JB 94-15

Entitled:

"A System Design for a Hybrid Network Terminal Using Asymmetric TCP/IP to Support Internet Applications"

Authors:

(with A. Falk)

Conference

Conference Technology 2004
Washington, D.C.
November 8-10, 1994

A SYSTEM DESIGN FOR A HYBRID NETWORK TERMINAL USING ASYMMETRIC TCP/IP TO SUPPORT INTERNET APPLICATIONS

Aaron Falk
Institute for Systems Research
University of Maryland, College Park, Maryland

Narin Suphasindhu
Institute for Systems Research
University of Maryland, College Park, Maryland

Doug Dillon
Hughes Network Systems
Germantown, Maryland

John Baras
Institute for Systems Research
University of Maryland, College Park, Maryland

PURPOSE OF THIS PAPER

The purpose of this paper is to describe the system design process used to build a prototype terminal which uses a hybrid satellite-terrestrial connection and "asymmetric" TCP/IP network protocols to give the computer user who has a minimum bandwidth modem connection to the Internet access to bandwidth hungry Internet applications.

INTRODUCTION

Using hybrid networking, the terminal will merge two connections, a bidirectional terrestrial link using a modem and a receive-only satellite link, so that the TCP/IP software above the device driver sees one "virtual" device. This design exploits three concepts: 1) satellites are able to offer high bandwidth connections to a large geographical area, 2) a receive-only VSAT is cheap to manufacture and easier to install than one which can also transmit, and 3) most computer users, especially those in a home environment, will want to consume much more data than they will generate.

This design supports any 80386 or 80486 processor PC, any TCP/IP package, access to any existing Internet host, and any commercial SLIP provider. These design drivers should maximize the commercial potential of the system.

In the hybrid network, a client will transmit requests over the terrestrial link to a server located somewhere on the Internet. The server will route the reply to a satellite uplink where it will be broadcast to all the clients on the system creating a virtual "ethernet in the sky." The client terminal will trap only the packets with the correct address and send them up to the application which requested them.

The potential user of this terminal would be a computer user in a home or small business without high bandwidth access to the Internet. High bandwidth access means anything higher than current modem speeds. A user with an IBM PC compatible and a modem connection to the Internet running the SLIP protocol has been used to define the user requirements.

Internet applications have been chosen as a user requirement for this design because there exists a vast number of enormous databases available on the Internet, much of it accessible by using applications such as FTP, Gopher, Archie, WAIS, or Mosaic. The Internet is the closest existing prototype of the National Information Infrastructure and if this hybrid, asymmetric link design has performance and cost advantages over other methods of accessing information on the Internet, then it may be instrumental in demonstrating satellites' significance in the development of the NII.

While the goal has been to provide the user with a terminal, it has been necessary to design an entire system of which the terminal itself is only a single subsystem in order to interoperate with the existing Internet. It would not be possible to create this design without considering the "big picture" since modifications of the hardware and physical, link, and network layers are all required to make the system interoperable with the existing hardware and protocols. Therefore, this design is an excellent example of the use of systems integration techniques to incorporate the super-system architecture into subsystem designs.

THE HYBRID NETWORK TERMINAL DESIGN

Students and faculty from the University of Maryland collaborated with engineers from Hughes Network Systems to create a prototype system that was demonstrated to offer up to 200kbps receivable data. The user terminal in the system used a Hughes device known as a BIC or DirecPC card to interface the computer with a 0.6 meter antenna.

The key problem to solve was how to force routing of packets on the Internet without requiring special modifications to the user application or the destination machine the user was trying to reach. Solving this problem meant manipulating the two primary Internet packet protocols: Transport Control Protocol (TCP) and Internet Protocol (IP) to affect the routing we wanted.

TCP/IP is a connectionless protocol where each packet is addressed with its destination and the network finds the best path each time a packet is sent into the system. Address Resolution Protocol (ARP) is used to locate machines on a local network once a packet has arrived at the gateway between the destination machine and the Internet. A typical client-server exchange includes the client creating a request packet which includes the server's IP address (the destination) and the client's own IP address (the source). Once the server has received the request and generated a reply, it merely swaps the two addresses from the request header and sends out the reply to the client.

The following sections describe in detail the design of the prototype system. The section on user requirements describes what the system must accomplish from the user's perspective. Following that are sections highlighting development of each subsystem.

The User & User Requirements

Developing the definition of user requirements is an iterative process with the goal of producing a set of system parameters which collectively satisfy the ultimate goal of the system. It is useful to explicitly (as much as is possible for a non-existent product) identify the potential user/customer, first. This will drive the definition of the user's requirements. With their experience in providing network services, Hughes Network Systems was instrumental in this task. The user for this system can be defined as the following:

> A personal computer owner located in a home or small business in the continental US who has an interest in accessing the Internet using applications such as FTP, Gopher, Mosaic, News, or Archie with the lowest delay possible. The user is capable of installing peripheral cards and software in his computer but may require professional assistance with the installation of a satellite dish. The user may be willing to spend the same amount of money on this system as for a high speed modem. The basic user hardware configuration is an IBM or IBM-compatible PC and a modem capable of at least 2400 baud. The basic user software configuration is DOS 5.0, Microsoft Windows 3.1, and a commercial TCP/IP package that includes client versions of the above mentioned Internet applications.

The user definition, in turn, generates several requirements for the system. These user requirements are system parameters that the system must satisfy in order to satisfy the user requirements. Each of these requirements are listed below:

- The system must provide significantly less delay responding to requests for large data files experienced using a modem.
- The system must work with any 386/486 33MHz machine.
- The system must work with any commercial TCP/IP package.
- The system must work with any commercial SLIP service provider.
- The system must be able to access any Internet host.
- The system must support Internet initiated connections.

Satisfying the user requirements while minimizing user cost and development time has justified the design of the different subsystems within this system.

How It Works

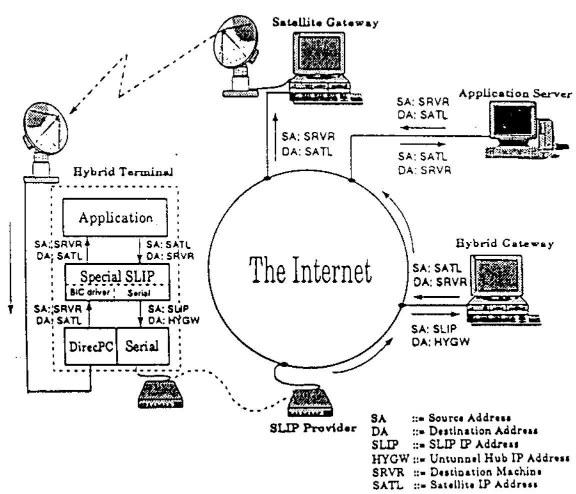

Fig. 1 The path travelled by a tunneled packet generated by the Hybrid Terminal. For simplicity, the diagram does not show that, before a packet can get from the Application Server to the Satellite Gateway, it must first be sent to the Hybrid Gateway to be encapsulated in a special satellite packet.

In this section we describe the general process of how a request from the user terminal is carried through the Internet and to a machine running a host application and how that machine's response is carried back to the user. The various subsystems are introduced here and will be described in more detail in the following sections.

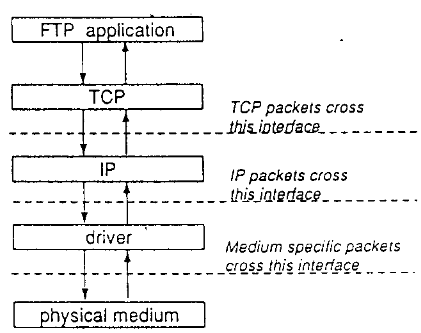

Fig. 2 The software stack when running FTP (File Transfer Protocol).

Before describing an example of how the system routes packets, it is important to point out that the user terminal is given two IP addresses. One IP address corresponds to the SLIP interface and would by typically be assigned by the SLIP service provider. The other IP address corresponds to the satellite interface and would be assigned by the hybrid service provider. These IP addresses correspond to completely different networks. Observe that the SLIP service does not need to know anything about the satellite IP address or even whether the user is using the hybrid service. If a host somewhere in the Internet is trying to deliver a packet to the satellite IP address by using the Internet routing scheme of routers, gateways, and ARPs, the *only* way that packet can reach the satellite IP interface is to traverse the satellite by being routed to the satellite gateway.

When requesting a data transfer, say using FTP, the user sends a request to a remote machine that is running FTP server software. This software receives file transfer requests and responds to them in the appropriate fashion. If a hybrid terminal user wanted to receive a file from a machine running FTP server (we'll call it the Application Server), every packet from the user terminal would take the following path:

1) Within the User Terminal, Hybrid Host, the FTP client software generate a request and pass it to the TCP/IP module. TCP/IP would place the request first in a TCP packet then in an IP packet which would then be passed to the Special SLIP driver software. This request would have a source IP address corresponding to the satellite interface and the destination IP address of the Application Server.

2) In Special SLIP, the IP packet is encapsulated, or tunneled, inside of another IP packet and sent over the modem connection to the SLIP server host. The encapsulation amounts to adding a new IP header in front of the original one with a source address corresponding to the SLIP interface and a destination address corresponding to the machine we are calling Hybrid Gateway.

3) SLIP server receives the IP packet analyzes the tunneling header and, thinking it is destined for Hybrid Gateway, uses standard Internet routing to send the packet to Hybrid Gateway.

4) When Hybrid Gateway receives the packet it strips off the tunneling header, revealing the true header with Application Server as the destination. The packet is then sent back out into the Internet.

5) Internet routing takes the packet to the Application Server which replies with the requested file and addresses the reply to the request's source IP address, i.e. the IP address of the User Terminal's satellite interface.

6) In order to find the user terminal's satellite interface, the Internet routing protocol will send the packet to the subnet containing a router/gateway connected to Hybrid Gateway. When that router/gateway sends out an ARP for any user terminals' satellite IP address Hybrid Gateway responds and says "send it to me."

7) Once Hybrid Gateway receives the reply packet, it encapsulates it in a special packet format that is used over the satellite link and uses the satellite IP address to uniquely identify the satellite packet's destination. Then Hybrid Gateway sends the packet over ethernet to the Satellite Gateway.

8) Satellite Gateway broadcasts over the satellite link any packets it receives from Hybrid Gateway.

9) The driver in Hybrid Host that services the DirecPC card scans all packets broadcast over the satellite looking for the satellite IP address in the header. Once it identifies one, it captures it, strips off the satellite header revealing the reply IP packet, and sends it to the Special SLIP driver.

10) The special SLIP driver calls the TCP/IP package notifying it that it has received an IP packet and passes up the reply, completing the transaction.

The User Terminal

The User Terminal has required the most development. Device driver software has been developed that will appear to an off-the-shelf TCP/IP package that the computer is connected to an ethernet card when it is actually connected to a satellite dish and a modem. At the same time it must appear to the SLIP Server that the computer has a single IP address assigned by the SLIP provider, and force the Internet to route IP packet replies to a different IP address than the requests originated from. For this last task the Hybrid Gateway is needed also.

The TCP/IP package includes some of the Internet applications that the user wants to run. It also, of course, contains the TCP and IP protocol stacks. In a normal configuration, the TCP/IP package would sit on top of a driver that would talk to an ethernet card, providing a fast connection, or a modem via the computers serial communications port, providing a slow connection. With a normal symmetric connection, TCP/IP would send and receive data over the network by passing and receiving frames across a software interface to the driver. The driver would handle the moving the frames back and forth over the physical connection to the network.

In the hybrid configuration the interface between TCP/IP and the driver doesn't change. It can't if we are going to support the user requirement of being able to use an off-the-shelf package without modification. However, instead of communicating with a single physical network, the driver for the hybrid terminal, which we are calling Special SLIP, communicates with two physical networks as shown in the following diagram:

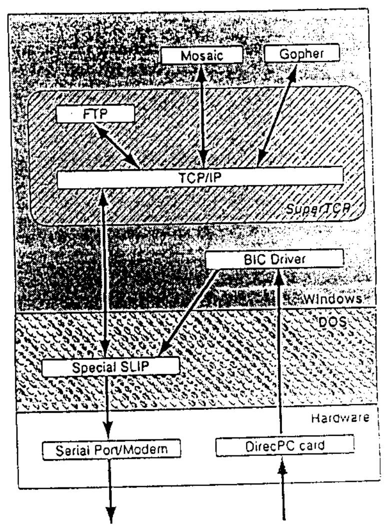

Fig. 3 The data paths within the User Terminal take data in off the DirecPC card and send data out to the modem.

Serial port handling

The serial port provides the physical connection to the modem and, through it, the terrestrial network. The Serial Line IP (SLIP) protocol will be used over the terrestrial connection. SLIP is the crudest form of IP protocol for serial lines. Its function is to delimit IP packets by inserting a control character (hex 0xC0) between them. To insure that a data byte is not mistaken for the control character, all outgoing data is scanned for instances of the control character which is replaced by a two character string. The protocol is described in more detail in [Romkey, 88].

An interrupt service routine is required to handle hardware interrupts coming asynchronously from the UART. On an interrupt, the routine should read or write a byte to the UART. This interrupt service routine (isr) needs to conform to the standard DOS calling conventions, i.e. it needs to chain off an interrupt vector stored in PC memory, service the interrupt as quickly as possible, and send an end-of-interrupt to the system to allow other interrupts to get through. This code should be configurable as which chip, connection parameters, and COM port are used and it should load at system boot time.

BIC driver call handling

The BIC driver has been developed by HNS. The driver's functions include scanning all packets transmitted over the satellite channel for one with a header corresponding to the IP address of the satellite interface, performing some error detection and correction on the packet, buffering the received packet, and passing it to the Special SLIP driver. It will call the Special SLIP driver using the DOS IOCTL_output_cmd() call and will pass the address and length of a received packet in the BIC driver's buffers. Special SLIP needs to copy the data out of the BIC's buffers as quickly as possible and pass it up to the TCP/IP package.

TCP/IP call support

There are two popular interfaces between network drivers and network software: the Crynwr/Clarkson Packet Driver Specification and the 3Com/Microsoft Network Driver Interface Specification (NDIS). The NDIS specification has greater complexity but also greater functionality and has somewhat greater industry acceptance. Also, the Frontier SuperTCP TCP/IP package that is being used for development supports the NDIS standard and not the Clarkson so that is what was used in this design.

Tunneling

One of the most innovative concepts incorporated in this design is the tunneling of IP packets to fool the Internet routing scheme. This idea was proposed by Doug Dillon of HNS and developed by the author with assistance from Narin Suphasindhu.

The reason for tunneling is this: the user terminal has two IP addresses associated with it—one for the SLIP interface which is assigned by the SLIP provider, a commercial service the developers have no control over, and the other corresponding to the satellite interface, assigned by HNS and essentially an extension of the uplink network. The way to get the Internet to route packets to the satellite interface when the request came from the SLIP interface is to set the source IP address in the request packet to be the satellite IP address. That way, when the Application Server forms its reply to the request, it addresses the reply to the source address, i.e. the satellite IP address.

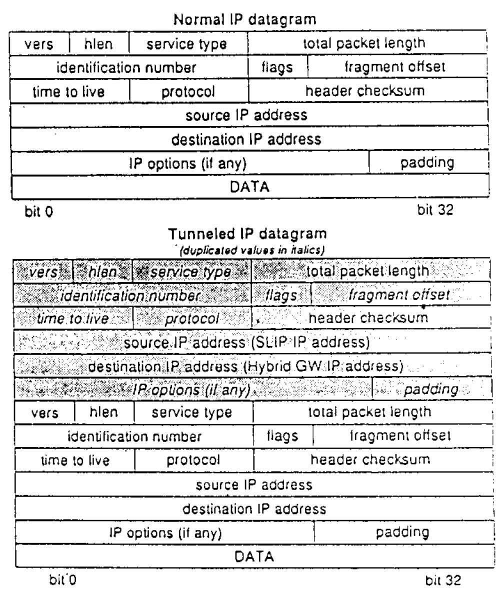

Fig. 4 When tunneling an IP datagram, a complete IP header is appended before the "true" IP header duplicating all values from the "true" header except the source and destination addresses, the total packet length, and the header checksum.

There is a complication, however. When SLIP service is purchased from a commercial provider, the provider assigns a single IP address and may pass IP packets containing the assigned address as the source address. If the SLIP provider thought that an entire network was going to connect through the account, the traffic rate would be higher and the provider could, reasonably, ask for larger fee. Therefore, it is not unreasonable to assume that the SLIP provider may be checking the source IP addresses of traffic flowing from the user terminal into the SLIP server. If the user terminal is changing the source IP addresses to match that of the satellite interface, the SLIP provider could declare a violation of the service agreement or maybe just drop the packets. Either event is unacceptable.

To cope with this possibility, IP encapsulation or tunneling is used to bypass the address checking that may occur in the SLIP server. In tunneling, every IP packet passed to the Special SLIP driver by the TCP/IP package has the satellite IP as its source address and some Application Server as its destination address and is encapsulated in *another* IP packet which has SLIP IP as its source address and Hybrid Gateway as its destination address. The "encapsulation" really just amounts to adding a new header in front of the true one. The effect of the new header is to route every packet to the Hybrid Gateway. At the Hybrid Gateway, the tunneling header is removed and the packet is sent back out into the Internet to be rerouted to its proper destination.

In forming the tunneling header, all of the values from the old header are copied into the new one with the following exceptions. Of course the source and destination address change. Also the total packet length grows by one header length. Additionally, the header checksum needs to be recalculated since some of the fields have changed.

Now, each packet experiences some additional mileage and this scheme clearly will add some additional delay to each transaction over the network. However, it satisfies the user requirement of making the system operable with any commercial SLIP service. The added delay can be minimized if the Hybrid Gateway is "well connected" to the Internet. The main arteries of the Internet operate at very high rates and if the Hybrid Gateway has a high rate connection to a main artery of the Internet, the added delay can be minimized. In the prototype a 1.5Mb/s T1 link was acquired to SURAnet, the southeastern United States Internet provider to minimize this additional delay.

ARP handling

As stated above, the Special SLIP NDIS driver declares to the TCP/IP package that it is an ethernet card. The TCP/IP package handles ethernet routing and when it is trying to send data to a new IP address, it tries to resolve that address to a hardware or MAC address using an ARP as described in the section on Internet Routing. The SLIP connection only carries IP level communication and MAC addresses have no meaning at the IP level. So, in order to satisfy the TCP/IP package's request for MAC address, each packet sent by TCP/IP needs to be checked to see if it's an ARP trying to resolve a new IP address. If TCP/IP does send an ARP, then the driver creates an ARP reply handing TCP/IP a bogus MAC address for the ethernet header of the packet to be transmitted. The contents of the MAC address are irrelevant since the ethernet header is stripped off to send the packet over the SLIP link.

ARPs don't occur that frequently because most connections will be made outside of the subnet of satellite IP addresses. In this case the packet is automatically sent to the local router. So, at least one ARP is required but not many more than that.

Segmentation

Since the TCP/IP is configured to talk to ethernet and we want to be able to receive the largest sized packets we can, TCP/IP is configured such that the Maximum Transmission Unit (MTU) of the network is as large as possible, 1500 bytes for ethernet. This specifies the largest packet size the network can handle. Our experience has shown that SLIP servers can have a much smaller MTU such as 512 bytes or even as small as 256 bytes per packet. Usually, the application is generating small packets to send over the SLIP link, like 60 byte acknowledgments. However, the tunneling header adds about 40 bytes to each packet and occasionally the application will generate some large packets to send. To handle this situation, the driver must implement it's own segmentation procedure. In segmentation, the packet is broken into pieces the size of the SLIP MTU and the header minus one header length is copied onto each piece with an offset value specifying where that particular piece goes in the original packet.

Once a tunneled packet is segmented, it is reassembled when it reaches the Hybrid Gateway. Only the tunneling header is copied onto the head of the segments.

The Hybrid Gateway

The Hybrid Gateway is allocated all the special network routing functions that must occur outside of the User Terminal. Untunneling is one of those functions but not the only one. Nonstandard packet formats are used to tunnel IP packets from the User Terminal as well as to send packets over the satellite link.

Because the Hybrid Gateway is a bottleneck through which traffic from all hybrid terminals must flow, the functions are kept as simple as possible to maximize throughput. Each function is implemented so that the processing requirements are minimized. In the prototype situation, only a single User Terminal is active at a time. However, in an operational situation, the demand on the Hybrid Gateway could be such that a single PC could not keep up and a more powerful platform or collection of machines should be considered.

Untunneling

Every IP packet from every User Terminal is tunneled and sent to the Hybrid Gateway for untunneling. The Hybrid Gateway should have good Internet connectivity to minimize the accumulated delay from having to route every packet via this machine. When a tunneled packet is received by the Hybrid Gateway, the length of the header is read from the IP header and those bytes are simply stripped off and the packet is sent back out into the Internet.

Segmentation and Reassembly

It's possible that a tunneled packet is segmented on its way from the User Terminal to the Hybrid Gateway. This can occur if segmentation occurs within the driver as described above or if the packet traverses a network with an even smaller MTU than the MTU of the SLIP connection. Since only the tunneled header is copied onto the head of each segment, the segments must be reassembled within the Hybrid Gateway before the packet can be untunneled and sent back out into the Internet.

Reassembly involves allocating several buffers for partially received packets and filling in the segments as they arrive. A time to live value is assigned to each packet and if all the segments don't arrive before the time to live timer expires, the packet is discarded.

ARP responding

The machine that forwards packets over the satellite is on the same network as the Hybrid Gateway. This network's router will receive packets with the user terminal's satellite IP address and will send an ARP to find out what MAC address to send them to. The Hybrid Gateway needs to encapsulate these packets and so it must respond to ARPs for any User Terminal satellite IP address so as to receive them.

This is implemented by specifying a range of IP addresses that will be assigned to User Terminals and having the Hybrid Gateway respond to ARPs for it's own IP address as well as any IP address in the specified range. Once the router gets an appropriate ARP reply from theHybrid Gateway for a certain IP address, it will send all packets with that destination IP address to the Hybrid Gateway.

Satellite Packetizing

The Satellite Gateway expects IP packets to be encapsulated first in a special satellite packet and then within an LLC packet. The special satellite header identifies the downlink and contains a sequence number and the packet length. The LLC header is used to send the packet to the Satellite Gateway which is on a token ring network. The Hybrid Gateway must prepare packets for the Satellite Gateway by appending the correctly configured headers to the front of the packet. The receiver in the User Terminal does not get the LLC header and identifies packets destined for it by the least significant byte in the satellite IP address. Therefore, the six byte satellite destination address is determined by reversing the order of the bytes of the satellite IP address for the user terminal and then padding the rest of the address with zeros. The sequence number is just a counter and the length is calculated from the packet header.

The SLIP Server

A SLIP Server was configured for testing purposes but in real operation the SLIP Server will be a commercial service. Its functions are to receive SLIP encoded IP packets from a modem connection with the User Terminal, uncode them, and forward them to the Hybrid Gateway via the Internet.

For development, we configured a PC with the Frontier TCP/IP package using two interfaces: a serial port connected to a modem and an ethernet card. The Frontier package acted as a router receiving packets from the SLIP connection and forwarding them to a router available on the ethernet connection with Internet access.

The Satellite Gateway

The Satellite Gateway has been previously developed by HNS. It consists of a PC with token ring connection and a DirecPC card configured for transmission. When the Satellite Gateway receives a token ring frame at its MAC address, it checks for the correct service access point identifier and sequence number and then sends it over the satellite.

The Application Server

The Application Server represents any server running an Internet application available on the Internet using the TCP/IP protocol suite. If the User Terminal is running a Mosaic client, the Application Server may need to support a variety of connections. The key element to this subsystem is that it should include *any* host on the Internet reachable by normal Internet communication.

CONCLUSION

In this project we have taken several existing pieces of technology, the commercial TCP/IP package, the DirecPC card, and the Internet and combined them (with a little software glue) to provide a level of data communication service previously unavailable to homes. This is an excellent example of the potential of satellites in communications as well as the field of systems integration.

Hughes Network Systems has plans to integrate Hybrid Internet Access into a group of new services/products that it will be marketing in the next year. Hybrid Internet Access will be combined with a software "package" delivery service for point-to-point distribution of off-line video or commercial software and a broadcast distribution service for sending out live, real-time video, audio or a news "ticker" service as a suite of products based on the DirecPC card technology. A PC with the DirecPC card and a modem could purchase any of these products. This fall, HNS plans a 100 site nationwide field trial.

This system is capable in its current state of producing over 200kbit/s per terminal. With the modifications suggested, we hope that the 1Mb/s barrier will be broken and true ethernet bandwidths will be available to the home user. This is sure to change the nature of computing for the general population. A growing number of Americans have computers in their homes and the utility and importance of the National Information Infrastructure will have a direct correlation to the average Americans ability to satisfactorily navigate the "net."

On a personal note, this project has meant a lot to me. Over the past two years my wife would accompany me to school and use the computers there to run Mosaic and visit art archives, opera databases, and other Internet treasures around the world. I wanted to give her (and others with her curiosity and bravery to enter the currently nerd oriented net) the ability to access this tremendous resource from our home at her leisure. I feel that there are many others like her and want to empower their curious impulses with this simple design.

.SOURCES CONSULTED

3Com/Microsoft Corporation. 3Com/Microsoft LAN Manager Network Driver Interface Specification. Version 2.0.1. 1990.

Adams, Phillip M. and Clovis L. Tondo. Writing DOS Device Drivers in C. Englewood Cliffs, NJ.: Prentice-Hall, 1990.

Blanchard, Benjamin S. and Wolter J. Fabrycky. Systems Engineering and Analysis. 2d ed., Englewood Cliffs, NJ.: Prentice Hall, 1990.

Borland International, Inc. Borland C++ 3.1 Programmer's Guide. Scotts Valley, Cal.: Borland International, Inc., 1992.

________, Turbo Debugger 3.0 Users Guide., Scotts Valley, Cal.: Borland International, Inc., 1991.

Comer, Douglas E. Internetworking with TCP/IP, Vol. 1: Principals, Protocols, and Architecture. 2d ed. Englwood Cliffs, NJ.: Prentice-Hall. 1991.

Dettman, Terry. DOS Programmer's Reference. 3d ed. Carmel, Ind.: Que Corp., 1992

Duncan, Ray, ed. The MS-DOS Encyclopedia. Redmond, Wash.: Microsoft Press, 1988.

DalBello, Richard. "The Role of Satellites in the National Information Infrastruture Initiative". Via Satellite. February, 1994. pp 48-56.

Eggebrecht, Lewis C. Interfacing to the IBM Personal Computer. 2d ed., Carmel, Ind.: SAMS, 1992.

Goodwin, Mark. Serial Communications Programming in C and C++. New York: MIS: Press, 1992.

Habayeb, A. R. System Effectiveness. Oxford: Pergamon Press, 1987.

Hunt, Craig. TCP/IP Network Administration. Sebastopol, Cal.: O'Reilly & Associates, 1992.

Intel Corporation. Intel486 Microprocessor Family Programmer's Reference Manual. Mt. Prospect, IL.: Intel Corp. 1992.

Kelly, Al and Ira Pohl. A Book on C: Programming in C. 2d ed. Redwood City, Cal.: Benjamin/Cummings. 1990.

Malamud, Carl. Stacks: Interoperability in Today's Computer Networks. Englewood Cliffs, NJ.: Simon and Schuster Co. 1992.

Microsoft Coporation. Microsoft MS-DOS Programmer's Reference (covers through version 6). 2d ed., Redmond, Wash.: Microsoft Press., 1993

Phoenix Technologies Ltd. System BIOS for IBM PC/XT /AT Computers and Compatibles. Phoenix Technical Reference Series. Reading, Mass.: Addison-Wesley, 1989.

Romkey, J. A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP. RFC 1055, June 1988, 6 pp.