

Illustrating Networking Concepts with Wireless Handheld Devices

Brad Richards and Nate Waisbrot
Computer Science Department
Vassar College
Poughkeepsie, NY 12604

{richards|nawaisbr}@cs.vassar.edu

Abstract

We are developing innovative new laboratory materials that give students a hands-on introduction to both traditional and wireless networking concepts. Our labs are designed around the Cybiko, a powerful multi-functional wireless handheld computer that sells for less than \$50. The materials can be used to augment existing networking courses or as the core of a new upper-level course. This paper describes the labs completed to date, our plans for the future, and presents the results of a comparative study using the labs to demonstrate routing algorithms.

1. Introduction

The tremendous growth of telecommunications and the Internet makes it increasingly important that undergraduate students be exposed to fundamental networking concepts and technologies. Yet little material for supporting networking courses is publicly available, as a search of the Computer Science Teaching Center [4] and Computer Science Education Links [8] sites reveals. Even less material exists to support wireless networking courses and topics. (The authors have been unable to find *any* such teaching materials.)

We strongly believe, as do many others [12], that students learn most effectively from active involvement in experience-based activities. Towards that end, we are developing lab exercises and materials that facilitate interactive, hands-on introduction to traditional and wireless networking concepts. These labs are designed around wireless handhelds, and can be used to supplement existing networking courses or as the core of a new upper-level course. Our efforts to date have focused on creating materials to support traditional networking topics, which are described in detail in Section 4 along with our plans for labs addressing wireless networking topics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ITiCSE'02, June 24-26, 2002, Aarhus, Denmark.
Copyright 2002 ACM 1-58113-499-1/02/0006...\$5.00.

Background

It is becoming common for larger schools, or those with more technical programs, to have purpose-built facilities to support networking labs [5,7,14,15]. Students gain first-hand experience building actual networks, setting up domain name servers and web servers, configuring routers, etc. The specific skills taught in these courses are often determined at least in part by consultation with industry [6,10,14]. However, many smaller schools either cannot afford such facilities or wish to focus on the fundamentals instead of the skills dictated by current technology.

A number of sophisticated network simulators exist and can be used to introduce basic concepts such as communications protocols, routing, bridging, error handling, etc [1,2,9]. But these powerful tools can have steep learning curves and, while excellent for doing in-depth evaluations or comparing algorithms quantitatively, often simulate a networking environment in far more detail than is necessary for a simple introduction of fundamental concepts.

We are producing materials that introduce the fundamentals in a simple, hands-on fashion, and at minimal expense. Our approach enables demonstrations to be performed in any classroom or lab space, and for students to arrange themselves such that their positions mimic a given topology (a larger benefit than one might at first imagine). Custom applications have been written to illustrate routing concepts in traditional wired networks, with plans for labs to demonstrate several topics in



Figure 1: Cybiko

wireless networking. This is in contrast to other work using handheld computers in education, which has focused primarily on enhancing the lecture environment instead of the lab [3,11].

2. Cybiko

The Cybiko is a powerful multi-functional wireless handheld computer designed for the teenage market. (See Figure 1.) Given its current list price of \$49.99¹, the feature list is impressive: a 32-bit processor, 512K of RAM (expandable to 1.5 MB), miniature QWERTY keyboard, and a 160×100 LCD display with four levels of gray. An expansion slot can accept memory cartridges or an MP3 player, with additional options being developed. More interestingly, the devices have built-in support for wireless communication with a range of up to 300 feet and throughput of 19.2Kbps. (A newer model has just been introduced with a faster processor, more memory, and an improved keyboard, for a list price of \$99.)

Cybikos have standard PDA-like functionality (*e.g.*, calendar, address book, notebook), but also provide a wireless text-based chat program, E-mail software, and a wide variety of games. New applications can be downloaded to a PC from the Cybiko web site², and then transferred to the Cybiko over a serial cable. (Currently, over 400 applications are available for free on the site.) Better still, for our purposes, Cybiko gives away a software development kit containing a C compiler and API documentation³.

The low price, wireless communication capabilities, and free development environment make this an extremely attractive platform on which to create teaching materials for illustrating networking concepts.

¹Sale prices have been seen as low as \$29.99.

²<http://www.cybiko.com>

³An enhanced SDK was recently released that includes a C++ compiler, but is no longer free.

2.1 Will Cybiko Survive?

The longevity of our teaching materials is independent of the Cybiko, as there are other platforms that could be adopted, but it is natural to wonder about the survival of the devices. Cybiko is a privately held company, formed in 1999 to market the Cybiko handheld, its only product. Given the positive press the Cybiko has received and the large and growing number of Cybiko-related web sites, there seems to be ample evidence of consumer enthusiasm.

The potential of the Cybiko has not escaped AOL, which is now a significant shareholder. The two companies are working together to add AOL features such as Instant Messaging to the devices, and to establish wireless Internet gateway services in shopping malls and other teen gathering spots.

Should the Cybiko fail, there will be a number of fallback platforms to which our materials could be ported. The Bluetooth wireless standard, while off to a slower start than anticipated, is making an appearance in consumer electronics and should continue to gain ground. Also, the 802.11 standard has become almost ubiquitous and is now showing up in less expensive, low-end devices. Wireless-enabled PDAs would make excellent platforms on which to implement wireless networking materials, though at somewhat greater cost.

3. Lab Details

The basic model for each of the lab demonstrations is the same: Individual students (or small groups) are each assigned a Cybiko and actively participate in the given lab activity. The session is supervised by a faculty member or teaching assistant who oversees the action from an “administrative” Cybiko and guides the students through a series of planned activities.

3.1 Traditional Networking

We have written applications illustrating flooding, distance-vector, and link-state routing algorithms. For all of these demonstrations, a virtual interconnection topology is created first on the administrative Cybiko. The administrator can alter this topology, modifying link weights and states as the lab progresses. Student Cybikos are assigned positions in the virtual topology automatically as they contact the administrative Cybiko, and are given the identities of their immediate neighbors. Routing applications run atop our Virtual Topology Layer (VTL), which hides the details of the Cybiko’s communications routines and makes it appear as though the devices are in a wired network. The VTL also makes calls into the application code when networking events occur. (*e.g.* message arrival, topology changes, etc.) This isolates our routing applications from the Cybiko OS,

which both simplifies the routing protocol code, and make it easier to port the code to another platform if it becomes necessary.

Introductory Lab

This first lab shows students how to download custom applications to their Cybikos and familiarizes them with the operating system. Students then pair up and use one of our custom applications to pass messages back and forth between Cybikos, displaying transfer and error rates. Students can investigate the error rate as a function of distance, and the effect of obstructions.

Flooding Lab

This lab demonstrates that flooding distributes data packets to all nodes in the network without knowledge of the entire topology, and that hop counts effectively limit this otherwise infinite process. Students take turns sending simulated data, which is initially flooded onto the virtual topology without hop counts or other optimizations. As a Cybiko receives and forwards data packets, it prints the identity of the neighbor who forwarded it and the intended destination, so students can see how data is moving. The unit also beeps so its owner and all lab participants realize it is forwarding data. It rapidly becomes obvious that flooding can continue indefinitely, even when packets are not forwarded back along the source link.

Data forwarding can be disabled on each Cybiko at the press of a key, so students are able to halt the infinite flooding. The administrative Cybiko can modify the virtual topology at any time to help illustrate its affect on the flooding. It can also instruct all student Cybikos to begin using hop counts. When in effect, students must specify the maximum count each time they send data. The administrator can guide the students in selecting values and show that unrealistically low thresholds prevent data from reaching all nodes, but that the process eventually stops even with large maximum hop counts.

Distance Vector Lab

In lecture, students are shown that local routing tables are sufficient to perform system-wide routing, and the basic mechanics of the distance vector routing algorithm are introduced. In lab, they run an application implementing distance vector routing and take turns passing data to remote destinations and observing as intermediate Cybikos help route the data.

Each Cybiko initially knows only its immediate neighbors. Distances to these hosts are determined via student-initiated *ping* messages (which are actually handled by the administrative Cybiko, to maintain central control over the virtual topology). Students can then

request that their Cybiko send distance vectors to its immediate neighbors, and watch as incoming routing information is incorporated and tables updated. (This application can also be used to demonstrate static routing, as each Cybiko can “cheat” and compute optimal routes when instructed to.)

Once all routes have been discovered and the tables have stabilized, the lab administrator demonstrates that the scheme can adapt to changing conditions by shortening one or more paths and watching as this information propagates between machines. Similarly, changes are monitored in response to disabled Cybikos. This offers an opportunity to present the count-to-infinity problem, and to demonstrate how the split-horizon optimization attempts to solve it. In addition to split horizon (with and without poisoned reverse), the application can also perform triggered updates. These optimizations are all enabled or disabled by the administrative machine, which can also force all student machines to clear their tables at any time.

Link-State Routing Lab

The distance vector lab illustrates clearly to students how the propagation of distance information can delay convergence of the routing tables, and unknowingly pass along “stale” information. In lecture, students are shown how accurate information is broadcast quickly to all nodes using the link-state routing algorithm, and how shortest path algorithms can build routing tables from the link state database. The lab then demonstrates that changes in routing information can propagate more quickly in this scheme than in distance-vector routing, and shows that the count-to-infinity problem no longer occurs. Students witness first-hand the increased volume of routing information as well.

3.2 Wireless Networking

We are continuing to develop Cybiko applications demonstrating wireless networking concepts. The first lab will implement a simple ALOHA-style environment with collisions being deduced by means of overdue acknowledgements. An improved protocol will add the equivalent of carrier sensing, and can be used to demonstrate both the hidden and exposed station problems. A final lab will extend this to a MACA-style protocol with RTS and CTS control frames to prevent collisions.

Additionally, a collection of Cybikos will be used to simulate wireless communication with fixed access points (*i.e.*, cellular communication). Several units will be designated as access points and tethered to PCs on a fixed network. The mobile Cybikos will broadcast probe frames to discover potential access points and present possible candidates to the student. Students can select an

Multiple Choice (88 points)			
	<u>First</u>	<u>Second</u>	<u>Change</u>
Cybiko	56.2	67.2	19.6%
Lecture	56.9	66.8	17.4%

Routing Problems (42 points)			
	<u>First</u>	<u>Second</u>	<u>Change</u>
Cybiko	26.9	35.8	33.1%
Lecture	29.4	33.6	14.5%

Combined (130 points)			
	<u>First</u>	<u>Second</u>	<u>Change</u>
Cybiko	83.1	103.0	23.9%
Lecture	86.3	100.4	16.4%

Figure 2: Assessment exam results

access point and observe communication traffic as the access point is notified and user data is transmitted. Students simulate motion between cells either by manually selecting another access point, or by allowing the application to force a transition if contact with the current access point is lost. (This could be achieved through intervention from the administrative Cybiko.)

3.3 Additional Projects

In addition to the Cybiko lab applications, source code will be made available to interested faculty and could be used as the basis of more involved activities. For example, students could be required to write all or part of the distance-vector routing code and verify that it works correctly. We have been careful to separate our library of communications and administrative routines from the rest of the applications, and this infrastructure could be used in projects such as implementing and evaluating slotted wireless protocols, tunneling IP packets across chains of Cybiko units, writing transport-layer protocols, or building a full 802.11-style wireless system.

4. Experiences

Our experiences with the Cybiko have been favorable overall. The devices appear to be well built and robust, and the free software development kit contains everything one needs to write applications for the device, including documentation for the Cybiko API. Unfortunately, the documentation describes the calls only briefly and the descriptions are occasionally incorrect. Thus, programming for the Cybiko has often required trial and error on our part. (Though others will now be spared this effort, as they can build atop our software layer.) In some cases the Cybiko development team have answered questions via E-mail, but in general their developers' message board has been more useful (and timely). Cybiko has recently begun selling a "professional" version of their SDK that contains libraries of graphics and sound files,

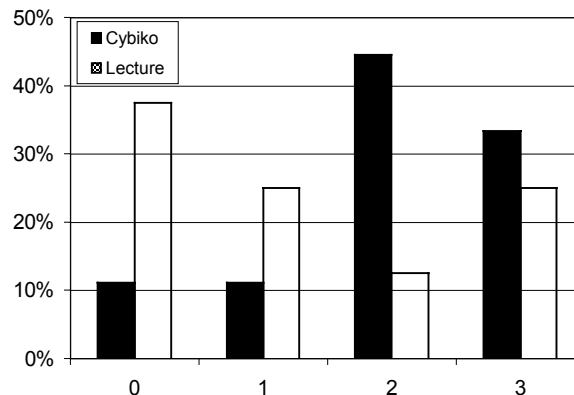


Figure 3: % wanting more examples, by number

and a compiler capable of generating native code⁴, but the API documentation is the same as that in the free SDK.

4.1 Evaluation

The three routing labs described in Section 4 were completed in time to use in our upper-level Computer Networks elective course in the fall of 2001. Until this trial, routing had been taught entirely as a lecture topic, and the trial began with a two-lecture introduction to flooding, distance vector, and link state routing. The entire class was then given an assessment exam to determine how well they knew the material. The group was randomly split into two halves — one of which received an additional more detailed lecture, the other explored the same material via the labs. The assessment exam was then given again to measure improvement.

The class consisted of 17 students, seven women and 10 men, with seven being students of color. While not a particularly large group, it was at least a diverse mix. Nine of these participated in the lab trials, including four women. The exam consisted of 22 multiple-choice questions worth four points each covering routing concepts and issues, and four detailed routing problems worth a total of 42 points. Each component was designed to be precisely gradable.

The results were extremely encouraging, though care must be taken to emphasize the small number of students involved. The exam results are shown in Figure 2. Both groups showed improvement from the first exam to the second, with the students using the labs improving by 24% to the lecture group's 16%. The scores on the multiple-choice portion improved by nearly the same amount in each group, but the lab group's improvement on the detailed routing problems was more than twice that of the lecture group (33% to 15%).

⁴The basic compiler generates compact byte-code programs whose performance is more than adequate for our purposes.

Students were asked, on a scale of 0 to 10, how much they had enjoyed both the first set of lectures and the additional lecture/lab. The lecture group's enjoyment decreased from 7.4 to 6.7 (down 10%) while the lab group's increased from 8.1 to 8.9 (up 11%). Students were also asked whether they wanted to see more examples of flooding, distance vector, or link state routing. Figure 3 shows the percentage of each group who answered yes to zero, one, two, or all three of these questions. Not only did the lab group demonstrate greater improvement and higher overall scores, but they were also far more likely to want to see additional examples.

Finally, the lab group was excited enough about the approach that they spontaneously suggested additional labs to illustrate topics such as connection establishment, virtual circuit setup, and reliable delivery.

5. Summary

We are developing innovative new laboratory materials that give students a hands-on introduction to both traditional and wireless networking concepts. Our experiences with the routing labs have shown that students learn the material at least as well via the labs, and enjoy the experience more than lecture. Our materials, including the source code for all Cybiko applications, are freely available to faculty interested in using the labs to supplement their networking courses or as the basis of more involved projects and simulations. Visit our web site at <http://www.cs.vassar.edu/cybiko>, or contact the first author for more information,

Acknowledgements

This authors wish to thank the NSF for supporting this work with DUE CCLI grant 0087723. Thanks also to Charles Nicholson, Arie Rubenstein, Kwasi Danquah, and Nate Waisbrot for their Cybiko programming assistance, and to the Fall 2001 CMPU 375 students for participating in the assessment trials. Susan Hert provided valuable feedback on earlier drafts of this paper.

References

- [1] Carniani, E., and Davoli, R. The NetWire emulator: a tool for teaching and understanding networks. In *Proceedings of the conference on Integrating Technology into Computer Science Education (ITiCSE)* (2001), ACM Press, pp. 153–156.
- [2] Chang, X. Network simulations with opnet. In *Proceedings of the 1999 simulation conference on Winter simulation (Vol. 1): Simulation — a bridge to the future* (1999), ACM Press, pp. 307–314.
- [3] Jipping, M. J., Krikke, J., Dieter, S., and Sandro, S. Using handheld computers in the classroom: laboratories and collaboration on handheld machines. In *Proceedings of the SIGCSE technical symposium on Computer Science Education* (2001), ACM Press, pp. 169–173.
- [4] Knox, D. L., Fox, E. A., Grissom, S. B., and Heller, R. Computer science teaching center (CSTC). Online. Internet. [Aug, 2001]. Available WWW: <http://www.cstc.org>.
- [5] Levin, M. H. A prototype for a data communications laboratory. In *Proceedings of the SIGCSE technical symposium on Computer Science Education* (1997), ACM Press, pp. 179–183.
- [6] Maj, S. P., Fetherston, T., Charlesworth, P., and Robbins, G. Computer network infrastructure design, installation, maintenance and management - a proposed new competency based curriculum. In *Proceedings of the third Australasian conference on Computer science education* (1998), ACM Press, pp. 9–18.
- [7] Mayo, J., and Kearns, P. A secure unrestricted advanced systems laboratory. In *Proceedings of the SIGCSE technical symposium on Computer Science Education* (1999), ACM Press, pp. 165–169.
- [8] McCauley, R. Computer science education links. Online. Internet. [Aug, 2001]. Available WWW: <http://www.cs.cofc.edu/~mccauley/edlinks/>.
- [9] McDonald, C. A network specification language and execution environment for undergraduate teaching. In *Proceedings of the SIGCSE technical symposium on Computer Science Education* (1991), ACM Press, pp. 25–34.
- [10] Mulder, M. C., and Lidtke, D. K. ISCC '99: Educating the next generation of information specialists in collaboration with industry, 1999. [Aug, 2001]. Available WWW: <http://www.iscc.unomaha.edu>
- [11] Myers, B. A., Stiel, H., and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. In *Proceedings of the ACM conference on Computer supported cooperative work* (1998), ACM Press, pp. 285–294.
- [12] NSF. *Shaping the Future: New Expectations for Undergraduate Education in Science, Mathematics, Engineering and Technology*. National Science Foundation, Arlington, VA, 1996.
- [13] Owen, G. S. Computer science courseware repository. Online. Internet. [Aug, 2001]. Available WWW: <http://www.siggraph.org/education/nsfscsr/nsfscsr.home.html>.
- [14] Rickman, J., McDonald, M., McDonald, G., and Heeler, P. Enhancing the computer networking curriculum. In *Proceedings of the conference on Integrating Technology into Computer Science Education (ITiCSE)* (2001), ACM Press, pp. 157–160.
- [15] Sivalingam, K. M., and Rajaravivarma, V. Education of wireless and ATM networking concepts using hands-on laboratory experience. In *Proceedings of the SIGCSE technical symposium on Computer Science Education* (1999), ACM Press, pp. 114–118.