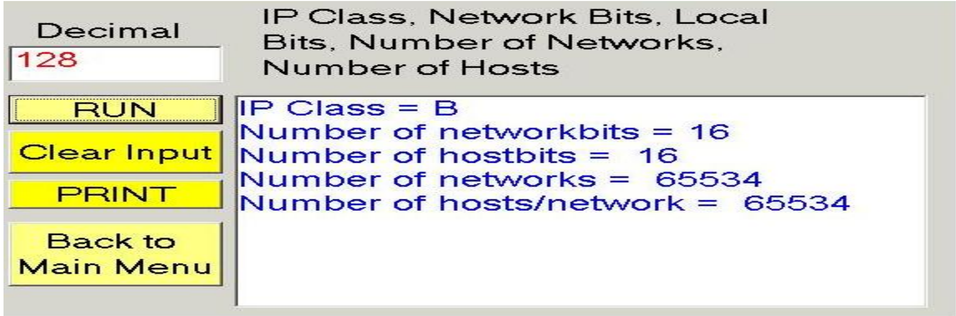


```
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

Dotted Decimal Notation

255 . 255 . 255 . 255

Figure 2. Dotted Decimal Notation



Decimal
128

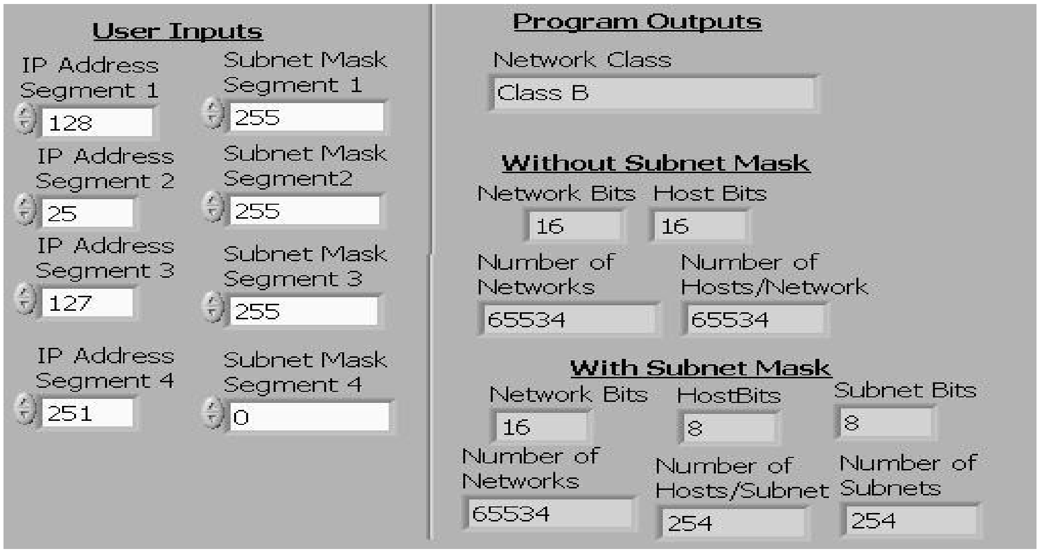
IP Class, Network Bits, Local Bits, Number of Networks, Number of Hosts

```
IP Class = B
Number of networkbits = 16
Number of hostbits =  16
Number of networks =  65534
Number of hosts/network =  65534
```

Fugure 3. IP Class, Network and Host Information without Subnet

B. LabVIEW VI for IP Class, Network and Host Information without and with Subnet

Figure 4 describes a LabVIEW panel for IP Class, Network, and Host information in two cases: with subnet and without a subnet



User Inputs

IP Address Segment 1
128

Subnet Mask Segment 1
255

IP Address Segment 2
25

Subnet Mask Segment2
255

IP Address Segment 3
127

Subnet Mask Segment 3
255

IP Address Segment 4
251

Subnet Mask Segment 4
0

Program Outputs

Network Class
Class B

Without Subnet Mask

Network Bits	Host Bits
16	16

Number of Networks	Number of Hosts/Network
65534	65534

With Subnet Mask

Network Bits	HostBits	Subnet Bits
16	8	8

Number of Networks	Number of Hosts/Subnet	Number of Subnets
65534	254	254

Figure 4. Front Panel

IV. Conclusion/Discussion

The LabVIEW VI and Visual Basic modules presented above are user friendly and performed satisfactorily under various input conditions. It helped the students to understand items relating to IP Addressing in more detail. It also introduced the students to program development using Visual Basic and Virtual Instruments. We believe that this introduction to Visual Basic and

Virtual Instrument will help the students to design modules for other courses. These modules can be used in conjunction with other teaching aids to enhance student learning of Networking and the Internet as follws:

There are number of books and research papers written on TCP/IP. Also, there are number of command line tools available to check the effective use of the Internet. Students can use these books and research papers to have theoretical understanding of TCP/IP and use the command line tools to have practical and hands-on experience. Unfortunately the command line tools and many books don't provide the students with the underlying algorithm and codes and therefore the students are deprived of developing his/her own program(s) to study and experiment with Networking and the Internet. We believe modules like these will assist the instructors/students to design and develop other networking modules and hence reinforce their theoretical knowledge.

V. References

[1]. Robert H. Bishop, Learning with LabVIEW, Addison Wesley, CA 94025, 1999.

[2]. Lisa Wells and Jeferey Travis, LabVIEW for Everyone, Graphical Programming Even Made Easier, Prentice Hall, NJ 07458, 1997.

[3]. M. Chugani, A. Samant, and N. Cerna, LabVIEW Signal Processing, Prentice Hall, NJ 07458, 1998.

[4]. Leonard Sokoloff, Basic Concepts of LabVIEW 4, Prentice Hall, NJ 07458, 1998.

[5]. Tobin, Gwendolyn & Swain, N. K., "Object Oriented Programming (Visual Basic) and Graphical Programming (LabVIEW) in study of Digital Arithmetic" – Proceedings of the NASA/MUSPIN 8th Annual Conference, NM, October 1998.

[6]. Swain, N. K., Korrapati, R.B. & Anderson, James, "Application of Graphical Programming, Object Oriented Programming, and Virtual Instruments in Education" – Proceedings of Conference on Information Technology (CIT 99), India, October, 1999.

[7]. Swain. N. K, Korrapati, R.B. Anderson, James, Singh Ajit, & Swain. M, "Internet Based Virtual Instrumentation" – Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications, November 2000, Las Vegas, Nevada, USA, pp. 296-301.

[8]. Comer, Douglas E., Computer Networks and Internets, with Internet Applications, Prentice Hall, NJ 07458, 2001.

[9]. Comer, Douglas E., Computer Networks and Internets, with Internet Applications, Prentice Hall, NJ 07458, 2001.

[10]. Panko, Raymond R., Business Data Networks and Telecommunications, Prentice Hall, NJ 07458, 2002.

Biography

Nikunja Swain is currently a Professor at the South Carolina State University. Dr. Swain has 25+ years of experience as as an engineer and educator. He has close to 50 publications in journals and conference proceedings and he has procured research and development grants from the NSF, NASA, DOT, DOD, and DOE. He has reviewed books on TCP/IP, Microprocessors, and Information Systems and a reviewer for ACM Computing Reviewes, IJAMT, CIT, ASEE, and other conferences and journals. Dr. Swain is also an adjunct faculty for Walden University, and Webster University. He is a registered professional engineer in the state of South Carolina.

Raghu Korrapati, Ph.D is a faculty in Applied Management and Decision Sciences Program in School of Management at Walden University, Minneapolis, MN, USA. His email address: rkorrapa@waldenu.edu.