Abstract—For today’s graduates in computer science and engineering, possessing a working knowledge of networking and data communications has become essential. While most computer science and engineering departments offer an upper-level course in this area, many are exploring ways to teach some of this material earlier in the curriculum. Doing this provides an opportunity for advanced study in this area, but it also presents some challenges. One such challenge is how to create labs and activities for students who may not have the typical upper-division course background. This paper presents activities that are suitable for use in a lower-division Computer Networks course or that can be used in programs where access to proper lab facilities may be limited. In addition, the projects are designed to use collaborative learning and discovery learning techniques. The structure of these activities is presented and a qualitative discussion of their effectiveness is given.

Index Terms - Computer networks, discovery learning, lab exercises, learning activities

I. INTRODUCTION

Today’s computing needs typically make extensive use of network services and facilities. In recent years this has grown to the point where networks pervade nearly all aspects of computing. Because of this, a working knowledge of networking is essential to computer science graduates - an opinion put forth in the Computing Curricula 2001 Computer Science guidelines [1]. While most computer science programs offer an upper-division elective course in data communications the need is often felt to teach this material sooner in the curriculum, and these guidelines make it a requirement instead of an elective within the overall program. One way to satisfy these needs would be to offer a sophomore level networks course or to incorporate essential network information into existing courses. An appealing advantage of the former is that it would give departments the opportunity to offer a second, more advanced networking course to upperclass students. A consensus is forming that supports having such an advanced class [9]. Either way, one problem with teaching networking at a lower level is that the students do not have the background necessary to accomplish many of the more “traditional” networking projects. If taken during the sophomore year most likely students will not have had operating systems, possibly not data structures and their programming skills would not be mature. Therefore, the challenge is to create lab assignments or activities that are meaningful, yet doable for these students. This paper presents activities that the author has used when teaching networking to both junior-senior students and also to sophomores. These activities require very little lab resources which also makes them desirable for courses where access to such facilities is limited.

The nature of these activities is quite varied. Some of them involve student-active learning methods similar to ones present in the computer science literature [3], [5], [8], while others involve studying and making simple modifications to sample protocol code, using existing software tools to extract information about network connectivity and doing Internet research to determine requirements and costs of network configurations. The focus is to design activities that teach basic networking concepts that use simple or no computer tools and that motivate students to learn through discovery. Discovery learning is known to improve self learning and to improve problem solving abilities [2]. A very popular way to teach networking is to consider each layer of the OSI model [10] and study the functionality, algorithms, and requirements for each of them. In keeping with that approach, this paper presents activities that work in conjunction with three of the more important layers - the Data Link, Physical, and Network layers. General descriptions of the projects are given, and interested parties are encouraged to contact the author for details.

II. DATA LINK LAYER ACTIVITIES

When studying computer networks the first serious treatment of software occurs when considering the Data Link layer. In the OSI model this is layer two (just above the Physical layer) while in the TCP/IP model this is combined with the lowest level (Host-to-Network layer). The objective of the software is to make it appear that the physical transmission media connecting two nodes is error-free [4] while in actuality it is not. Protocols are developed that involve a series of packet exchanges between two nodes that include data...
transmissions, acknowledgments, and data re-transmissions. Simple protocols that use unrealistic assumptions are typically taught first followed by a progression of more complex ones that deal with problems that actually occur. What follows are descriptions of two activities that the author has used to teach these protocols.

A. Protocol Learning through “Discovery” Activity
This activity begins by splitting the class into two groups with one being named the “sender” while the other is called the “receiver”. The objective is for the sending group to transmit a message to the receiver group where the message is a 4-5 word sentence that the Instructor has provided. The sending group uses small slips of paper to send the message with the Instructor acting as the transmission medium. They are told that this medium is capable of losing and/or corrupting the messages. While this type of activity is similar to the simulation game discussed in [8] in that it teaches Data Link layer protocols it differs in the way that it is taught. In that activity the students have received information about the basic principles of the protocol being used, a discussion of problems that may arise and suggestions for solutions prior to performing the activity. In the activity presented here the Instructor provides virtually no initial discussion of these protocols nor does he discuss problems and solutions. Rather, the goal is stated and the only “rule” given is that just one word of the sentence can be transmitted on a slip of paper at a time. It is stated that the goal is for the message to be received correctly by the receiver and the the sender knows that this occurred.

After the Instructor has finished giving the brief instructions the groups are sent into different locations. What typically happens next is that the students express uncertainty of what to do. They know the goal but are unsure of how to meet it. Consequently, each group is forced into brainstorming about how to proceed to accomplish the activity goal. The sending side knows that they must write the first word of the sentence on the paper slip. However, they realize that they must think about what the receiver side needs to correctly receive not only this individual piece of information (called a packet) but also to correctly reconstruct the entire sentence. The Instructor checks on the group but plays a passive role. In some cases it does become necessary to steer a group away from a poor idea that would squander time or cause unwanted digressions. Otherwise, the Instructor lets this group make its own decisions on how to create its data packet. On the receiving side the immediate issue is whether it should wait for a packet or send a “request” packet asking the sender to transmit data. Other issues are how to handle an arriving packet, how to tell if this packet has been corrupted, and how to respond to it. Again, the Instructor listens to their discussions and only interferes if necessary to prevent very poor choices. What is occurring is that each side is forced to think through the same areas that the designers of the actual network protocols had to consider.

Eventually the sending group creates a packet and passes it to the Instructor for transmission. The Instructor can deliver it (representing an error-free transmission), corrupt it, or not deliver it (lost packet). It works out better if the first packet is delivered without error. Now it is the receiver’s job to take some action. They read the paper slip and try to ascertain what the message is and if it is correct. At this point they start to see that there needs to be some agreement with the sending side on the numbering system to use (for ordering the packets) and on the form of error detection. Having no agreement, they do their best in formulating a response. While they may not know the term for what they are doing, they realize that they need to send an acknowledgment packet to the senders. The form and content of the acknowledgment becomes the discussion. Concurrently, the sending side must decide if they should send the second word of the sentence or wait for a response regarding the first word. The term for this in networking is “flow control” and the Instructor does tell them that it is possible for the receiver to not be able to hold or process more than one message at a time. Eventually the receiver group will formulate their packet and give it to the Instructor for transmission. Again, the Instructor can deliver the packet unaltered, can change its contents or simply not deliver it. If the packet is delivered, the sending group will receive it, note its contents and determine their next step. They may decide to send the next word (satisfied that the first word was received correctly) or they may think that something may have happened and their original transmission was corrupted. If they have waited for a length of time and have not received any packet from the receiver group they have another issue to confront - namely whether they should “time out” and re-transmit the first word.

The activity continues in this manner with the Instructor delivering, altering, or losing packets to simulate various events that can actually happen when two computers are communicating. After a while, the Instructor calls both groups back into the same room and a discussion of the activity and its relation to actual network protocols ensues. Students are told that the problems they faced are actual problems that do occur. Furthermore, protocol designers have developed solutions for not only these situations but for many others as well. Lastly, ways are discussed to increase efficiency and to allow both sides to act as senders and receivers concurrently. If time permits, the simulation can be repeated using these discussed techniques.

B. Protocol Code Analysis and Modification Activity
Once students grasp the basic functionality of the data link layer protocols it is good for them to see some actual program
code that implements them. While samples of such code can be obtained from a variety of sources, the website supporting the Tanenbaum book [10] provides protocol code along with a simulator. Having these available provides the basis for this activity. Downloading and installing the necessary files is a simple process. The protocols provided range from the simple (p1) to the complex (p6) with each providing enhanced functionality and robustness. The protocol given that most closely resembles the “stop and wait” protocol that the previous activity simulated is p3.

In this activity students analyze this p3 protocol. Then, they simulate it with various parameter sets. These parameters are altered to demonstrate how this protocol works under various conditions (higher error rate, various time-out values, etc.). Next, they are given an assignment where they need to perform some simple modifications of the code. One such activity is to turn the original “sender-driven” code into a “receiver-driven” protocol. Once done, they simulate their new protocol and compare it to the “sender-driven” version. Making the modifications requires mostly code shuffling that is well within the capabilities of sophomore level students. By doing enough simulations they can determine when such a protocol fails, and how it compares with the sender driven one. Ultimately they learn why the sender-driven protocol is preferred.

III. PHYSICAL LAYER ACTIVITY

This physical layer involves the actual network hardware and the signaling of data that occurs. Along with the theoretical topics taught at this layer are practical ones such as the transmission media (coaxial cable, twisted pair, fiber optics, etc.), the various network topologies, and the hardware required to build such networks. The nature of these topics lend themselves well to creating an activity suited for lower division students. While it is easy to obtain network equipment and have a “show and tell” demonstration, what has proven more beneficial is if the students perform a little design and then do an analysis of the hardware cost to implement it.

Students are given an example scenario to consider (such as a company residing on three floors of a building that wants to network 20 computers on each floor) and are instructed to design a local area network, or LAN, to meet the requirements. The design phase requires them to consider the network topologies available (star, ring, bus, others) to determine which might work the best. In doing so, they have to also consider the cabling requirements, hardware for each workstation including the network card, the need for hubs and repeaters along with any other hardware required to build the network. After creating a drawing showing the topology and listing the hardware needs, the students get approval of their design by the Instructor. Next, they perform an Internet search to locate and price each item and determine an overall hardware cost. At the end of the activity students share their solutions and compare costs. By doing so they learn that there are several possible solutions each with its own benefits and deficiencies. A discussion ensues about which network topologies could be expanded to handle additional computers with the lowest additional cost, the so-called scalability of the network.

IV. NETWORK LAYER ACTIVITIES

The Network layer is responsible for the source-to-destination delivery of a packet across the links in a network. Typically the Internet or another wide area network, WAN, is used for example purposes here. Topics studied at this layer include the logical addressing of network computers and the routing of packets in the network. In order to route a packet from a source to its destination some knowledge about the overall connectivity must be maintained. Hence, an individual machine must be configured to send packets to one or more intermediate routers. Routers, in turn, send the packets to other routers and so on until the packet reaches its destination. This routing process fosters a discussion of how the network is connected and how these routers know where to forward incoming data packets. Both of these topics are the basis for good activities suitable for lower level students.

A. Network Connectivity Activity

Local area networks, and indeed the Internet as a whole, is set up so that details of its configuration are hidden. However, many easy to use software tools exist that allow one to uncover much of this information. Some of these tools exist on standard personal computers and Unix workstations. Commands such as traceroute, netstat, ifconfig and others provide information that is useful in determining not only how the local host is configured but also the route that a packet takes from a source to its destination. In addition, many websites exist that provide similar tools that will show the intermediate locations, or hops, that a packet takes when moving from place to place [7].

In this activity, students are charged with discovering as much as they can about how their local area network is configured as well as how it connects to the Internet. They are given basic instruction on how to use the software tools mentioned and are provided with additional website addresses that contain other tools. To learn about their workstation and lab connectivity they will need to use the software tools in a myriad of ways. Additionally, they usually will have to work on several different machines within the lab. As an “extra” part of the activity, students are encouraged to learn how other computer labs on campus are configured. What typically happens by doing this project is that students begin to learn that the machines in the same lab have much in common although they are different from machines in other labs.
Furthermore, they learn about the existence of routers that connect various labs and that provide the campus connection to the Internet. Lastly, the use of Internet Protocol addresses and the movement of packets is clearly demonstrated.

B. Routing Table Activity

While working on the above activity students often inquire about how a router knows where to transmit its incoming packets. The theory says that a routing table is used but then the question of how this routing table is built arises. There have been several algorithms that have been used in the history of the Internet to collect and build this information. Two of the more popular ones are based on the exchanging of information between routers. An individual router knows the networks to which it directly connects. At various time intervals each router tells its neighboring routers what it knows about the network. Much like the analogy of friends telling friends gossip, eventually knowledge spreads to each router. From this information, each router builds its own routing table.

Two activities have been developed to help teach this concept. The actual algorithms are Distance Vector Routing, DVR, and Link State Routing. For brevity, only the former will be discussed although the author has made activities for each algorithm. The activity begins with each student given the task of acting as a router and being given information about what routers he/she is connected to and the “distance” to each of them. The student is to create a routing table based on this information. Hence, at the beginning a router only knows about the routers that are connected to it. The Instructor explains what information needs to be exchanged between neighbors and provides the proper format. The students use slips of paper to create their DVR information packets and then exchange them with their neighbors. Upon receiving these packets each student will use them to update their own routing table. Information kept is knowledge of other routers, the distance to them, and which neighbor to use as an intermediary table. Information kept is knowledge of other routers, the packets each student will use them to update their own routing table. Upon receiving these packets each student will use them to update their own routing table. Knowing what routers he/she is connected to and the “distance” to each of them. The student is to create a routing table based on this information. Hence, at the beginning a router only knows about the routers that are connected to it. The Instructor explains what information needs to be exchanged between neighbors and provides the proper format. The students use slips of paper to create their DVR information packets and then exchange them with their neighbors. Upon receiving these packets each student will use them to update their own routing table. Information kept is knowledge of other routers, the distance to them, and which neighbor to use as an intermediary to reach them. The process of creating new DVR packets, exchanging them, and updating routing tables continues until each person has a routing table that does not change after a round of packet exchanges. It is said that the table is now stable.

At this point students will have learned how the algorithm works and why it is used. However, the Instructor should not stop the activity at this point. Rather, some catastrophic networking events should be introduced. For example, what happens if a workstation suddenly stops responding? How should that be treated by the neighbors? How are the routing tables adjusted? How long until everyone knows that this machine is “dead”? All these questions can be discussed within this activity and students will clearly see the issues and how they are resolved.

V. DISCUSSION

The author is convinced that these activities are useful tools for teaching many of the complex topics in computer networks. Additionally, they require either no or minimal lab resources making them suitable for a lower-level course or when available facilities are at a minimum. The game-type simulations have been used many times and the student response is always overwhelmingly positive. Additionally, on several occasions I have had peer faculty members participate and provide input. Each time they shared the students enthusiasm and expressed a better understanding of the subject matter. At the end of each activity a discussion followed where the networking concepts are tied to the activity results. This is quite beneficial as later during lecture periods references can be made to the activities - a link that improves the student’s understanding. Indeed the technique of having students actively involved (as opposed to being passive viewers or listeners) has been identified as important in creating understanding [6].

The data link layer simulation is typically the most popular of these activities as student find that they can figure out the issues that must be addressed by these complex protocols. From a teaching perspective, it is a good activity because it requires group cooperation, is discovery learning, and requires no lab equipment. By following this activity with the code analysis activity students get a chance to see how the ideas they learned are actually implemented. Since they know what ideas to look for, reading and learning the code is much easier. The physical layer activity has also proven to be a good activity. Students typically have an interest in building a network but have no real idea on what is involved. This activity provides them with an opportunity to design and price a network. In the follow-up discussion they are curious to learn of classmates designs and the costs of alternate configurations. The only requirement of this lab is an Internet connection.

When studying the Internet, the issue of connectivity and routing of packets is always of great interest to the students. The activity of learning how routers know where to send packets greatly interests them. Because the Internet contains distributed information this activity is useful in teaching students how such information propagates through the routers. By acting as routers and exchanging information, students get acquainted with a distributed form of algorithm - something that the vast majority of lower-level students will not have seen. Because the Internet is composed of individual local area networks, a hosts connection to the LAN and ability to get packets to the router is important to study. By using the software tools mentioned, students can learn how data moves from an application on a host through its LAN and out into the Internet. The lab requirements of these projects are a basic PC or Unix lab and access to the Internet.
VI. Conclusion

In computer science departments it is becoming essential that students acquire a working knowledge of computer networks and data communications. As this material has typically been taught in an upper class elective, the challenge is to either make the course a requirement or to fit the material into other existing courses. In either case, it is important that there be creative activities to supplement the textbook and lecture material. In this paper, activities have been presented that can be used to sophomore-level students and beyond and that require only minimal lab equipment. In addition, the projects were designed to do the following:

- Serve as a tool for understanding the non-trivial Data Link Layer Protocols.
- Provide an opportunity to better understand the hardware requirements of an actual LAN.
- Give students the opportunity to learn about network connectivity and how a distributed system exchanges information while working to efficiently route packets.
- Promote learning through discovery and collaborative learning.

REFERENCES


