121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014

Paper ID #10087

A Multiple-Access Message-Exchange Course Project for a Networking Course in a BS Computer Engineering Program

Dr. Edward W. Chandler P.E., Milwaukee School of Engineering

Dr. Chandler is Professor of Electrical Engineering and Computer Science at Milwaukee School of Engineering (MSOE). He received the Ph.D. degree in electrical engineering from Purdue University in 1985 and is a registered Professional Engineer in Wisconsin. He previously was a Member of Technical Staff at L-3 Communications and currently performs systems engineering consulting in the area of communications for DISA (U.S. DoD). He is a Senior Member of the IEEE, and teaches courses in circuits, signals, communication systems, and networking.

Prof. William Barnekow, Milwaukee School of Engineering

Prof. William Barnekow is Professor of Electrical Engineering and Computer Science at Milwaukee School of Engineering. He earned the M.S. in Electrical Engineering from University of California – Berkeley. His primary areas of interest are in embedded microcomputer-based systems, digital circuits and systems, advanced digital design using VHDL, senior capstone design projects in software and computer engineering, and computer networks.

A Multiple-Access Message-Exchange Course Project for a Networking Course in a BS Computer Engineering Program

Abstract

Since the 2009-10 academic year, the seniors in the computer engineering program at Milwaukee School of Engineering (MSOE) have been required to complete two networking courses, Networking I and Networking II. Each carries three credits on the quarter system and each includes a project-based laboratory. The first of these two courses concentrates on the physical and data link layers of communication networks, and the second concentrates more on higher layer protocols, with emphasis on those used in Internet applications. The first course includes a course project in which student teams of up to three students each are formed. Each team designs and implements a network node or terminal for short message exchanges, which is expected to interoperate with nodes implemented by other teams.

For the course project in the first course, the network medium has a bus topology. Both electrical busses and wireless optical media have been used for node-to-node connectivity in different years of the course offerings. The stated purpose of the network is to allow the exchange of short text messages between connected nodes. The students implement lower layer network protocols that are covered in the lecture portion of the course, including carrier-sense type channel monitoring, collision detection, random back-off times for retransmissions, message-packet error detection, and a specific signal-coding technique such as Manchester encoding, which varies from year to year. An interoperability standard is used by the teams. In some years the standard has been developed by the students, and in more recent years it has been defined by the faculty members teaching the course. Student teams are required to draft the test procedures for verifying different parts of the required network operations, which are developed incrementally throughout the course.

This paper presents an overview of the microcomputer-based platform that has been used for the Networking I course project, and several specifics on the signaling techniques that have been used in different years of the course offerings. The paper also discusses the degree to which course project success was achieved based on assessments including successful project milestone demonstrations and student surveys, several problems that were encountered, and the actions that were taken to address the problems.

Introduction

At Milwaukee School of Engineering (MSOE), the ABET-accredited computer engineering curriculum includes two required three-credit courses on the topic of computer networks. The first course concentrates on the physical and data link layers of networks and the second course has more emphasis on higher layer protocols common in Internet applications. Each of these two courses includes a laboratory. The focus of this paper is on the course project within the laboratory of the first course.

For the course project in the first networking course, the student teams of up to three students each are formed. Each team designs and implements a network node or terminal that is expected to interoperate with nodes implemented by other teams. The network nodes are required to transmit and receive short text messages over a specified network medium, using specified signaling, addressing, and medium access control (MAC) protocols. The required signaling, addressing, and MAC protocols are specified in an interoperability standard, and the network nodes are expected to comply with the requirements stated in the standard.

Most student teams choose to implement the network node on a microcomputer platform with which the students have familiarity and have used in earlier courses. However, the students are not required to use any specific implementation hardware and methodology. Each student team is free to choose any hardware and, if applicable, any software language, in the network node implementation. The Atmel ATmega32 microcontroller and the C programming language are currently used in an earlier embedded systems course in the curriculum, and are therefore currently chosen by most student teams.

This expository paper describes course project details such as the types of network media that have been used in various years of this course offering, the types of signaling that have been used, the types of protocols that students have been required to implement, and the suggested techniques that have been given to students for implementation of some of the protocols such as collision detection. The primary value offered by this paper is (a) its identification of some types of network configurations, applications, and protocols that can be successfully implemented in a networking course that focuses on signaling and lower layer protocols, (b) its description of innovative techniques for collision-detection on networks having a bus topology, and (c) its description of project activities that greatly contribute to students' exposure to real-world networking activities, such as design and adherence to standards, and test procedure development and execution for requirements verification.

This paper is organized as follows: The Networking Course Project Overview section provides general information about the duration of the course, the schedule of laboratory sessions allocated to the project, and the use of an interoperability standard to capture the requirements for nodes operating on the network. This section also describes the network application, network topology, network node software components, network media, and message transmission format. The section titled Line-Coding Formats, Collision-Detection Techniques, and Collision-Detection Testing describes each of those aspects of the project, including a method for generating collisions to test random back-off times that are required for retransmission attempts. A brief section on Error Detection Protocols describes the protocols of this type that have been incorporated so far, and ideas for future expansions to include data link control protocols. A section on Project Experiences and Feedback from Student Surveys describes the degree to which course project success was achieved and identifies some problems that were encountered and the actions that were taken to address the problems.

Networking Course Project Overview

General information and schedule of project milestones: MSOE has academic terms on the quarter system, with each quarter spanning eleven weeks. The networking course project is described to students in the first laboratory session for the course, and six of the remaining nine weekly laboratory sessions are allocated for project milestone demonstrations. Table I shows the schedule of project milestones that each project team was expected to demonstrate for the 2013-14 offering of the course. During each laboratory period that has a project milestone, student teams each demonstrate network node compliance with the expectations for the specified milestone. The use of scheduled milestones results in an organized, incremental development of the various components of the network node functionality. Students are expected to work in teams on their implementations at times outside of the scheduled laboratory sessions, and the scheduled sessions are intended to accommodate the demonstration of the implemented milestones.

Table I. Schedule of Project Milestone Demonstrations for 2013-14 Course Offering

Week	Laboratory Topic
1	Laboratory Safety and Course Project Description
2	Experiment 1 on Digital Signal Line Coding including Manchester
3	Project Milestone: Demonstrate operation of Channel Monitor function that continuously determines state of network channel/medium (Idle, Busy, or Collision)
4	Project Milestone: Demonstrate operation of Transmitter function that sends properly formatted signals on the channel/medium (transmission of short sequence of characters required; entire message transmission not required; header on each transmission not required)
5	Project Milestone: Demonstrate operation of Transmitter function operating in conjunction with Channel Monitor function (previous two milestones working together; channel must be idle to transmit; random back-off upon collision required)
6	Experiment 2: Channel Bandwidth Required for Digital Data Signals
7	Project Milestone: Demonstrate operation of Receiver function that receives signals from the channel/medium and outputs decoded text messages (for testing, can use Transmitter function developed by same team or can use ARB generator having message transmissions preprogrammed by instructor; reception/interpretation of header not required; interoperability with other teams not required)
8	Experiment 3: Bit Error Rate (BER) Testing on a Digital Data Link with Noise
9	Project Milestone: Demonstrate transmission and reception of complete text messages (Week-5 and Week-7 milestones working together; header transmission and reception/interpretation required; no interoperability with other teams required)
10	Project Milestone: Demonstrate transmission and reception of complete text messages (Week-9 milestone repeated, but interoperability with other teams is required/tested using a network hub); and optional demonstration of CRC-based Error Detection function in transmitter and receiver (for extra credit in the lab)

An interoperability standard is used by the teams. In earlier years the standard had been developed by the students, with one student from each project team serving as the team representative on an interoperability standards committee. Although the development of the interoperability standard was a very useful component of the networking project experience, it resulted in a stable standard being first available at too late of a date in the academic term to accommodate the teams' implementation of the required protocols. In more recent years the interoperability standard has therefore been developed and provided to the teams by the faculty members teaching the course.

Network application: For each year that this course project has been assigned, the network application has been the exchange of short text messages between nodes. The text messages usually have a specified maximum length of approximately 200 characters. Each node has a unique address, and there is also a defined broadcast address. Each node is expected to receive and display only those messages sent to its address or to the broadcast address. Each node when transmitting a message is expected to identify its address as the source address and is expected to identify a destination address. This is a relatively simple application. Course time constraints have discouraged the use of more complicated applications, such as a real-time data-streaming service for a real-time feedback control application. Such an application would seem better suited for an independent study or perhaps a follow-on elective course, which has not yet been developed or offered at MSOE, or perhaps for a semester-length course at a different institution.

Network topology: For each year that this course project has been assigned, the network has used a bus topology, although, as in the case of some real-world networks having a logical bus topology, it is implemented through a central hub. The network therefore has a bus topology logically but physically has a star topology, as shown on Figure 1. Each network node shown on Figure 1 is required to output a continuous high level when it is idle (that is, not transmitting). If any node outputs a low level then the network bus will be at a low level.

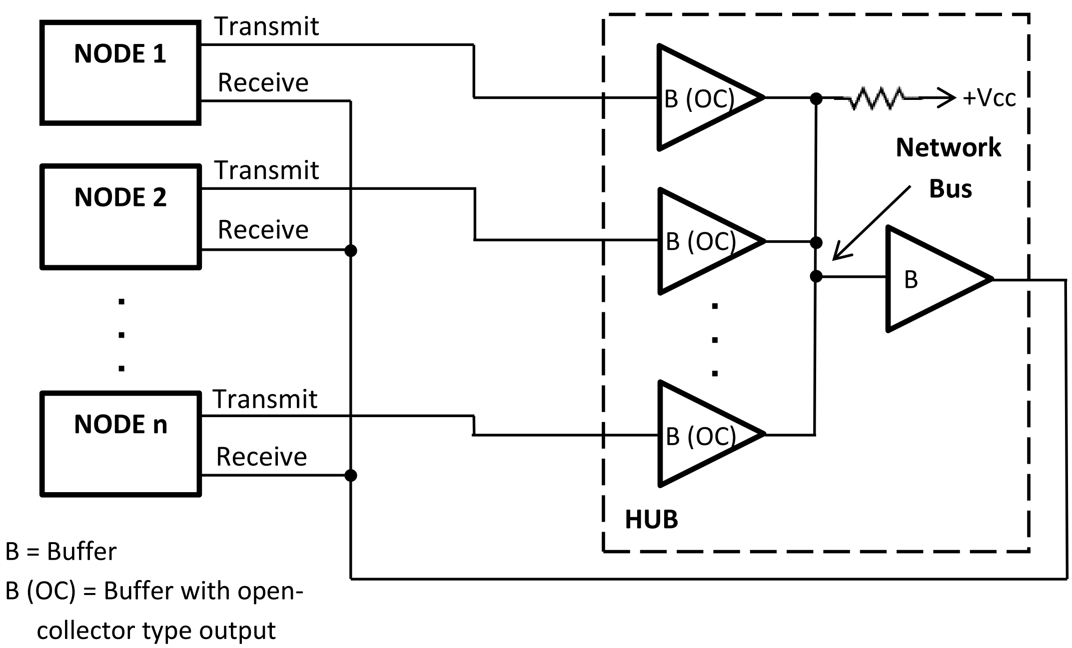

Figure 1. Logical bus topology, physical star topology, with utilization of a hub.

With the logical bus topology, an Ethernet-like medium access control (MAC) protocol is used. This MAC protocol has a carrier-sense function that determines whether the bus is idle or busy, and a collision-detect function that determines when there are collisions on the bus. In a future year, a course networking project that uses a ring topology might be tried. This could be an interesting deviation from what has been used so far. The reasons for the consistent (so far) use of a bus topology are (1) the bus topology is the best topology to use if carrier-sense and collision-detect protocols are to be implemented, and (2) for a course that focuses on lower layer network protocols and signaling, the widespread real-world use of the Ethernet protocol provides motivation for including carrier-sense and collision-detect protocols within the project.

Network node software components: Using the bus topology, the networking course project requires each student team to implement three components within its network node. These components are typically software components and are called the Transmitter, Receiver, and Channel Monitor functions. The Channel Monitor function will be described here. The Channel Monitor is expected to maintain three flags to indicate the bus status: the IDLE flag, the BUSY flag, and the COLLISION flag. When actively operating on the bus, at any particular time, exactly one of these three flags should be asserted. The IDLE flag indicates that there is no activity on the bus (that is, no carrier present) and this informs the Transmitter function that it is allowed to transmit. The Transmitter should never begin a transmission unless the bus is idle. The BUSY flag indicates that a carrier has been detected and a transmission is in progress on the bus. This should inform the Receiver function to begin receiving data from the bus (unless the same node is transmitting). The COLLISION flag indicates that two or more nodes are attempting to transmit at the same time on the bus. When a collision is detected by a node, the node is expected to stop transmitting and wait a random back-off time after which it is to wait for an idle bus and then attempt a retransmission.

Wired versus wireless network media: For most of the approximately seven years that this networking course project has been assigned, a wired, electrical bus has been the network medium used. During two of the years when the course was offered, the project used an optical medium instead of an electrical bus. When an optical medium was used, the electrical signals that served as transmit outputs from the network nodes were connected to infra-red drivers, and all network-node drivers were aimed at a reflective surface. Each network node also had an infra-red receiver aimed at the same reflective surface. The infra-red receivers provided electrical signals as receive-signal inputs to the nodes. Having wireless optical signals made the project a bit more interesting to some students. However, the network-node implementations were otherwise the same as they would have been with an electrical bus. It was decided that the additional complexity and logistical difficulties in setting up the wireless optical network outweighed the positive aspects of having implemented a wireless network.

Message transmission format: Each transmission for the message-exchange course project application contains one message that is expected to be encapsulated in a packet containing a header segment. The header contains the source address and the destination address. To accommodate a variable message length, for some course offerings the header has included a message-length field and for other offerings an end-of-message control character has been used within the message/payload segment. For some of the years that this networking course project has been used, one or more fields for error-detection frame-check sequence(s) have been included.

Line-Coding Formats, Collision-Detection Techniques, and Collision-Detection Testing

Line-coding formats: The assigned signaling technique, also called line-coding format, has varied from year to year. Several line-coding formats have been used. Two examples that have been used are Manchester encoding and a technique called Return-to-High (RH) coding.

A Manchester encoding technique that has been used with this project is described as follows:

> Manchester encoding: A logic-0 bit shall be transmitted as a high signal level (nominally +5V) during the first half of the data bit interval followed by a low signal level (nominally 0V) during the second half of the data bit interval. A logic-1 bit shall be transmitted as a low signal level (nominally 0V) during the first half of the data bit interval followed by a high signal level (nominally +5V) during the second half of the data bit interval.

A Return-to-High (RH) encoding technique that has been used with this project is described as follows:

> Return-to-High (RH) encoding: A logic-0 bit shall be transmitted as a low signal level (nominally 0V) during the first half of the data bit interval followed by a high signal level (nominally +5V) during the second half of the data bit interval. A logic-1 bit shall be transmitted as a high signal level (nominally +5V) during both halves of the data bit interval.

Innovative collision-detection techniques associated with line-coding formats: Consider the above defined Manchester encoding. When this Manchester encoding is used, during a transmission without a collision, the bus will never be at a high level for more than one data bit interval, and will never be at a low level for more than one data bit interval. Therefore if the bus is high for more than one data bit interval, the bus is to be declared as being in the IDLE state. Upon detection of any high-to-low transition on the bus, the bus is to be declared as being in the BUSY state. If more than one node transmits at the same time, unless all transmitted data from the two or more colliding nodes are identical and in perfect synchronization, the bus at some point in time will go low for more than one data bit interval. Therefore if the bus is low for more than one data bit interval, the bus is to be declared as being in the COLLISION state.

If Manchester encoding is used, one simple algorithm for the Channel Monitor function is as follows: Assuming that the bus is initially idle (at a constant high level), and that the Channel Monitor in the node initializes in the IDLE state, switch to the BUSY state upon the first high-to-low transition on the bus. While in the BUSY state, switch to the IDLE state if the bus goes high for more than 1.05 data bit durations. The additional five percent in the 1.05 factor can accommodate clock errors at nodes. While in the BUSY state, switch to the COLLISION state if the bus goes low for more than 1.05 data bit durations. While in the COLLISION state, switch to the IDLE state if the bus goes high for more than 1.05 data bit durations.

There are many ways that the above described channel-monitor algorithm can be implemented, for example using interrupts in a microprocessor-based embedded system. The implementation and demonstration of the Channel Monitor function is what would be expected of each student team. For most of the expected project milestones scheduled throughout the term, each team is expected to develop the algorithm(s) to be used, implement them, develop a test procedure for verifying the expected operation, and use the procedure to demonstrate the milestone. In recent course offerings, the first project milestone, which is the Channel Monitor function, is expected very early in the term (see Table I), and the students are, for this milestone, given suggestions for the algorithm (such as the one given above) and are also provided a test procedure to use. For all of the subsequent required milestones, the students are not given suggestions for the algorithm and are not provided with a test procedure. They are expected to develop these.

Example test-procedure step for verifying expected collision detection: Following is an example of a portion of the test procedure for the Channel Monitor function, using the above-defined RH line-coding format, assuming that the nominal data rate on the bus is 4800 bits per second (bps) and that the first data bit in every transmitted byte is a logic-0 bit (which is transmitted as a low signal level during the first half of the data bit interval followed by a high signal level during the second half of the interval). The following example procedure assumes that the COLLISION state should be entered if the bus level goes low for a time interval longer than one-half of a data bit interval, and the node is expected to have implemented a collision threshold between 52 and 62 percent of a data bit interval.

> Specific example of COLLISION detection verification: After first verifying that a constant high level into the node's receive input results in the node indicating a continuous IDLE state, and a constant low level into the node's receive input results in the node indicating a continuous COLLISION state, a 2400-Hz square wave having 75% duty cycle, as shown on Figure 2, is presented to the node's receive input to verify that the node then indicates a continuous BUSY state. Then the frequency of the square wave is gradually decreased until the COLLISION indicator goes on (and ideally the other two indicators – IDLE and BUSY – are both off, however it is acceptable if the BUSY indicator is on more dimly, that is, a smaller fraction of time, than the COLLISION indicator). Determine the frequency at which this COLLISION indication first occurs. This frequency must be between and 1935 Hz and 2308 Hz to pass. [Note: 2308 Hz, with 3/4 duty cycle, has a signal-low time of approximately 52% of the bit interval T_b, and has a signal-high time of approximately $1.56T_b$; whereas 1935 Hz, with 3/4 duty cycle, has a signal-low time of approximately 62% of T_b, and has a signal-high time of approximately $1.86T_b$.]

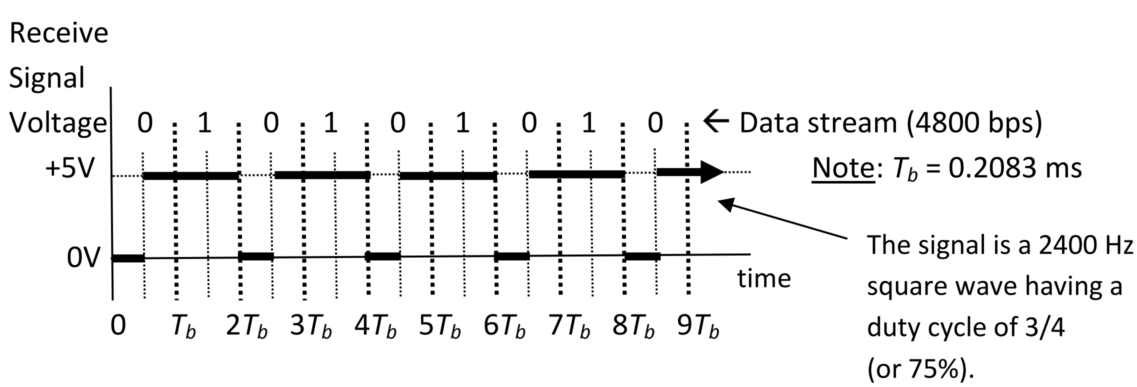

Figure 2. Test signal for verifying BUSY state and COLLISION detection.

Verification of back-off when collision occurs during transmission. A description of a test procedure for testing the detection of collisions is given just above. However, in order to test that a transmitting node will, upon collision detection, back off (that is, stop transmitting) and wait a random back-off time before retransmitting, it is necessary to create the collision condition after the node being tested has started its transmission. If the collision condition already exists at the time that the node is commanded to send a message, the node will not begin any transmission because the bus will not be in the IDLE state. In order to accommodate this test condition, a simple circuit, shown on Figure 3, was developed to serve as a collision generator.

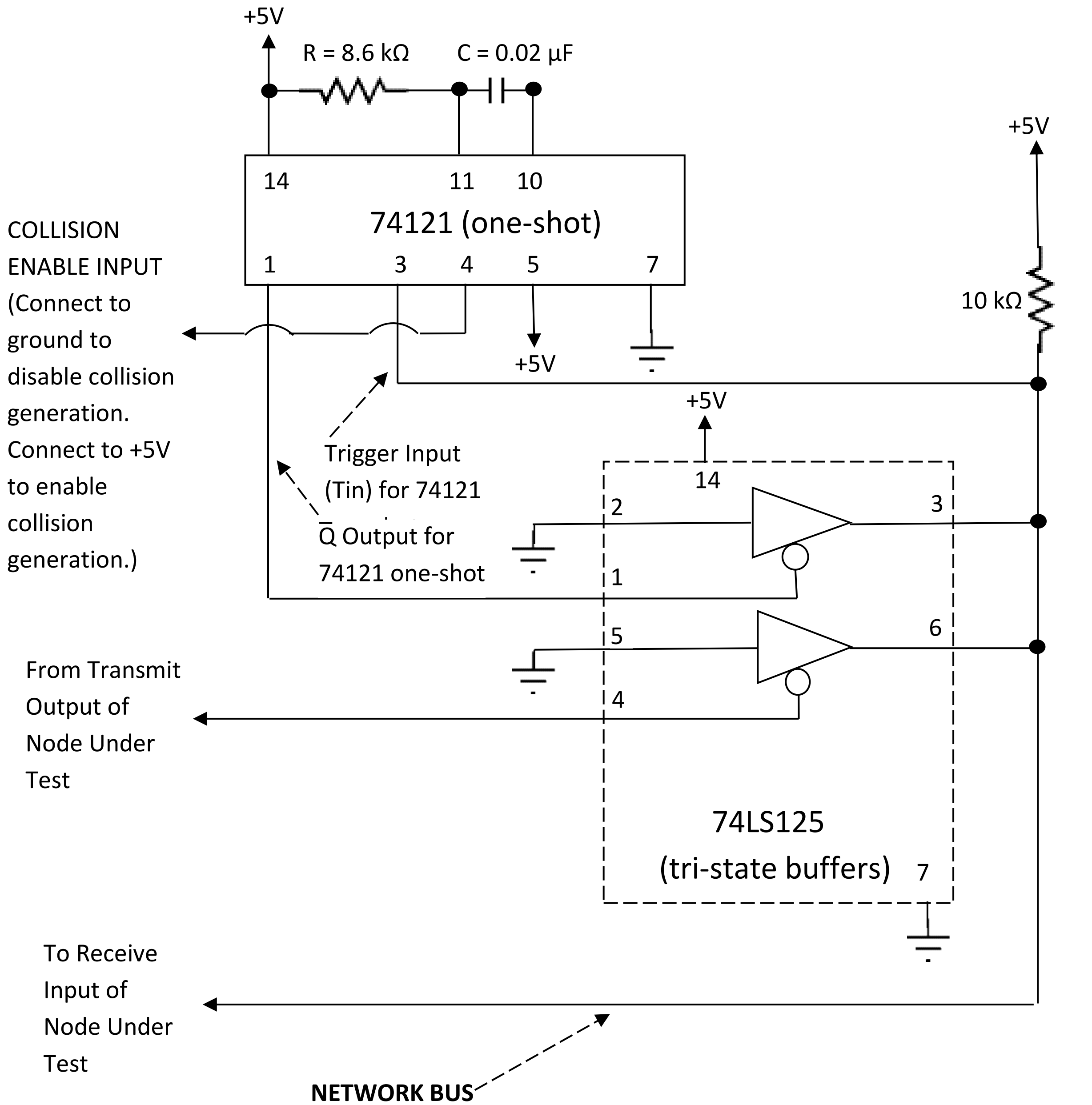

Figure 3. Collision generator for verifying random back-off upon COLLISION detection.

The circuit shown on Figure 3 creates a collision condition after the node under test has started its transmission. The edge-triggered one-shot has an output at pin 1 that is normally high. When the node under test starts transmitting, the first high-to-low transition on the bus triggers the one-shot and causes it to bring the bus low for a predetermined time that is expected to be recognized as a collision. On Figure 3, that predetermined time is 0.7 RC, or 0.1204 millisecond, which is 58 percent of a data bit interval at 4800 bits per second. After that predetermined time, the collision generator no longer holds the bus low, and the node under test should find the bus in the IDLE state after its random wait time expires. The node under test is then expected to attempt a retransmission. A photograph of the prototyped collision generator is shown on Figure 4.

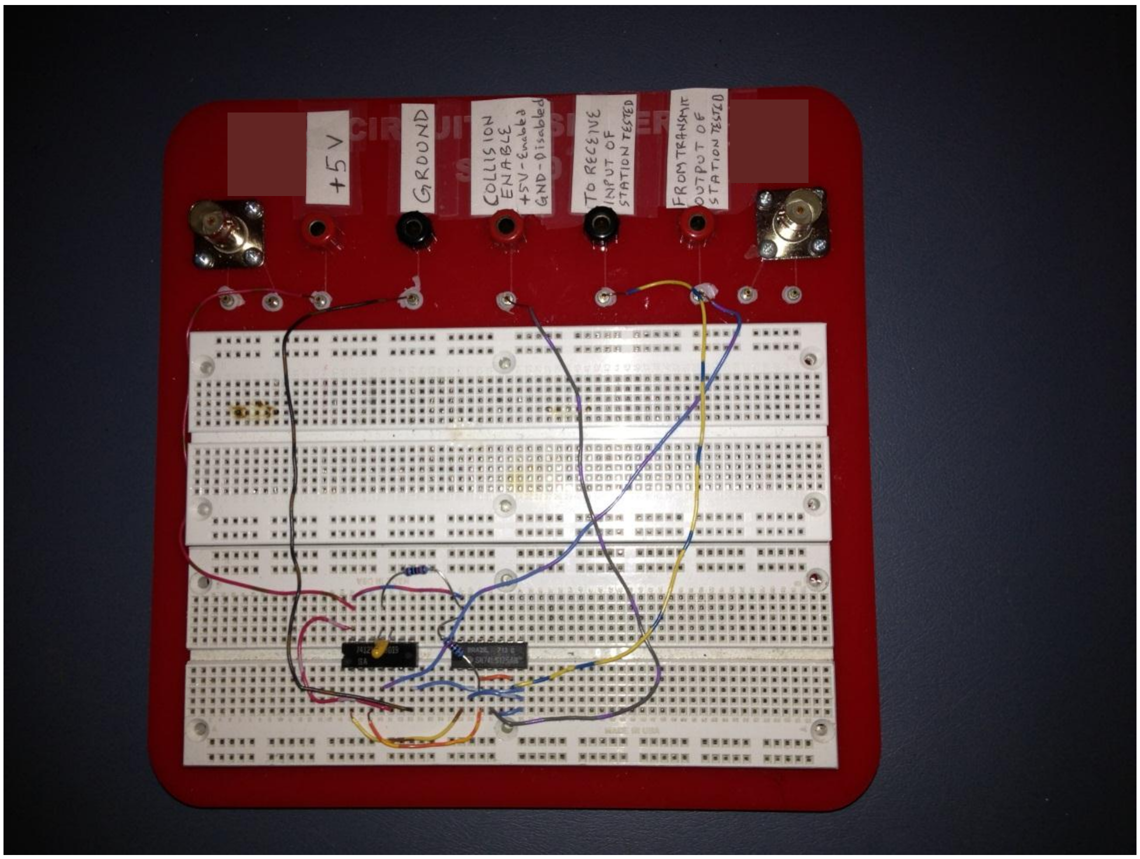

Figure 4. Photograph of collision generator.

Error Detection Protocols

For some of the years that the networking course project has been used, error-detection protocols based on a cyclic redundancy check (CRC) were included in the project. In such cases, the most commonly used CRC error checking included a CRC field within the header for detection of errors in the header, and a trailer segment that followed the message/payload segment. The trailer segment then included a single field that contained the CRC for detection of errors in the message. This networking course project has so far not included the transmission of acknowledgement messages (ACKs) or requests for retransmissions (that is, negative acknowledgements or NAKs). The inclusion of a data link control protocol involving ACKs and possibly NAKs is a feature that could be included in a future offering of the course.

Project Experiences and Feedback from Student Surveys

Feedback from student surveys and actions taken: For each course in which a student is enrolled, MSOE has each student complete a Class Climate survey at the end of the course. The survey for each course consists of several items for which students provide numerical ratings, and a place for students to type in positive comments and/or comments on things that need improvement. The survey items that are given numerical ratings by each student and that are most directly related to the networking course project are the two items identified in Table II. Although the two survey items listed in Table II are quite general, and are expected to have student responses based on many aspects of the networking course (not just the course project), these are the items on the standardized survey that are most related to the course project. The data in Table II span three academic years over which the wording on those survey items was identical and therefore allowed meaningful comparison. The data in Table II indicate that the 2011-12 offering had slightly better results than the other two years. However, the survey questions pertain to many aspects of the course, not just the course project, and therefore the only reliable conclusion from the data is that the responses to these two survey items were generally good in all three years.

Table II. Networking course survey results (items with numerical ratings)

Survey item:	2010-11 Results:	2011-12 Results:	2012-13 Results:
Assignments and exams were representative of the material covered in class.	n = 34 average* = 4.0 std dev* = 0.85	n = 25 average* = 4.3 std dev* = 0.6	n = 27 average* = 4.1 std dev* = 0.9
The assignments were helpful in increasing my understanding of the course material.	n = 34 average* = 3.8 std dev* = 1.1	n = 25 average* = 4.1 std dev* = 0.9	n = 27 average* = 4.0 std dev* = 1.1
* The survey numerical responses were numbers between 1 and 5, where 5 meant strongly agree, 4 meant agree, 3 meant neutral, 2 meant disagree, and 1 meant strongly disagree.			

Over the same academic years between 2010-11 and 2012-13, the student comments applicable to the project and the actions taken are provided in Table III.

Project success based on student performance on the project: Table IV and Figure 5 show, for each of three academic years, the number of student teams that successfully demonstrated only one of the six milestones, the number that demonstrated two of the six, etc., up to the number that demonstrated all six of the milestones. The performance indicated in Table IV and Figure 5 shows that approximately 67.7 percent of the student teams successfully demonstrated at least five of the six milestones in 2011-12 and 2012-13, and that percentage improved to 76.9% in 2013-14. Those student teams having very poor performance, that is demonstrating three or fewer of the expected project milestones, were relatively small in actual numbers, making it difficult to analyze trends. For example, there were three such teams in 2013-14 and they consisted of eight of the 33 students in the course that year. In 2011-12 and 2012-13, there were three teams and four teams in that category, respectively, and those teams consisted of seven of the 51 students in 2011-12, and six of the 44 students in 2012-13.

Table III. Networking course student comments from surveys and actions taken

Comment from student:	Action taken:
2010-11: The wireless optical signaling for the course project did not work reliably.	The optical signaling format has not been used since 2010-11, and will be used again only if transducers that work reliably are first identified and are tested to be reliable prior to adopting their use.
2010-11: There were too many things expected in the lab with both the project and the other lab experiments.	One of the four “other lab experiments” that had been included in the past was changed to an optional exercise for extra credit.
2011-12: The project was, by far, the most work in the class and was only worth 14.4% of the overall grade. I think it should be worth 25% or 30% . . .	The weighting for the course project was increased to 22.22%.
2011-12: The standards committee should be removed, as it was too much of a hassle. . . . Furthermore, the standard needs to be finalized by milestone 2 so we can build our labs to it rather than try and modify them later to fit the standard because it may require rewrites. . .	Beginning in 2012-13, the standards committee that had included one student representative from each team has no longer been used. An instructor-written standard has instead been distributed to students and then modified when warranted by comments from the student team(s).
2012-13: Give standards at the beginning of the project.	The first draft of the standard, which had been distributed to students between Week-3 and Week-4 in 2012-13 (and had some minor updates as late as Week-8 of that 2012-13 term) was distributed to students one week earlier in 2013-14, and had no updates made or requested throughout the term in 2013-14.
2012-13: The course project material was not well laid out for us and descriptions of the milestones were not very informative.	Prior to 2013-14, the only written documentation of each project milestone expectation was a brief description similar to that in Table I of this paper. Beginning in 2013-14, in addition to these brief descriptions of each milestone, a more detailed description of the milestone expectations was written and distributed to the students for most of the milestones.

Table IV. Project success: Milestone-demonstration completion rates

Academic Year	Number of student teams that successfully completed a specified number of the 6 expected milestone demonstrations						
	1 completed	2 completed	3 completed	4 completed	5 completed	5.5 completed	6 completed
2011-12	0/21	2/21	1/21	4/21	2/21	7/21	5/21
2012-13	1/21	0/21	3/21	3/21	1/21	4/21	9/21
2013-14	1/13	0/13	2/13	0/14	1/13	2/13	7/13

Note: The average number of milestones completed for the teams in 2011-12 was 4.83, in 2012-13 was 4.90, and in 2013-14 was 5.00.

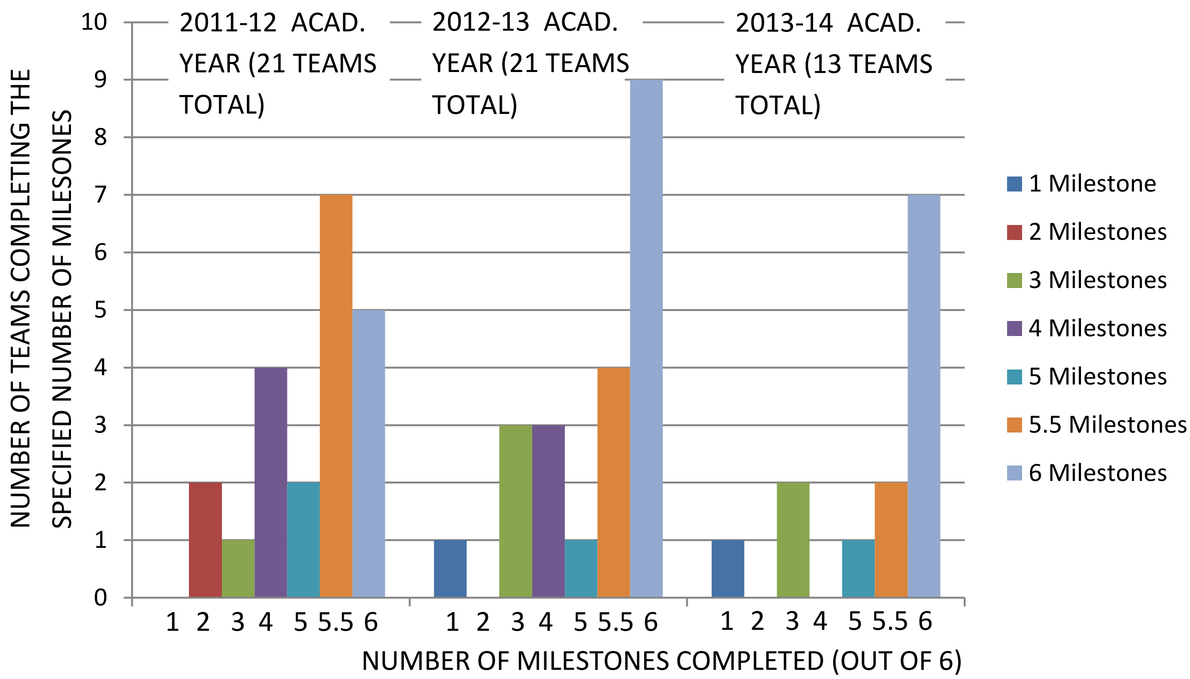

Figure 5. Project success: Milestone-demonstration completion rates

Conclusion

A networking course project was developed and has been incorporated successfully into a computer networks course that focuses on the lower layer protocols of a network. The project has incorporated innovative techniques for collision detection. Each year, the project has used an interoperability standard that was in earlier course offerings developed by students, but has more recently been developed by the faculty so that a stable standard is available at an earlier time during the course. The bus network topology has been used consistently due to its accommodation of Ethernet-like protocols. However, in the future, a course networking project that uses a ring topology might be tried. Although wireless optical media have been used in some previous years, the additional complexity and logistical difficulties in setting up the wireless optical network outweighed the positive aspects of having implemented a wireless network.

Course time constraints have so far prevented the inclusion of any network applications more complex than simple message transfers, and have precluded the inclusion of data link control protocols. However, alternative applications and the inclusion of a data link control protocol involving ACKs and possibly NAKs are features that could be included in a future offering of the course. Project experiences and feedback from students have contributed to changes in the networking course project. The overall success rate for teams measured in terms of the number of expected project milestones successfully demonstrated, has improved slightly over the past few years.

Bibliography

[1] Jianyo Dong and Huiping Guo, “Enhance Computer Network Curriculum Using Collaborative Project-Based Learning,” 2011 ASEE Annual Conference.

[2] Mohammad B. Dadfar and Sub Ramakrishnan, “System Projects for a Computer Science Course,” 2003 ASEE Annual Conference.

[3] Jianyo Dong and Huiping Guo, “Effective Collaborative Inquiry-based Learning in Undergraduate Computer Networking Curriculum,” 2013 ASEE Annual Conference.

[4] Maurice F. Aburdene, Thomas J. Goodman, “Probability, Computer Networks, and Simulation,” 2005 ASEE Annual Conference.