Building the CU Community Wireless System

Sponsored by:

Professor Ralph Johnson

Project Team:

Title	Name	Email
Website Admin	Ryan Calvert	rmcalver@uiuc.edu
Project Manager	Aaron Hays	aaronhays2000@yahoo.com
Technical Expert	Darin Riggs	riggsdr@hotmail.com
Client Contact	Aaron Tubbs	atubbs@uiuc.edu

Table of Contents

Introduction

Executive Summary

The purpose of this document is to provide a comprehensive overview of the computer science senior project assigned to Team 2 for the Fall 2002 thru Spring 2003 school year. It will explain the nature of the project, the need for a solution, the research done to prepare for a solution, the work that was accomplished, and what the future holds after our project comes to a conclusion. In addition, special attention will be paid to the challenges encountered during the course of the project, as well as the lessons learned in working through these problems.

In short, our project aims facilitate a community wireless network for the city of Champaign. The initial network will be developed along Green Street, and will enable community members to share resources such as the Internet among themselves.

Document Organization

The document begins with the background of the project, and a description of the problems the project aims to solve. It explores the underlying vision that motivates our research, design, and decisions along the way, as well as several alternatives that were used for inspiration. Next comes a more careful discussion and analysis of our requirements as we solve the problem, followed by a description of our solution.

Specifically, the technologies and methodologies that we have been assigned to use along with those we have selected to use are explored in detail. Attention is paid to the difficulties we encountered during the duration of the project. The document concludes with a log of our activities and schedules, and some considerations for the future. Throughout the document, we have attempted to provide footnotes with links to additional information where it may be helpful.

Project Description

Loosely put, the purpose of this project is to develop the technological infrastructure to support a wireless network to serve the residents of Champaign-Urbana. Unlike many small hotspots created by commercial entities, this network aims to provide coverage to all of the Champaign-Urbana community, and do so via community-built and supported systems. Instead of merely servicing a few coffee-drinking customers at a trendy café, the network aims to service all users, from customers in restaurants to local residents sitting in the comfort of their own homes. Indeed, these home users will provide the foundation for the network, both in terms of technology and community support.

The project will be initiated with a prototype network along Green Street, starting near Prospect and extending east towards Neil. While a relatively small area in terms of coverage, this prototype will be fully functional, and serve as a base from which to grow the community network to reach all of the Champaign-Urbana community. The concepts developed and proven at the smaller scale will be used to grow the network into something that can be appreciated by everybody.

It is important to note that the CU Community Wireless System is hardly the first attempt at creating a wireless network to share resources and provide enhanced means of communication within a community. In fact, the inspiration for our project comes directly from the CU Grassroots Wireless Internet Project[1]. Both of these projects share similar goals, and our intent is for the two systems to cooperate as both expand throughout the area. In addition to interoperation, it is possible to draw resources in terms of hardware, software, and experience from this group in order to advance our goals. The collaboration between the two groups will provide a foundation for the future of wireless connectivity in the area.

There are also additional influences beyond those in the local community. Large established community wireless networks and organizations provide an additional resource for advice, experience, and ideas. Some examples of these resources include the Seattle Wireless[2] organization, the Bay Area Wireless Users Group[3], and Freewan.org.[4] Each of these organizations has an established presence both within their communities and online, and through examining their web sites and collaborating with their members, we will save a great deal of time and prevent as many setbacks as possible.

---

[1] http://wireless.cu.groogroo.com/
[2] http://www.seattlewireless.net/

[3] http://www.bawug.org/
[4] http://www.freewan.org/

The Vision

The CU Community Wireless System is driven by a vision of the future, in which a robust ubiquitous wireless network links the community. Connecting to the network is inexpensive, and the access provided is free and unrestricted. Businesses can offer connectivity to their employees and clients, and residents can surf the Internet for a reasonable cost. Computer buffs can hack at the protocols and have a place to play with a wireless network. Schools and libraries can offer electronic resources for less money than is currently possible. Mobile users can remain connected from their backyards, the community pool or a park bench. More than a simple system to interconnect neighbors, the goal is to provide a framework allowing the sharing of any type of resources and communication within the community without boundaries. In short, the vision is connectivity and resource sharing for everyone, anytime and anywhere.

Alternatives

Before we began, and during the course of our project, we encountered and considered many alternatives to our approach. Our sponsor, Professor Ralph Johnson, provided us with the requirement to utilize the NetBSD operating system as our software platform. For hardware, he supplied us with old commodity computers discarded from the computer science department. He also provided us with Linksys wireless cards and semi-directional patch antennas.

These provisions and requirements limited our ability to directly draw experience and software from some of the larger wireless organizations. Luckily, there are enough similarities in requirements between our project and the CU Grassroots Wireless Internet Project to allow us to directly borrow some of the software and configurations that they have already developed. However, the information and experience provided by wireless community networks like the Seattle Wireless network and the Bay Area Wireless User Group remain invaluable as a resource. In the event of difficulty, or when a decision about how to solve a problem has to be made, leveraging the past experience and present advice of these groups is critical to efficiently moving forward.

A more detailed explanation of some of the alternatives considered and examined during the course of the project is provided later in this document, in the discussion of plans for the future of our network.

Analysis of Requirements

Introduction

The analysis of the requirements for this project has been one of the most difficult aspects – both because the gathering of requirements is an inherently difficult process in any situation and because of the unique nature of this project. Specifically, unlike most senior projects, this project is not centered on software development. Instead, its focus is on hardware and software configuration, specifically the creation and configuration of computers, the installation of antennas, and the configuration of software to support a wireless network. Through discussion with our sponsor, with members of the CU Grassroots Wireless Internet Project, and the staff of the class, we have been able to isolate the requirements for our project. With these requirements in mind, we will be able to design solutions to fulfill them.

The Problem

The CU Community Wireless System solves the problem of providing a way to inexpensively and effectively connect and facilitate communication between Champaign-Urbana community members. This problem is confronted through realizing the aforementioned vision, providing access to resources that are currently unavailable, as well as facilitating new and improved means of accessing resources previously accessible through less effective channels.

The Client

One of the challenges has been to identify our client. Initially it would seem as though the network’s end user is our client. However, while this might seem the obvious choice, we believe that there is a better answer. Eventually, the end users of the network will hopefully be average citizens of Champaign-Urbana; people who currently have no idea what we are doing. It is difficult, if not impossible, to have a client who doesn't know they are a client. Instead, there is a group of people who are committed to seeing this network grow. We don't yet know them by name, but we do know that Ralph Johnson represents them and provides us with a point of contact. This group of community members dedicated to the growth and improvement of the network in the long run is our client. We are responsible for helping them to overcome the technological hurdles that will be encountered in initially creating and building this network by providing them with a prototype network.

Goals of the CU Community Wireless System

To provide a concise purpose and focus to our project, we have identified the following goals for our network. Namely, the network should:

- Allow communication between users
- Allow sharing of resources by users
    - Local network access
    - Broadband Internet connections

    - Users should be able to control the level of resource sharing
- Be robust
    - No single node failing should bring the entire network down
        - Segmentation that results from node failure should be automatically reintegrated when the node(s) come back online
    - Hardware should be chosen to minimize the chance of failure
    - Software should be configured to minimize the chance of failure and maximize the ability of the network to recover from problems
- Be expandable
    - New nodes should be automatically discovered and integrated into the network
    - No changes should have to be made to existing nodes on the network
- Have acceptable performance
    - Without reasonable performance, the network will never be used

Benefits

Providing the technological infrastructure to support a community wireless network has the obvious advantage of making it possible to build such a network. With this infrastructure in place, it becomes possible to enhance electronic communication between users, provide the ability to share resources such as inexpensive broadband Internet access, and supports the possibilities of expanding into as of yet unrealized technological frontiers in the future. Much like bringing the telephone to a new community in years past, we see the creation of a community network with access for all to be a way to significantly enhance communication and collaboration both within the community and beyond its boundaries.

Users of the Network

Currently, we see two major classes of users, which we label as routing users and end users. A routing user provides a component of the wireless network that forwards packets to other portions of the network, whereas the end user only consumes the resources provided by the network. This distinction is important, as the routing user will require additional equipment, as well as a more thorough understanding of how the network works and how they contribute to its support and growth.

The end user of the network is an average person wanting to have access to a growing electronic community of neighbors as well as the possibility of access to inexpensive broadband. The user will need special equipment such as a computer, wireless network card and possibly an antenna, but should not require any special computer knowledge or skills beyond what is necessary to attach to and begin using the network.

User Principles and Constraints

While there will certainly be constraints to the usage of the network, many of the final limitations will be realized much later in the growth and evolution of the project. Still, one of the goals of the project is to identify and minimize the possible constraints that will be faced in the future. By understanding and considering these limitations ahead of time, we can ready our implementation to accommodate them. Doing so can prevent an incompatible solution, or one requiring a great deal of additional work, from developing.

How Will the CU Community Wireless System Be Used?

Effectively, there are as many potential uses for the wireless system as there are potential users. Our goal is to create a wireless network that can support as wide a user base as possible. While it is impossible to guess all of the long-term uses of the network, we have tried to enumerate some of the most common usage patterns in the following section, by enumerating the types of users we envision to be most common.

Who Will Use the Network?

The Surfer

By far the most typical user, the Surfer will be a resident wanting to be able to browse the web, check email, and communicate with friends via instant messaging. These services should be provided at speeds faster than those experienced via dial-up connections and at a much lower cost. To reach these goals, the user is willing to make the initial investment to connect to and utilize the network by purchasing a wireless networking card and an antenna and installing it in their home computer. To remain effective, the network must provide this user access that remains inexpensive and reliable, or else there will be no incentive to switch.

The Traveler

The Traveler is a visitor, typically on business, to Champaign-Urbana who reads about the wireless network and is intrigued. A vendor rents him a wireless card for his laptop and internet access for the duration of his visit so that he is able to access all the resources his company offers online and stay in touch via email at all times during his visit. He is not limited to dialing in from his hotel room at night. He can remain connected while working at the client site, enjoying a cup of coffee at a café, or relaxing by the hotel pool. Because the Traveler will be paying a larger fee up front and will only be accessing the system for a limited amount of time, it is of the utmost importance that this user is provided with access that is reliable, or else it becomes extremely unlikely that business users will continue to pay for the service.

The Computer Enthusiast

The computer enthusiast likes to play with computers, and she wants more than the mere ability to surf the Internet and participate in the more typical activities of the Surfer. With the intent of providing infrastructure to actually support and participate in the wireless network, she buys or resurrects an old, clunky PC to be a router, buys a couple of wireless cards and antennas, and downloads our ISO image for a diskless router. Without needing a great deal of technical expertise, she transcends the boundary of a normal end user and becomes a routing participant on our network. She connects her new router to her home network and she is surfing the net from all her machines while at the same time supporting and strengthening the reliability and performance of the network as a whole. Perhaps even this is not enough for the computer enthusiast; she may want to get under the hood. So she starts poking around with her router, checking out the source, playing with the configuration, tweaking, and hacking. Pretty soon, she is going to CU-Wireless group meetings and helping guide the direction of the community network as it grows and matures.

The Home Business

Jane's home business is booming. From the corner of her bedroom, the business has grown to take up the extra bedroom and part of the basement. She has several computers on a network, a couple printers, and a problem – her only Internet access is from a single computer and it is on her fax line. She needs a way to get her whole network on the Internet and keep her fax line free, but she also wants to keep her monthly expenses in check. She hears about the CU-Wireless network and decides to check it out. It sounds like the perfect solution, but Jane is not a technical person; luckily, her next-door neighbor's son is. He set up her current network, did a good job and didn't cost a fortune. She gives him a call, buys an old PC, a wireless NIC, an antenna and a pizza and he grabs an ISO image from the CU-Wireless group website. A few hours later, Jane has an extra computer on her network connecting her to the CU-Wireless network, her fax line is free and she can hit the Internet from anywhere, all for a lower monthly cost than her previous dial-up connection. Because the network is now providing service to local businesses as well as the Traveler, it is again important to focus on making the network as reliable as possible, or else there will be no incentive for the business to use the network compared with more traditional alternatives.

The Hacker

The Hacker is a driven programmer. He is trying to develop a new streaming media format for wireless networks, but has only had a small local wireless LAN to test on. He hears about the CU-Wireless network and sets up a node (probably as a routing user). Now, he can collaborate very easily with his friends across town who are also working on the protocol and he has a realistic test network to push content across. Working diligently, he and his friends get into the beta stage and start pushing content. A local group of media enthusiasts hears about the project and use the new protocol to broadcast a weekly local news and entertainment show. By joining the network, he is able to develop new software and resource for the community, and will also possess the technical expertise to assist in the most demanding and difficult problems related to network design, growth, and software development.

User Summary

These are just a few of the many possible uses for the wireless network; from simple internet access to a test bed for the development of new technologies, the network should serve the entire community. It is important that each of these users and their needs are considered while making design decisions for the project. While there are likely many users we have not yet considered in the above categories, it is essential that at least these base users be supported. Without each user type, the network will be crippled and future growth will be stifled.

Architecture

We have made several high-level architectural decisions during the course of the project. Each node on the network consists of a computer with a wireless card configured to communicate with the rest of the network. In addition, some nodes are configured for the aforementioned routing users to route and forward packets between nodes that cannot directly see each other due to wireless range constraints. Routing nodes will have multiple radios operating on different channels and subnets to connect a group of end users to the network. Unlike projects such as LocustWorld, in which every user of the network helps to propagate the signal, only systems that participate as routing nodes will actually provide the infrastructure for the network. An end user with a laptop will not provide any additional integrity or resources for the network.

IP address assignment will ideally be dynamic, but we are unsure that a totally dynamic system is achievable within our time and resource constraints. An alternative has been suggested in which our routing nodes are statically addressed, but our end user nodes are dynamically addressed and allocated. Other completely dynamic solutions are being considered as well. More concrete decisions on this topic will be considered as the project progresses past its current state of development.

More detailed discussion of the technical and architectural aspects of this design are considered in the next portion of the document.

Design

Introduction

The design of our project is a critical portion of the project, as it provides a specific focus and outline for what will be done before we begin any of the actual implementation. We are looking for four things in our design: reliability, flexibility, expandability, and ease of use. At each stage of the design, we consider the goals and vision outlined in the previous sections as we make decisions about how the network will be created in the context of these critical points.

NB: Due to the nature of our project, this design section is far shorter than for a traditional software development project. Our hardware and software considerations make up a far larger portion of our actual design when discussing a network, as opposed to an application.

Reliability

The attractiveness of any networking solution is diminished if it is not reliable. Unreliable networks are almost as bad as no network at all because they cannot be counted on for critical applications, such as submitting homework. Presented with an unreliable network and a costlier, reliable one, most consumers will gladly pay the premium for reliability. Our design addresses reliability in two ways.

The first is inherent in our operating system. NetBSD is an operating system designed for reliability. Crashes are rare, and can usually be rectified with a simple reset. Additionally, depending on the nature of the crash, it may be possible to initiate a reset from a remote location. We also make a goal of minimizing crashes related to hardware problems, by attempting to secure reliable hardware and reduce the points of failure. One example of this is our intent to use CD-ROM discs to boot the computers, rather than relying on hard drives, which are prone to failure and data corruption. The computers will initially boot off of their CD-ROM drives, and then the system will be loaded entirely into memory, such that the only moving parts that remain active during normal operation are the fans to cool the processor and power supplies.

Secondly, the design of our wireless and network infrastructures, which will be discussed later, has been constructed to overcome the loss of a single computer for any reason. All computers within range of each other can implement a network, regardless of the number of computers. Only when a computer linking two disjoint network segments goes down is the network as a whole affected. In this case, both network segments remain operating, but are no longer connected. The network will be configured, in the event of such a segmentation, to repair itself as soon as possible.

Flexibility

For many reasons, the design of our system must remain flexible. Primarily, geography must be considered. Because of a 2.4GHz signal's poor ability to propagate through foliage and other obstacles, growing the network in a certain direction may simply not be feasible. The nodes are designed so the signal can be sent and received in any direction, even all directions. In fact, to reach a desired effect, a node can be designed to take anywhere from 1 to 3 cards in order to propagate the network in multiple directions. Thus, by adding nodes, one can simply go around difficult obstacles.

Additionally, by using a full operating system, actual computers, and common software, our nodes themselves remain flexible. As technology advances, our system can be upgraded instead of wholly replaced. This saves both downtime and cost when implementation of new technology becomes desirable. If a new type of hardware is released, or if new technologies in the wireless arena make our current system obsolete, it should be a natural transition to update the network the new standards. As a side note, the network we are creating is totally compatible with standard Ethernet. Thus, it is also flexible in its use, as it forms a system completely interoperable with already deployed networks and standards.

Expandability

The broad vision for this project is that some day it may become very large. This may in fact be the most difficult aspect of the project. For a user to reach a distant resource on the network, packets must be transmitted through interconnected nodes. At each node, the packet must be disassembled to discover its destination, which must then be cross-referenced with the local routing table to determine the appropriate interface to be forwarded on. Once this is complete, the retransmission can occur and the process is repeated until the packet reaches its destination. Due to the processing time required to handle each packet, this adds to the total time it takes to transmit the packet. In some cases, with the wireless configuration, it may be necessary to retransmit the packet on the same interface it was received on in order to reach the next node. This can cause duplicate packets to clutter the bandwidth, limiting the throughput of the network. Over a large network, this can cause noticeable delays in packet delivery. A routing algorithm must be chosen to minimize the path and the amount of packet re-transmission.

In practice, it may become desirable to utilize some form of "backbone" which extends to distant areas within the network. A higher bandwidth technology, possibly 802.11a, could be implemented on selected nodes, providing a higher-speed path over which to transmit distant packets once they have reached the backbone. However, such considerations are outside the scope of this project; we include them here for future consideration because our project would be compatible with such an implementation. Also, with an expanded network, routing tables grow in size and complexity. It has thus far been impossible it determine if this will have a large influence on the performance of our network. In any case, expandability bears an important consideration on both reliability and usability. Specific to our project, we are trying as much as possible to maintain compatibility with the Champaign-Urbana Wireless Users Group's solution. This is to make easier the hopeful eventuality that our networks will meet and be able to interoperate, thus expanding both networks.

Ease of Use

For the network to gain acceptance, it must be usable not only by those who have helped design it, but also by the average consumer, after minimal instruction.

Fortunately, the technology we are using, 802.11b, is widely available in the commercial market. It has been widely accepted, and compatibility with it has been included in most major operating systems. The design should be such that the consumer's computer is presented with an interface that is indistinguishable from another network, meaning that no additional software must be installed. Though the system is designed for this purpose, we have not yet been able to do extensive testing to examine this issue.

Setup

To accomplish the design goal of our project, each node must be configured with two interfaces, each pointing down opposite directions of Green Street toward the next node. To do this, we will be installing antennas on opposite sides of the customers' houses, so that the signal runs parallel to Green Street. These are to establish the direct-links to the other nodes. The customer then has a number of options. They may connect their Local Area Network directly to the Ethernet port of the node to gain access to the network. Otherwise, they may install a third wireless interface to distribute the signal either inside their home, or to the outside world via an omni-directional antenna mounted on the outside of their house. They can also elect to use some combination of the above options. A client who is not a host to one of the nodes could technically tap into a direct-link signal, but additional configuration would be required on their computer to make useful access to the network.

The computer will establish a determined configuration to talk with the nodes it is aimed at. This includes setting the correct Extended Service Set Identification

(ESSID) and channel number. Based on the CU Grassroots Wireless Internet Project scripts, the interfaces now determine their IP addresses based on the MAC address of the interface card. They are given a reserved address in the 10.x.x.x range set aside by Internet standards. This process is not perfect, as two cards could technically hash to the same IP address because IP addresses have a smaller number of bits than a MAC address. However, this implementation is both pseudo-random and static, and is currently seen as the best solution. Any additional cards are also given an address based on a different scheme that will give them 255 internal addresses to assign to computers accessing the network through them. Client machines on this interface will be given use of a DHCP daemon in order to configure themselves. This is to ease configuration for client machines.

The Zebra software then implements the Open Shortest Path First[5] (OSPF) routing algorithm on the interfaces. This begins compiling a table of all other nodes on a network by listening to traffic and making Address Resolution Protocol[6] (ARP) queries. This protocol will automatically detect when a node goes down, and change its table accordingly. It also can detect which path is the shortest to the destination, and decide accordingly which interface is best suited to the task. There are other protocols being developed which cater directly to other concerns with wireless networking such as signal strength, but as they are still in development stages, we have chosen to wait to utilize them. The machine then activates packet forwarding and waits for packets.

To accomplish the diskless system, the configuration creates a RAM-disk in memory that keeps track of any changes to the file system. This is because it is not

---

[5] http://www.firewall.cx/ospf.php
[6] http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html

possible to write the changes to the CD-ROM that the system was loaded off of. This also lowers the amount of use of the drive, adding to its lifespan.

Additionally, the system loads an SSH[7] daemon to provide remote access. This proves useful for such diverse applications as remote monitoring, remote restarting, and even the temporary changing of the routing algorithm.

This is the basic design of our system. Its purpose is to fulfill our design goals by efficiently passing network traffic to its desired destination. This will allow users on the network to communicate with one another at speeds that are fare greater than that of even DSL or Cable modems. At some point, a bridge to the Internet could even be added to the network to give users access to its content. The best way to implement this would be to establish tunnels through our network to the Internet source. This way, controlled and secure access would be provided. The effects of such an implementation have yet to be studied.

Wireless Networking Subnet Considerations

The specific nature of the Ethernet is implemented in a wireless medium causes some design considerations. Because of the way Ethernet networks are designed, if a host receives a message destined for a different machine that is on the same subnet, it assumes that this message has been delivered. This is because situations that would cause a packet intended for a particular subnet to reach a host would be when that host is participating on a repeated or bridged network. If machine A needs to talk to

[7] http://www.openssh.org/

machine B by passing through machine C and they are all on the same subnet, B will receive the message but not forward it because it assumed C received it as well. Because of this, we will have to carefully allocate IP addresses and subnets, using a long bitmask to prevent conflicts of this nature from developing.

Security

Wireless networks broadcast into a shared medium when they use their microwave signals. Since anybody with wireless networking hardware can capture the packets transferred on a wireless network, 802.11 has provisions for an encryption scheme known as Wired Equivalent Privacy[8] (WEP). Unfortunately, work[9] by Fluhrer, Mantin, and Shamir (Lecture Notes in Computer Science vol. 2259) has demonstrated that there is a weakness in WEP's key scheduling algorithm. Wireless networks using only WEP encryption are effectively just as susceptible to data interception as if they weren't using any sort of encryption.

Because of this, a further layer of security has to be provided by a wireless network if data integrity and information privacy is a concern to the network's users. Our project doesn't focus on security, as our primary concern is to provide the infrastructure to support various services, as opposed to the services themselves. However, in trying to keep our network as flexible and extensible as possible, we met with Madhur Nigam, one of our sponsor's graduate students. Mr. Nigam's thesis focuses on establishing a virtual private network system based on public key encryption

---

[8] http://www.psionteklogix.com/assets/downloadable/80211_Security.pdf
[9] http://citeseer.nj.nec.com/fluhrer01weaknesses.html

and SSH. Specifically, his system is designed to work on the NetBSD platform, and is being designed to be compatible with our and other community wireless networks based on this platform. While the implementation of his additional security layer is beyond the scope of our project, making sure our system would be compatible with his was of the utmost importance.

Mr. Nigam explained the function of his system, which is based on the use of TFTP and BOOTP protocols to distribute keys across the network for automatic configuration of the virtual private network. Assuming that keys can be distributed securely and safely through this mechanism, it will allow users of our network to communicate without any fear that their data might be seen by other members of the community network, or by a random person with wireless networking hardware. Further, this use of a virtual private network could be used in the future to restrict access to the network to only people that should be utilizing the network, as a connection without a virtual private network layer could be restricted from having any rights on the network.

It has already been made evident that further work in this direction will be necessary for the network to thrive. During our second semester, a meeting with Chris Harris, another community member, made it very clear that many users would be extremely uncomfortable with the idea of being part of the community network unless they had a guarantee of security.

Reflector Nodes

Accomplishing the goals and achieving the vision of the wireless network, propagating a signal down the street, and eventually throughout the neighborhoods, requires a certain amount of ingenuity. Many of the wireless networks we surveyed use a single channel to spread coverage over a very large area, by repeating that same signal again and again. In order to work with the existing technologies while achieving the maximum amount of usable bandwidth, we plan to work with a concept that we've dubbed a reflector node.

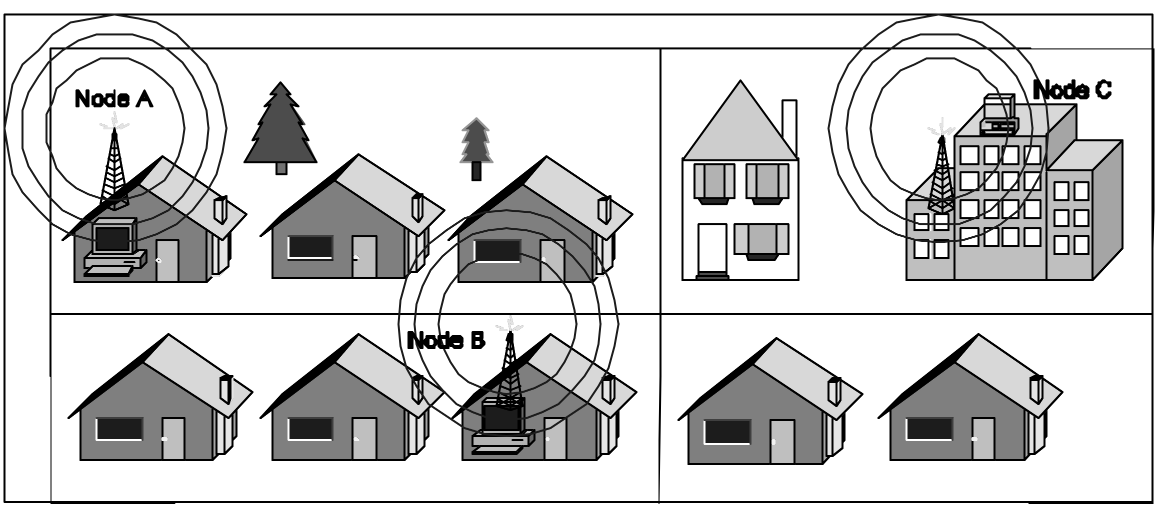
This idea was suggested by our sponsor, with the idea being to use small semi-directional patch antennas on either side of houses along the street, such that the signal arrives at one side of the house, and then it is transmitted out the other side. The big advantage to this approach is that different channels can be used on either side of the house, and that there is not a shared medium all the way down the entire street. As an example of where this can be a big benefit, if two neighbors next door to each other want to transfer data back and forth, more traditional wireless networks would achieve this by having both these neighbors and everybody else on the network sharing the same medium. During normal circumstances this might not be a problem, but during high use, either the data transfer between the houses will be impacted, or the performance of all of the other users on the network will be degraded. In the worst case, all parties would experience noticeable performance degradation.

To achieve this, each reflector node works like any other node computer, but instead of a single omnidirectional or radiating antenna, the reflector node will have two separate wireless interfaces, each using a separate antenna, on opposite sides of the house. While not all nodes on the network will be reflector nodes, utilizing these nodes periodically will isolate segments of the network, and will work better in dense environments where mounting an omnidirectional antenna is impractical. Further, the use of the low-profile flat antennas with these types of nodes makes reflector node installations particularly low impact for homeowners not looking to have a large antenna in a highly visible location on their house.

Some of the challenges that come in this node design are in figuring out where to send signals on the network. Instead of being able to just dump data transmissions not intended for a particular node out on the network, the reflector node has to know which way to send the signal. To facilitate this, the routing tables will have to be carefully configured to prevent against unnecessary delays or data loss.

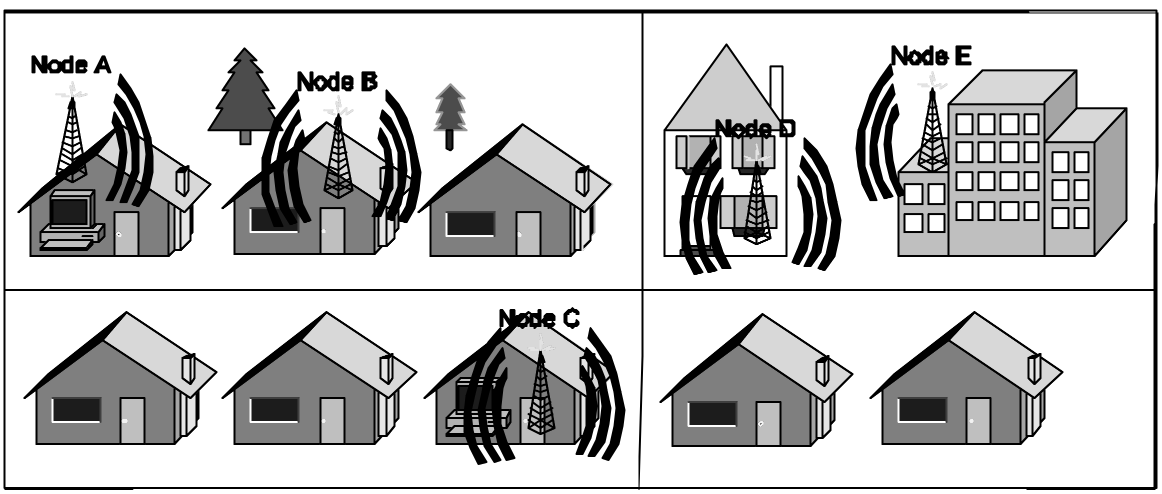
For illustration of the differences between the layout of a more typical community wireless network and our reflector node network, figures 1 and 2 are provided below.

Figure 1: Typical Community Network Topology



In the above diagram, we are looking at an arbitrary neighborhood, provided with a wireless network, in accordance with the grassroots network topology. Each node consists of a computer driving an omnidirectional antenna, and all nodes communicate on the same wireless channel. The advantage to this system is that a small number of antennas and computers serve a relatively large area (this diagram is not necessarily showing the scale of the area that could be covered by this approach). The disadvantage is that, since all nodes are transmitting on the same channel, that the shared medium limits how many nodes can be communicating at any given time.

Figure 2: Our Network Topology



Our network topology is different in that we use semi-directional patch antennas that tend to aim a signal in one particular direction. As can be seen in the above diagram, Node A is sending and a signal with Node B, B does the same with C, and so forth. In the diagram above, nodes B, C, and D are all functioning as reflector nodes, propagating signals across the network from one end to the other. The signal from A to B is on a different channel than the B to C link, which is different from the C to D link, and so forth. This is facilitated by the semi-directional nature of the antennas, which allow us to bounce the signal down the street, rather than radiating it everywhere. With this channel isolation, much greater network utilization should be possible, as there is not a global shared medium for the entire network.

Network Control and Monitoring Tools

At some point, it will become necessary to develop some suite of network control and monitoring tools. This will range from tools that will allow global maintenance, adjustment, and monitoring of the network performance and utilization to utilities to configure how an individual user participates on the network. For example, if a person is sharing an Internet connection, he may want to limit the shared amount to only 25% of his bandwidth. We will either need to find or develop tools to perform such a task. We will also need a way to control who can and cannot use the network if it ever becomes a subscription service. This is covered somewhat in the security section of the paper, but could be considered an important consideration that is entirely separate.

At present, we do not have any network monitoring tools. In order to check if a node is functioning properly or in order to change the way a certain machine is configured, a method of remotely logging in to the nodes could be very beneficial. Doing so would prevent the need to visit individual houses, which would be inconvenient for both the administrator and the homeowner. It is essential that these sorts of tools are made available in the long run, so as to prevent frequent visits to the houses of users that operate nodes on the network for constant testing and configuration purposes.

Hardware and Software Considerations

The primary software used to operate the network is the NetBSD[10] operating system, which provides support for a large number of hardware platforms, as well as a wide range of peripherals and accessories. Contained within the operating system is also a large amount of the functionality necessary for implementing the software portions of our project. Currently, we are using the 1.6.X series of NetBSD, and are continually investigating new innovations to NetBSD, such as the –CURRENT development and testing environments.

We are using the NetBSD operating system because it is desirable for many reasons. First, it has been designed from the ground up for reliability. This is important for all nodes, but especially for nodes that are located at an important network juncture and may be carrying a large volume of network traffic. Testing we have done as well as other parties indicates that it can remain running almost indefinitely, precluding hardware failure. Secondly, it can run on almost any hardware. This makes it easier for anyone to use extra hardware they already own to access the system and become a node. Third, and most importantly, it is small. It can run on minimal memory and hardware resources and still be usably fast for our purposes. This brings down the cost of operation and makes the project feasible.

[10] http://www.netbsd.org/

The hardware we are using to implement this project is by no means proprietary; the computers that make up routing nodes on the network cover a very wide spectrum. Essentially, they can be any system that will run the NetBSD operating system when booted from a CD-ROM and support a wireless networking card. For our development, we will be using old Pentium computers, and we are developing our system based on i386 computers, but there is no reason to be constrained in the future.

The wireless standard to be used is the IEEE 802.11b[11] protocol, which operates on the 2.4GHz band and provides performance roughly equivalent to 10baseT Ethernet. For our development environment, we are utilizing Linksys wireless networking cards, but any 802.11b radio should be compatible with our network.

Specifically, the base computers are older model Dell OptiPlex GMT+ 5166 machines. They have 166MHz processors and 32MB of RAM. In the final solution, we will be running them with no hard drives, only a floppy and CD-ROM drive. They are stripped of everything except the wireless cards and Ethernet NICs, and are thus not even capable of sound other than the internal PC speaker. The wireless hardware is available at many computer stores. They are Linksys model WMP11 mini-PCI wireless adapters. These are fully compatible with the 802.11b standard and will be configured properly to communicate with other units. Such a modest approach benefits us two-fold. First, the equipment is readily available, meaning that anyone can go out and purchase equipment to use on the network. Secondly, it is inexpensive, meaning that the network will be cheap to grow. This hardware can conceivably be left in someone's basement, without even a keyboard or monitor, cutting both hardware and electricity

---

[11] http://grouper.ieee.org/groups/802/11/

costs. Minor configurations can be done remotely, while major ones would be prepackaged on a CD-ROM that could be replaced. In case of a power outage, there is no corruptible media, meaning that when it reboots, it will automatically go back to the default configuration.

For many of the routing tasks necessary for the network that extend beyond the base capability of the NetBSD operating system, we will be utilizing a software package known as Zebra[12] which enables a user-level daemon to interface with the routing parameters in the NetBSD kernel. This program abstracts routing algorithms to make configuration and implementation easy. This is to avoid both making complicated changes to the operating system's kernel and to prevent difficult to locate bugs from infecting the configuration. With this program, it is quite a simple matter to implement complex routing algorithms and to change those algorithms when better ones become available.

To configure the machines, we are relying heavily on the scripts and building blocks established by the Champaign-Urbana Wireless Users Group. This is both to speed development time and to ease the interconnecting of our groups' networks if and when they finally grow to within range of each other. However, due to the nature of our design, we do have to make a number of changes to their default configuration.

Both the decisions that have made thus far and those that will be made in the future are constrained by cost, in that the expense of the system must be kept to a minimum, or else it will become prohibitive to grow and expand the network.

---

[12] http://www.zebra.org/

Implementation Issues

Introduction

There have been a number of challenges we've faced during trying to implement this system. Some of them were expected from the start, but others came up as a surprise, and a large amount of our time has been spent trying to combat these issues, both during the end of the first semester and during the majority of the second semester.

Setup Issues

In order to place antennas on the outside of customers' houses, it is necessary to utilize LMR-400 coaxial cable to run from the computer, through the walls of the houses, to the antennas. LMR-400 is expensive and difficult to use, but it is necessary to carry the complex microwave signal for this application. We will be drilling 5/8" holes through the houses in order to accommodate the cable. The location of the node computer itself will vary depending on the house, but will hopefully be centrally located to limit the amount of signal loss due to cable runs. On the inside and outside, the cable will have to be properly terminated, which requires the correct equipment. An adapter cable will be necessary on the inside to connect the coax to the external antenna connector on the interface card. On the outside, the cable will be connected to the antenna and properly weatherproofed to prevent large signal loss caused by water in the connectors. Then the antenna will have to be properly mounted to both retain the aesthetic look of the houses, and to reach the next node with sufficient signal. The holes in the houses will also need to be properly weatherproofed to avoid expensive water damage to the homes.

Figure 3: LMR-400

This is an example of the LMR-400 coaxial cable we are using. It is shown next to some loose change to provide an example of just how thick this cable truly is. The diameter and rigidity of the inner conducting wire is evident when observed in this fashion.

Figure 4: LMR-400 Spool

This is a spool of the LMR-400 wire we will be working with. These spools are extremely heavy, but carry enough wire for several installations.

Figure 5: Patch Antennas

These are examples of a couple of patch antennas. While we are using slightly different antennas, they maintain this general shape and size, which makes them relatively unobtrusive when compared to more traditional varieties.

Getting Working Computers

From the start, one of our biggest implementation issues has been the procurement of working computers, both for the purpose of a laboratory and for actually implementing nodes for our network. Initially, we were provided with four computers, from which we managed to piece together two systems with working hard drives, floppy drives, and CD-ROMs. After doing so, we acquired two more systems, but have been unable to create another fully functional system since then. All told, we have been unable to procure any more working hard drives, requiring we attempt to boot other semi-functional machines via CD-ROMs, which have to be generated via the working machines (which also leads to another issue, discussed below).

Additionally, our functioning machines only work intermittently; both occasionally will not boot, will not recognize hardware that is present, or will behave in an erratic manner. While this behavior is often cleared up via cycling power, it makes reliable testing conditions difficult to reproduce, as we will sometimes think we have discovered a software or implementation issue, when in fact the hardware is merely misbehaving.

During our second semester, we procured additional machines from the Computer Science department. These machines were far more modern than the machines we were using during the first semester, but we found that they introduced new challenges. It took several weeks to acquire the machines, and upon retrieving them, we discovered that they had no memory. This required the purchase of memory before they could even boot. Once they came online, we had to reset the BIOS, as the machines had password protection enabled, precluding us from using their CD-ROM drives as a boot device. Upon trying to get our CD to boot these machines, we discovered that the CD-ROM drives were SCSI, and that this configuration was not handled out of the box by our base CU-Wireless discs. To try and accommodate this, we started using the old IDE CD-ROMS in these machines instead, which while slow seemed to facilitate a boot.

Unfortunately, once the new machines came online and their wireless interfaces were configured, they instantly started encountering I/O conflicts, and hardware interrupt errors. From experimenting with the cards and machine, it doesn't seem that NetBSD properly allocates interrupt request numbers when there is more than one wireless network interface in the computer. Disabling the onboard Ethernet controller seemed to help matters somewhat, but it did so only in bringing the overall number of problems down, not eliminating them.

Clearly, this problem suggests the need for reliable hardware while trying to create the network. While it is important that the systems used to create and grow the network are inexpensive, the decision will have to be weighed in the future as to whether expense or reliability is the primary concern. The less expensive systems will ultimately cost more if it becomes necessary that they require more maintenance.

Getting NetBSD to Install Properly

During the initial installation process, it was somewhat difficult to get NetBSD to properly install via an existing wireless infrastructure, since there was no wire-bound network available. Through the availability of many helpful documents and websites, we were able to discover the appropriate configuration parameters, and able to configure the miniPCI Linksys cards to properly function from the installer. It turns out that this required a more recent version of NetBSD than we previously had available, but selecting the then in-development (and now stable) 1.6.0 distribution helped make this process much less painful. Operating with bleeding-edge releases of NetBSD exposed us to several issues, as we'd occasionally encounter a problem related to the NetBSD installation and a fundamental bug in the new release, when we again thought we might be experiencing a software bug of our development. This was especially true since we were using drivers for our wireless cards that were very new and not heavily tested.

Packet Loss During Testing

While initially proving the concept of using the wireless systems to transfer packets from wire-bound machines, we started with two functioning wireless nodes, and then connected a laptop to each one. Surprisingly, when we pinged one laptop from the other, we discovered a significant amount of packet loss, on the order of at least 50%. If this sort of problem were to propagate across an entire network, it would quickly render the entire system useless, as compounding a loss of half of the packets would make any hop more than a few nodes completely useless, as nearly all of the bandwidth would be dumped into the ether without actually transferring anything.

Initially we suspected that we might be experiencing packet loss due to software configuration problems, or the inability of the wireless nodes to keep up with the packets on both sides, but further investigation helped us to determine that the machines should be sufficient to accomplish what we were trying, and that our configuration at that point was simplistic enough that it should not be encountering such errors. Further inspection of the system logs provided no explanation either.

Perplexed with what the problem was, we finally discovered that there seemed to be some interference issues between the two wireless nodes; their proximity on the test bench, which was under a meter, was far too close, causing the signals between the two antennae to not properly communicate with each other. We expect packet loss on this order to evaporate as an issue once antennas are placed sufficiently far away, though further testing will be necessary to verify this hypothesis. The lack of space in our lab along with cramped space on the test bench makes it difficult to accomplish this.

Lack of Sufficient Computers to Verify Reflector Concept

Related to our lack of a large number of computers for our test bench mentioned above, we are limited in testing our concept of a reflector node, which is designed only to bounce signals from one end of a network to another, without sharing any additional resources. Because we only have two fully functional computers, building another system to bounce packets between them has been, to this point, impossible. In order to verify that our network works as expected when we start scaling it, it becomes of the utmost importance to procure more functional computers; without them, we end up wasting a large amount of time in developing a solution for a particular direction the project is taking, only to discover what we thought might work in theory is fatally flawed in practice. All told, other than trying to meet with community members, this has been one of our biggest stalling points, as it ultimately leads to a large amount of time spent that doesn't move us closer to our goal of a fully functioning community network.

Reflector Node Issues

During our second semester, we procured additional computers, and were finally able to verify the reflector node concept. As discussed previously, our new machines indicated that booting up with two wireless Ethernet cards caused significant problems in terms of interrupts and bus I/O. Having a functional old machine around (it was extracted from its installation site after experiencing problems), we tried booting it with two interface cards, and discovered the same problems, supporting the idea that NetBSD has a fundamental limitation at this point in time which doesn't allow it to properly configure two wireless networking cards at once. Unfortunately, this seemingly innocuous problem is a dramatic setback for our project, which relies on the idea of using reflector nodes with two wireless network interfaces in order to propagate the signal along the network.

To try and figure out what we were doing wrong, we contacted local wireless experts David Young and Sascha Meinrath, as well as several mailing lists of groups working with NetBSD and wireless technologies, and discovered that working with two wireless cards under NetBSD has been a problem for quite a while. Specifically, Sascha mentioned that they had been trying to work with two cards in a NetBSD machine almost two years ago, and encountered a significant amount of problems. David mentioned that he's had a long outstanding PR related to this problem, and that it exists as a familiar unsolved problem. Unfortunately, this suggests that either some serious work needs to be done at the kernel level in NetBSD to work with two wireless cards at the same time, or that we need to wait for a more mature version of NetBSD or consider a completely different operating system that has more useful wireless device support. Because it took us quite a while to learn these things about NetBSD, a certain amount of our project is inescapably flawed. To fix this flaw, either the reflector node concept needs to be abandoned, or we have to find a way around the problem, which would involve either extensive work on NetBSD or a complete replacement of NetBSD with an alternative operating system with more mature wireless support.

With further investigation, it appears that this I/O problem is only present on Dell machines; according to NetBSD PR 18794[13], it has, in fact, only been produced on those systems; other machines are unaffected. Because of this, a third option appears, in that it may be possible to continue as originally planned, with the use of machines that are not manufactured by Dell, or machines that have been made recently by Dell that are unaffected by this behavior. At the very end of the semester, a couple of machines were made available to work around this issue, but the amount of time remaining to accomplish something once these machines were made available limits what can be done.

[13] http://www.netbsd.org/Misc/query-pr.html

Serial versus Mesh Network Issues

Our initial plan was to distribute the signal of our network in a linear fashion, injecting resources along the chain, and then bouncing packets back and forth down the street, as if each node on the network were a stop on a subway. As discussed in the implementation issue of channel allocation, the benefit of this sort of organization came in that there need be no holdups between nodes while other nodes are negotiating whose turn it is to communicate on a specific channel.

However, as appealing as this serial design sounded, we have been discovering more and more implementation issues related to this design, in that operating a network with a serial design ultimately leads to difficulties when abstracted beyond a very small scale. The first problem is that in communicating from one end of the chain to the other, a packet must pass through an extremely large number of hops, when considered in the context of a very large network. Just like trying to make a large number of connections during bus, car, or air travel, a few hiccups along the way can cause the entire process to fail or be delayed beyond an acceptable boundary. Secondly, a serial system like this has redundancy and reliability issues, as a single break in the chain can cause the network to become disconnected. If a node were to lose power, have a software failure, or become overloaded with traffic, the two ends of the network on either side of that node would no longer be able to communicate with each other, resulting in two segregated networks. As the ultimate goal is connecting the entire community, rather than just creating disjoint hot spots of network connectivity, this situation is entirely unacceptable. Further, performance across many connections in this manner cannot be distributed, or take alternate routes, meaning that it could become very expensive in terms of latency to attempt to use another community member's resource, such as the Internet. If a user were trying to use an Internet connection provided via eight wireless hops, it is unlikely that their performance will be at all acceptable, especially when they are no longer the only users attempting to utilize this resource. This, again, is undesirable, as it is essential for the success of the network that the entire network is interconnected with both reliability and performance as critical goals.

To work towards resolving this issue, we are attempting to investigate a network that conforms more to a mesh topology, wherein no single node failing can break the network, and packets can take multiple paths, so as to improve performance. However, just switching to a mesh design leaves a lot of issues to resolve; with multiple antennae operating on the same channel in a certain region, only one station will be able to communicate at any given time. Further, it becomes much more difficult to coax the network into routing packets properly, as the path is no longer just a matter of coming in on one interface, and going out on the next interface. Instead, intelligent routing protocols that adjust the path of packets in the network have to be considered, which creates a great deal more complexity in the system, causing more software development difficulties. This will continue to be one of our largest ongoing issues, and our attempt to figure out how to get around all of these considerations will likely occupy much of our development efforts.

Lack of Real World Testing Conditions

In attempting to verify our system, we need to examine how it functions in a real-world environment. This includes the issues directly related to installation of several nodes in houses throughout the community, combined with less tangible problems such as interference from rain, trees, consumer electronics, traffic, and the like. As good as our test bench is for verifying simple concepts, the perfect lab conditions can give us a false sense of success even though it's possible that when we attempt to deploy our system in a real environment, that considerations we haven't taken into account will cause things to fall apart in practice. While it would be ideal if our experimental results transferred directly to the real world, we can't reasonably expect everything to work perfectly. Other issues related to this begin with the unpredictability of how reliable antennae will be in communicating with each other when not connected by ideally short runs; we suspect that it will not always be possible to position nodes in the most optimal position within the house, resulting in long LMR400 runs that will decrease the sensitivity of our radios, and therefore decrease the effective ability of two nodes to communicate.

As such, it is of the utmost importance that our meetings with community members do take place, and that we are able to verify that our experimental results from the lab transfer directly to real-world results when we actually deploy nodes.

IP Addresses

Currently, we are placing each node on its own subnet. Doing so can prevent some of the problems inherent in wireless communication. The main problem with having numerous machines on the same subnet occurs when the entire subnet is not within radio range of all the machines. When a packet is sent from one machine to another machine on the same subnet, it is assumed by default that the two machines can communicate directly. This is not necessarily the case with wireless networking, as explained in the design section.

In order to determine IP addresses, we are using the CU wireless MAC-to-IP script. The script uses the MAC address of the wireless card and hashes it to an IP address. While the network remains small, this method will work fine. However, as the network grows the hash function will generate collisions. Since a collision will make routing impossible, the MAC-to-IP solution will not work in a large network.

We have begun to research various alternatives. A DHCP server would normally be used to solve the problem. Unfortunately, several problems can potentially arise when working in a wireless network. Since only machines within radio range of the server would have direct access to it, outlying nodes would depend on intermediate nodes to function properly. When adding one node at a time, this shouldn't be a problem. However, if a power failure wipes out a large section on nodes there could potentially be problems with outlying machines starting up again. They would not be able to communicate with the DHCP server until everything in-between was working properly. It could take considerable time for startups to propagate through the entire system, and it is unknown what kind of behavior would result. We have only been able to perform very basic experiments in this area, as we do not have a large enough network to test it thoroughly.

Creating a Bootable CD

In order for our nodes to function properly without hard drives, we needed to create a bootable CD. The CD should load the operating system, configure the required hardware to act as a node, and begin communicating on the network without the need for any manual input. This task turned out to be far more difficult than what we had originally anticipated.

Initially, we tried to create the bootable CD by making an ISO image of the machines in the test lab. This proved to be far more complicated than it sounds. Since the provided computers are old, there are no BIOS settings to allow them to boot directly from a CD. Therefore, a boot floppy is required to load a kernel and basic drivers before the CD can be mounted and accessed. The difficulty occurs when switching between the floppy and the CD. Since some of the boot sequence is performed from the floppy and the rest from the CD, numerous changes must be made to the default boot sequence. Several additional changes must also be made in order for everything to function properly, such as mounting the main memory to act as a writeable disk. The order in which everything must be done is critical for a successful startup. We spent a considerable amount of time researching how to make all this happen in the correct way.

Unfortunately, the only way to test if what we are doing is correct is to burn actual CDs. Doing so requires a lot of time. It generally takes 10 to 15 minutes to create the ISO image, 10 minutes to transfer the image to the machine with a CD burner, and an additional 10 minutes to burn the image. Once we created the CD, we would test it and if it didn't boot properly try to find out what caused the mistake, correct it, and spend 30+ minutes making a new CD. Unfortunately, tracking down the bugs is extremely difficult. There were also several places along the path of CD creation that could have caused failure aside from not having the OS changes made properly. Some of these factors include faulty ISO creation; bad transfer to the machine with the CD burner, bad CD creation, or hardware failure, which was experienced regularly. To further complicate the problem, our lab did not contain a CD burner, so someone had to bring their machine to the lab or we had to move one of the lab machines somewhere else.

After several hours of research and making changes and several more hours of creating faulty CDs, we decided to change strategies and try to create a bootable CD based on the CU-Wireless CD. After a few failed attempts, we finally found a way to create bootable CDs by using some of the work they had previously done. For the time being, we plan to continue creating CDs in this manner, but use our scripts and configuration files in order to create the network with our layout.

During our second semester, we attempted to base our CDs on the more modern CU-Wireless CD. The CU-Wireless project updated their CD to use C-based startup programs that had a much lower footprint than the previously Python-based systems. While the transition seemed at first to be straightforward, we found that any changes we made to these discs resulted in machines that would not boot at all. Because of this, we fell back to our older discs, in the interest of trying to create a workable network as a goal more important than using the newest and latest technology.

While creating the new discs during the second semester, we encountered two major setbacks. The first was that, despite making changes to the files before we re-created the new ISO and recorded it to a compact disc, the final burned disc did not seem to reflect the changes. Some related weirdness came in that the original disc image was on the order of about 150 megabytes, but the newly burned discs were almost completely control, which didn't make much sense at all. We mounted the disc images that we created before burning utilizing a kernel facility known as a loopback device, which allows us to simulate a hardware drive on the computer. Doing so, the changes to the image appeared in place, and it would seem to suggest that everything was correct with the images. However, upon burning, the changes did not take. From talking to local NetBSD experts, the only viable suggestion that came back was that the loopback device wasn't flushed or properly unmounted before burning, which we carefully verified not to be the case. To work around this, we often accepted what changes we had made, and were forced to manually configure the settings to finalize configuration.

A much stranger problem came in trying to actually boot the discs we had created. Before installing the discs into our provided machines, we verified that they would boot in our own machines, and they did so quickly and effortlessly, though they failed at configuring the wireless network, since our computers lacked wireless capabilities. Upon placing the discs in the old computers for the project, we observed that boot from the discs was extremely slow. Though the systems did eventually come online, fully configured, it took almost 45 minutes to do so, which is completely unacceptable performance for a production wireless environment. Aside from practicality issues for a real wireless network, this also made testing new discs very expensive in terms of time, since any change to a disc would take at least an hour to verify when mastering, burning, and bootup were considered.

Figuring this might be a problem with the old systems only, we attempted booting the new systems with these burned discs, but encountered the exact same problem. Because our newer systems had only SCSI hardware, we utilized IDE CD-ROM drives from the older computers to boot these new machines. This would seem to suggest a reasonable explanation for why the two systems exhibit the same behavior. Things become strange, however, in that we placed an old CD-ROM in one of our home machines, and were able to boot at normal speeds. Additionally, the original CDs on which we based our custom discs would also boot in the new and old machines without delay. We again sought the help of local experts on this problem, and received the suggestion that old CD-ROM drives do not read CD-R discs reliably, and that if for some reason a couple of sectors were bad, that they would have to be re-read an inestimable number of times, which could cause the boot process to be as slow as we are observing. It is possible that the use of a brand new CD-ROM drive, or at least one more recent, would cause this problem to disappear. Our only hesitation in suggesting this route is that the boot time was normal on a home system, using and identical CD-ROM drive, which would seem to suggest some sort of common problem between the wireless network machines, though architecturally their differences are substantial enough that the only commonality we can find is that they are both made by Dell.

Routing Considerations and Difficulties

The group has spent considerable time researching various routing strategies. Ideally, all packets should be routed along the quickest path. However, the previous statement cannot occur without several trade-offs occuring. The group compared the advantages and disadvantages of different routing methods to arrive at what we consider to be the best routing strategy.

Initially, the network will be linear and relatively short from end to end. Within the network, only one path will exist from the source to the destination. Also, due to the small number of nodes and unchanging topology of the network, manually predetermining routes will not be a difficult issue. Therefore, until the network either becomes non-linear or begins to grow at a quick pace, static routing is the best approach. Static routing has little processing overhead when compared to dynamic routing. It also does not produce any additional network traffic. A good rule or thumb is to use static routing when you can and dynamic routing only when you must. The down side of static routing is that all routes must be manually determined. Since we are using machines that will require no user input once installed, all routing must be resolved in the lab before the bootable CDs are created. Our test lab is currently using static routing, which should be easy to reproduce when we create bootable CDs to place in the community.

As the network begins to grow, static routing quickly becomes impractical. Every new node must be manually configured before the boot CD can be created, and this can quickly become very laborious. At this point, dynamic routing provides an attractive alternative. Dynamic routing allows for the automatic discovery of new routes. A good algorithm will determine the quickest route to a destination, making packet flow as efficient as possible. New nodes can also be easily added with no additional work from the user. The problem with dynamic routing is that additional processing power is required in order to determine which route to take. This could become an issue as the network begins to grow to a very large scale, as we are using older machines with low processing power. A good algorithm will keep this additional overhead as small as possible. Another problem with dynamic routing is the production of extra traffic. A node's routing information must be propagated through the network in order for new routes to be discovered. If a good algorithm is used, the additional traffic can also be minimized.

The two most commonly used dynamic routing algorithms are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). RIP is a distance-vector algorithm that is used extensively throughout the Internet. Every node contains a table with the distance to every other node in the network. At a pre-specified time interval, a node transmits its table to all the other nodes in the network. The receiving nodes examine the distances in the transmitted table and determine if a shorter path exists and updates its tables if need be. As one can imagine, RIP has high overhead on a large network. The tables can become excessively large since each node contains information about every other node. A large amount of extra traffic is also generated since each node must send its entire table to every other node in the network. For these reasons, we decided RIP would not be a good choice.

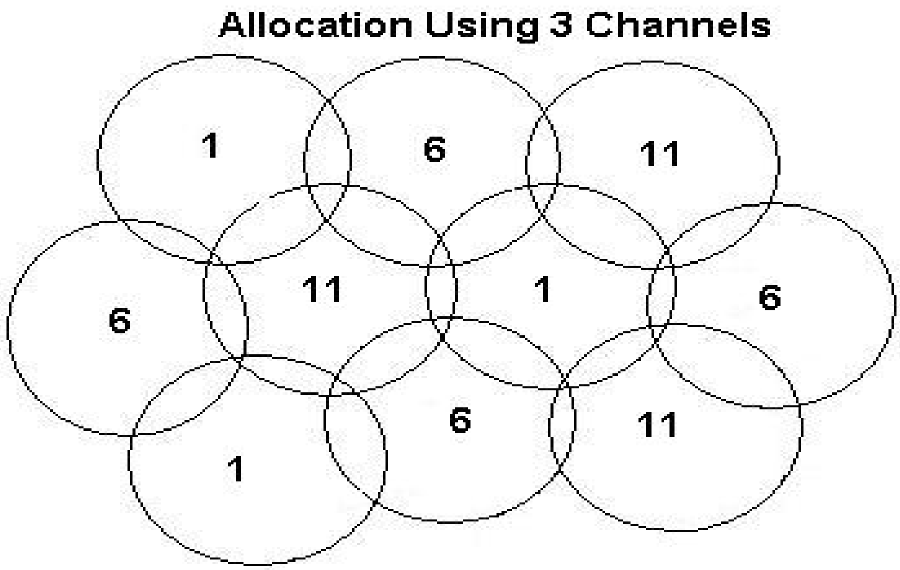
The other dynamic routing algorithm is OSPF. As the name implies, Dijkstra's Shortest Path First algorithm is used to find the quickest route to a given node. A node will occasionally send a table with all its directly connected neighbors to the rest of the nodes. Each node then collects these tables and finds the topology of the network. Dijkstra's algorithm is then run to find the shortest path, and packets are routed accordingly. We decided that OSPF is the better choice of routing algorithms. It requires slightly more processing time than RIP, but produces less network traffic and requires less memory. A key factor in the network is minimizing unnecessary network traffic, so OSPF was the superior algorithm. Additionally, since CU-Wireless is currently using OSPF routing, the two networks should have increased compatibility once they meet.

In order to implement OSPF, we are using a piece of routing software called Zebra. Zebra is an open-source program that is widely used in Linux and Unix routing boxes. It supports the common routing methods, such as OSPF and RIP. It also allows an administrator to remotely access a machine and check that it is working properly. A large amount of information is available on the web concerning Zebra, so support should not be an issue. It is also known to run on NetBSD. In order to make Zebra run properly, two configuration files need to be written, one for Zebra in general and the other for the routing algorithm to be used. Only minor changes need to be made to these files from node to node, and these changes can easily be implemented with scripts. Therefore, once the basic configuration files and scripts are created, Zebra will allow us to add or remove nodes in the network with no additional user input. Zebra using OSPF routing is currently implemented.

Channel Allocation

One of our largest concerns deals with channel allocation. Adjacent nodes must be set on the same channel in order for them to communicate. However, if either of these nodes is able to receive signals from an unintended station on the same channel, interference and therefore packet loss will occur. For our initial linear network, channel allocation will not be a large issue. We can simply alternate between two or three channels as we move from block to block. To set up this system, we will assign the first wireless interface channel X and the other interface channel Y. The antennas would be installed so the proper interface is pointing in the correct direction.

The real problems begin when the network grows into a second dimension, going to side streets. Only 11 channels are available for use within the 802.11b spectrum. To further complicate the problem, adjacent stations contain some overlap in frequencies, so to minimize cross interference it is recommended to leave a space of 5 channels between nearby stations. This leaves us with only 3 usable channels, 1, 6, and 11. We researched the mobile phone system to gain a better understanding of how it allocates channels since it is faced with a similar issue. However, we found that the mobile system has 832 usable channels, but typically uses 8 channels full time. Since we do not have 8 available channels without running into interference, this model is not applicable. Assuming overlap is kept to a minimum, channels can be allocated as illustrated below:



The wireless interface would be assigned a channel according to which zone it was trying to communicate with. For example, a reflector node could be assigned channel 1 in one direction and 6 in the other. If a third card was implemented adding communication to side streets, it could be assigned channel 11. However, as the next paragraph illustrates, problems with this method can occur.

To further complicate the problem, we also want the machines to be able to detect what channel to use automatically. Finding a way to have the network automatically determine its own channel layout is a major challenge. Active channel detection is another issue we are encountering. A channel scanning utility is provided with NetBSD, but initial experiments using it have produced poor results. A method other than scanning for available channels may need to be devised.

As a further complication to this, discussion with local wireless experts suggested that attempting to utilize multiple channels from one device, even with semi-directional antennas, would cause a significant amount of problems between the two devices so as to make the network unusable. While our network is not large enough to verify these claims yet, such a setback would require a catastrophic change in our topology and design.

While the network merely operates in a straight line down a street, if we are able to use at least two channels without this theoretical interference, things should work without much need for concern. However, once we expand in another dimension, away from the main network artery down Green Street, complications come in that we would now most likely be considering systems with three channels of wireless communication, which is almost guaranteed to encounter interference problems.

Our current solution to the channel allocation issue involves knowing the location of surrounding channels during installation. The first interface is configured use channel 11, the second interface uses channel 6, and if a third interface is present, it uses channel 1. When the node is installed, it is important to place the antenna in the direction of the corresponding neighboring channel.

Arranging Meetings with Community Members

While not a technical issue per se, one of the difficulties with trying to establish an initial testing environment is in arranging a meeting with community members. As people outside of our group will ultimately provide houses and electricity to support the infrastructure for this network, it becomes essential that we establish a testing environment that shares resources for the community as quickly as possible. To this point, we have had difficulties in coordinating with our client and community members in order to arrange a meeting and summarily a test installation. For resolution, we've tried to make our schedules as open as possible, in order to try and fit our meeting to when is convenient for community members, but we have not succeeded in arranging a meeting thus far.

We were able to arrange installations with Professor Ralph Johnson and his next door neighbor, Les Carlton. These installations consisted of locating a place within the house to store the computer, running LMR400 coaxial cable to the exterior antenna site(s), and then installing the semi directional patch antennas exterior to the house to create the network. Because the idea of just dropping a box in their houses combined with drilling holes in the walls is a little scary to a homeowner, it was important that the initial meetings explain the process as much as possible, and leverage the benefits over the possible detriments. An additional installation did not take place at another location further down Green St, at the house of Chris Harris, because of concerns over security. Having access to a much greater pool of community members would have allowed us a much larger test bed, as well as more possibilities for scheduling tests and installations. Having this resource during our first semester, as well as our second semester, would have allowed us to isolate many of the problems we encountered towards the end of the project in a much more timely fashion.

Project Log

Introduction

Below appears our project log, as a description of what happened during the course of our project. While our original schedule anticipated more progress through this time period, the project log is the most accurate indication of our work and what actually happened during the two semesters. It is arranged in reverse chronological order.

Project Log

Week of May 11

We have a final meeting with our client scheduled to discuss the final status of our project. We also have a meeting with our client and his neighbor to complete the node installation using hard drives if the CDs are still not booting properly.

Week of May 4

Aaron H:

2H Group Meeting

Aaron T:

2H Group Meeting

Darin:

2H Group Meeting

Ryan:

2H Group Meeting

We completed work on the final document.

Week of April 28

Aaron H:

4H Group Meeting

Aaron T:

4H Group Meeting

Darin:

4H Group Meeting

Ryan:

4H Group Meeting

In the past week, we have discovered the problems we have been encountering with two wireless cards is an unresolved problem on Dell computers. We are currently trying to acquire different machines. We also found out why some of our CD modifications were not appearing on newly burned CDs, and fixed this problem. In the group meeting, we prepared for the presentation and worked on the final document.

Week of April 21

Aaron H:

3H Group Meeting

Aaron T:

3H Group Meeting

Darin:

3H Group Meeting

Ryan:

3H Group Meeting

We are still trying to figure out why the CDs are suddenly having problems. We are testing them on newer machines to see if newer hardware can read the CDs any better. We are also double checking startup scripts to verify everything is behaving properly.

Week of April 14

Aaron H:

3H Group Meetings modifying and burning CDs

5H Burning and testing CDs

Aaron T:

3H Updated project documents

Darin:

1.5H Installing and testing node

2H Group meeting

Ryan:

3H Group Meetings modifying and burning CDs

We have been spending a lot of time trying to create a working CD. New problems we have not experienced before keep popping up. For example, the most recently created

CDs will boot in the new machines, but load extremely slowly. We think it is a hardware problem.

Week of April 7

Aaron H:

6.5H Group Meetings

Aaron T:

6.5H Group Meetings

Darin:

6.5H Group Meetings

Ryan:

6.5H Group Meetings

We held three group meetings this week. The first two were quick organizing meetings to discuss the status of the new machines and set a time to install a node. We then met on Sunday to install the node. While testing the new node, a problem we had never experienced before regarding interrupts arose. We spent the rest of the afternoon trying to solve this problem. No solution has been found yet.

Week of March 30

Aaron H:

1.0H Group Meeting

Aaron T:

1.0H Group Meeting

Darin:

1.0H Group Meeting

Ryan:

1.0H Group Meeting

Everyone was very busy this week, so we only held a short meeting. We found out that new memory should be purchased for the machines. We hope to find out what kind of RAM to get and have it by next week. We also discussed what we would like to accomplish by the end of the semester.

Week of March 23

Spring Break

Week of March 16

Aaron H:

1.5H Group Meeting

Aaron T:

1.5H Group Meeting

Darin:

1.5H Group Meeting

Ryan:

1.5H Group Meeting

We finally received new machines. The machines were moved to the lab so we could begin getting them setup. However, we discovered that the machines had no memory, and the memory in the old computers was not compatible. Aaron T. emailed the appropriate people to check on getting new memory, but we have not yet heard what action should be taken.

Week of March 9

Aaron H:

1.5H Group Meeting

1.0H meeting with grad student to discuss network security.

Aaron T:

1.5H Group Meeting

1.0H meeting with grad student to discuss network security.

Darin:

1.5H Group Meeting

1.0H Node Install

Ryan:

1.5H Group Meeting

1.0H Node Install

We discussed possible application programs that could be written for the system. We still only have one working computer in our lab, so we cannot easily test anything with the network.

Week of March 2

Aaron H:

1.5H Group Meeting

Aaron T:

1.5H Group Meeting

Darin:

1.5H Group Meeting

Ryan:

1.5H Group Meeting

We discussed alternate methods of IP address assignment. Aaron T. has been trying to acquire machines for our lab so we can experiment with what we are researching.

Week of February 23

Aaron H:

2H Group Meeting

Aaron T:

2H Group Meeting

Darin:

2H Group Meeting

Ryan:

2H Group Meeting

The meeting was a coordinating meeting. We worked on a plan for the rest of the semester as well as general strategies about some of the problems encountered to this point.

Week of February 16

Aaron H:

2H Group Meeting

Aaron T:

2H Group Meeting

Darin:

2H Group Meeting

Ryan:

2H Group Meeting

We discussed the channel allocation problem, including possible solutions.

Week of February 9

Ryan:

2H researching DHCP and how it could be used in our project.

Week of February 2

Aaron H:

1.5H node install

Aaron T:

1.5H node install

Darin:

1.5H node install

Week of January 26

Aaron H:

1H Client Meeting

Aaron T:

1H Client Meeting

Darin:

2H Client Meeting

Ryan:

2H Client Meeting

Aaron H and Aaron T met with Chris Harris, a potential customer, to explain our project. Darin and Ryan met at Prof. Johnson’s house and installed the first node.

Week of December 1

Aaron H:

1H Group Meeting

4H Preparing final document and presentation

Aaron T:

1H Group Meeting

4H Preparing final document and presentation

Darin:

1H Group Meeting

4H Preparing final document and presentation

Ryan:

1H Group Meeting

4H Preparing final document and presentation

In the group meeting, we discussed the content of the presentation and final document and how we would split them up.

Week of November 24

Thanksgiving break

Week of November 17

Everyone was busy finishing projects and assignments before the break, so we didn't work on the project. We had planned on meeting with interested clients in the community to discuss potential installs, but we did not hear back from the clients.

Week of November 10

Aaron H:

0.75H Meeting with client

Aaron T:

0.75H Meeting with client

Ryan:

0.75H Meeting with client

1.0H reading about dynamic channel allocation and MACAW

Week of November 3

Aaron H:

3.5H Group Meeting (Monday)

3H Group Meeting (Friday)

3.5H Group Meeting (Saturday)

Aaron T:

3.5H Group Meeting (Monday)

3H Group Meeting (Friday)

Darin:

3.5H Group Meeting (Monday)

3H Group Meeting (Friday)

3.5H Group Meeting (Saturday)

Ryan:

3.5H Group Meeting (Monday)

3H Group Meeting (Friday)

3.5H Group Meeting (Saturday)

In the group meetings, we studied the CU wireless scripts to gain a better understanding of how they solved some of the problems we were encountering. We also tried to create more working machines out of parts from several of the junk computers. A significant amount of the time on Friday and Saturday was spent trying to create bootable images. This turned out to be very time consuming.

Week of October 28

Aaron H:

2H Group Meeting

Aaron T:

2H Group Meeting

Darin:

2H Group Meeting

Ryan:

2H Group Meeting

In the group meeting, we tested a machine with two wireless cards that will act as a reflector node. We also studied the CU Wireless CD to get a feel for how their CD boots and the type of scripting run so we will have an idea of what is required when we build our boot CD. This coming week, we plan to create our first CD that will allow us to begin installing computers in the community.

Week of October 21

Aaron T:

2H Researched card-based access point functionality, OpenAP, IBSS

0.25H Met with Alejandra to discuss requirements document

Darin:

0.25H Meeting with Alejandra

1H Looking at Zebra

1H Installing another copy of Zebra

Ryan:

0.25H Meeting with Alejandra

2H Researching methods to measure network speed and throughput

Week of October 14

Aaron H:

2H group meeting

Aaron T:

2H group meeting

Darin

2H group meeting

Ryan

2H group meeting

1H reading about Zebra configuration files

Week of October 7

Aaron H:

1h Documents

4h Admin

Aaron T:

1h Met with Darin to research wireless access point support

2h Researched Zebra

1h Researched mkisofs to make bootable cds

Darin:

2h Pickup computers and disassemble/verify function

2h research NetBSD Network Address Translation

Ryan:

3.5h Researching Zebra configurations and capabilities

Week of September 30

Aaron H:

1.5 Hours group meeting

1 Hour meeting with client

1.5 hours admin work

5 hour documentation

Aaron T:

1.5 hours group meeting

1 hour meeting with client

Darin:

1.5 hours group meeting

1 hour meeting with client

1 hour install Zebra

Ryan:

1 hour meeting with client

1.5 hour reading about wireless routing and how to configure Zebra

Week of September 23

Aaron H:

1.5H meeting with David Young

Aaron T:

1H arranging meeting with David Young

2H researching media bridging and packet forwarding on NetBSD

Darin:

1.5H meeting with David Young

Ryan:

1.5H meeting with David Young

1.5H reading about OSPF and Zebra

Week of September 15

The group met for 1.5 hours on Wednesday. We continued working on different ways to get the computers to talk.

Everyone met again on Friday for 1.5 hours. We tried different bridging and routing configurations. We are planning a meeting with David Young, a local expert on the wireless project, next Wednesday evening.

Week of September 8

The group met for 2 hours on Monday. We found 2 computers that worked and installed NetBSD and a wireless card on one of them. The OS did not recognize the cards, so before the next meeting we should find out if NetBSD supports the LinkSys WMP11 cards.

Darin spent 2 hours learning how to configure NetBSD to work with the wireless cards. He got one computer talking to Shipwreck, his local wireless network.

The group met for 1.5 hours on Wednesday. Darin showed us how he configured the first computer and we got a second computer set up and talking to Shipwreck. We then tried to get the two computers to talk independent of Shipwreck.

A major problem we will be faced with is routing, so everyone in the group should start researching methods of wireless routing with NetBSD

Week of September 1

The group met on Wed. for 1 hour to meet each other, begin discussing the project, and assign jobs. Group met again on Fri. for 1.5 hours with our client. Ryan worked for 1 hour learning html and setting up the web page.

First Semester Future Plans

Introduction

At the end of the first semester, there still remained a great deal of work left to complete this project. The plan for both semesters is included in the following section, to demonstrate what the forecast of activity was from the conception to the conclusion of the project. It is useful as a contrast to the actual project log as it played out, as detailed in the previous section.

To provide a concrete organization for the project, we broke it down into several phases. While some portions of the project could be done out of sync with the rest of the project, those phases with a number greater than any uncompleted phases could be finished only when all numerically lesser phases were completed.

NB: References to LANs refer to wired networks, not wireless.

Overview

These are the phases we have defined for each portion of the project:

Phase 1: Proof of concept.

Phase 2a: Initial Installations.

Phase 2b: Dynamic Networking.

Phase 3a: Diskless Routers with Dynamic Networking. (2b)

Phase 3b: Network Tool Creation. (2a?,b?)

Phase 4: Upgrade installed routers (3a,b)

Phase 1: Proof of concept

Dependencies: *None*

Description

In this phase, we need to set up a lab in which to work and create a mini-network. We will need to have three machines, two end-user machines (A and C) and one reflector node (B). The goal is to get the machines configured so that packets can be transmitted from A to C through B and vice versa. From a machine on a LAN with A to a machine on a LAN with C again through B. Finally, from a machine on a LAN with B machines on LAN with A and C.

From a configuration point of view, for this phase both routing and IP addressing can be static. The test lab machines will have hard-drives for development, but before we proceed to Phase 2, we need to have bootable CD for diskless workstations. We will obviously test these in the lab before Phase 2.

Documents

In this phase we will need to generate the following documents:

- Requirements: Specify the goals and scope of the project.
- Configuration Manual: Describe the process needed to create a test machine for the lab and to recreate the bootable CD. This will serve as a guide for others who may modify the configuration in the future.

- CU-Wireless Networking Standards Describe the standards that we are adopting for our network. Some of these will come from the CU-Wireless group; others may be set by our client or us.
- Installation Manual - DRAFT describes how we think an installation should work.

Tests

As this is a proof of concept, the tests are fairly light – we just need to prove that this network configuration can work. We will probably use a ping utility through out this phase to establish that we have connectivity. Once we get to the end of this phase we will want to test the network using more realistic network programs. The most important aspect will be to test that we have connectivity from every machine to every other machine to insure that there are no holes in the routing.

Phase 2a: Initial Installations

Depends on Phase 1.

Description

Once Phase 1 is complete, we can begin installing our infrastructure (routing users) for users in our client group (see Requirements). This phase will be ongoing throughout the life of the project as we get more willing participants. In addition, this phase includes updating our client base with the newest version of our station software.

This is placed as a level two phase because we wanted to start the installations as soon as possible as we feel that there is a great deal that we will learn from having a working network up that we will not see in the lab.

Documents

- Final Installation Manual: The final version of the document that describes how to perform an installation. This document should include guidelines for interacting with the homeowner, running cable, placing antennas, testing the installation, and maintaining the station.

Tests

Unknown at this time, but will be part of the draft of the Installation Manual. It is possible that we will need to write some testing utilities or scripts to be run as part of the installation process.

Phase 2b: Dynamic Networking.

Depends of phase 1a.

Description

There are two main goals of this phase; first, dynamic routing using an efficient algorithm; second, dynamic assignment of IP addresses. The dynamic routing is the easier of the two as there are established software packages and algorithms out there even though wireless is a new technology. We may need to tweak a bit to make things work with our network design, but the foundation exists. Dynamic IP addressing is a different story. DHCP was not designed for a wireless environment and there several know problems combining the technologies. We will need to find a solution to this issue.

Documents

- Configuration Manual: The configuration manual will need to be updated to reflect the changes to enable the dynamic routing and IP addressing.

Tests

The primary test will to set up a network using the dynamic protocols and then attempting to add a node to it without rebooting any machines. The node should integrate seamlessly with the existing network.

Phase 3a: Diskless Routers with Dynamic Networking.

Depends on Phase 2b.

Description

The purpose of this phase is to make the necessary changes to our lab machine configuration to allow stations to boot from a single master CD with the dynamic configuration designed in phase 2b. We want to avoid custom CDs for each station, so we need to come up with a way to configure each machine properly without resorting to fully specified configuration files on the CD, as this would vary slightly from machine to machine. The current ideas are to script everything or to have a series of servers hold configuration information to be queried by the stations on boot. We prefer the first option as it would make the network more robust and avoid issues should the network become fragmented.

Documents

- Configuration Manual: The configuration manual will need to be updated to include sections on the scripts or the servers and on how to create the bootable CD from a lab machine.

Tests

The best test here would be to burn several copies of the master CD and boot our lab off of the CD and rerun all of the previous tests. Everything should work correctly if the CD is correct.

Phase 3b: Network Tool Creation.

Depends on Phase 2a.

Description

This is a nebulous phase at this point. We believe that we are going to need to write several utilities for managing our stations and allowing the users to control access to their resources.

Documents

- User Manual: We will need to write a manual on how to use any network tools that we create.

Tests

Unknown at this time.

Phase 4: Upgrade installed routers.

Depends on Phases 3a and 3b.

Description

This represents our final official update of the installed stations. Once we have done everything that we can, we need to implement the changes in the field.

Documents

- All: We will need to finalize all of our documents in this phase.

Tests

We will need to rerun all of the tests from Phase 2a after upgrading the stations.

Future Plans

Introduction

As is made clear by the contrast between the actual project and the original plans for the second semester, our project suffered a number of setbacks, many of which are explained in the Implementation Issues portion of the document. The most critical setback came in the complete failure of our reflector node concept, which invalidated our entire plan for the way the network would operate and propagate. Still, our work provides a strong foundation on which to grow a wireless network, and provided each member of the project with a great learning experience, both in terms of working as a team, and in terms of discovering a great deal about NetBSD and wireless networking.

Because we don't want to see the dream of a wireless network in Champaign evaporate, we explore here some ideas for how to grow the network in the future. First we consider what would be necessary to continue our work directly, and then examine some other proposals that involve a definite shift in focus. The proposals range from slightly different plans to starting from scratch.

Continuing Our Work

The path that follows most directly from our work on the project is to continue to work with the ideas that we have explored while trying to create a wireless network using NetBSD. By using our experiences, setbacks, and currently finished work, another senior project group, or private enterprise, could take our ideas another step further into a more viable network. Even at the time of writing this document, some of the challenges we encountered in things such as the reflector node concept are being worked out, and investigated as possible fundamental problems with the NetBSD operating system. If some of the remaining issues were surmounted, another team could expand upon what we've already developed, and quickly bring a viable network online. On the other hand, if it turns out that these issues continue to be a source of constant challenges that cannot be resolved in a simple and elegant fashion, it may prove that this option is impractical in the long run. Additionally, assuming this network continues to grow along with the CU Grassroots Wireless Internet Project, there will soon be competing networks between the two towns, with stifles the possibilities for collaboration and expansion that come with a single unified network.

To continue our work, the first step that will be necessary is testing our reflector node concept using non-dell machines. By using machines such as those that have been made available recently from the department, it is possible that the boot up and I/O problems that we had while using the CD-ROMs in the older machines will cease to be a problem. On the other hand, if the problem persists, despite claims from the NetBSD problem reports suggesting that the problem is specific to Dell computers, it suggests that there continues to be a fundamental problem with dual Linksys wireless cards under NetBSD. If this proves to be the case, it will be easiest to examine the reflector node technology, and decide whether or not this sort of topology makes sense in the future.

Once this stage has been figured out, the best thing to do will be to start installing more nodes, and getting a functional network up and running. Our current discs are very close to this stage right now, and should require minimal alterations at best to facilitate the growth of the network. There should be enough hardware now available (two working old machines, two working new dell machines, two working micron dual processor machines) available in the lab. Once this network is up, it will then be possible to consider possible applications and tuning for the network.

To achieve these goals, a group continuing our work should first start by researching wireless technology, NetBSD, and how the two interoperate. It is also critical to quickly master the creation of bootable CD-ROMs, as not doing so will surely slow down any attempts to continue the project. New CD-ROMs can be made by duplicating our images on disk, and then using the mkisofs and cdrecord tools to generate a new version of the disc, with any changes applied. These changes can be made on any system that supports these tools, such as Linux. It is not necessary to utilize NetBSD.

Alternative Future Paths

It's possible a future group continuing to try and implement our vision will elect not to use NetBSD, and perhaps not even the hardware we have been using in our project. Presented in the following section are several possible alternative routes that could be attempted, which may lead to a more successful project.

Adopting the Grassroots Wireless System

Another option is to convert our hardware and installations to systems that would be suitable as members of the aforementioned CU Grassroots Wireless Internet Project managed by David Young. Since the hardware we are using currently is quite similar to that used by this project, the transition to this pre-existing solution would only require slight modifications to installed locations to grow the network. Because this network already exists and has several deployed stations, it is already achieving many of its goals, and has solved a lot of the problems that we are still facing. It already has a community following that supports and expands the network regularly, which is important once the original network creators are no longer available. Further, this solution will avoid competition between wireless networks, and facilitate far faster adoption across both cites. Having collaborated with Mr. Young several times on challenges we have faced throughout the project, we feel his expertise helps to fulfill the vision shared by both of our projects.

Starting From Scratch

Though a drastic option, completely rethinking how the network is designed and implemented may provide a more viable solution in the future. Starting from the requirements for the network, and then selecting network topologies, hardware, and software based on those requirements could facilitate a system that is much more likely to work. A complete restructuring of the project could still build on our experiences, but could leverage operating systems with much more sophisticated and complete support for wireless networking (such as linux). Further, this would facilitate consideration of a network topology that does not require reflector nodes, a source of constant headaches during the creation of our network.

The availability of newer technologies in the 802.11 group may enable approaches that were previously unworkable, or they could invalidate approaches that at the time made sense. As an example, 802.11g provides much of the benefits of 802.11b, but with a much increased bandwidth, which would make usage of a network based on a moderate number of omnidirectional antennas a still workable solution, as bandwidth exhaustion would cease to be as critical of a concern.

Open Source Turnkey Solutions

In addition to the solutions offered by several established wireless groups, there are open source projects in existence that focus on creating the software to support wireless networks. An example of this is LocustWorld[14], which provides a platform for ubiquitous wireless networking through a heavily customized Linux installation that allows individual laptops, desktops, and specialized mesh networking hardware to create a wireless computing grid. With the need for wireless networks constantly growing, even more solutions like this will likely become available in the future.

Hardware Solutions

Consumers have driven the need for an abundance of wireless networking access points and routers in the past months. It is now possible to inexpensively

---

[14] http://www.locustworld.com/

acquire sophisticated wireless networking hardware for a fraction of the cost of a dedicated computer that facilitates a wireless network, even if the computer is an outdated commodity devices and the wireless hardware is inexpensive. It is now possible to economically create a wireless network by establishing a mesh of inexpensive hardware wireless devices, instead of trying to emulate these devices using software and commodity hardware. The largest advantages of this system are not requiring nearly as much time spent developing and adapting software to the system as well as removing the necessity to constantly develop, maintain, and update the software that supports the network itself. These changes would allow development to be focused on the development of more resources for use on the network, rather than being focused indefinitely on just supporting and creating the network.

Conclusion

At the end of the second semester, we did not yet have a functioning network within the community. We had, at that point, successfully implemented a small-scale testing environment within our laboratory that proved all of our concepts except for the reflector node, and allowed us to investigate many of the design considerations for the network in a very controlled fashion. This allowed us to solve several problems we expected, as well as discover other problems and find solutions to them before deploying a network. Many of the goals we developed at the beginning of our first semester were quite aggressive, and did not fully take into account the difficulties we would encounter during the initial phases of developing and building the network.

During the second semester, we began deploying are experimental network into the real world, to see how our results transferred from theory into practice. We deployed nodes at Professor Ralph Johnson's house, and then at his next door neighbor, Les Carlton. We discussed further expansion of the node installations with Chris Harris, who lived several houses down the street, but his concerns about security prevented an installation at his house. With the installations finished, we expected the process of testing and improving the network to be a straightforward one, but, as explored in the Implementation Issues portion of the document, this was not the case. Numerous troubles with our hardware, software, and overall project concept left us constantly behind schedule, and faced with many insurmountable problems. Despite seeking the assistance of experts, and working to resolve these problems as expediently as possible, the project ultimately failed to achieve its goal, and our wireless network is still little more in practice than a lot of theories and ideas, many of which failed in practice.

Even though we failed to achieve the final goals of the project, we do feel that we achieved a great deal in our own way. All of the members of the group are now intimately familiar with wireless technology, and have a far greater understanding of NetBSD and the hardware that we were working with. More important than this technical knowledge, however, is that our group has made great strides in learning about the importance of communication, seeking the help of experts, and valuing leadership and organization within a project. Because of our experiences working on this project, we will all be far more successful on projects in the future, and will be better able to recognize when our projects are not going well and we should seek help. Further, we now know the value of being frank with our managers and sponsors, as doing so makes the entire project run more smoothly, and prevents failure to deliver expectations. This knowledge will make us better team players in both the academic and business world in the future.

Glossary

Following is a list of terms used throughout the paper and their respective meanings. More information is also available throughout the document by referring to the footnotes along the way.

- **User -** A computer or Ethernet that sends and receives data over the wireless network. Also, the person in charge of such a computer or network.
- **End User -** A user who does not forward packets bound for other users over the wireless network.
- **Mesh/Grid Network -** A wireless network that is joined by a mesh of wireless antennas, creating a grid of connectivity. This type of network expands in two dimensions.
- **Routing User -** A user who forwards packets bound for other users over the wireless network.
- **Reflector Node -** The basic unit for one method of routing. A dedicated router with two wireless cards and an Ethernet interface. Each radio communicates with a different subnet and forwards packets between them.
- **Serial Network -** This is a type of network that consists of links connected in one dimension, forming a straight line down a path.