E90 Proposal: Shuttle Tracker

Nick Guerette

December 1, 2004

Abstract

I propose the implementation of a system using long-range RFID and ethernet/IP network communication to track the location of Swarthmore College shuttle vans, in order to reduce student frustration with shuttle unreliability and save time for Public Safety officers. I began work on this project during the summer of 2004, and will complete this work during the Spring semester of 2005 at a cost of $200.

1 Introduction

Swarthmore College runs two shuttle vans each at twenty-minute intervals every evening to safely transport students to and from off-campus dormitories. It is common for the shuttles not to run on schedule, and this causes students who are waiting for a shuttle to get frustrated, and often call the Public Safety department to inquire about the shuttles' current locations. The purpose of this project is to address that problem, and both reduce students' frustration and save time for public safety officers by implementing a system to track the location of the shuttle vans.

Over the summer, with Professor Erik Cheever as my adviser, I designed and built prototype hardware for a such a system, which I dubbed the Shuttle Tracker. This system uses long-range radio frequency (RF) identification to detect when shuttles are at their stops, and distributes that information to display devices, a webpage, and client programs over an ethernet/Internet Protocol (IP) network. In this report I first describe all the parts of the system, the work that has already been done on them, and what work remains to be done. Following that, I lay out a plan and timeline for completing and integrating each part of the project. Finally, I state the skills which qualify me for carrying out the work described, and predict the costs in time and money for completing this work.

2 Technical Overview

2.1 High-Level Description

The Shuttle Tracker is divided into four main functional components:

- RF transmitters installed in each van
- RF receivers located at each shuttle stop
- a computer acting as a central server
- display clients reporting on the shuttles' locations which report the most current available information about the shuttles' locations, and may take the form of dedicated wall-mounted displays, a web server, or programs to be installed on personal computers.

The transmitter installed in each van which will periodically transmit a short string of serial data, including an ID number unique to each van. The receiver at each shuttle stop will detect transmissions from nearby vans and measure the amplitude of the carrier, and when a transmission is received and the carrier amplitude is above a threshold, will send a User Datagram Protocol (UDP) datagram to the central server reporting the identity of the van from which the transmission originated, at what time the transmission was received, and a measure of the relative amplitude of the transmission carrier. The server will log this information and send UDP datagrams out to the display clients that it is aware of, indicating the new information about the location of the van in question. The server will also maintain the current status (last known locations) for all shuttles, and provide this information to any display client which is newly coming online or which sends a request for the status of all shuttles.

2.2 Long-Range RFID Considerations

The task of identifying a van based on a short string of serial data transmitted over an RF link reliably while avoiding false positive identifications requires overcoming obstacles including random receiver noise, overlapping transmissions, and transmissions emanating from sources other than a Shuttle Tracker van transmitter. Most work on this task was completed during the period of May through August 2004.

2.2.1 Choice of RF Modules

I decided to use RF modules designed for transmitting serial data, of which there are many on the market. I settled on the Linx LR series due to the fact that this model has a stated range of up to 3000 feet, and that the receiver has an analog output that indicates the amplitude of the carrier and can be used to provide a relative indication of van proximity. At the time of beginning to work on this project, no LR series transmitters were available from distributors. Since the documentation claimed that the long range of the LR series was mostly due to receiver sensitivity, I chose to use the compatible Linx LC series transmitter modules.

2.2.2 Random Receiver Noise

When there is no carrier present, the receiver module outputs random data. In order to avoid detecting spurious valid data, the presence of a shuttle will only be acknowledged when the carrier amplitude is above a threshold. The threshold will be a value between the median of daily minimum amplitudes during the previous seven days and the median of daily maximum amplitudes during the previous seven days.

2.2.3 Encoding Method

I found several vendors that manufacture specialized hardware for encoding and decoding data or button presses to be sent over a serial wireless link, but to minimize cost and hardware complexity I decided to use the universal asynchronous receiver transmitter (UART) modules built-in to the microcontrollers in the receiver and transmitter hardware. A difficulty with this method is that while a standard UART is inactive, its output is high, and because the RF modules being used use simple on-off keying, simply connecting the output of a UART to a Linx LR or LC transmitter would cause the carrier to be constantly present when data is not being transmitted. This would interfere with transmissions from any other transmitter, and whenever more than one van is present at a stop, none of the vans would be able to be detected.

In order to deal with this problem, the transmitter microcontroller's UART is disabled while data is not being transmitted, and the output to the RF transmitter is set low to disable carrier generation. When the transmitter microcontroller is ready to send identification data, it enables its UART to set the output to the RF transmitter high so that the RF transmitter begins to generate a carrier. It waits in this state for the period of time necessary to transmit one byte, in order to assure the receiver senses a stop bit from any data it had been reading out of random noise and is ready to receive a new byte. The transmitter microcontroller then sends two bytes over its UART. The first byte is the same for all transmitters and identifies the transmission as emanating from a Shuttle Tracker transmitter. The second byte is an identification for the van.

2.2.4 Overlapping Transmissions

The Shuttle Tracker needs to be able to deal with multiple vans at a single stop, but if two van transmitters are transmitting simultaneously, they will interfere with each other and neither will be detected. In order to avoid this problem, each transmitter will transmit at different intervals ranging from approximately two to four seconds, with each transmission lasting 13 milliseconds (in the prototypical configuration, with a data rate of 2400 baud; I plan to experiment with higher data rates and more robust error-checking methods that may require sending more data in each transmission). In order to accomplish this, the transmitter's ID, which is stored as an 8-bit unsigned integer, will be bit-reversed, and the resulting number used to calculate the length of time beyond the minimum of approximately two seconds between transmissions. Using this method, transmitters that are numbered sequentially starting at 1 will have quite different intervals between transmissions until the number of transmitters in the system becomes large. The result is that if two transmissions overlap once, they are guaranteed not to overlap again for some time.

2.3 Ethernet/IP/UDP Communication Considerations

The receivers will use a connection to an ethernet/IP network to notify the server of vans nearby, and the server will use the ethernet/IP network to distribute van location information to display devices. Both will use the connectionless transport-layer protocol UDP. The only configuration that will be required for the receivers and display clients to find the server on the network will be the server's Domain Name Service (DNS) name (i.e., "shuttle.swarthmore.edu"). The dedicated receivers and display clients will obtain network configuration information using Dynamic Host Configuration Protocol (DHCP), and use the DNS name of the server to obtain its IP address.

2.3.1 Choice of Embedded Ethernet Hardware

There are several vendors that sell easily programmed embedded ethernet communication devices designed for use with microcontroller-controlled systems. However, to save costs and also to give me the opportunity to explore low-level network programming, I decided to simply use an ethernet controller like those found on computer ethernet cards which has an 8-bit mode allowing it to be driven by a microcontroller. Several examples of microcontroller programs for operating two common such ethernet cards, the Realtek RTL8019AS and the Crystal CS8900, are available.

A problem with this approach, though, is that the two ethernet controller integrated circuits mentioned above are both in surface-mount packages, and proper facilities are not available to me for mounting surface-mount IC's on printed circuit boards (PCB's). Also, it would be an extra cost to have PCB's made for mounting surface-mount IC's. Because of this, several small companies sell network interface cards designed for microcontrollers that consist simply of an ethernet controller and ethernet jack with supporting components mounted on a small PCB that is intended to connect to another PCB using headers. I chose to use such a network interface card from EDTP Electronics, the "Packet Whacker". Were the receiver and display devices to be mass manufactured, however, it would be less expensive and more efficient to purchase ethernet controllers and jacks and their supporting components separately and mount them directly on the device PCB's.

2.4 Transmitter Hardware

The Shuttle Tracker van transmitters will contain a PIC16LF873A microcontroller and a Linx LC series RF transmitter module, will be powered through a 12 volt automotive “cigarette lighter” plug, and will have a simple wire antenna. I chose the PIC16LF873A microcontroller due to my existing familiarity and satisfaction with 16F87XA series of PIC microcontrollers and the availability of these microcontrollers in the Swarthmore Engineering Department for development. The 16LF873A is a low-power-consumption version of the least expensive microcontroller in this series, and is suitable for use in controlling the van transmitters. Using it in its low-power mode with a slow oscillator frequency (38.4 KHz) along with a low-quiescent-current voltage regulator allows the transmitter device to draw a total average current of much less than 1 mA, so that it can be left powered from a vehicle battery without fear of significantly draining the battery.

2.5 Receiver/Display Hardware

Installed at each of the main shuttle stops will be a device that incorporates both the receiver and display client functionality. It will contain a PIC16F877A microcontroller, a “Packet Whacker” network interface card from EDTP Electronics, a Linx LR series RF receiver, and a light emitting diode (LED) display showing the current time as well as the time and location of the last transmission seen from each of two shuttle vans. The PIC16F877A microcontroller was chosen for, again, my familiarity and satisfaction with the 16F87XA series of PIC microcontrollers. Also, the variety of peripheral modules built into the 16F877A, including a UART, synchronous serial port, and analog to digital converter, which I’m using for, respectively, RF identification of shuttle vans, communicating with LED display drivers, and reading received signal strengths from the RF module, allow for simpler circuitry and cost savings. An LED display was chosen over a liquid crystal display (LCD) to allow the display to be quickly and easily read from a distance of 10-15 feet under varying lighting conditions, despite that the requirements in power and circuit complexity are greater for an LED display.

2.6 Server Software

The server software will be written in Python, since this allows for rapid software development, and for the final program to be run on any computer for which a Python interpreter is available, which includes all commonly used architectures and operating systems.

2.7 Web Page

A web page will be constructed which shows the current location of the shuttle, and uses Javascript to update automatically and provide the option of pop-up alerts for shuttle arrivals or departures. An html-only version of the webpage that is free of client-side scripting will also be available.

2.8 Design Constraints

2.8.1 Economic

In desiging and choosing components for the transmitter and receiver/display hardware, much care was given to making the devices inexpensive enough to produce so that with the $500 equipment portion of the Richard Hurd grant and the $200 E90 budget, one transmitter could be installed in each van and one receiver/display could be installed in all three of the buildings at which the shuttle normally stops with an extra one available that could possibly be installed in the Public Safety building for notification of when a shuttle is waiting there for a driver to arrive.

2.8.2 Ethical

If the Shuttle Tracker is eventually installed permanently, users may become accustomed to having information about the Shuttles' locations readily available, and could thus feel less safe if the system fails, say due to problems with the ethernet/IP network. That situation would, however, in reality be no worse than the current situation.

If the devices involved in the shuttle tracker system were to be mass manufactured, I would want to investigate the suppliers of all components to assure that revenue generated by the product is not supporting unfair labor practices.

2.8.3 Sustainability

The use of resources involved in the Shuttle Tracker is very low, since the hardware is small and simple. However, if the system were to be mass manufactured, I would again want to investigate all component suppliers to assure that resources for manufacturing hardware for the system are not being obtained by means that sacrifice the future availability of resources or well-being of living things.

2.8.4 Political

As I am proposing installing the Shuttle Tracker system permanently at Swarthmore College, the project needs to be presented to authorities at Swarthmore in such a way that it will be clear that permanent installation is worthwhile. Also, shuttle drivers may oppose implementation of the system as an invasion of privacy. Since they are being paid to drive College-owned vans, however, knowing when the vans arrive at and leave each building provides accountability for drivers accomplishing their task. That accountability as well as the inconvenience caused when the shuttles do not run on time to those who expect the shuttle service to be available are good defenses to the invasion of privacy argument.

2.8.5 Aesthetic

The receiver/display devices will be in plain sight in College buildings, and thus need to be visually appealing in a way that will match their surroundings.

2.9 Possible Extensions

2.9.1 Display Client Software

A simple text display client could be written in Python that would run on any computer for which a Python interpreter is available and receive real-time updates from the Shuttle Tracker Server. Also useful for a large portion of possible users would be a display client for the popular Microsoft Windows operating system that displays an icon in the Windows "notification area" or "system tray" indicating the current locations of the shuttles.

2.9.2 Satellite Receivers

Receivers could be built that are simpler than the receiver/display devices described above, and connect to a computer, say a public lab computer, in buildings at which the shuttle does not normally stop, but to which it makes frequent visits. These receivers would be simpler and lower in cost since network communication would be accomplished by the computer.

2.9.3 Prediction of Shuttle Arrivals

The functionality to predict the arrival times of vans at their next stops based on past departure and arrival times and send this information to display clients regularly could be added to the server software. Development of this functionality would require data gathering over a period of several weeks to assure that it is robust.

3 Project Plan

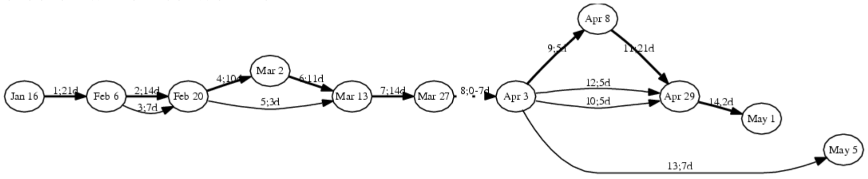
Work on this project was begun during the period of May – August 2004 with funding from the Richard M. Hurd '48 Engineering Student Research Endowment. During this period, the following was accomplished:

- Overall system design was finalized.
- The RF identification subsystem was designed and tested.
- The embedded ethernet hardware was selected and considerations for microcontroller ethernet communication were researched.

Table 1: Activities Summary

Activity	Needs	Feeds	Effort	Duration	CPM Duration
1	–	2,3	75 h	21 d	21 d
2	1	4	25 h	7 d	14
3	1	4,5	25 h	7 d	7 d
4	2,3	6	25 h	7 d	10 d
5	3	7	10 h	3 d	3 d
6	4	7	3 h	11 d	11 d
7	5,6	8	10 h	14 d	14 d
8	7	9,10,11,12,13	0–25 h	0–7 d	0–7 d
9	8	11	3 h	5 d	5 d
10	8	14	15 h	5 d	5 d
11	8	14	60 h	15 d	21 d
12	8	14	15 h	5 d	5 d
13	8	–	40 h	6 d	7 d
14	10,11,12	–	6 h	1 d	1 d

Figure 1: Critical Path Diagram. Durations of some activities were extended so this diagram would make sense, but since all work is to be done by the same person, in fact progress on all tasks is critical. Progress cannot be made on more tan one task in parallel, but when there are two parallel tasks there is a choice of which to work on.



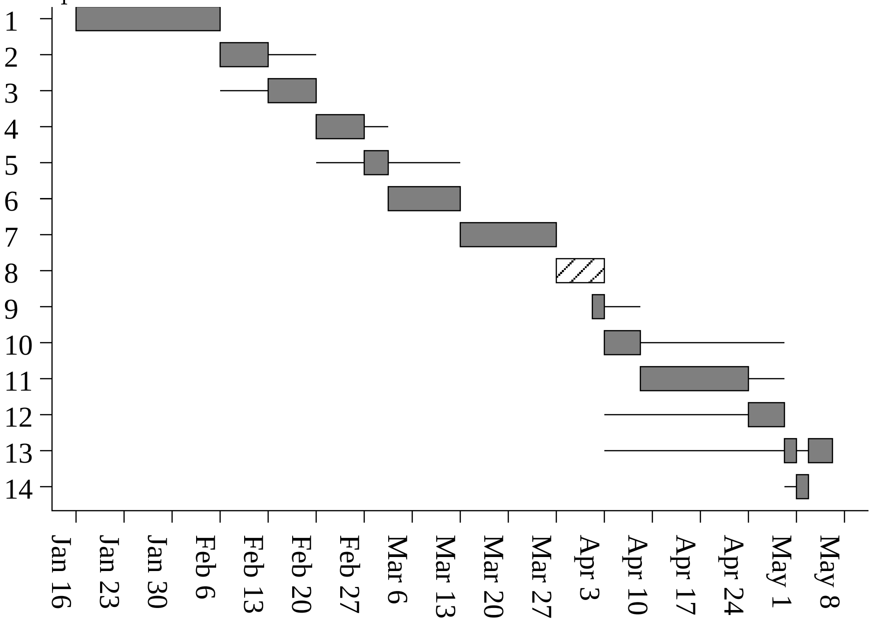
- The transmitter and receiver/display devices were designed.
- Prototype transmitter and receiver/display devices were built, but have not been tested.

The tasks remaining to be accomplished are enumerated in Table 1 and explained in the remainder of this section. Figure 1 shows task dependencies, and Figure 2 shows when I intend to accomplish each task. Durations were calculated based on spending three hours each weekday and ten hours each weekend working on this project, for a total of 25 hours per week.

3.1 Microcontroller Ethernet/IP/UDP Communication

PIC routines to send and receive data using the Realtek RTL8019AS ethernet controller on the Packet Whacker network interface card will be written. Numerous examples of such routines are available. A factory-made

Figure 2: Modified Gantt Chart. Boxes indicate time I actually intend to work on each task, and lines indicate time to which work on each task could be moved due to parallelism of tasks.



demonstration board from PIC manufacturer Microchip with a PIC16F877 microcontroller and a Realtek RTL8019AS ethernet controller is also available for testing.

3.2 Test Prototype Transmitter and Receiver/Display Hardware

The prototype transmitter and receiver/display devices that have been built will be tested for RF communication functionality, ethernet communication functionality, and display functionality and debugged.

3.3 Write Server Software

The Shuttle Tracker server software will be written once the microcontroller ethernet/IP/UDP communication is functioning. The method of receivers and display clients automatically discovering the server on the network using DNS lookup also will be tested at this point.

3.4 Lab Test, Calibrate RFID Subsystem

The method devised of robustly detecting the presence of a shuttle at a stop will be tested under simulated real-world circumstances with the prototype hardware, and the median filtering of carrier amplitude minima and maxima and carrier amplitude thresholding algorithms need to be tuned. Having a receiver/display device with functioning ethernet/IP/UDP communication abilities and LED display will aid in the collection of data involved in this step.

3.5 Create Web Page

A web page will be created that displays information from the Shuttle Tracker server about when the shuttle arrives at and leaves Parrish Hall, as well as general information about the Shuttle Tracker system to the Swarthmore College community.

3.6 Preliminary Testing

With the transmitter and receiver/display device prototypes functioning and communicating on the ethernet/IP network and the server software installed on a computer on the network and tested, a preliminary real-world test of the system will be performed. This will involve temporarily installing the prototype transmitter in one of the shuttle vans and temporarily installing the prototype receiver/display in Parrish Hall. Operation of the system will be observed through data gathered by the Shuttle Tracker server.

3.7 Obtain Community Feedback

The existence of the Shuttle Tracker system and the web page will be announced to the Swarthmore College community. Feedback will be elicited from the community about the Shuttle Tracker system.

3.8 Revise System Based on Feedback

Based on feedback from the College community, changes to the hardware and web page will be considered.

3.9 Order Parts for Final Hardware

Parts will be ordered for the final versions of transmitter and receiver/display hardware.

3.10 Plan for Ultimate Installation

Permanent installation of the Shuttle Tracker system will be proposed to relevant persons and departments at Swarthmore College, including Facilities, Information Technology Services, and Public Safety. If permanent installation proves feasible, installation will be scheduled, the costs of installation will be examined and plans for any future system maintenance will be made.

3.11 Construct Final Hardware

Purchase hardware for and construct final transmitter and receiver/display devices.

3.12 Update Web Page

The web display client will be revised, considering feedback obtained from the College community.

3.13 Write Documentation

Thorough documentation will be written that would allow others to perform repairs, maintenance or reinstallation on the Shuttle Tracker components.

3.14 Perform Installation

The final transmitters will be installed in the shuttle vans and the final receiver/display devices will be installed at the shuttle stop locations.

3.15 Expand System

If there is time and money available, build satellite receivers and write a Windows display client.

4 Qualifications

I am a double major in Engineering and Computer Science at Swarthmore College, and lived in the off-campus Mary Lyon dormitory for two years, during which time I almost never tried to use the shuttle due to its unreliability. The facilities necessary to complete this project include a basic electronics workbench, available for my use in the Swarthmore College Engineering Department.

5 Project Costs

Most of the parts for four receiver/display devices and two transmitters have already been purchased. The remaining parts and printed circuit boards can be purchased for approximately $200. A detailed report of costs of materials is attached, with quantities already purchased highlighted. Approximately one hour of time of assistance from a member of the Facilities Department will be necessary to determine a method of mounting the transmitters in the shuttle vans, and approximately four hours of assistance from a member of the Facilities Department will be necessary for installing receiver/display devices in Parrish Hall, Palmer Hall, the Mary Lyon dormitory, and the Benjamin West house. I estimate this project will take 375 hours of my time.