

*Technical report for*

**Wireless Internet Post Office:  
Providing Rural Access to Text based Digital Communication  
using Wireless Multi-hop Mesh Networking**

*Sponsored by*

**PAN ASIA ICT R&D Grants Programme**

August 29, 2003



**Department of Computer Science & Engineering  
Indian Institute of Technology, Delhi**

# Contents

<b>1</b>	<b>Overview</b>	<b>4</b>
1.1	Motivation . . . . .	4
1.2	Aim . . . . .	5
1.3	Project Description . . . . .	6
1.3.1	Internet Gateway Station . . . . .	7
1.3.2	Wireless Communication Architecture . . . . .	8
1.3.3	Wireless Communication Protocols for the Relay Stations . . . . .	9
1.3.4	Application Modules . . . . .	10
1.4	Cost Issues . . . . .	11
<b>2</b>	<b>Project Design</b>	<b>12</b>
2.1	Design of the Internet Gateway Station . . . . .	12
2.1.1	Hardware . . . . .	12
2.1.2	Software . . . . .	13
2.2	Design of Wireless Communication Architecture . . . . .	17
2.2.1	Overview of the IEEE 802.11, 802.11b specifications . . . . .	17
2.2.2	Use of Multiple Network Interfaces on WPO's . . . . .	21
2.2.3	Different Kind of Antennae . . . . .	21
2.3	Design/Selection and Adaptation of Wireless Communication Protocols . . . . .	22
2.3.1	Infrastructure vs. Ad-hoc Mode . . . . .	22
2.3.2	Routing Protocols . . . . .	23
2.3.3	Routing in Ad-Hoc Wireless Networks . . . . .	25
2.3.4	Description of AODV Routing Scheme . . . . .	26

---

2.3.5	Current Implementations of AODV . . . . .	28
2.4	Design of Application Modules . . . . .	30
2.4.1	Application Framework . . . . .	30
2.4.2	Components of application framework . . . . .	31
2.4.3	Communication interface between WPO and handheld . . . . .	32
<b>3</b>	<b>Current Setup</b>	<b>33</b>
3.1	Internet Gateway Station . . . . .	34
3.1.1	Hardware . . . . .	34
3.1.2	Software . . . . .	34
3.2	Wireless Communication Architecture . . . . .	35
3.2.1	Hardware . . . . .	35
3.2.2	Issues . . . . .	36
3.3	Wireless Communication Protocols . . . . .	39
3.4	Application Modules . . . . .	39
<b>4</b>	<b>Appendix</b>	<b>40</b>
4.1	Appendix I : Setting Up CISCO Wireless Client Adapter on Linux . . . . .	40
4.1.1	Installing drivers for the client adapters in Linux . . . . .	40
4.1.2	Configuration Using the CISCO Client Utility . . . . .	42
4.1.3	Configuration using iwconfig and ifconfig . . . . .	42
4.1.4	Monitoring the performance of the client adapters . . . . .	44
4.2	Appendix II : Infra Red . . . . .	45
4.2.1	Infra Red Configuration for Linux . . . . .	45
4.2.2	Infra Red Configuration for a Handheld running Windows CE . . . . .	48
4.3	Appendix III : Dynamic Host Configuration Protocol (DHCP) for Linux . . . . .	49

# List of Figures

1.1	Pictorial View of WIPO Idea . . . . .	6
2.1	Picture depicting non availability of path from a relay station to the Internet Gateway Station . . . . .	14
2.2	Managed or Infrastructure mode in wireless networks . . . . .	22
2.3	Ad-Hoc mode in wireless networks . . . . .	23
2.4	Picture showing RREQ/RREP message cycle in AODV . . . . .	26
2.5	Picture showing upstream & downstream nodes . . . . .	27
3.1	Picture showing multiple R/F links from the same node . . . . .	37

# Chapter 1

## Overview

### 1.1 Motivation

The project aims at designing and developing a prototype model for a Wireless Internet Post Office using the technologies from Ad-hoc Wireless Networks and Mesh Networks. The central concept behind this project is to replace the existing system of exchanging physical media such as letters and postcards with their electronic equivalents such as emails.

We propose in this document, a practical means by which we can connect remote villages together and bring them in touch with the digital age. For most people in rural areas, the only form of communication with the outside world is either through one-way mediums such as television or radio, or through the only two way communication mechanism available to them, i.e. telephone lines. However not all necessary information can be retrieved using these available communication models. Thus the need for connecting the rural areas with the data networks. Internet connectivity in such a situation may seem impractical and inappropriate, but this report presents a systematic way of bringing about the Digital Age for all people in a rapid and cost effective manner.

As in case of snail mail, in the proposed system also a postman distributes the mails by going around in the village. Here we envisage that the village postman or an official person designated by a NGO/charitable organization or any enterprising villager, who, using a PDA, distributes personal emails or other messages to the rest of the villagers. The messages can vary from weather

---

predictions to expert advice for protecting crops from various diseases to early warnings of natural disasters.

Economic issues which are relevant to a rural, agrarian society such as current, fair market prices of various goods can potentially revolutionise their economic standing in society. Informing villages of weather forecasts can now be done on a village-by-village basis. This kind of information will add value to the farmer community, increasing their efficiency and will enable the masses to gain from advances in digital communication. It is hoped that the state or national government can someday adopt this system, allowing us to utilize the wide network of (physical) post offices and post men which already exists.

## 1.2 Aim

The main focus in this project is to develop a prototype system for text based messages (which could be emails, news articles, bulletins etc) to be transmitted over the installed infrastructure wherever available and using off-the-shelf 802.11b hardware available for other locations. This system thus increases the penetrability into the remotest areas.

Extending the infrastructure to support higher bandwidths can then extend the same architecture on to provide more sophisticated applications. As we see today, there are already specifications extending the 802.11b architecture for higher bandwidths over the WLAN specifications e.g. 802.11a that already supports upto 54 Mbps is now available in the market.

In this system, the conventional post office is replaced with a Wireless enabled PC, a *Wireless-Post-Office* (WPO). Each such WPO is in direct communication with similar such *Wireless-Post-Offices* nearby through a wireless link forming a mesh network amongst themselves.

The *Wireless-Post-Office* machine also provides an interface to communicate with the handheld device for the postman/villagers, so that the emails/messages to be delivered or to be sent can be transferred to and from the handheld device. The postman/villagers thus connect their handheld

---

device to the nearest *Wireless-Post-Office* periodically to upload and download the requisite information and then goes around in the village distributing the information to the appropriate people, similar to the process of distributing postal mail.

### 1.3 Project Description

The figure below depicts the above application. Note that since each *Wireless-Post-Office* also relays the information for other nodes in the neighborhood we call them as relay stations, using the term interchangeably with *Wireless-Post-Offices*.

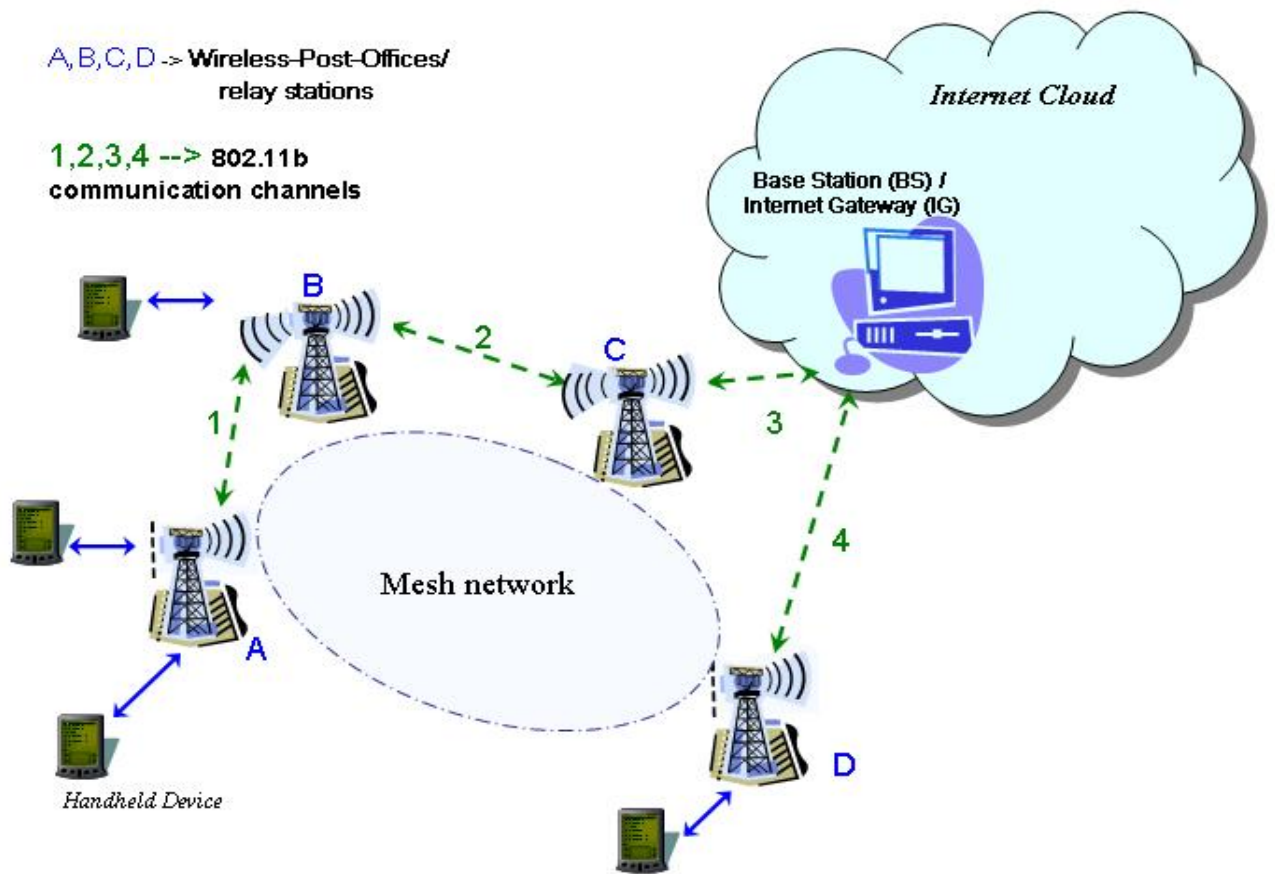


Figure 1.1: Pictorial View of WIPO Idea

---

In the above figure, green colored arrows indicate a 802.11b based R/F communication between the nodes, while blue colored arrows indicate a communication interface between the *Wireless-Post-Office* systems and the handheld device, which can be through infra-red or a direct physical connection similar to a cradle on a Palm-type device.

As a part of this project, we plan to design and develop the underlying technology for a Wireless Internet Post Office which can then be implemented across rural areas etc. The main modules being developed in this project are

- Internet Gateway Station
- Wireless Communication Architecture
- Wireless Communication Protocols for the Relay Stations
- Application Modules for the handhelds and the relay stations to up and down load the emails/text messages etc.

### 1.3.1 Internet Gateway Station

Note that in figure 1.1 the device named BS/IG acts as a gateway between the different *Wireless-Post-Offices* and the Internet. A gateway by definition is an entity used to interconnect two or more networks. In our case the two networks being Internet and the Wireless Mesh network formed by the Relay Stations. The wireless mesh network can have more than one Internet Gateway Stations.

- The basic task of this gateway/base-station is to relay the emails/text messages etc. between the handheld devices via the relay stations and the Internet. Apart from this, this gateway will also be responsible for
- IP Address allocation to the various relay stations.
- It will also be the Domain Name Server for the network formed by the relay stations.

For this purpose, it needs

- 
- At least one wireless interface to communicate with *Wireless-Post-Offices* e.g. the BS in figure 1.1 needs to talk to nodes marked C & D.
  - One network interface to connect to the Internet.
  - Software to route data between the Relay Stations and the Internet.
  - Necessary software and configuration to receive messages from the Internet on behalf of the Relay Stations and forward them to appropriately or store them for later retrieval by the relay station as need be.
  - Necessary software and configuration for it to act as a Domain Name Server for the network.

### 1.3.2 Wireless Communication Architecture

The only practical form of electronic communication between remote, isolated and scattered villages is a wireless one. The biggest advantage of a wireless communication mechanism is that it allows a node to fall off-line or out-of-touch with the rest of the group, without adverse effects to any other. Telephone and power lines, in contrast, cannot recover by themselves if an intermediary node in their infrastructure fails. It is possible to shut down power to a village if inclement weather is expected, but not possible to wrap up wires spread over kilometers. For these reasons, a wireless communication medium becomes a realistic means of communication between the different *Wireless-Post-Offices*. The average distance between two villages is estimated to be 3-4 kilometers. Since we are focusing on text based messaging and email application, the bandwidth requirements from these nodes is not high and a bandwidth of 2-3 Mbps would serve several PDAs at a time and also provide the necessary reliability in the system.

The wireless communication architecture thus comprises of

**Network between Relay Stations** Relay Stations do not have a direct connection to the Internet. They sit on the inside of the wireless mesh network and exchange messages with the Internet Gateway Station on behalf of the handheld devices. These relay stations will use 802.11b based communication mechanism amongst themselves.

---

**Wireless Interface for Internet Gateway Station** The Internet Gateway Station communicates with the relay stations using a wireless interface (802.11b).

As a part of this study we need to establish the usability of 802.11b wireless communication technology for such an application. This would involve research and development of the following components:

- A study for the hardware modules like antennas/cables etc will have to be carried out in order to use off-the-shelf 802.11b infrastructure for the purpose. Although the ranges we propose are longer than commonly used, available equipment makes it possible without significant problems.
- Necessary configuration and system set-up for validation of the different components.
- Addressing the issues of interference due to multiple communication links on each *Wireless-Post-Office*.
- Following this a study combined with experimentation is required to evaluate the performance of these networks over the distances desired.

### 1.3.3 Wireless Communication Protocols for the Relay Stations

A wireless medium provides the flexibility for the nodes present in the system to move around, this also implies that the nodes can join or leave the network at will, thus forming what is known as an Ad-Hoc network. Note that nodes falling off-line due to power failures or environmental disturbances add unforeseen dynamism to the network, because effectively these nodes are also leaving and subsequently joining the network at random, unpredictable times. Also to reduce power consumption by the system, we do not want all *Wireless-Post-Offices* to be powered and running all the time. We can thus consider a *Wireless-Post-Office* without power as a node which has moved out of the network and is no longer available. So in order to handle such scenarios, there is a crucial need for appropriate communication protocols which can handle and adapt to such network changes.

The two main components of this network are:

- 
1. Configurability of the nodes, which includes IP address allocation for each of the *Wireless-Post-Offices*. One of the schemes which is commonly used for the configuration in ordinary PC LAN systems is a Dynamic Host Configuration Protocol (DHCP) based scheme. This scheme is not intended for use on Mobile Ad-Hoc Networks however this can be used for configuration purposes in the first stage. Other schemes proposed for Mobile Ad-Hoc Networks can also be tested if the need arises.
  2. Routing mechanism to route the data across the network in the dynamic sense. In order to facilitate communication within the network, a routing protocol is used to discover the routes between nodes. Since the network is changing continuously, this calls for routing schemes that can discover routes adaptively in case of network changes such as some nodes along a path going down due to power failures etc without many overheads Some such schemes have already been proposed for Mobile Ad-Hoc wireless networks namely "Dynamic Source Routing", "Ad-Hoc On Demand Vector Routing", "Temporally Ordered Routing Algorithm" etc. The paper in [12] compares some of these routing mechanisms for wireless ad-hoc networks.

Two implementations for Ad-Hoc On-Demand Distance Vector Routing (AODV) are already available and have been tested out for some of the scenarios. As a part of the project, one needs to evaluate the applicability and performance of the two scheme for the above application.

### 1.3.4 Application Modules

In this module, we will look at schemes to deliver email, messages and other such content over the wireless communication architecture. The idea is to be able to deliver atleast text-based information content from the Internet to people in rural areas.

In addition to the conventional method of accessing emails, where in all the emails are stored on to a central server and fetched on demand, theres a need for distributing the emails available at the server to the appropriate WPOs.

The different subtasks of the application module are:

- Email/Information retrieval from the server to the relay stations and finally to the handheld devices.

- 
- Email/Information sending capability from the various relay stations and/or handhelds
  - User interface on the handheld device for composing, sending and reading emails or messages
  - A communication interface between the relay stations and the handheld device.

## 1.4 Cost Issues

Since one of the main motivations of the project is to build a cost effective and economical solution, we will be using off the shelf 802.11b hardware available in the market (below \$100) for the communication architecture. This hardware coupled with the directional antennas can provide a communication range of a few kilometers. Note that these antennae are also available within \$150.

For the PC's to go on into the *Wireless-Post-Offices* a single board computer with a battery back up should serve the purpose. There are many such platforms available in the market with the price range varying from \$100 to \$300. The design and system issues in integrating these systems to run the wireless architecture and using the Linux platform reliably would be one of the targets of the project.

# Chapter 2

## Project Design

### 2.1 Design of the Internet Gateway Station

The Internet Gateway Station (or Base Station) is a PC Server and email relay connected to the Internet and acts as the gateway between the Internet and the roaming PDA's via relay stations. Since text messages do not consume much bandwidth or storage, a single PC can serve thousands of PDAs distributed over a large area covered by a couple of hundred relay stations. Some measurements showed that a desktop machine (Intel PIII CPU @ 800 MHz, 256 MB RAM, 100 Mbps LAN connection, a 20 GB Hard disk) is able to handle more than 15000 mails on a daily basis, while the average load on the machine (as given by the uptime utility in Linux) remained below 0.04. In the subsequent two sections we describe the hardware and software requirements for the implementation of the Internet Gateway Station. Note that there could be more than one Internet Gateway Station for one such WIPO network as shown in figure 1.1, however the hardware and software configuration defined below will be sufficient to handle such a scenario with appropriate setup.

#### 2.1.1 Hardware

In the simplest form the gateway could be a simple desktop PC running Linux with a Internet connection which should work for most scenarios, in case of a very large deployment (> 1000 relay stations) a higher performance PC server with a high speed dedicated Internet connection will

---

suffice. As identified previously, this PC will have atleast one wireless interface to communicate with the relay stations and atleast one interface to connect to Internet. The wireless interface would be in the form of a 802.11b compatible adapter, while the medium for Internet could be either through a ISDN connection or a LAN based access, as is available.

### **2.1.2 Software**

The Internet Gateway Station will work with Linux platform. The following subsections cover the various software modules required on this machine to provide the necessary functionalities desired from the gateway.

#### **E-Mail Exchange Server**

For providing email access on the handheld devices through the network formed by the relay stations and the Internet Gateway Station, we will use an email server called sendmail on the gateway machine.

For sending mails, the handheld device will send the composed email to the relay station (IrDA or Serial/USB) which will be running a SMTP server. The relay station in turn will forward this mail to the Internet Gateway Station which is the main mail server with all users having their primary mailboxes there. The main mail server will also acts as the SMTP server for the mails from different relay stations and destined outside the WIPO network. This email server will be responsible to receive all the emails destined for different PDA devices in the network(for which this machine serves as the gateway).

sendmail is a powerful electronic mail transfer agent and a version has been available for most UNIX based operating systems. In addition, it is open-source and then fully customizable. "sendmail" uses Simple Mail Transfer Protocol(SMTP) for sending emails and hence the mail sending server is also referred as SMTP server. We will configure sendmail as the email server on this Linux machine. One important reason for using sendmail as the email server software was the familiarity with the system due to usage in the Networking Laboratory for quite some time. The exact version and detailed setup of the "sendmail" server has been provided as a part of the current setup chapter.

More detailed information about sendmail can be obtained from referring to [2] & [8].

Note that it could happen that the relay stations may not be able to reach the mail server at

a particular instant due to non availability of some node on the path to the mail server (Internet Gateway Station in our case). Refer to figure 2.1, when node B is powered off, node A loses its connectivity to the Internet Gateway station or the mail server. However, the defined application framework handles this scenario by providing a mechanism to download all the available emails on the server whenever a connectivity is available.

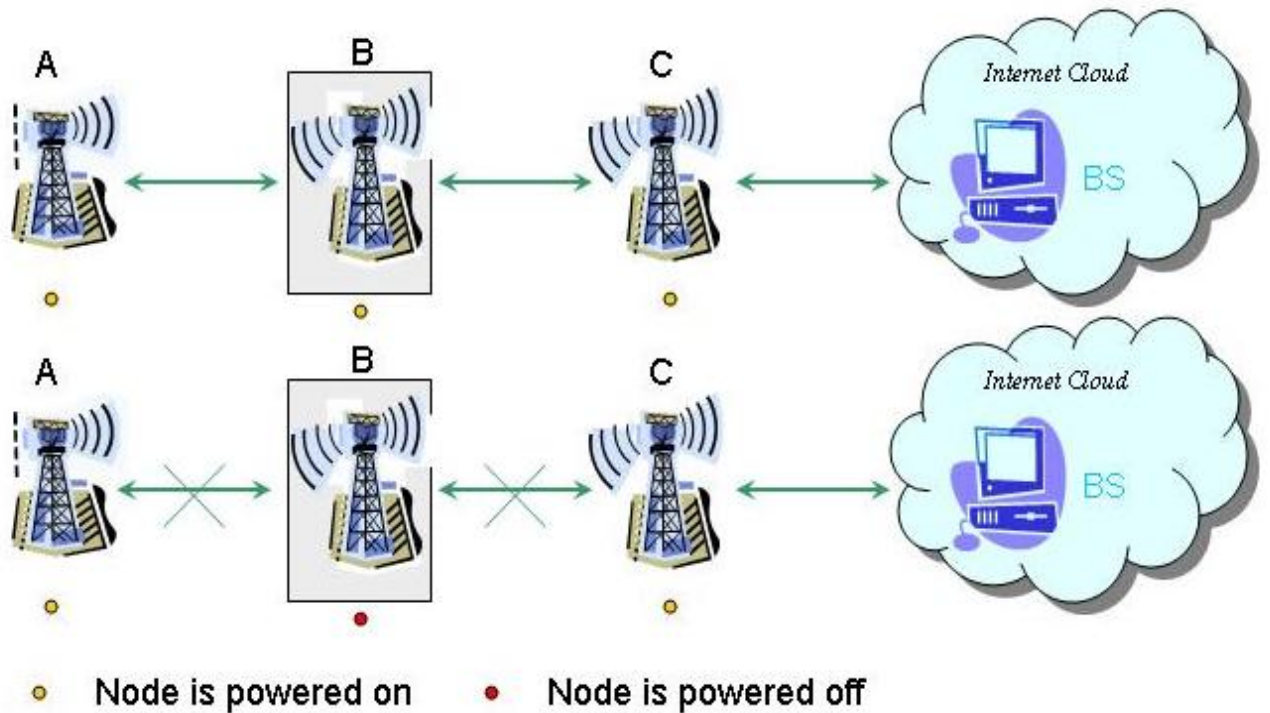


Figure 2.1: Picture depicting non availability of path from a relay station to the Internet Gateway Station

### POP 3 Server

In our design, the emails for users would be stored on the gateway station until the appropriate relay station downloads the emails. In order to allow relay stations to download emails from the mail server, the mail server will support the Post Office Protocol 3 (POP3).

The downloading relay station in turn will act as a POP3 server for the handheld devices who want to download the emails/text messages for the users of that handheld device.

---

## Domain Name System

A domain name system is used to convert the hostnames (along with their domains) to IP addresses for any machine connected to the network. It maps from a hostname to an address or from an address to a hostname. A similar system will be required for the network formed by the Relay Stations as the email distribution from the Internet Gateway Station will be based on the domain name allocation as per our design. This implies that when an email is received at the Internet Gateway Station for a user in the WIPO network, the email will be delivered to the appropriate handheld via the corresponding relay station. This mapping of the email addresses to the appropriate relay stations is being done using domain name system. This framework is described in the section on Design of Application module.

We will configure the gateway machine to be used as a domain name server for the mesh network formed by the relay stations. More information is also available from the Linux DNS-HOWTO [3].

In this case also, we will encounter the problem due to lost connectivity to the Internet Gateway station, refer to figure 2.1. This can be worked around by using multiple DNS servers within the network. Some of the more reliable relay station nodes can also be configured as secondary name servers.

## IP Address Allocation server

In our design all the nodes of the network use IP as the communication protocol at the network layer. In IP each node of the network is identified with a locally unique IP address (as long as they are not connected to the Internet directly e.g. relay stations), where by locally unique we mean that no two nodes in the same network can have the same IP address.

Note that in the WIPO application, to provide a reliable and easy-to-setup communication mechanism we need to reduce the configurability aspects of the network to a bare minimum. The two choices available to us for IP address allocation of the relay stations are:

- A Static IP address assigned when the relay station is first configured. The relay station then uses this same IP address for communication whenever it boots up.
- The other mechanism is by using dynamic IP address allocation scheme, in which case whenever a node is powered on, it gets an IP address allocated to itself. The IP address assigned

---

to a machine at power-on maybe different from the IP address allocated to this machine previously i.e. before it was powered off last. Several schemes have been proposed for dynamic IP address allocation. The most commonly used of them being Dynamic Host Configuration Protocol. Refer to section 4.3 for more details.

In our case, we should first use a static IP address allocation scheme for the testing purposes. There are some issues with using a static IP address allocation scheme, such as - the scalability of the system reduces since we cannot reuse the IP address once allocated even if that relay station is not powered on at this instant. This scheme will require the static IP address of the relay station to be configured whenever the relay station is installed.

Thus for the purpose of IP Address allocation to the various relay stations, we are currently judging the suitability of the Dynamic Host Configuration Protocol. More details on DHCP are available from [4].

The Internet-gateway machine should be configured as a server for DHCP. The server has to be configured with a pool of available IP addresses, that it can use while allocation IP addresses to its clients. Each machine that is powered on tries to contact this server and asks for an IP address to be allocated. The server then selects an IP address from the pool of available IP addresses and informs the client of its newly allocated IP address. Note that it may happen that the node that has been powered on is not able to contact the DHCP server. In such a scenario, the machine may not be able to communicate with neighbors etc. due to lack of an IP address. This problem has been discussed in detail below. A detailed how-to to setup the Dynamic Host Configuration Protocol on a server as well as a client is included in section 4.3 and a detailed howto is available from [4].

As mentioned above, DHCP based scheme would fail if the relay station being powered on is not able to contact the DHCP server due to non-availability of some relay station at that point of time on the path to the DHCP server (in our case the Internet Gateway Station). In such a case, the requesting relay station would not be able to obtain an IP address for itself. This is depicted in the figure 2.1 above

To overcome this difficulty, other schemes proposed for IP address allocation for Mobile Ad-Hoc Networks should be considered. Note that Mobile Ad-Hoc networks, which inherently have a high amount of dynamism, cannot use the DHCP type of scheme for the purpose of IP address allocation primarily because it would result in a single point of failure. Two of the schemes proposed for the

---

IP Address allocation problem in Mobile Ad-Hoc Networks that can be evaluated are:

1. IP Address Autoconfiguration for Ad Hoc Networks, proposed by Charles E. Perkins, Jari T. Malinen et al. This scheme has been proposed as an Internet-Draft to the Internet Engineering Task Force.
2. Configuration in a Mobile Ad-Hoc Network, proposed by Sanket Nesargi and Ravi Prakash.

Another possible scheme is one based on the hardware addressing of the network interface cards. Note that each network interface card has a globally unique hardware address called the MAC address. Since we will always be using standard (off-the-shelf) network interface cards for connectivity, we can use their MAC addresses to generate a unique IP address for each node in the network of relay stations. The only problem with this scheme is that some of the network interface cards allow changing the MAC address of the card by reprogramming the EEPROM. Thus the uniqueness of the MAC addresses cannot be guaranteed.

## **2.2 Design of Wireless Communication Architecture**

### **2.2.1 Overview of the IEEE 802.11, 802.11b specifications**

A network where a mobile user can connect to a Local Area Network using wireless (radio) communication is generally referred to as Wireless Local Area Network (WLAN). 802.11 is a family of specifications for Wireless Local Area Networks developed by a working group of IEEE. The different specifications in the family are: 802.11, 802.11a, 802.11b and 802.11g, 802.11i and 802.11x. Of these 802.11, 802.11a, 802.11b are developed and commercial systems based on them are available in the market, while 802.11i and others are in different stages of development. 802.11, 802.11a, 802.11b and 802.11g use the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for path sharing across various nodes in the network.

The earlier 802.11 specification provided 1 or 2 Mbps data rates and worked in the unlicensed 2.4 GHz radio band. The modulation used was either Frequency Hopping Spread Spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS). FHSS is the repeated switching of frequencies during

---

radio transmissions, often to minimize the effectiveness of any attempts on interception or jamming of the signal. DSSS has been explained below.

Wi-Fi (short for wireless fidelity ) is a cheap and fast way of connecting computers to each other using high frequency wireless communication in a local area network. The term Wi-Fi is normally used interchangeably with IEEE 802.11b specification. 802.11b networks operate in the unlicensed 2.4 GHz radio bands, with an 11 Mbps data rate (supports fallback to 5.5, 2 and 1 Mbps for backward compatibility with 802.11).

IEEE 802.11b uses Direct Sequence Spread Spectrum (DSSS) encoding. DSSS works by taking a stream of ones and zeroes and modulating it with a second pattern called the chipping sequence. In 802.11 this sequence is known as the Barker code, a 11 bit sequence (10110111000) is ideal for modulating radio waves due to certain mathematical properties. Complementary Code Keying (CCK), used in 802.11b uses a series of codes called Complementary Sequences, which allows higher data speeds and is less susceptible to interference in multi-path propagation. More details on modulation schemes can be obtained from [1].

Recently a new specification called the 802.11a has also been included in the broad Wi-Fi networks definition. As compared to 802.11b, 802.11a works in the unlicensed 5 GHz radio band. 802.11a uses a modulation scheme called the orthogonal frequency division multiplexing (OFDM). This modulation scheme makes it possible to achieve data rates as high as 54 Mbps. Note that since 802.11a works in the 5 GHz range, 802.11a is not backward compatible with 802.11b hardware. While 802.11g provides the same throughput (54 Mbps) as the 802.11a it also provides backward compatibility to the currently dominant 802.11b standard as it works in the 2.4 GHz radio band.

Since WLAN's use a wireless medium for communication, which is essentially a broadcast, security in such networks becomes critical. Signals travel through the air, and can be trapped by anyone having the required equipment. In order to avoid such eavesdropping, some security mechanism becomes indispensable. Basic WLAN has a standard for security called Wired Equivalent Privacy (WEP). WEP algorithm is used to protect wireless communication from eavesdropping and

---

also prevent unauthorized access to the network in question. This scheme has some flaws and is easily breakable. More information about these loopholes can be obtained from [5]. However some enhancements to the scheme to make it more secure have been implemented by different manufacturers. But since there is no standard for security, hardware from different vendors poses an interoperability risk more often than not.

While there are other developments like user authentication etc. to increase the security of the network, 802.1i and 802.1x promise to overcome this problem of security with the current standards. However these standards are still emerging and analysts say that it would be at least two years before we see any commercial hardware based on these standards. The two most commonly used user authentication schemes are Lightweight Extensible Authentication Protocol (LEAP) and Protected Extensible Authentication Protocol (PEAP).

Hardware complying to the 802.11a, 802.11b standards are available from various vendors like Cisco, Intel, Proxim, 3Com, Avaya and many more. This hardware is mainly in the form of an extension card easily pluggable into notebooks, hand-helds, desktop PCs and other such devices on the node end and in form of access points for the operations in the infrastructure mode(explained below). Two of the most commonly used interfaces for the extension cards are the PCMCIA & USB interfaces.

With the adoption of the technology, and reduction in costs with time, more and more devices are now coming out with embedded chips for Wi-Fi networks and hence provide seamless communication capabilities in mobile environments.

The 802.11 nodes in any WLAN can work in two modes, **Ad-hoc** (peer to peer communication mode), or **Infrastructure** (access point mode). The need for these two modes of communication for mobile nodes as explained below in the section on Design/Selection of Wireless Communication Protocols. The routing requirements for the two modes are also different and have been explained in a section below.

The 802.11b standard defines 14 frequency channels in the 2.4 GHz radio band.

Channel	Center Frequency (GHz)
1	2.412

---

Channel	Center Frequency (GHz)
2	2.417
3	2.422
4	2.427
5	2.432
6	2.437
7	2.442
8	2.447
9	2.452
10	2.457
11	2.462
12	2.467
13	2.472
14	2.484

#### **802.11b Channel frequencies**

A channel actually represents the center frequency that the transceiver uses (e.g. the center frequency for channel 1 is 2.412 GHz and that for channel 2 is 2.417GHz). The separation between two adjacent center frequencies is only 5MHz (2.417GHz - 2.412GHz), however an 802.11b signal occupies approximately 30 MHz of the frequency spectrum. The signal occupies approximately 15 MHz on each side of the central frequency. This clearly indicates that each 802.11b signal overlaps with several (at-least 3 on each side) adjacent channel frequencies. So if we want to use multiple channels in the same proximity, in order to reduce interference one will need to select the channels as far apart as possible i.e. ideally channel 1, channel 7 and channel 13 should be used for avoiding interference between three simultaneously active channels in the same proximity. However note that various countries limit the use of these channels (U.K allows only channels through 1 to 13, Japan allows use of all 14 channels and so on).

802.11b specifications define a network id(called ESSID in case of Cisco cards), configurable for each 802.11b interface. This id is used for the Ad-Hoc mode operation of the cards. Only nodes with the same network id can communicate with each other.

---

### 2.2.2 Use of Multiple Network Interfaces on WPO's

As depicted in 1.1 above, some of the *wireless-post-offices* (like node A and node D) communicate with only one node each (node B and C respectively). However, note that some of the *wireless-post-offices* (like B & C) communicate with multiple *wireless-post-offices*. In order to be able to talk to multiple *WPO*'s simultaneously, nodes such as B & C require multiple wireless network interfaces. However if the multiple interfaces use the same frequency, simultaneous transmissions on different links from different interfaces will interfere with each other. The various schemes possible to avoid this problem are

1. Use of different 802.11b channels and hence different frequencies for different interfaces
2. Use of 802.11b network ids to separate the two networks on two interfaces(notice in this case the interference will still reduce the available bandwidth but will not mangle the data on the two different links)
3. Use of filters (available in iptables, netfilter [7]) to separate the two networks on two interfaces.

### 2.2.3 Different Kind of Antennae

There are various kinds of antennae available in the market for the 2.4 GHz 802.11b cards. The different types of antennae are designed to perform in variety of environments. The right choice of the antennae for an application can greatly improve coverage and performance. The three basic features provided by any antenna are gain, direction and polarization. More the power of the antenna, higher is the gain.

But as the gain increases, there is some tradeoff in the coverage area of the antenna. Directionality of the antennae determines the transmission pattern in the wireless system. The polarization of the antenna is rated in comparison to the isotropic and dipole antennae. More detailed information about the polarization ratings is given in [9].

The antennae are broadly classified as

1. Omni-directional antennae, providing a 360 degree transmission pattern and hence equal range in all directions.

- 
2. Directional antennae, the coverage pattern is dependent on the shape of the antennae. Some of the common directional antennae are Yagi antenna, patch antennas, parabolic dishes etc. Directional antennae basically redirect the signal (energy) in one particular direction, thus increasing the range in that particular direction.

## 2.3 Design/Selection and Adaptation of Wireless Communication Protocols

### 2.3.1 Infrastructure vs. Ad-hoc Mode

There are two approaches that allow two wireless stations to communicate with each other. The first one is to introduce a third fixed party (a base station) that will hand over the offered traffic from a station to another, as illustrated in Figure 2.2 below e.g. the towers named BTS acts as the base station between the source and destination pairs(e.g. BTS1 server source S and destination D1, BTS2 serves source S and destination D2). This same entity will regulate the attribution of radio resources, for instance. When a node S (source) wishes to communicate with a node D (Destination), the former notifies the base station, which eventually establishes the communication with the destination node, via another base station if need be, or directly if the destination is within the reach of the first base station. At this point, the communicating nodes, in our case S & D1/D2, do not need to know of a route for each other.

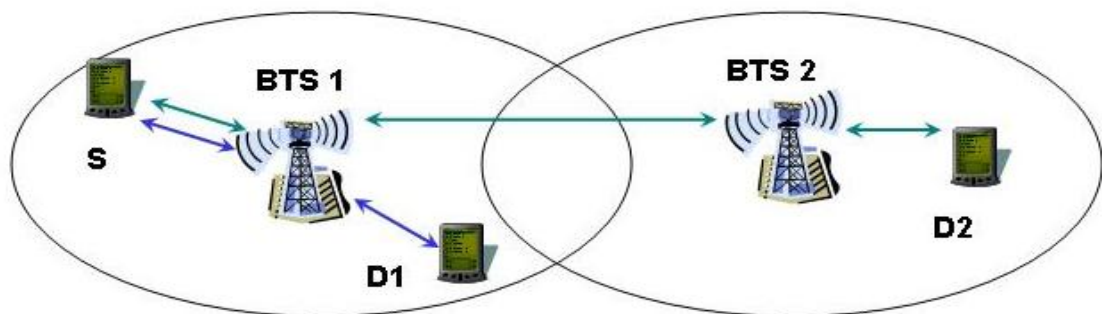


Figure 2.2: Managed or Infrastructure mode in wireless networks

---

The second approach, called ad-hoc, does not rely on any stationary infrastructure. The concept behind these infrastructure less networks is the collaboration between its participating members, i.e., instead of making data transit through a fixed base station, nodes consequentially forward data packets from one to another until a destination node is finally reached. Typically, a packet may travel through a number of network points before arriving at its destination. E.g. in the figure 2.3 below, node S (source) sends the data to the node D1/D2 (destination) through one or more similar nodes. The in between nodes are called as intermediate hops. In this case source S needs to know that the route to destination D is via which intermediate hop.

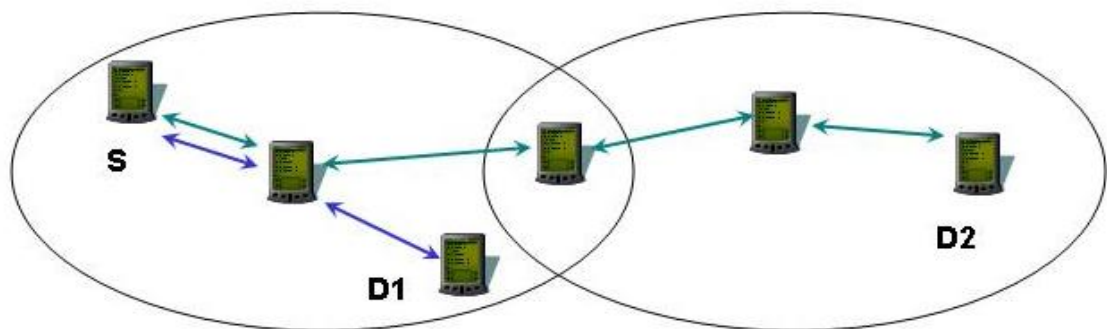


Figure 2.3: Ad-Hoc mode in wireless networks

Because of the improvised nature of Ad-Hoc networks, routes are built dynamically as and when nodes are regrouping (due to mobility). Each node in an Ad-Hoc network keeps updating the information about its neighbors. Hence Ad-Hoc networks are more responsive to changes in topology as compared to the infrastructure mode networks.

### 2.3.2 Routing Protocols

A number of protocols have been in use for routing of information between the network nodes based on the application in use. This is because a single routing scheme does not give optimal output under different communication mediums and different scenarios e.g. a routing scheme designed optimally for wired networks may give very poor performance when used for wireless communication.

One of the classification used to classify different routing schemes proposed is on the basis of whether it is a proactive or a reactive scheme. The differences between the two categories are as

---

follows

### 1. Proactive Schemes

- (a) Determine the routes to various nodes in the network in advance, so that the route is already present whenever needed.
- (b) Route Discovery overheads are large in such schemes as one has to discover all the routes.
- (c) Consumes bandwidth to keep routes up-to-date
- (d) Packet forwarding is faster as the route is already present.

Examples of such schemes are the conventional routing schemes, Destination Sequenced Distance Vector (DSDV).

### 2. Reactive Schemes

- (a) Determine the route when needed
- (b) Smaller Route Discovery overheads.
- (c) Employs flooding(global search)
- (d) A node trying to transmit a packet may have to wait for route discovery.

Examples of such schemes are Dynamic Source Routing, Ad-Hoc On Demand Distance Vector Routing (AODV) etc.

Different proactive, reactive and even hybrid (reactive as well as proactive) schemes e.g. Zone Routing Protocol, have been proposed for Wireless Ad-Hoc networks. The main challenges for any routing protocol for Wireless Ad-Hoc Networks are

- Lower routing related overheads
- Discovering the optimal routes (may be different than shortest routes depending on the constraints of the application)
- Discovering stable routes yet allowing mobility of nodes

---

However, in general, reactive routing schemes are more common because of low routing overheads. Ad-Hoc Wireless Networks when used in compact handheld devices are limited in hardware capabilities in terms of storage, communication and power, making reactive schemes ideal for this type of application.

### 2.3.3 Routing in Ad-Hoc Wireless Networks

As described in [12], routing protocols for wireless ad-hoc networks are generally classified as:

**Table Driven Routing Protocols** In each of these protocols up-to-date routing information from each node to every other node in the network is maintained on each node of the network. The changes in network topology are then propagated in the entire network by means of updates. Destination Sequenced Distance Vector Routing (DSDV), Wireless Routing Protocol (WRP) are two schemes classified under the table driven routing protocols head.

**Source Initiated On-Demand Routing Protocols** As the name suggests, the routing protocols classified under this category, create routes only when desired by the source node. When a node requires a route to a certain destination, it initiates what is called as the route discovery process. This process basically comprises of packets with a description of the destination (address information of the destination etc.) being forwarded from one hop to the next. Any node receiving such a request, looks into its available routing table to find if it has a route to the described destination. If a route to the destination is present, the node returns this route to the source and the process ends else the request packet is forwarded to its neighbors continuing the route search process. Once a route is found, it is temporarily maintained in some form (typically the routing table) and then subsequently removed after either a timeout, or if the destination node leaves the network etc. Some of the schemes classified under this head are Ad-Hoc On Demand Distance Vector Routing (AODV), Dynamic Source Routing (DSR), Temporally Ordered Routing Algorithm (TORA) etc.

Some of these schemes have been studied in detail as part of various student projects and have been evaluated. Based on these studies we have selected AODV to be used for the purpose of routing between the relay stations.

---

### 2.3.4 Description of AODV Routing Scheme

Ad-Hoc On Demand Distance Vector Routing (AODV) is an on demand algorithm, which means that it builds routes between nodes only when desired by the source node. It maintains these routes as long as they are needed by the source. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting and scales to large numbers of mobile nodes. AODV builds routes using a route request/route reply query cycle. The figure 2.4 shows a very simplistic view of the route request/route reply cycle.

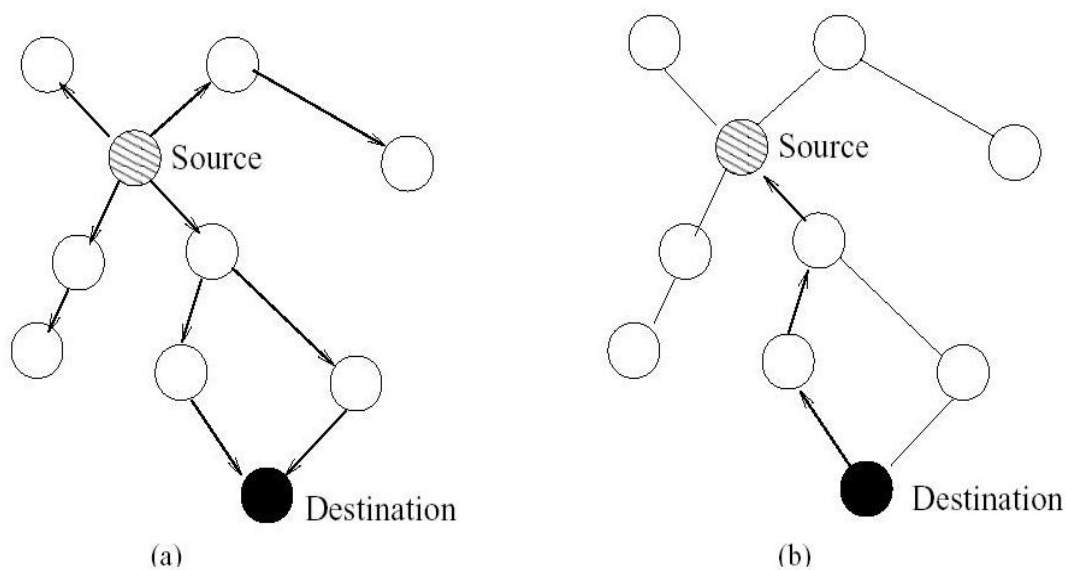


Figure 2.4: Picture showing RREQ/RREP message cycle in AODV

When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. This broadcast is received by all the nodes in the vicinity of the source then. In addition to the source node's IP address, current sequence number, and broadcast ID(used to control the amount of flooding in the network), the RREQ also contains the most recent sequence number for the destination of which the source node is aware. Nodes receiving this packet update their information for the source node and set up backward pointers to the source node in the route tables. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with

---

corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

As the RREP propagates back to the source, the intermediate nodes also set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables.



Figure 2.5: Picture showing upstream & downstream nodes

Nodes monitor the link status of next hops in active routes. If a link break occurs between any two nodes while a route on that link is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After

---

receiving the RERR, if the source node still desires the route, it can re initiate route discovery.

Route table information must be kept even for ephemeral routes, such as those created to temporarily store reverse paths towards nodes requesting RREQs. AODV uses the following fields with each route table entry:

- Destination IP Address
- Destination Sequence Number
- Hardware Interface (such as eth0, eth1 etc)
- Hop Count (number of hops needed to reach destination)
- Last Hop Count
- Next Hop for the destination
- List of Precursors
- Lifetime (expiration or deletion time of the route)
- Routing Flags

Managing the sequence number is crucial to avoid routing loops, even when links break and a node is no longer reachable to supply its own information about its sequence number. A destination becomes unreachable when a link breaks or is deactivated. When these conditions occur, the route is invalidated by operations involving the sequence number and metric (hop count).

### **2.3.5 Current Implementations of AODV**

An implementation of Ad-hoc On-Demand Distance Vector Routing protocol used in Mobile Ad-Hoc Networks, developed by Uppsala University [11], Sweden has been studied and tested in the Ad-Hoc network setup in the lab. This implementation is available under the GNU General Public License. We have worked with version 0.5 of this AODV package.

---

Another implementation of AODV developed by National Institute of Standards and Technology (NIST) available at [10] has also been studied, configured and tested. The main differences between the two implementations are as follows

1. AODV implementation from Uppsala University

- This AODV implementation runs as a user-space daemon, maintaining the kernel routing table.
- Netfilter is used to capture data packets.
- Filtering is done in user-space, so there may be some performance penalties.
- Stable operation has higher priority than performance.
- The code has been successfully tested in a real ad-hoc environment using up to 5 nodes (4 hops) without problems.

2. AODV implementation from National Institute of Science & Technology

- Managed Internet gateway.
- Monitor wireless signal strength.
- Fixed buffer size for /proc Route Table.
- Fixed Expanding Ring Search.
- Supports multiple interfaces.
- Since it is a Kernel module it runs in the Kernel space instead of the user space, which allows for better access to resources.
- Uses Netfilters from the 2.4 Kernel to capture packets going in and out of the node instead of using the libcap library.
- Uses a Proc file to update the user about current routes and statistics for that node.

These implementation have been tested for a single network interface devices, however to use it for the WIPO application, we need to modify this implementation to work for multiple network interfaces (note that some of the wireless-post-offices will have multiple wireless network interfaces)

---

## 2.4 Design of Application Modules

The design of the application module will incorporate the application features like email transfer and retrieval, retrieval of text messages from the Internet to the handheld devices carried by the postman etc.

### 2.4.1 Application Framework

The application framework has been designed using email as the base for all text message exchanges. Under this framework every user has 2 mailboxes. One is the primary Mailbox, which is located on the internet gateway station. The other is the secondary Mailbox, which is located on one of the RS, depending on the location of the user.

The handheld device runs an email client like Outlook Express. This email client is configured with the details of the user account(s) for which it is to send and retrieve emails. The accounts domain is the domain of the network. It uses Simple Mail Transfer Protocol (SMTP) to send mails to the SMTP server and uses Post Office Protocol 3 (POP3) to retrieve mails from the POP3 server. It accesses the secondary Mailbox on the relay station while doing this.

It is the responsibility of the relay station to which the handheld device is connected (through IrDA or serial/USB), to relay all mails to the gateway, and also periodically retrieve all emails from the gateway for all users having a secondary Mailbox on it. Note that finally its the relay station who has to allow the handheld device to send emails and download the newly received emails from the users secondary mailbox, for this each relay station will also have to work as SMTP and POP3 server.

The gateway is the node, which receives all emails, whether for internal domain or any external domain, where an external domain means for some user outside the WIPO network. An email for an external domain is relayed to the appropriate destination. An email for a host in the internal domain is kept in the users account(primary mailbox) on the gateway. All emails from the outside network are received on the gateway. The appropriate relay stations would retrieve these periodically and

---

whenever the complete path is available.

The reasons for working with the above application framework in case of outgoing and incoming mails are:

### **Outgoing Mails**

- Since there's a possibility of the handheld not being able to reach the Internet Gateway Station during some period, if the emails to be sent are uploaded directly to the main server, the emails would be held in the PDA itself.
- But, with limited memory and storage available on the PDA, we cant keep on storing the outgoing mails on the PDA for a very long time.

Hence this framework has been developed so that the handheld device uploads all the emails to the relay stations, who in turn either stores it or hands it over to the Internet Gateway Station whenever a path is available.

### **Incoming Mails**

- A users primary mailbox is maintained on the Internet Gateway Station , which receives the mails destined for the user. Further on, whenever the relay station corresponding to that user has a complete path to the Internet Gateway Station, these mails are downloaded to the users secondary mailbox.
- This approach does not depend on the power availability along the entire path from the handheld to the gateway, whenever it wants to read mails. This approach reduces the power consumption of the relay stations considerably.
- Also this allows a user to connect to a particular relay station(who downloads the mails for this user) through any other nearby relay station or from the same.

## **2.4.2 Components of application framework**

**Email/Information retrieval from server** For the purpose of email retrieval from the mail server on to the relay stations, we will use Post Office Protocol 3 (POP)on the server. This

---

module will also involve the design of a mechanism for email address assignment for the users. The email addresses will also depend on the domain name allocation for the WPOs.

**Email/Information retrieval from relay station** For the purpose of email retrieval on the handheld devices, the relay stations will have to serve as POP3 servers for them.

**Email/Information sending** For the purpose of sending mails, we will be using sendmail utility.

**User interface on handheld** For the user interface, we will initially work with the Windows based environment on the iPAQ (handheld) devices. A utility similar to Outlook Express for reading, composing and sending mails is already available in the default installation. An iconized user interface for some of the other applications (like crop prices, plant diseases) can be designed using Linux platform. Note that Linux port on the iPAQ devices is already available. Some of the other common handheld devices have also started supporting Linux platform.

### 2.4.3 Communication interface between WPO and handheld

We will work with the IrDA interface (available on almost all the common handheld devices) as a medium for communication between the WPOs and the handhelds. Note that IrDA interface is not so commonly available with recent desktop PC platforms. However IrDA extension cards for serial as well as USB ports are commonly available. We are using an old IBM laptop with IrDA interface for our experimentation. Optionally, communication between the handheld and WPOs can also be using:

- Serial port based communication, most of handheld devices provide a serial port interface embedded in the docking station/cradle.
- USB port, available with the docking station/cradle

# Chapter 3

## Current Setup

The setup currently being used for the testing purposes in the laboratory makes use of the following hardware

- Three Pentium III machines with 128 MB RAM and 20 GB hard disks each. Two of these nodes have three network interfaces while the third machine has two network interfaces. Each of these machines has one wireless 802.11b PCI network card and the other cards are 10/100 Mbps Intel Ethernet LAN cards. Note that for this application, we only need one wireless and one landline LAN interface on all these machines.
- One Compaq Laptop with 128 MB RAM and a 20GB hard disk. This node also has one wireless network interface in the form of a Cisco 350 series PC Card and a 10/100 Mbps Ethernet adapter.
- One IBM laptop with similar network architecture and an IrDA interface.
- A pair of yagi antennae
- Cisco 350 Series PCI/PCMCIA 802.11b client adapters.

The Yagi antennae have been tested using two desktop machines. The 350 Series Aironet PCI cards come with a detachable antenna and the connector is RP-TNC. The Yagi antennas, which also have a RP-TNC connector can thus be directly connected to the WLAN card using appropriate cables.

---

## 3.1 Internet Gateway Station

### 3.1.1 Hardware

One Pentium III machine with 128 MB RAM and a 40GB hard disk is being used for the Internet-gateway. This machine (depicted as a BS in figure 1.1) is currently running RedHat Linux distribution, version 8.0. The linux kernel version on the machine is 2.4.18-3.

We have used a standard off-the-shelf Cisco Aironet 350 PCI Series 802.11b client adapter for the necessary wireless interface on this machine. The details about 802.11b are provided in the section on Wireless Architecture. More details about setting up the card in Linux are provided in section 4.1.1. This card has been configured for Ad-Hoc mode of operation and the data rate setting has been configured to auto.

For Internet connectivity we have used a 10/100 Intel EEPro network adapter to hook onto the LAN in the Networking Laboratory.

### 3.1.2 Software

#### Mail Server

For mail server we are using sendmail-8.12.5-7 configured on the machine being used for Internet gateway server. Sendmail for Linux is available both in the form of precompiled binary package as well as a source package. The package that we have used is sendmail-8.12.5-7.i386.rpm ( a binary package). Another package called sendmail-doc-8.12.5-7.i386.rpm provides all the documentation for sendmail including the sendmail FAQ. A detailed HOWTO on sendmail (and configuration) is available at [2] & [8].

#### IP Address Allocation Scheme

Currently for IP address allocation we have configured static IP addresses for all the machines. We are also testing the Dynamic Host Configuration Scheme. The machine acting as Internet gateway station has been configured as a DHCP server. The pool of IP addresses available to this server is: 192.168.3.1 - 192.168.3.254

---

To enable a DHCP server on a Linux machine one needs the dhcpd package. For the RedHat distribution this package is available both in the form of precompiled binary version as well as a source version in the form of a rpm. For RedHat Version 8.0 the package is called dhcp-2.0pl5-8 package and is included with the distribution itself.

A detailed write up on configuring the machine as a DHCP server is given in section 4.3.

## DNS Server

Currently we are using the DNS Server package provided with the RedHat distribution. This package is called "bind-x.x.x.i386.rpm" (binary package), where x.x.x is the version number. We are using 9.2.0-8 version of this package. Section ?? provides an outline on setting up a Linux machine as a DNS Server.

## 3.2 Wireless Communication Architecture

### 3.2.1 Hardware

The main hardware requirements for the wireless architecture of this project are as follows:

**802.11b cards** WLAN cards from 4 different vendors were studied for their specifications. The brands evaluated were: Orinocco, Intel, D-Link and Cisco. The main points of evaluation of the hardware were

1. Extent of support and programmability in Linux
2. Range coverage of the cards, both in office/closed environments space and open areas.
3. Transmit Power
4. Support for Ad-Hoc mode operation
5. Support for data Encryption
6. Support for 802.1x

---

We selected the Cisco Aironet 350 Series cards because of their larger coverage range, larger transmitting power, easy availability and better support. Intel hardware based on the Spectrum chipset posed a lot of problems when working in Linux.

Both the PCMCIA version for laptops and PCI version for desktop machines are being used for testing and development purposes. Another reason for choosing Cisco hardware was the good support for the PCI versions of the card in Linux. The adapter drivers available in standard Linux distribution from RedHat worked without difficulty for both the PCI and the PCMCIA hardware. Specifications for the Cisco Aironet 350 series cards used:

Transmit Rates	1,2,5.5 and 11 Mbps
Transmit Power	100mW
Coverage(Open Space)	800 ft 11 Mbps
Encryption Supported	128 bit WEP
Antennae	
PC Card(PCMCIA Interface)	Integrated diversity Antennae
PCI Card	RP-TNC connector

**Directional Yagi Antennae** For the purpose of long-range communication between the various wireless-post-offices or relay stations we tested with off-the-shelf directional 2.4 GHz Yagi Antennae (model number 2415AB) from Telex available with the Networking Laboratory, DCSE, IIT Delhi. The 2415 is a 13.5 dBi directional antennae with a half power beam width of 30 degrees.

**Micro-Strip Antennae** Currently we are trying to develop a micro-strip antenna with the help of CARE, Dept of Electrical Engineering, IIT Delhi. The main reason for developing a micro-strip antenna is because of the difficulty in procuring off-the-shelf hardware.

### 3.2.2 Issues

- Problems faced while using Yagi Antennae. We started by using the available pair of Yagi Antennae within the lab environment (a range of approx 200 m). During this testing phase, several performance measurements were done to analyze the signal strength and the available

bandwidth. One of the problems encountered while using the Yagi Antennae for longer range was the unavailability of the required cables to connect the Yagis to a Cisco 350 Series Card locally. The same had to be imported. Currently we are testing the Yagi Antennae over a larger distance by mounting a pair of these antennae on roof-tops.

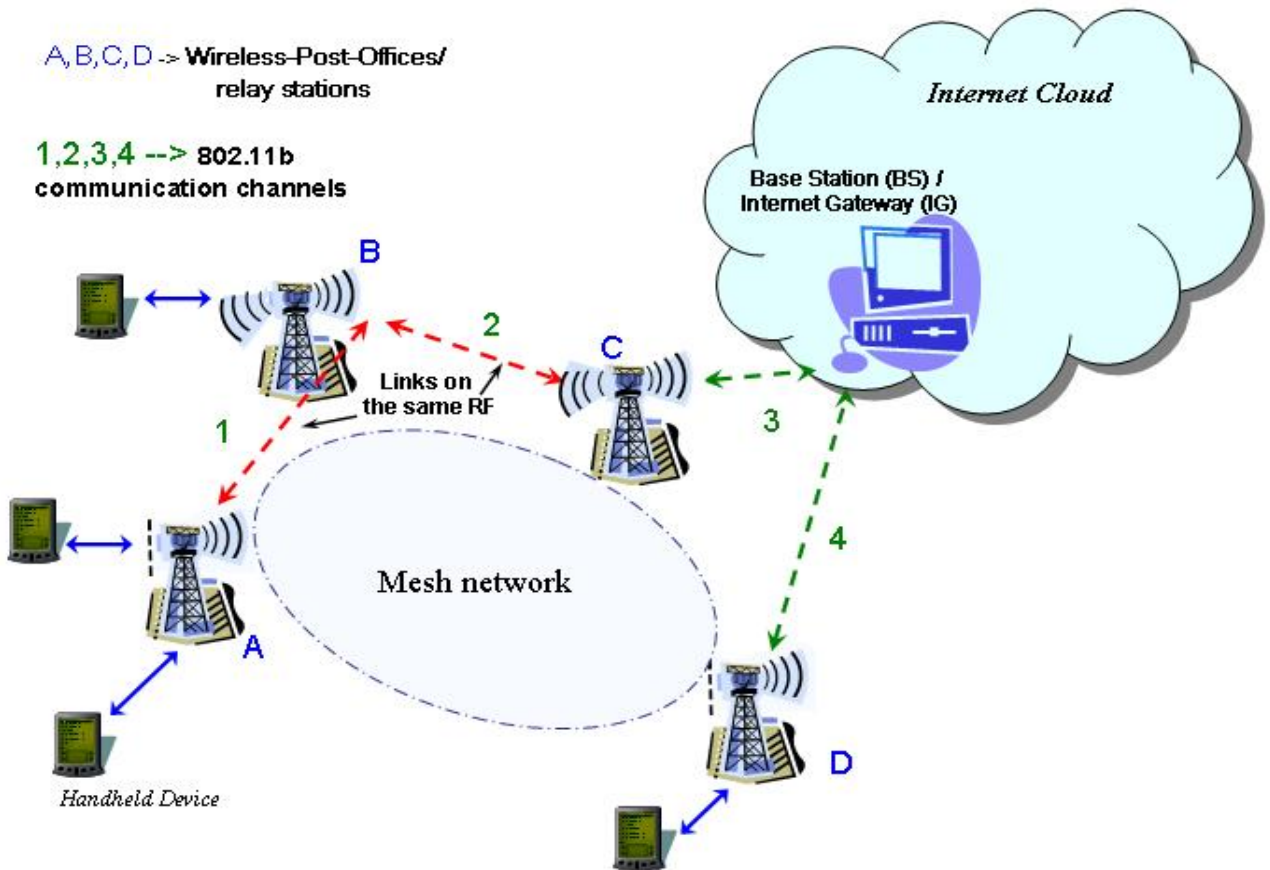


Figure 3.1: Picture showing multiple R/F links from the same node

- Note that due to the multi-point to multi-point nature of the mesh network being designed, some of the nodes use multiple interfaces i.e. an intermediate node will have multiple links (typically the number of other nodes it is connected to (degree) will be 3-4). If all of these links work on the same R/F frequency, it causes interference amongst the multiple data paths centered around the same node e.g. as depicted in figure 3.1, path 1 & path 2 will interfere with

---

each other if both were using the same frequency channel for communication. This interference reduces the effective bandwidth of the node because each communication link now gets a smaller timeslot to transmit without interference. Note that 802.11b uses Collision Avoidance (CSMA/CA) method which means that if there were 3 nodes in communication with a single node and transmitting on the same frequency, theoretically, the available bandwidth will be reduced to 1/3rd. This is because the transmission along any link will have to wait longer till it is able to find an empty slot to transmit its data.

In order to overcome the interference problem, in our first approach we tried using different transmitting frequencies (i.e. different 802.11b channels) along different links originating/traveling through the same node. The idea is to select channels so that there would be no interference among all the transmissions at a single node for all the different links, hence allowing simultaneous communication on all such links. The available bandwidth per link will remain unaltered in comparison to the situation with only a single link.

This can be tested using existing hardware because the 802.11b architecture supports 14 different frequency channels that can be used for transmission as shown in Table 2.2.1 previously. The Cisco 350 Series Aironet Cards however only support 11 channels numbered channel 1 through 11. So for our purpose, in order to reduce interference between 3 simultaneous and distinct transmissions links at any relay station, the optimal choice would be to use channel 1, channel 6 and channel 11 for the three transmissions.

We performed some experiments using the available utilities to determine the viability of 3 channel simultaneous transmission, however we were unable to change the default frequency channel used by the card. By default, the cards use Channel 6 for communication. We are determining the constraints preventing us from using other frequency channels by examining the source code for the driver and the firmware in the card, etc. The same issue is also being discussed with the support team at Cisco.

---

### 3.3 Wireless Communication Protocols

As discussed earlier we are working with AODV as the communication protocol between the relay stations. We have used the NIST implementation in our setup. Small modifications have been made to the available package to suit the needs of the application.

The modified source code for the NIST implementation will be soon released on the project webpage.

### 3.4 Application Modules

The components of the application framework as discussed in section 2.4.1 have been implemented as follows:

- Email retrieval from the server to the relay stations is done by enabling the POP3 server on the Internet Gateway Station which also serves as the mail server for the WIPO network. The steps required to enable a POP server on a RedHat Linux 8.0 machine are as follows:
  1. Enable the POP3 services in `"/etc/xinetd.d/ipop3"` file. One just needs to change the field `"disable=yes"` to `"disable=no"` as root.
  2. Restart xinetd by typing `"/etc/rc.d/init.d/xinetd restart"` as root.

In order to get the mails onto the relay stations using POP3 client, we used `"fetchmail"`.

- Email/information retrieval from the relay stations is done by enabling POP3 services on each of the relay stations as described above. We used the Outlook Express utility in WinCE environment on the iPAQ handhelds to retrieve the mails from the relay stations to the handhelds.
- IrDA has been used as a communication interface between relay stations and iPAQ handhelds. A detailed step-by-step procedure to setup IrDA on a linux based relay station is given in section 4.2.1.  
A how-to on setting up IrDA for communication on the handheld device is discussed in section 4.2.2.

# Chapter 4

## Appendix

### 4.1 Appendix I : Setting Up CISCO Wireless Client Adapter on Linux

#### 4.1.1 Installing drivers for the client adapters in Linux

The drivers for the CISCO wireless client adapter are already provided on the Linux2.4.18 kernel distributed with RedHat. The following steps are needed to compile the driver module, assuming that the user is familiar with Linux kernel compilation. For reference on kernel compilation, the Linux Kernel Compilation How-To Refer to [6] will provide a good start.

1. Go to the kernel source directory, henceforth called as `kernel_source`
2. Type `make xconfig` at the prompt. This will produce a GUI interface for compiling the kernel with a lot of options. In case you do not have a X server running on the Linux machine, type `make menuconfig` instead.
3. Select the option called Network Device Support from the list of buttons on the GUI. This will pop up another window listing various kind of Network devices for which support is available in the Linux kernel.
4. Select the button labeled Wireless LAN (non hamradio). Selecting this will pop up another window listing all the Wireless LAN devices that are currently supported.

- 
5. Select the following options in this window
    - Wireless LAN (non-hamradio)
    - Cisco/Aironet 34X/35X/4500/4800 ISA and PCI cards, this option may also be selected as a module.
    - Cisco/Aironet 34X/35X/4500/4800 PCMCIA cards, this option may also be selected as a module.
  6. Click OK to close this window.
  7. Then from the Network Device Support window opened in step 3 above, select the PCMCIA Network Device Support Button
  8. This will open another pop-up window with the same name.
  9. Select the following options from this window
    - PCMCIA network device support
    - PCMCIA Wireless LAN
  10. Click OK to close this window.
  11. Click on Main Menu button on the Network Device Support window.
  12. On the main window click on Save and Exit button.
  13. On the prompt type `make dep`.
  14. Type `make bzImage;make modules`. This command will compile a new kernel image called `bzImage` in the `kernel_source/arch/i386/boot` directory. It will also compile the drivers selected as modules.
  15. Type `make modules_install` to copy the modules at the right place. Note that you need to have root access to install the drivers etc.
  16. Copy the `bzImage` in the `/boot` directory of the system and update LILO or GRUB configuration file as the case maybe.

- 
17. Rebooting the system in the newly compiled kernel should load the modules for the Aironet cards.

Once the driver modules are installed and loaded, the Cisco client adapter on Linux can be setup by two methods:

1. Using the CISCO Client Utility
2. Actually configuring it through `iwconfig` and `ifconfig` tools available with the Linux distribution.

### 4.1.2 Configuration Using the CISCO Client Utility

The CISCO Client Utility can be downloaded from CISCO website. From the web it can be downloaded from the following address :

<http://www.cisco.com/cgi-bin/tablebuild.pl/aironet-utils-linux>

Configuring the wireless client using the CISCO Client Utility is simple and done through the GUI interface which takes you step by step through the set up process.

The README file for the Client Utility as well as technical notes about the Utility are also downloadable from the above address. These documents contain the stepwise procedure to make the wireless client adapter work in Linux. They also contain information about the tools for monitoring the operation and functioning of the wireless adapter.

### 4.1.3 Configuration using `iwconfig` and `ifconfig`

Once the drivers for the card are installed the following steps can be carried out in order to get the card working:

1. first, we need to figure out which interface has been assigned to the wireless card. For this type in  
`iwconfig`

---

which will give you all the interfaces with a wireless extension. Say the interface returned was : eth2.

2. Then the IP address of the card needs to be set. This is done through the following command:

```
ifconfig eth2 inet 192.168.3.9
```

3. Next we need to set operation parameters for the card. First of all we specify the data rate of operation by the following command:

```
iwconfig eth2 rate 11Mbps
```

802.11b also gives the option of automatically setting the data rate according to noise conditions, etc. This can be done through the command:

```
iwconfig eth2 rate auto
```

4. Another very important parameter that needs to be set is the ESSID. Two wireless cards with different ESSIDs cannot communicate with each other. So the ESSIDs of all the cards that need to talk to each other must be the same. The ESSID can be any number e.g. 100. To set the ESSID you type in

```
iwconfig eth2 essid 100
```

5. Now, we can set the mode which the card will work under: 802.11b offers two modes of operation - Ad-Hoc and Managed.

6. The Managed mode works from a centralized point, with the central arbitrator called the Access Point. Each communication occurs through this Access Point. To set the card in Managed mode the command used is:

```
iwconfig eth2 mode Managed
```

7. The access point is specified by its MAC address. The command used to designate the access point is:

```
iwconfig eth2 ap AB:CD:EF:GH:IJ:KL
```

8. The Ad-Hoc mode is a distributed mode where all nodes first choose a common cell number before communication starts. To use the nodes in Ad-Hoc mode the following command is used:

---

```
iwconfig eth2 mode Ad-Hoc
```

For this project, we used the Ad-Hoc mode.

9. Finally, the interface itself is activated using the following command:

```
ifconfig eth2 up
```

10. At this point the card should be functional. If it does not work, run the iwconfig command again to display the current settings. If the settings do not match the above, please rerun the corresponding step.
11. If a card is functioning properly, its green light will be on continuously and the amber light will flicker to show activity. Once all cards are working, they can be verified by doing a broadcast ping using

```
ping -I eth2 -b 192.168.1.255
```

where 192.168.1.255 is the broadcast address for your wireless network. If all hosts are echoing the ping packets, they are working fine.
12. If you are using a laptop, you may have to restart the PCMCIA daemon after inserting the card. To do this use the command:

```
/etc/rc.d/init.d/pcmcia restart
```

#### 4.1.4 Monitoring the performance of the client adapters

Once the cards are working, we would need to monitor certain performance characteristics. The following parameters are the ones which are usually monitored.

**Signal Strength / Rx Power** The terms Signal Strength and Rx Power are used synonymously in the 802.11b parlance. One of these is displayed when a iwconfig command is issued while the card is working. The Cisco Client Utility also gives an option of displaying in either format. The signal strength is specified on a scale from 0 to 100 whereas Rx Power is on a scale from -90dBm to -45dBm.

**Noise Level** This is similar to the signal level but its sensitivity can be set using iwconfig.

---

**Link Quality** This parameter can be displayed by using the `iwconfig` command. It is a ratio with denominator 10, a higher ratio specifies a worse signal.

**Data Rate** This can be measured using a FTP transfer. We observed rates of about 60% of the actual rate specified.

**Tx Power** This parameter can be set using `iwconfig` utility. Only discrete values of transmitting power is supported by the CISCO card. These values are 100mW, 50mW, 30mW, 20mW, 5mW, 1mW. These values are specified in the Cisco Aironet 350 Series cards documentation.

## 4.2 Appendix II : Infra Red

### 4.2.1 Infra Red Configuration for Linux

The following steps should be followed to configure infrared support on a machine with IrDA port attached to a serial port and running Linux:

1. Download the following files from `synce.sf.net`, where X.X represent the current version number:
  - (a) `synce-librapi2-X.X.tar.gz`
  - (b) `synce-libsynce-X.X.tar.gz`
  - (c) `synce-dccm-X.X.tar.gz`
  - (d) `synce-serial-X.X.tar.gz`
2. Compile `libsynce` by issuing the following commands in the `/tmp` directory:
  - (a) `tar zxf synce-libsynce-X.X.tar.gz`
  - (b) `cd synce-libsynce-X.X`
  - (c) `./configure`
  - (d) `make`
  - (e) `make install`

---

(f) cd ..

3. Compile librapi2 by:

(a) tar zxf synce-librapi2-X.X.tar.gz

(b) cd synce-librapi2-X.X

(c) ./configure

(d) make

(e) make install

(f) cd ..

4. Compile dccm by:

(a) tar zxf synce-dccm-X.X.tar.gz

(b) cd synce-dccm-X.X

(c) ./configure

(d) make

(e) make install

(f) cd ..

5. Compile serial by:

(a) tar zxf synce-serial-X.X.tar.gz

(b) cd synce-serial-X.X

(c) ./configure

(d) make

(e) make install

(f) cd ..

6. Start the irda protocol stack. Install the appropriate module support.

- 
- (a) `insmod irda`
  - (b) `insmod ircomm`
  - (c) `insmod irtty`
  - (d) `insmod ircomm-tty`
  - (e) `modprobe ppp_async`
  - (f) `modprobe ppp_generic`
  - (g) `modprobe slhc`
  - (h) `depmod -a //`To install all dependencies.
7. Download and install the IRDA utilities package (`irda-utils-0.X.X.tar.gz`) from `irda.sf.net`.
  8. Locate the serial port on which your infrared port is attached and then run the following on the infrared port, where `n` is the number of your serial port:  
`irattach /dev/ttySn -s`
  9. Run:  
`synce-serial-config ircomm0`
  10. Change the IP address of the DNS server in the file `/etc/ppp/peers/synce-`, and set it to your own DNS servers IP address.
  11. Run:  
`dccm (dont run it as root)`
  12. Run:  
`synce-serial-start (needs to be run for every new connection)`
  13. To build support for iptables (if already not present), install the following modules:
    - `insmod ip_tables`
    - `insmod ip_conntrack`
    - `insmod ip_conntrack_ftp`

- 
- `insmod iptable_nat`
  - `insmod ip_nat_ftp`
  - `insmod ip_tables`
  - `insmod ip_conntrack_irc`
  - `insmod ip_nat_irc`
  - `insmod ipt_MASQUERADE`
  - `insmod iptable_mangle`
  - `insmod iptable_filter`
  - `insmod ipt_REJECT`
  - `insmod ipt_tcpmss`
  - `depmod -a`

14. To enable support for IP forwarding, run:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

15. Enable masquerading by:

```
iptables -t nat -A POSTROUTING -s 192.168.131.201/32 -j MASQUERADE
```

## 4.2.2 Infra Red Configuration for a Handheld running Windows CE

Following steps are to be taken on the handheld device, in our case a iPAQ 3950 running Windows CE, to configure the Infra Red support:

1. Tap on Start → Settings → Connections → Connections.
2. Tap on 'My Network Card Connects To' and select 'The Internet'.
3. In Internet Settings, tap on Modify → New.
4. Enter a name for the new connection we are going to create: such as 'infrared'.
5. Select to the modem to Generic IrDA Modem.

- 
6. Set the Baud Rate to 115200 bps.
  7. Tap Advanced → Port Settings. Check 'Enter Dialing Commands Manually'. Set Parity to none, number of data bits to 8, Stop Bits to 1, and Flow Control to Hardware.
  8. In TCP/IP settings, select IP compression, select software compression, and uncheck SLIP.
  9. For server settings, select Server-Assigned IP address, and select Server-Assigned Name Server.
  10. Save this setting, and use this to connect to the Internet. Open ActiveSync from Start Menu.
  11. Align the infrared ports of the laptop and the PDA.
  12. In ActiveSync, tap on connect → Use Infrared to connect to the laptop.

### 4.3 Appendix III : Dynamic Host Configuration Protocol (DHCP) for Linux

Dynamic Host Configuration Protocol as the name suggests, is used to configure vital networking parameters of hosts (running clients) with the help of a server. The following steps are required to configure a Linux machine (connected on a LAN to the hosts to which it has to serve) as a DHCP server

1. login as root
2. open the /etc/dhcpd.conf file
3. a typical dhcp.conf file will look like the following

```
global parameters...
subnet 204.254.239.0 netmask 255.255.255.224 {
    subnetspecific parameters...
    range 204.254.239.10 204.254.239.30;
}
subnet 204.254.239.32 netmask 255.255.255.224 {
```

---

```

        subnet specific parameters...
        range 204.254.239.42 204.254.239.62;
    }
    subnet 204.254.239.64 netmask 255.255.255.224 {
        subnetspecific parameters...
        range 204.254.239.74 204.254.239.94;
    }
    group {
        group specific parameters...
        host zappo.test.isc.org {
host specific parameters...
        }
        host beppo.test.isc.org {
            host specific parameters...
        }
        host harpo.test.isc.org {
            host specific parameters...
        }
    }
}

```

Global options can be the options which are common to all the various subsections created(eg name servers). Each subnet declaration indicates addresses from which the dhcp server can assign addresses to its clients.

The range keyword specifies the range of addresses picked up from a subnet.If the host is known (ie its mac address is known) the host keyword can be used to assign static IP addresses.

4. After making appropriate changes to the dhcpd.conf, start the dhcpd server

```
/etc/init.d/dhcpd start
```

To configure a RedHat Linux client to pick up its networking information from a server using DHCP protocol, the following steps are required

1. login as root

- 
2. run `redhat-config-network`
  3. select the interfaces on which dhcp client is to be run.
  4. Enable the option for obtaining IP address automatically using dhcp

# Bibliography

- [1] 802.11b wireless lan. [http://www.vocal.com/data\\_sheets/full/802.11b.pdf](http://www.vocal.com/data_sheets/full/802.11b.pdf).
- [2] Chapter 18, sendmail, linux administrators guide. <http://www.tldp.org/LDP/nag2/x-087-2-sendmail.html>.
- [3] Dns-howto. <http://www.tldp.org/HOWTO/DNS-HOWTO.html>.
- [4] Dynamic host configuration protocol.
- [5] (in)security of the wep algorithm. <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>.
- [6] Linux kernel howto. <http://www.tldp.org//HOWTO/Kernel-HOWTO/index.html>.
- [7] Netfilter howto.
- [8] Sendmail howto.
- [9] Cisco aironet antennae reference guide, 2002. [http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/agder\\_rg.htm](http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/agder_rg.htm).
- [10] Luke Klein-Brendt. Implementation of aodv from nist. [http://w3.antd.nist.gov/wctg/aodv\\_kernel/index.html](http://w3.antd.nist.gov/wctg/aodv_kernel/index.html).
- [11] Erik Nordstrom and Henrik Lundgren. Implementation of aodv from uppsala university. <http://user.it.uu.se/henrikl/aodv>.
- [12] Elizabeth M.Royer & C-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks.