ABSTRACT

Title of Thesis: A SYSTEM DESIGN FOR A HYBRID NETWORK DATA COMMUNICATIONS TERMINAL USING ASYMMETRIC TCP/IP TO SUPPORT INTERNET APPLICATIONS

Name of degree candidate: Aaron David Falk

Degree and Year: Master of Science, 1994

Thesis directed by: Dr. John Baras, Professor, Department of Electrical Engineering

Access to the Internet is either too slow (e.g. dial-up SLIP) or too expensive (e.g. switched 56 kbps, frame relay) for the home user or small enterprise. The Center for Satellite and Hybrid Communication Networks and Hughes Network Systems have collaborated using systems integration principles to develop a prototype of a low-cost hybrid (dial-up and satellite) network terminal which can deliver data from the Internet to the user at rates up to 160 kbps. An asymmetric TCP/IP connection is used breaking the network link into two physical channels: a terrestrial dial-up link for carrying data from the terminal into the Internet and a receive-only satellite link carrying IP packets from the Internet to the user. With a goal of supporting bandwidth hungry Internet applications such as Mosaic, Gopher, and FTP, this system has been designed to support any Intel 80386/486 PC, any commercial TCP/IP package, any unmodified host on the Internet, and any of the routers, etc. within the Internet. The design exploits the following three observations: 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 generate. IP encapsulation, or tunneling, is used to manipulate the TCP/IP protocols to route packets asymmetrically.

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

by

Aaron David Falk

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1994

Advisory Committee:

Professor John Baras, Chairman/Advisor
Professor Anthony Ephremides
Professor Steven Marcus

DEDICATION

To my wife, Caroline.

Table of Contents

Section	Page
List of Tables	iv
List of Figures	v
Purpose of this thesis	1
Introduction	2
The Need for a Hybrid Terminal	5
How the Internet moves data around	9
The Internet	9
TCP/IP	10
Ethernet Routing (ARPs)	13
IP Routing	14
The System Design Process	17
The Design Team	20
Top-Down Design, Bottom-Up Implementation	21
The Hybrid Network Terminal Design	23
The User & User Requirements	27
How It Works	31
The Subsystems	36
The Communications Links	58
Guide to Installing a Prototype User Terminal	61
Hardware	61
Software	62
System Costs	65
Opportunities for Future Development & Research	67
Spoofing TCP	67
Dropping ACKs in the hybrid host	68
Initiating Connections from the Internet	69
User Capacity of the System	70
Conclusion	72
Sources Consulted	74
Appendix A: Annotated Internet Index	76

List of Tables

Number Page
Table 3.1: Win Treese's Internet Index 6
Table 8.1: Transponder Cost per Megabyte 65

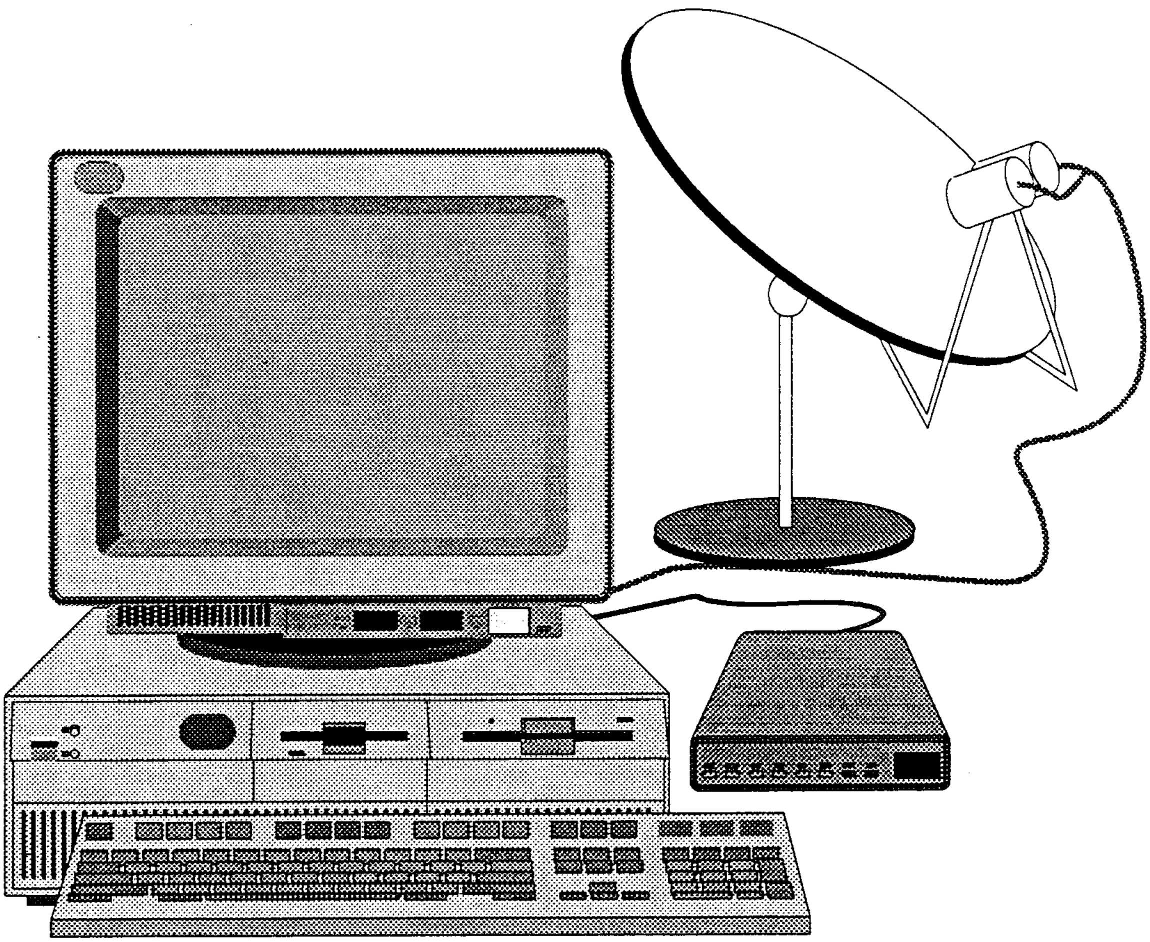
List of Figures

Number	Page
Fig 1.1: The Hybrid Internet Terminal	1
Fig 4.1: The Internet Cloud Representation	10
Fig 4.2: The Conceptual Layers in TCP/IP	11
Fig 4.3: Packet Encapsulation in TCP/IP	12
Fig 4.4: The Internet Network/Gateway Relationship	15
Fig 5.1: The System Design Process	17
Fig 6.1: The Hybrid Network System Schematic	32
Fig 6.2: The FTP Protocol Stack	33
Fig 6.3: The User Terminal Software Stack	38
Fig 6.4: Special SLIP's Entry Points	41
Fig 6.5: IP Packet Tunneling	48
Fig 6.6: Hybrid Gateway Functional Diagram	52
Fig 6.7: Satellite Packet Encapsulation	54
Fig 6.8: SLIP Server Functional Diagram	56
Fig 6.9: Satellite Gateway Functional Diagram	57
Fig 6.10: Application Server Functional Diagram	58

1. Purpose of this thesis

The purpose of this thesis is to use a "top-down design, bottom-up implementation" system design process 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.

Fig 1.1: The hybrid terminal consists of a PC, a modem, a receive-only satellite terminal and the software required to establish a TCP/IP connection over the Internet where all data is transmitted over the modem and received over the satellite.



2. 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 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 a PC compatible and a modem connection to the Internet running the SLIP protocol has been used to define the user requirements. A more detailed description of the user will be included later.

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 only 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 the 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.

The background portion of this thesis will present the need for a hybrid terminal in the context of the National Information Infrastructure, the way in which data travels through the Internet, and some general system design principles. Following the background will be a detailed description of the design of the hybrid terminal and the system in which it operates. In the conclusion, some cost calculations will be presented with several ideas for future design modifications and further research.

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.

3. The Need for a Hybrid Terminal

United States Vice-president Al Gore has proposed a National Information Infrastructure initiative or NII. Sometimes called the "infobahn," the NII is supposed to be a seamless integration of communication technologies with the goal of providing ubiquitous, high-bandwidth connectivity to citizens and businesses throughout the country. As a form of industrial policy, the NII initiative has several focal points. It hopes to encourage private sector investment; preclude a situation of information "haves" and "have-nots" by ensuring that the developing information resources are affordable; encourage innovation in technology; allow for seamless access to the infobahn; and better manage the radio frequency spectrum.

One of the problems the NII faces is that the proposal requires a tremendous investment in new communications infrastructure to provide services for which there is no demonstrated market. The business pages for the past year have been full of announcements of trials and demonstrations by big communications technology companies hoping to prove (or disprove) the existence of a market to justify the investment necessary to create a seamless network.

The closest thing to the infobahn currently in existence is the Internet. Here are some recent statistics on Internet connectivity:

Table 3.1: Win Treese's Internet Index, based on "Harper's Index." Sources are annotated in Appendix A.

The Internet Index

Annual growth rate for Gopher traffic	997%
Annual growth rate for WWW (Mosaic) traffic	341,643%
Average time between new networks connecting to the Internet	10 minutes
Number of newspaper and magazine articles about the Internet during the first nine months of 1993	over 2300
Number of on-line coffeehouses in San Francisco	18
Cost for four minutes Internet time at those coffeehouses	$0.25
Date of first Internet mail message sent by a US President	2 March 1993
Date on which first Stephan King short story published via the Internet before print publication	19 Sept 1993
Number of mail messages carried by IBM's Internet gateways in January, 1993	about 340,000
Advertised network numbers in October, 1993	16,533
Advertised network numbers in October, 1992	7,505
Date after which more than half the registered networks were commercial	August, 1991
Number of Internet hosts in Norway, per 1000 population	5
Number of Internet hosts in United States, per 1000 population	4
Number of Internet hosts in October, 1993	2,056,000
Round-trip time from MIT to mcmvax.mcmurdo.gov in McMurdo, Antarctica	640 milliseconds
Number of hops for above trip	18
Number of USENET articles posted in two weeks during December, 1993	605,000
Number of megabytes used	1450
Number of users posting	130,000
Number of sites represented	42,000
Number of Silicon Valley real estate agencies advertising with email addresses	1
Terabytes carried by NSFnet backbone in February, 1993	5
Number of countries reachable by electronic mail	137 (approx.)
Number of countris not reachable by electronic mail	99 (approx.)
Number of countris on the Internet	60
Amount of time it takes Supreme Court decisions to become available on the Internet	less than one day
Date of first National Public Radio program broadcast simultaneously on the Internet	21 May 1993
Percent of Boardwatch Top 100 BBS systems with Internet connectivity	21

[source: Win Treese, see Appendix A]

Some companies which hope to offer services on the infobahn are developing similar services available through the Internet. A great surge in Internet popularity has been felt since the introduction of the software application Mosaic which combines several previously difficult to use Internet applications into an intuitive point-and-click interface.

Early visions of the infobahn were considered hybrid in the sense that the different terrestrial media operated by different companies would be seamlessly integrated. That "hybrid" vision soon included terrestrial wireless technologies. Satellite technology was not included in the discussion for some time because of the long propagation delays in geosynchronous satellite communications. Historically, high data rate networks have been terrestrial, usually fiber, and have had low delay. Lately, satellite companies have become interested in being part of the NII and have been promoting the advantages of satellite communications as part of a national network.

The advantages of including satellites in the NII are that: (1) satellites can cover large geographical areas very effectively, (2) satellites can be reached by mobile platforms, (3) satellites are insensitive to the distance between transmitter and receiver, (4) satellites are very efficient at point-to-multipoint and broadcast applications, and (5) satellites are resilient to natural and man-made disasters.[Dal Bello, 94]

However, from a users point of view, satellites are expensive to use because of the costs involved in purchasing and installing a terminal. These costs have placed satellite communications outside of the reach of the consumer with the exception of analog TV reception. As this prototype will hope to demonstrate, satellites can have a role in consumer access to the Internet. Most of the discussion regarding bringing high data rates to the consumer have focused on one of three technologies: fiber optics, a modified cable TV plant, or ISDN. All three of these technologies are many years from being ubiquitously implemented. The Washington Post recently estimated that to wire every house in the country with fiber optics would cost 200 billion dollars, but every estimate has the caveat that the real cost is unknown. The only entity with the resources to undertake such an expensive effort is the US government which is not in the business of wiring homes for networking. Modifying the cable TV plant for bidirectional transmission of data signals would also require a massive deployment of resources costing billions. Similarly, ISDN is a data transfer technology that has been atrophying because of tariffing issues and the cost of deploying the new technology.

Regardless of who does the installation or how much it will cost, if it ever happens, the fact is that it will take a tremendous amount of time to connect the millions of homes in the nation to the NII. In many rural areas, it may never be cost effective to run a cable from the network to the home. Until these homes get a high bandwidth cable connecting them to the NII, satellites offer a very cost effective way to deliver high bandwidth data. For this reason, more than any other, satellites and hybrid terminals both deserve a place in the NII.

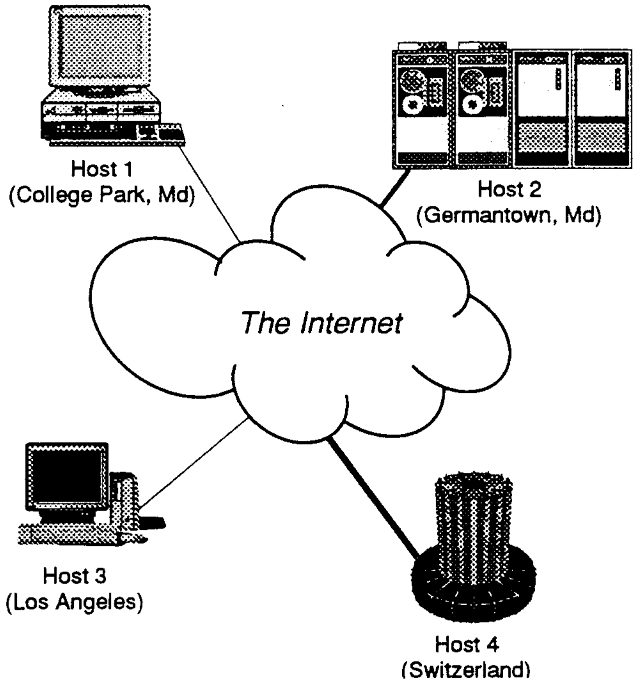
4. How the Internet moves data around

In order to understand how asymmetric TCP/IP is accomplished it is necessary to understand how the Internet moves data around and the encapsulation hierarchy of the payload. We provide next a *brief* description of the Internet protocols and a discussion on how packet routing is accomplished on the Internet.

4.1. The Internet

In his excellent book on internetworking, Stacks, Carl Malamud defines the Internet as "the collection of autonomously administered backbones and regional networks, and many thousands of local networks...[Consider it] a single meta-network because all the hosts share a common suite of protocols." In network diagrams (including those here) the Internet is represented as a cloud and with good reason. It is formed of a collection of heterogeneous hosts supporting a variety of operating systems and directory structures with connections that have varying delays and bandwidths. There is an enormous amount which cannot be specified about the Internet, but what can be stated is that the hosts share the same suite of networking protocols known as Transmission Control Protocol/Internet Protocol or TCP/IP.

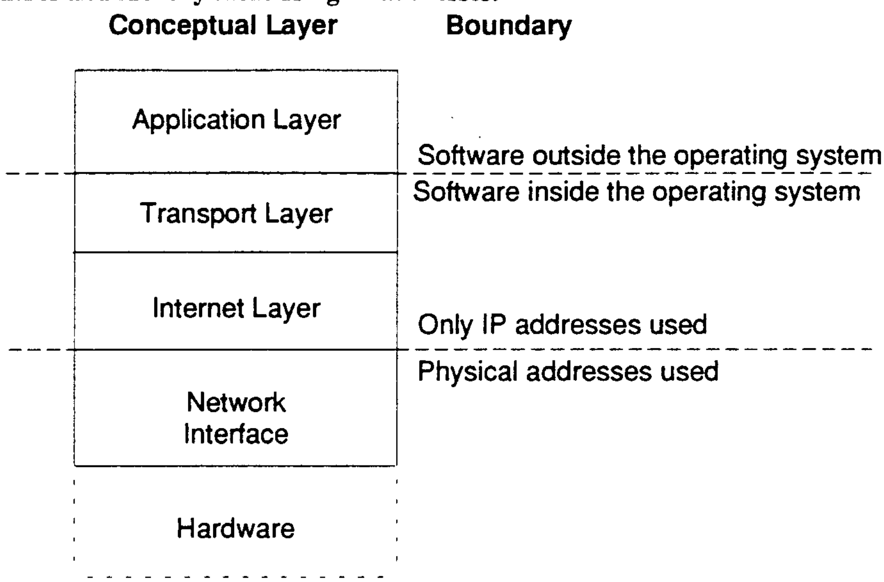
Fig 4.1 The cloud representation of the Internet implies the heterogeneous interconnection of a wide variety of machines and network architectures sharing a common network protocol—TCP/IP.



4.2. TCP/IP

Doug Comer, author of the basic text on Internet protocols, Internetworking with TCP/IP, Vol. I, states that the basic goal of TCP/IP is to "provide a virtual network that offers a connectionless IP datagram delivery service." TCP/IP refers to a set of protocols of which TCP and IP are but two. These protocols are implemented to varying degree on systems connected to the Internet to provide just what Comer referred to: connectionless delivery of packets.

Fig 4.2 The Application ,Transport, and Internet layers are unaware of the actual physical addresses of other machines and identify them using IP addresses.



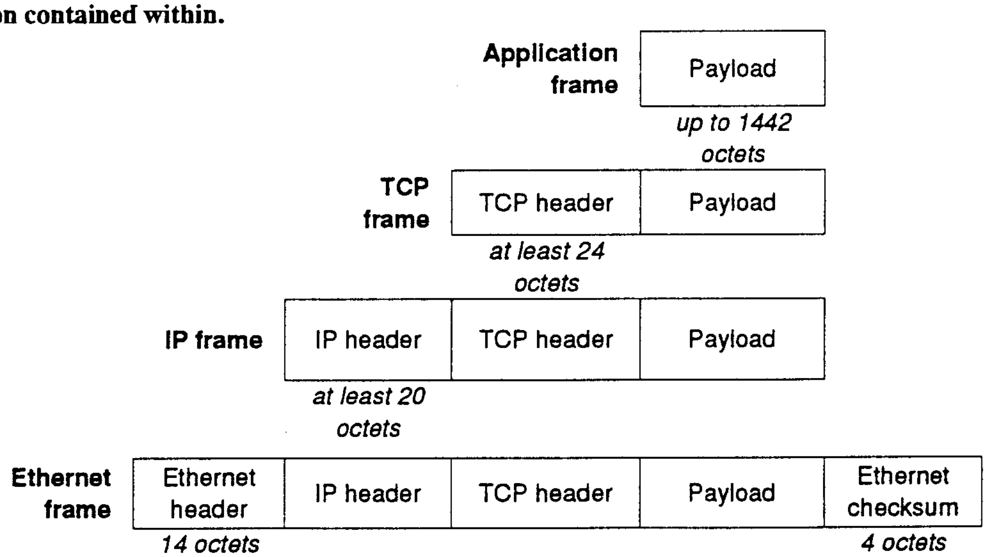
source: [Comer, 91]

IP packets are the basic unit of data found on an Internet network. While they vary in sizes, all IP packets have a header which contains, among other things, fields identifying the 32 bit IP address of the sender and destination of the packet. The IP delivery service provided by the Internet is not a reliable service. In other words, the network will make a "best effort" to deliver packets to the destination IP address but does not guarantee delivery or, if delivered, that the packets will arrive without errors or in order. These tasks are left to higher level protocols which are encapsulated within the IP packet.

TCP and UDP are examples of protocols which are encapsulated by IP. UDP is the Unreliable Data Protocol. It is a simple protocol which doesn't have an acknowledgment scheme and is useful for non-critical data transfer applications. TCP, the Transport Control Protocol, assures delivery of sequenced packets from sender to receiver. The added functionality of TCP means added overhead in packet header size and processing delay. Nevertheless, this protocol is used widely as a method to reliably deliver packets across the Internet.

All the Internet applications mentioned in the introduction use these protocols to send data around the Internet. A client application would generate a packet destined for a server application residing on a machine elsewhere in the Internet and identified by its IP address. The application would pass its packet to TCP or UDP which would encapsulate it as the payload within a TCP or UDP packet and hand it to IP. IP would in turn add an IP header to the packet and hand it to the driver. The driver would send the packet to the first hop towards its destination. If the physical medium the client was connected to was ethernet, the driver would encapsulate the IP packet within an ethernet packet.

Fig 4.3 The frame of data from an application is encapsulated by the TCP, IP, and ethernet protocols. Each protocol treats the frame passed to it entirely as payload and remains ignorant of the header information contained within.



4.3. Ethernet Routing (ARPs)

Ethernet is a special situation for routing IP packets because it is so common. Almost all LAN development in this project has been on ethernet LANs and an understanding of how packets move around on an ethernet is useful for understanding this project.

So far, IP addresses have been discussed as the source and destination address of a packet. An IP address is a virtual address which locates the machine as part of a logical network. It can be reallocated to a different physical machine if necessary. Ethernet physical addresses or Media Access Card (MAC) addresses are 48 bit addresses which are completely unique. The MAC address identifies an ethernet card on the physical network and cannot be reallocated to another machine without moving the ethernet card to the other machine. So, it can be seen that there are at least two levels of addressing in TCP/IP and a mapping is necessary between these two levels.

Now the question arises as to how a machine which has only the IP address of the desired destination finds out how to reach that destination. This is where ARPs come in. To reach a machine that may be on the same physical network, the source machine broadcasts a special type of packet called an ARP (for Address Resolution Protocol) packet. The ARP contains within it the IP address and MAC address of the machine sending it and the IP address of the destination.

All the machines on the physical network will receive the ARP broadcast. If the destination machine is on the same physical network, it will respond to it, inserting the correct MAC address. The destination machine will save the MAC/IP address combination of the sender in an ARP table and when the sender receives the reply it will do the same. Now, the two machines can communicate since they both have a mapping between their MAC and IP addresses. If the sender does not receive a reply to the ARP, the sender will assume that the machine cannot be reached directly.

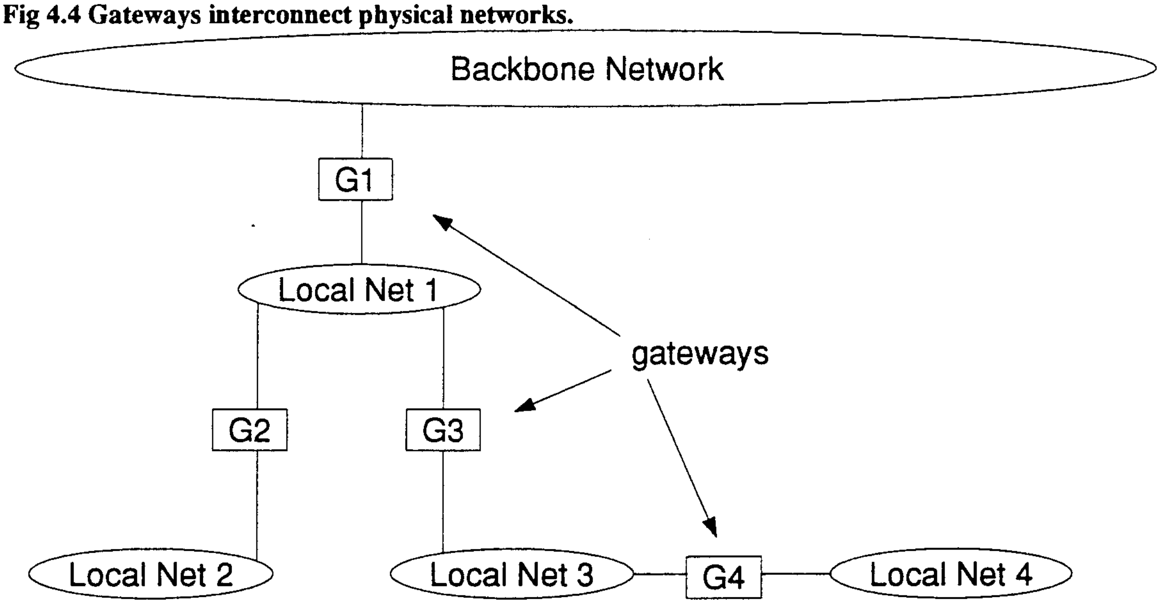
4.4. IP Routing

Of course, the whole point of the Internet is to interconnect machines across physical networks. Many physical networks can be combined to create a logical network. Physical networks refer to machines which are connected by a common wire. Logical networks are networks with the same beginning IP address. Logical networks are allocated by an Internet management body and refer to all the machines within the educational, governmental, or commercial organization. IP addresses have the form A.B.C.D where A, B, C, and D are decimal numbers between 0 and 255. These are decimal representations of the four bytes which make up the 32 bit IP address. Of the 32 bits, a portion of the high order bits identify the logical network while the remaining low order bits identify the specific machine. For instance, in the prototype network used in this project, the network is identified by 198.77.116.X and the specific machine would occupy the last decimal entry, i.e. the last 8 bits of the address. Each logical network has an associated domain name. At the

University of Maryland the domain name is umd.edu. The domain name for the prototype network is direcpc.com.

Physical networks are interconnected using gateways . Gateways will route IP packets across different physical networks. Each machine on a physical network can tell by examining the IP address whether or not the destination machine is on the same physical network as itself. If the destination is on the same physical network, the source machine will use ARP to send the packet to the destination. Otherwise, the machine will use a routing table to determine the MAC address to which to send the packet so that it can be forwarded.

Fig 4.4 Gateways interconnect physical networks.



source: [Comer, 91]

The routing table is a table with a mapping of network IP addresses to gateway IP addresses. The host searches for the network of the destination address and finds the gateway that it should send the packet to. Gateways have an IP address on each network they connect and so they are always reachable on the same physical network. Therefore, their IP addresses can be resolved using ARP. Note that in the table the network IPs identify entire networks of machines by their network IP prefix and so have a form similar to A.B.C.0 where 0 is used as a wildcard stating that any machine with the prefix A.B.C can be reached by that gateway. A gateway is a specific machine that the packet needs to be forwarded to in order to reach the destination network. Once the packet reaches the gateway that is physically connected to the destination machine, the packet is delivered to its destination using ARP.

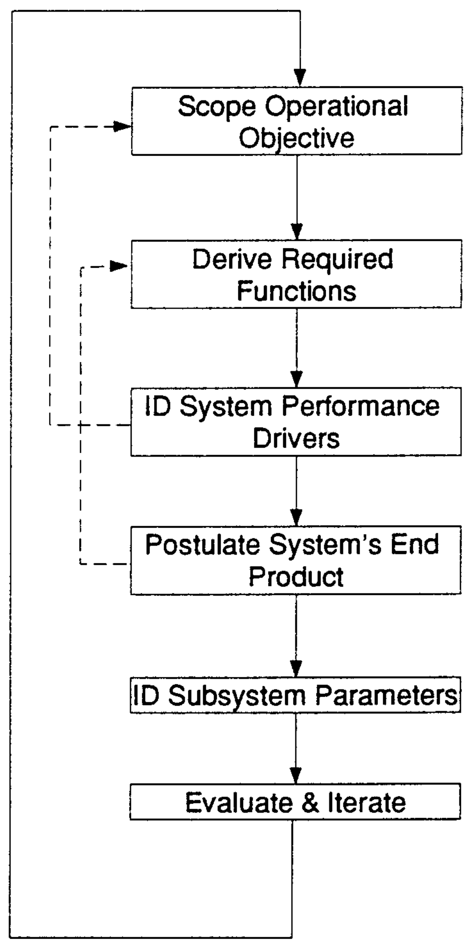
Topics such as routing on media other than ethernet, domain name to IP number resolution, the Internet organizational structure (or lack of it), and how the routing tables are created and maintained are beyond the scope of this discussion, however [Comer, 91] is an excellent primer on the topic. More sophisticated readers are referred to [Malamud, 92].

5. The System Design Process

Professor A. R. Habayeb, in his lectures on System Effectiveness to the System Engineering program in the Fall of 1993, presented an outline of the system design process. These elements of the process of bringing a system into being were used informally in the prototype terminal development and are described briefly below.

The seven step process of designing systems can be summarized as follows:

Fig 5.1 The system design process is multiply iterative. As the design is refined, changes are made and previous tasks must be repeated to insure the user's needs will be met.



Step 1: Scoping the operational objective. Articulate the user's need identifying the ultimate goal the system should accomplish. This typically will start as a vague "want" or "desire" and can be refined by iterations between the system designer and the system user into a concrete operational objective which the system must fulfill. This step serves to insure that the user's needs drive the system design before other considerations. It is necessary to proceed in this way to avoid designs that have "technology for technology's sake."

Step 2: Derive required functions. Analyze the operational objective to identify the functions the system needs to perform.

Step 3: Identify system performance drivers. Find the key parameters describing how the required functions need to perform. This can include physical characteristics, bandwidth, cost, reliability, design flexibility, and technical, bureaucratic, or cost risk.

Step 4: Postulate the systems end-product. Synthesize the above elements into a "virtual system." Allocate functions of the system to subsystems. This is the most creative part of the iterative process where the virtual system undergoes constant revision as the design definition progresses.

Step 5: Identify subsystem parameters. Determine the critical performance drivers of each subsystem. Similar to Step 3.

Step 6: Determine measures of adequacy. Find parameters of the resulting system that can be used to determine if the operational objective is being met. The more detailed or revealing the measure of adequacy, the easier it is to identify the modifications needed to meet the operational objective.

Step 7: Evaluate and iterate. Focus on the design of the subsystems and meeting the user requirements.

In order to accomplish Step 7, Habayeb presents a five step focusing process for the selection and evaluation criteria for systems as follows:

Step 1: Operational requirement assessment: Determine the subsystem functional parameters. What are the functions that each subsystem need to perform (from Step 3 above)?

Step 2: System performance specification and subsystem identification: Determine the building blocks of the subsystems.

Step 3: System and subsystem parameter specification: What parameters will drive the design?, i.e. What will the system do that can be measured?

Step 4: Subsystem design parameters: For each subsystem, assign values for the parameters identified in Step 3.

Step 5: Subsystem design specification: Complete the design to satisfy parameters derived in Step 4.

This process defines a top-down design process where the high-level operational requirements propagate through the design process to the system parameter specification, subsystem functional parameter specification, subsystem design parameter identification, and finally to the subsystem design specification.

Some criteria are necessary to determine if the subsystems are being designed to adequately support the user's requirements. One way to evaluate and select subsystems is based on some combination of the following seven factors: 1) performance, 2) cost, 3) reliability, 4) operational readiness, 5) technical or bureaucratic risk (as it applies to system cost) 6) physical parameters and 7) design flexibility. How these factors are applied to the system design depends on the particular system and user requirements.

5.1. The Design Team

The design team for the prototype development consisted of a partnership between Hughes

Network Systems and the Center for Satellite and Hybrid Communication Networks (a NASA Center for Commercial Development of Space) at the University of Maryland. This project was funded using a Maryland Industrial Partnership grant which was matched by HNS. CSHCN director Dr. John Baras supervised the long term goals and assistant director Tim Kirkwood acted as industrial liaison between the CSHCN and HNS and SURAnet. A graduate student (the author) and an undergraduate student, Narin Suphasindhu, worked closely with Hughes engineers Doug Dillon, Ilya Faenson, and Bill Stanton on the actual design and implementation of the prototype terminal.

5.2. "Top-Down Design, Bottom-Up Implementation"

At the initiation of the project, the University of Maryland was interested in creating a proof-of-concept prototype incorporating the idea of using an asymmetric link to bring high bandwidth data over satellite to a user. HNS could provide technology in the form of the DirecPC card but was only interested if a prototype could be developed quickly. Early discussions between the two groups resulted in a project definition including the following: very quickly complete a system design and build a prototype using the DirecPC card to provide high bandwidth access to Internet applications for a user with only a modem for transmit capabilities.

The design team used the principles of the system design process discussed above starting with the user requirements to system functions, allocate functions to subsystems, and define subsystem functional and design parameters. However, to reduce time spent in iterations the team chose a bottom-up implementation instead completing subsystem specifications.

Fortunately, the development required for subsystem creation consisted of obtaining existing hardware and creating or modifying software for it. For each subsystem, once the interfaces and design parameters were determined (although not necessarily fixed) the implementor would start writing code. The coding process frequently revealed design issues that needed to be resolved resulting in iterations between definition of the interfaces and parameters and software implementation and testing.

Arguably, this was not the most elegant method of system design since it resulted in code that was not completely thought out from the beginning. However, it did make for rapid implementation and testing since many design problems could be solved on the spot. This certainly would not have been possible without the assistance of the experienced HNS engineers made available for this project.

6. The Hybrid Network Terminal Design

NCSA Mosaic is an application which allows users to browse through lists of such files, using a point and click interface, and download them from anywhere in the Internet. Mosaic calls on applications such as FTP (File Transfer Protocol) and Gopher (an Internet navigating application) to support its intuitive interface. These applications operate best with high bandwidth connections because they are frequently used to transfer (and present) digitized image, audio, or executable files all of which can be large, on the order of megabyte. Often the user would not know before downloading and examining the file whether he was interested in it or not in the same way a researcher in a library pulls a book from a shelf to examine.

That delay experienced waiting for a file transfer can be small in the case of the ethernet on the order of seconds but quite daunting for a dial-up configuration. For instance, using a 2400 baud modem to transfer a 500k byte file would take around 30 minutes, hardly a data transfer rate that would encourage "browsing"! These applications have been developed in an environment where bandwidth was plentiful. Most university LANs are ethernet, running at a (theoretical) maximum bandwidth of 10Mb/s. At 10Mb/s it would take about 1/2 second of delay to transfer the same 500k file. These calculations do not take into account the delay encountered by the data as it crosses the Internet but only that delay experienced at that last hop from the local network's router to the user terminal.

These facts essentially relegate useful access to the "browseable" data on the Internet to only those users who have access to a high speed LAN. This is the same problem the NII faces: how does one get the data to cross the last link or "mile" to the user's site? And it is exactly this problem that this system design wishes to address. This design will show how computer users equipped with dial-up SLIP connection and a receive-only satellite terminal can access any Internet host that is available to a user with full ethernet connectivity by exploiting the three following observations: 1) satellites are a very cost effective way to bring high bandwidth to users who are geographically diverse; 2) receive-only VSATs are cheaper to manufacture and install than bidirectional satellite terminals; and 3) most users wish to receive, or consume, much more data than they generate in effect creating an asymmetric demand for bandwidth.

Most home computer users do not have access to ethernet connectivity because of proximity issues. Suppose your employer, or professor, was willing to let you access the company LAN from home. How would you get an ethernet connection? Ethernet requires running a coax cable from one computer on the LAN to the next. The total length of the coax on a subnet of ethernet is limited to several hundred feet for reasons of propagation delay at the high data rates. Therefore, if all other costs were equal, it would be impossible to access a high speed LAN without being in close proximity to the router or hub.

Another solution, besides running an ethernet cable from your office to your home, would be to purchase a high bandwidth connection from the telephone company. Today, the phone company prices its data connections in multiples of voice connections. Therefore, if your high speed data connection could carry 3 simultaneous phone conversations, telco would charge you a rate roughly equivalent to three phone lines. A 1.5Mb/s connection (T1 rate) is capable of carrying (assuming 64kb/s per voice connection) about 24 voice conversations! Additionally, telco adds distance charges sometimes calculated by the 1/4 mile. This is because the phone company did not design their network to carry this kind of traffic and, even though they are willing to charge you and provide the service, sometimes they have to actually buy and install the central office equipment and lay the cables to do it. Additionally, the hardware required for a home user to connect a PC to a T1 line can easily cost over $3000—much more than the PC itself!

Satellites, on the other hand, sacrifice the proximity requirement for one of delay. With a satellite connection, the transmitter and the receiver can be thousands of miles away. And, even though there is a mandatory 500 millisecond delay when using a geosynchronous satellite for the connection, the delay is not proportional to the distance between transmitter and receiver. Sender and receiver can be on opposite sides of the country and incur the same costs as those within the same city. This is advantageous for services where the customers may be spread out. Also, satellites cost less, not more, to operate as the number of users increase. There are potential queuing delays at the transmitter but, besides that, more customers just means a cheaper service because the fixed costs can be spread over a larger group.

Hughes Network Systems has explored the user costs for VSATs and has determined that a major portion of the cost in purchasing a satellite terminal is installing and aiming the antenna. Often antennae must be mounted on the customer's roof, resulting in carpentry, risk, and potential violation of neighborhood housing codes. Having the antenna exposed to the wind on the roof also means there is an upper limit to the size of the antenna because of the wind shear. It is a fact that receive-only antennae are cheaper to manufacture requiring less electronics, by about one third. They can also be made smaller and are easier to aim. To aim a transmitting antenna requires aiming the focus of the beam so that it is on an impossibly tiny speck of a satellite in the distance. To aim a receiving antenna, one must turn the dish so that it is optimally aligned with a very large beam coming from the satellite. The signals received by a satellite from earth are competing with the noise generated by the earth's thermal radiation, thus increasing the size and/or power requirements (translate: cost) of a transmitting terminal.

By considering what a typical user does on the Internet it is possible to understand the asymmetric link assumption. When most users are on the Internet, i.e. outside of their own LAN, they are either navigating the Internet or transferring data. Navigating includes moving from place to place, logging in to other machines, changing directories, etc. Transferring data includes requesting information, FTP'ing files, using gopher, Archie, WAIS, WWW, Mosaic or some other Internet application. An obvious exception to this observation would be when the user runs an application over the network, that is, when the user runs a process on a remote machine and has the local machine act as the display, such as with Telnet. This form of traffic is more symmetric and, on a hybrid terminal, would operate at the rate of the slower of the two links.

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. The subsystem section describes the functions and design of each functional subsystem. The communications links section describes some of the parameters of the links interconnecting the different subsystems used in the prototype.

6.1. 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 discussed below:

The system must provide significantly less delay responding to requests for large data files experienced using a modem. This requirement captures the essential of the goal of the system: that the user wants rapid access to large amounts of data available in the Internet.

The system must work with any 386/486 33MHz machine. The goal is to support the largest group of potential users as possible. This requires fixing a potential hardware platform. For the proof-of-concept, we will fix a specific model of PC but the design must support all 386 or 486 based machines. Other hardware platforms, such as Apple Macintoshes, can be supported in the future but will require substantial effort in software and hardware development. At this time the HNS receive-only satellite hardware (the DirecPC card or BIC) is only designed to support microchannel IBM PCs. However, release of an ISA bus card is immanent and this would allow PC "clones" to be used with the card. The 33MHz processor speed is necessary to support the significant real-time processing required for handling the network protocols. Theoretically, the protocol handling could be placed in hardware in the future and allow a relaxation of this requirement. However, computer processor speeds are increasing and future platforms will probably have more computing power, not less.

The system must work with any commercial TCP/IP package. This year, at least two companies (Frontier and Spry) have released software packages described by the trade magazines as "Internet in a box." These companies are clearly trying to capitalize on the recent popularity of Internet and have priced these packages significantly below previous TCP/IP application packages in an apparent attempt to gain market share. These companies are pursuing the same customers as our defined users and so these are the types of TCP/IP packages our users may have. If this turns out to be a big market, which is necessary for commercial success of this system, there will be more “Internet-in-a-box”-type packages. Therefore, this system should be flexible enough to work with any of them.

This means that no modifications should be required of the user's TCP/IP package, since we don't know what it will be. This also implies that modifications to the user's system should be contained in a single loadable module, if possible, such as a device driver and should support a popular, standard software interface.

The system must work with any commercial SLIP service provider. Since the user can be located anywhere in the continental US, we don't want to assume anything about the kind of access he has to the Internet. SLIP is the most primitive form of a modem connection to the Internet and is available from some national Internet access providers with the potential for more widespread availability in the future. A minimal SLIP service contract would cover access to the Internet by a single user machine. We are assuming in this case that the SLIP service provider may only allow traffic originating from that one machine to cross the modem link from the user's site to the SLIP providers host.

The system must be able to access any Internet host. The Internet is vast and heterogeneous. However, as described in previous sections the common feature of all (or most) sites available over the Internet is supporting the common network access protocols of TCP/IP. To require that this system use only specific, modified hosts available on the Internet would undermine the great virtue the of Internet, i.e. its incredibly wide accessibility. Therefore, a user requirement for this system is to require no modification to an Internet site for access by the user.

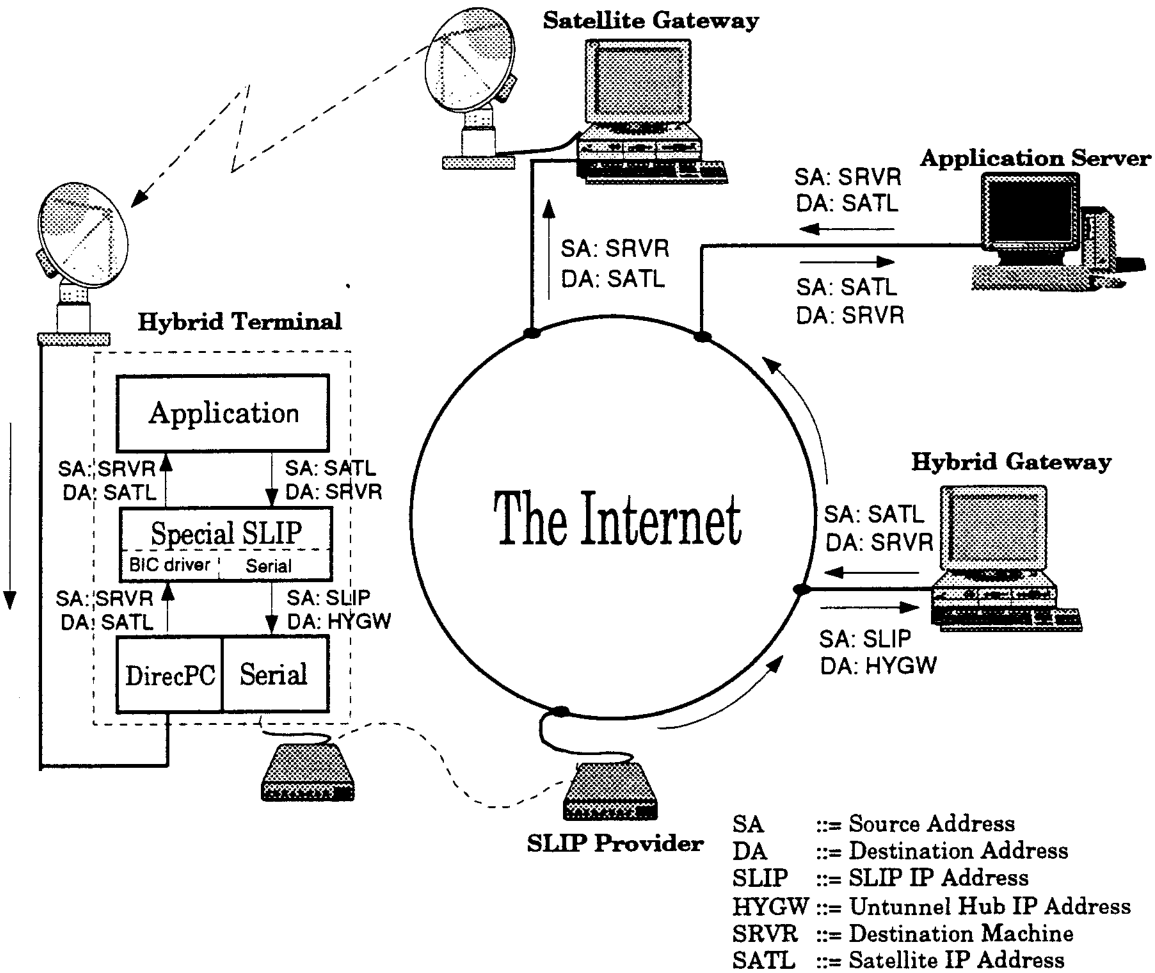
The system must support Internet initiated connections. To allow outside sites to initiate a connection to a user terminal connected to the Internet using this hybrid system would be a significant qualitative advantage over a modem-only connected system. In an Internet initiated session, a session request would arrive over the satellite link and would cause the terminal to initiate dial-up procedures to close the bidirectional connection by connecting to the SLIP server. At this time no other TCP/IP package allows dial-*in* operability. This could potentially allow Email delivery or remote file access without a pre-existing modem connection to the SLIP service.

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

6.2. How It Works

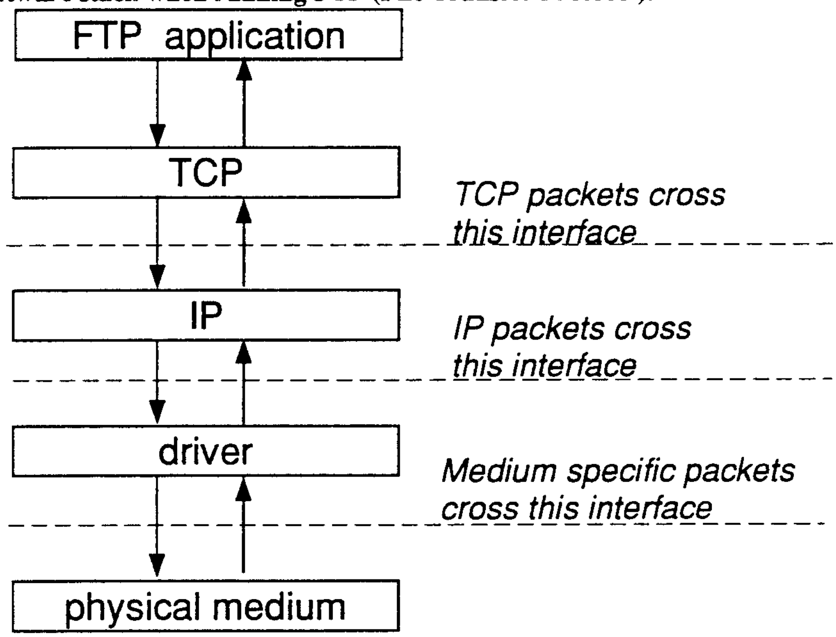
In this section we describe the general process of how a request from the user terminal is carried through the Internet and to a machine running a host application and how that machine's response is carried back to the user. The various subsystems are introduced here and will be described in more detail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

Fig 6.1 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.



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.

Fig 6.2 The software stack when running FTP (File Transfer Protocol).



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 re-

quest 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.

6.3. The Subsystems

In this section each of the prototype system's subsystems are described ordered by the amount of development effort required to bring the prototype into being. While this prototype does not satisfy every user requirement, the design does not exclude incorporating the additional development required to satisfy the remaining ones.

Specifically, the prototype may not provide faster access than some advanced modems and it does not support Internet initiated connections. Both of these issues will be described in the chapter on opportunities for future work.

There are five subsystems in this system design: the User Terminal, the SLIP Server, the Application Server, the Hybrid Gateway, and the Satellite Gateway. As stated in the user requirements definition, the SLIP Server and the Application Server require no modification, however, they perform essential functions in the system and are included for completeness.

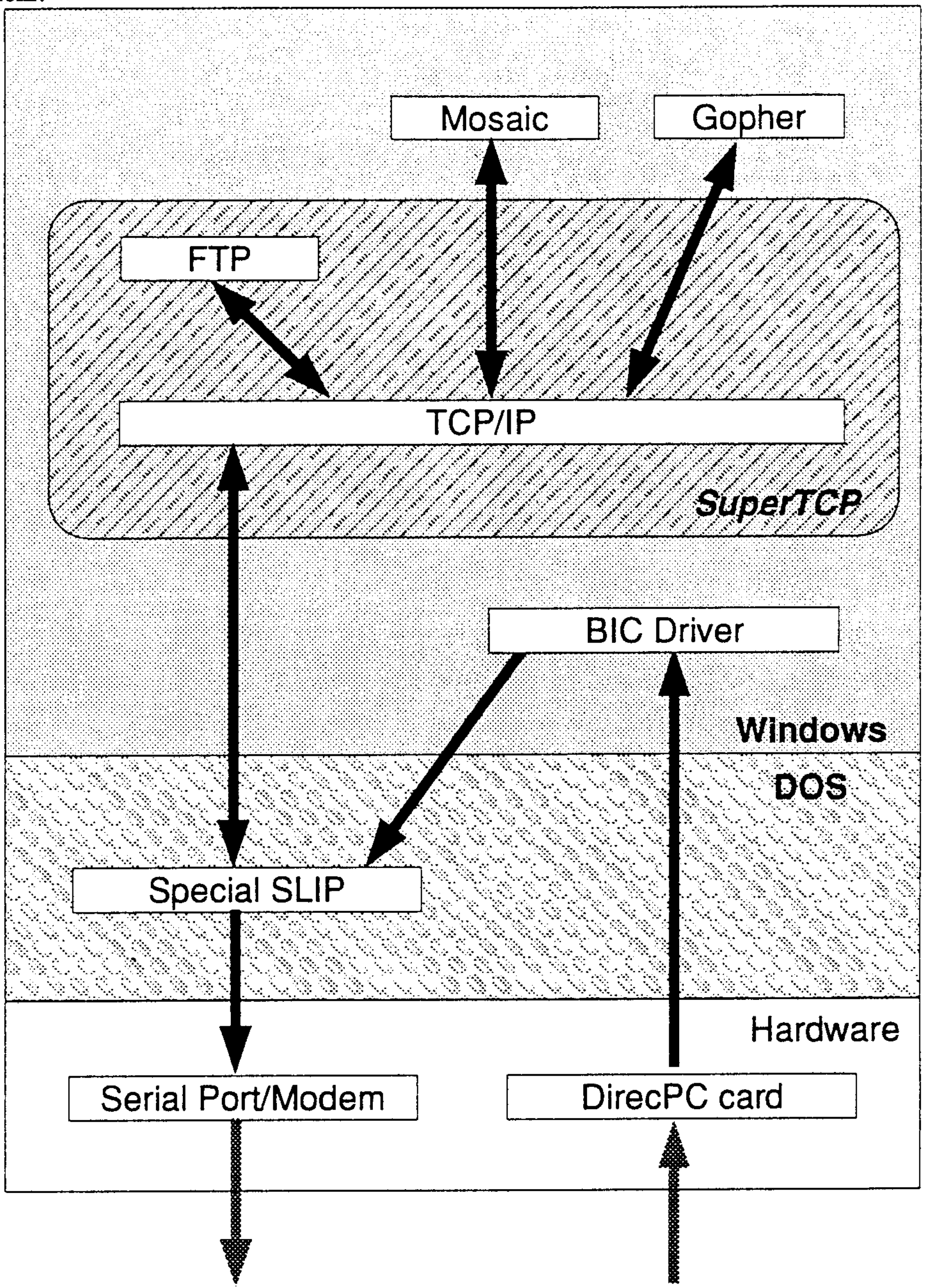
The User Terminal

The User Terminal has required the most development. A driver 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 6.3 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].

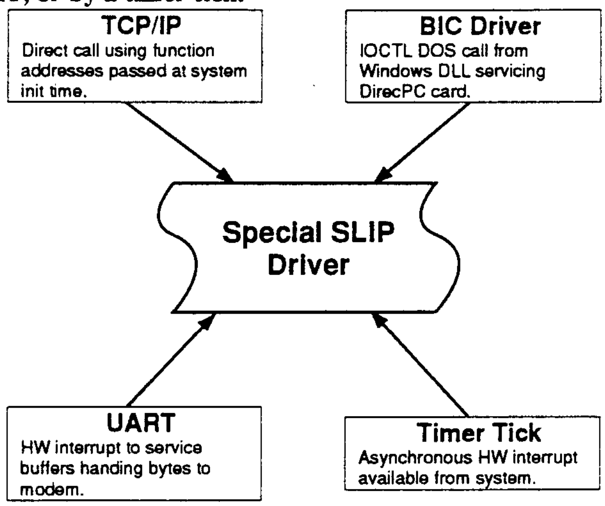
The serial data is sent and received through the RS-232 connector by one of two chips, the U8250 or the U16550 also called Universal Asynchronous Receiver Transmitters (UARTs). These chips handle the timing and some of the signaling necessary to send data over the serial interface. Both chips are identical except that the U16550 has a sixteen byte receive buffer while the U8250 has a one byte buffer. Sometimes the U16550 is called a "high speed serial interface" since high speed data can be received over it and serviced without overflows caused by delays in the operating system. There are 10 registers on the chips which are used to configure the connection rate, number of data bits, stop bits, parity and other parameters of the connection. The RS-232 interface provides control lines to support timing on the connection, however, use of these control lines are optional and are not implemented in this prototype design.

To send data over the serial port, a byte is written to the memory address corresponding to the register that holds data to be transmitted. The chip will then send the bits in the byte out over the serial line, one bit at a time, at the configured rate with the configured stop bits, data bits, and parity. When the entire byte has been sent, the chip sets a hardware interrupt notifying the CPU that the chip's transmit register is empty. The driver software needs to handle that interrupt and either write another byte to the register (if there is one to transmit) or disable the Transmitter Buffer Empty interrupt.

In the same fashion, data is received over serial line one bit at a time. When an entire byte has been received, the chip sets a hardware interrupt (Receive Buffer Full) to request service by the CPU. In the U16550, this interrupt can be delayed to occur when a configurable number of bytes (up to 16) have been received, reducing the frequency of the hardware interrupts. The register acts as a FIFO, so each read empties the buffer of the next buffered byte. This interrupt goes away when all the data have been read from the receive buffer register.

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.

Fig 6.4 The Special SLIP driver can be entered synchronously from TCP/IP or asynchronously from the BIC driver, the UART, or by a timer tick.



DOS driver call support

The DOS operating system communicates with device drivers using a set of carefully specified function calls. These calls are used for the operating system to communicate with storage, or block, devices such as disk drives where data is moved in multi-character blocks as well as character devices such as the serial port or keyboard where data is moved one character at a time. DOS device driver calls include functions such as init_cmd() to initialize a device as well as input_cmd() and output_cmd(), which are used to input data into the device from the operating system and output data from the device to the operating system, respectively.

DOS calls are used for several purposes in Special SLIP. At boot time the UART needs to be initialized with the correct parameters and this command will come via a DOS call.

Also, received packets from the BIC are passed from the BIC driver to the TCP/IP package by first passing them to Special SLIP using a DOS call. In this fashion the two physical links are combined within Special SLIP and this is how the two links are made to appear to the upper layers of software as one.

DOS calls are also used in Special SLIP to pass control commands into the driver during run time. At boot time, the device driver initializes and then terminates and stays resident. This means that control of the CPU is released for other uses but the driver's executable code remains in memory. The only way to communicate with the driver after that is either to use a DOS call (as the BIC does) or call a function within the driver by using its memory address (as TCP/IP does using the NDIS interface). It is sometimes convenient to alter some of the functions of the driver during run time. This can be done by passing a character into the driver using the DOS redirect command. Therefore, in order to pass a control character, say X, into a driver named HYSLIP$ the following command could be used from a DOS prompt: echo X>HYSLIP$. The DOS redirect command (">") will call the function input_cmd() in the driver which can check the character being inputted and can take appropriate action.

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.

Timer tick handling

The DOS timer is a useful mechanism for initiating asynchronous processes. It is a regular interrupt that occurs in every PC and can be used to handle process that may go idle and need to be restarted later. For instance, in some situations data will be queued up in Special SLIP to be sent and the transmitter will have been made idle. On every timer tick (19 times each second) a check is made as to whether the transmitter needs to be kick started. Also, the driver interface specification being used declares that the TCP/IP package sometimes needs to receive an asynchronous interrupt, meaning that the TCP/IP package can sometimes say "Interrupt me at some unknown time after this call." This is sometimes used to do additional processing on received data. The timer is an easy means to get asynchronous interrupt and can trigger the requested call.

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.

By supporting the NDIS specification, it is possible to declare that our driver is for an ethernet card fooling TCP/IP into thinking that it can send and receive at 10Mb/s data rates. This is a great advantage on the receive side because we will be receiving data at high rates. However, caution is required on the transmit side since the modem is not capable of such a high rate. Fortunately, the nature of ethernet communication is that there are collisions on the physical cable reducing the maximum transmission rate and the TCP/IP package can handle situations where transmission goes very slowly. Another issue is that TCP/IP is sending ethernet packets to the driver, i.e. the IP packets have been encapsulated in an ethernet packet. The SLIP connection is for IP packets, meaning that the ethernet header needs to be stripped off each packet as TCP/IP sends it.

By using NDIS, two entities on either side of the interface (Special SLIP and SuperTCP) pass each other the addresses of the functions they want to call. Tables are created containing the addresses of all the functions that are callable by the other entity and the address of the table is passed to the other entity at init time. Suppose TCP/IP wants to transmit data over the network. The NDIS specification insures that TCP/IP will have been passed the address of the Special SLIP function TransmitData() which TCP/IP will call in order to send data.

Queue handling (including overflow)

As data is transmitted from TCP/IP or received from the BIC driver, it is copied and buffered within Special SLIP. Therefore, Special SLIP needs to manage a fixed size transmit and receive queue. The fixed size is important because device drivers under DOS cannot allocate memory dynamically. A certain amount is requested at init time and that must be managed by the driver.

As mentioned above, there may be situations where no buffers are available, say, for transmission because TCP/IP is sending packets to transmit faster than the modem can send them over the serial line. In this case the queue handler software must gracefully drop the packets and return an error to the calling function so that the rate of transmission can be decreased.

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.

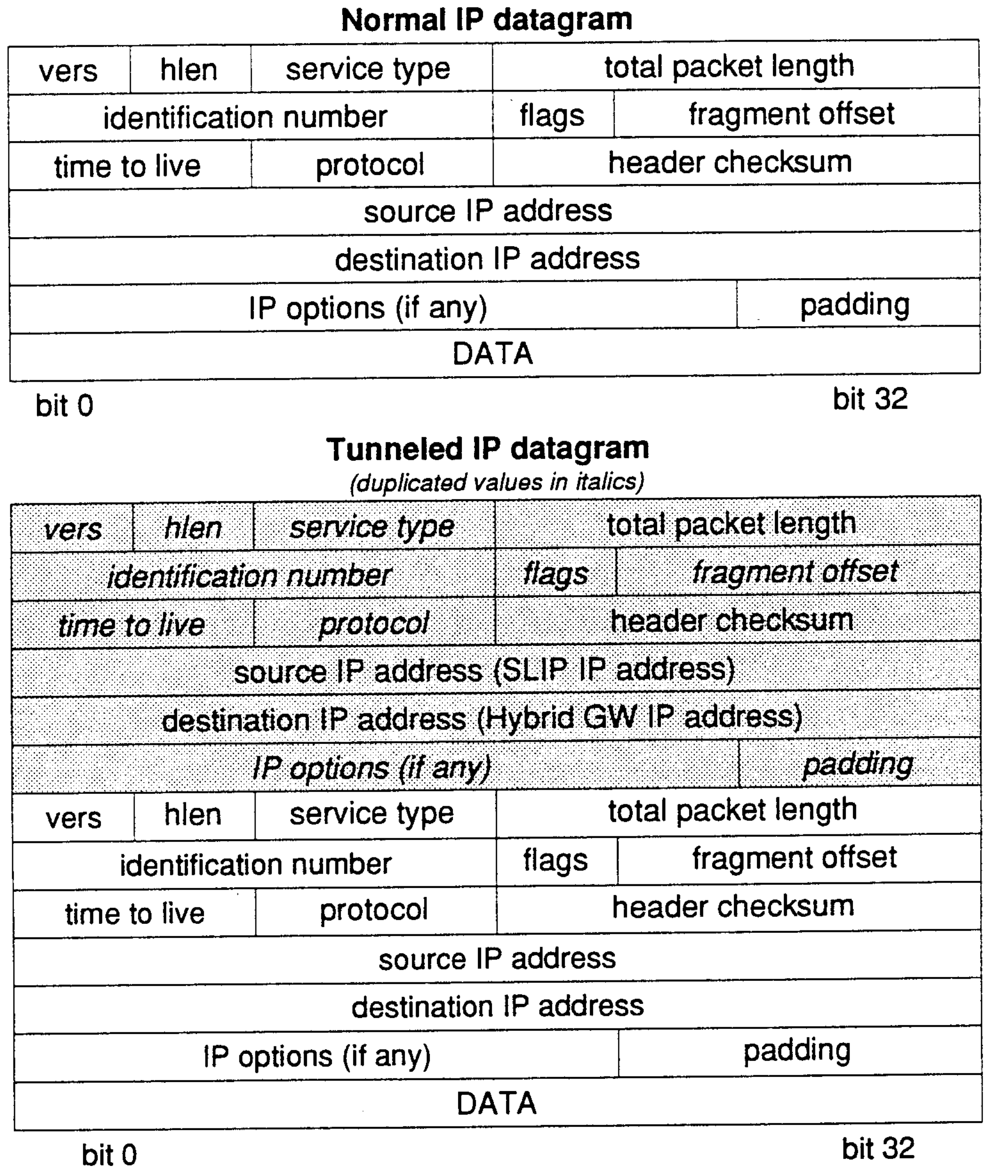
The reason for tunneling is this: the user terminal has two IP addresses associated with it—one for the SLIP interface which is assigned by the SLIP provider, a commercial service the developers have no control over, and the other corresponding to the satellite interface, assigned by HNS and essentially 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.

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.

Fig 6.5 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.



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.

Dialing

SLIP services are dial-up services which require some sort of login procedure for security. If an automatic script is not incorporated in Special SLIP (which is unlikely since we don't know the service or login parameters to initiate the connection) then some mechanism for dumb terminal communication must be provided to allow modem dialing and login control. This is a subtle problem since, at system boot time, the driver grabs the serial port hardware interrupt for sending IP packets and this action needs to be disabled in order to send and receive ASCII data over the modem. This is accomplished using the DOS redirection command discussed above. The command "echo d>HYSLIP$" disables the interrupt service routine in the Special SLIP, allowing another interrupt service routine to handle the login. This command can be entered from a DOS window in Microsoft Windows. Then, the Windows Terminal program can be started using the Windows dial functions and dumb terminal screen to initialize the SLIP session. Once the session has been initialized, the user quits from Terminal and enters the command "echo S>HYSLIP$" to re-enable the isr. This is an admittedly clumsy implementation, however, it is functional and can be cleaned up later.

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.

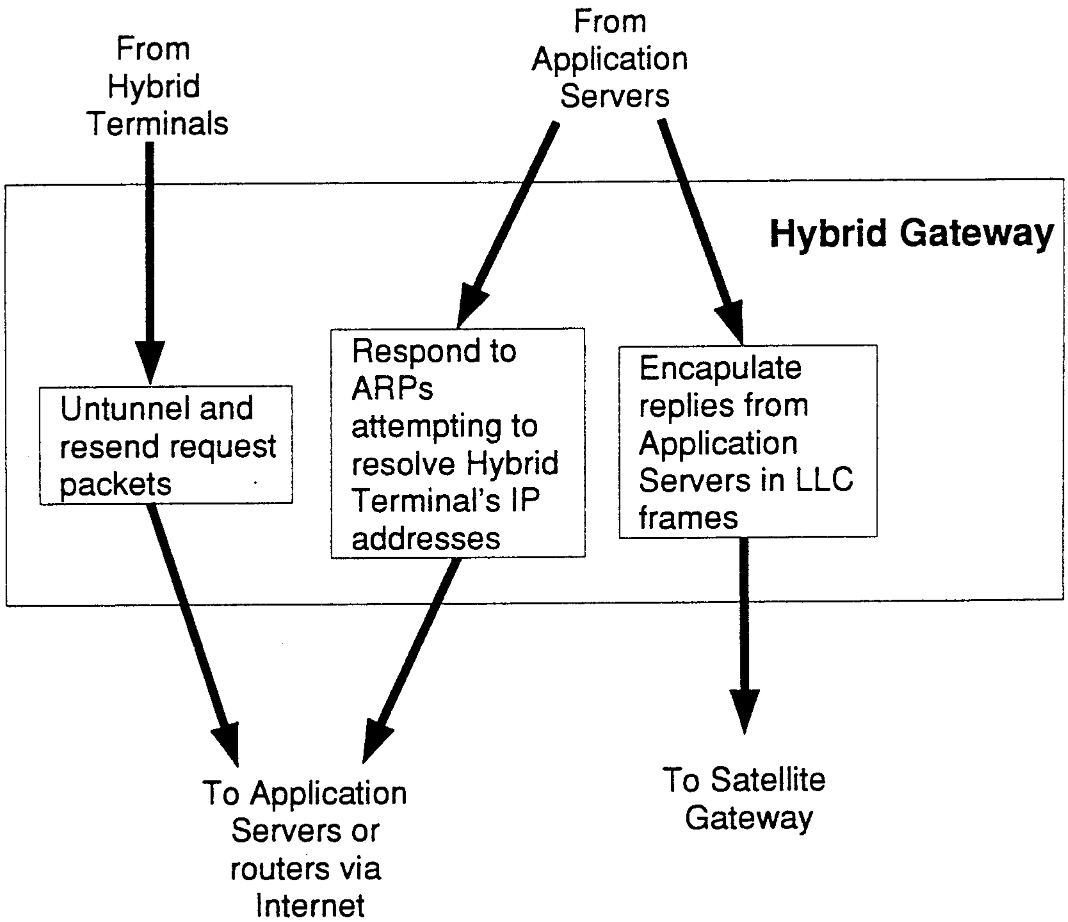
To handle this situation, the driver must implement it's own segmentation procedure. In segmentation, the packet is broken into pieces the size of the SLIP MTU and the header minus one header length is copied onto each piece with an offset value specifying where that particular piece goes in the original packet. Once a packet is segmented, it is not reassembled until it reaches its destination. 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 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.

Fig. 6.6 The Hybrid Gateway accomplishes all the protocol functions required to enable the routing of reply packets to the satellite interface on the User Terminal.



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 SLIP server to the Hybrid Gateway. This can occur if segmentation occurs within the driver as 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

Fig 6.7 The IP packet from the Application Server has two headers appended: the inner one addresses the packet for its final hop from the Satellite Gateway to the User Terminal and the outer one addresses the packet for the Satellite Gateway which is on a token ring network.

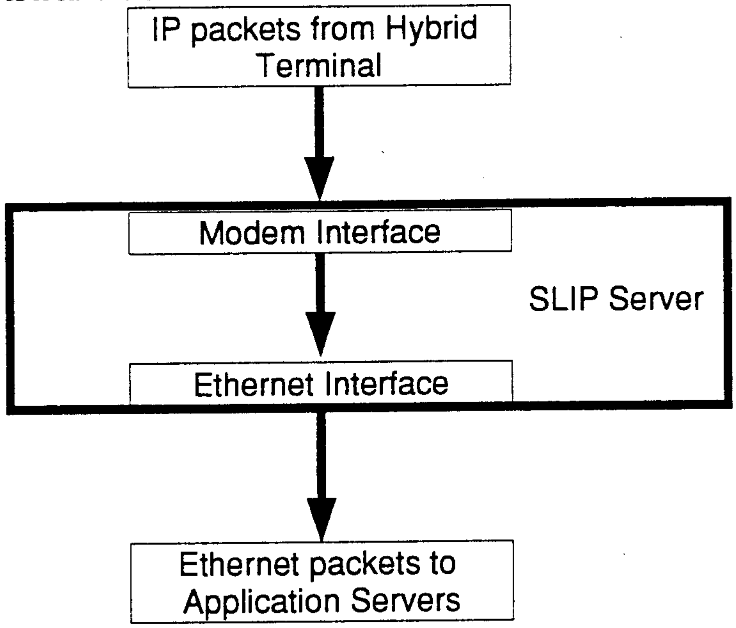
LLC Header	Satellite Header	IP datagram(payload)
Delivers packet to Satellite Gateway	Used to ID correct receiver terminal	Destined for TCP/IP package in User Terminal
Stripped off in Satellite Gateway	*Stripped off in BIC driver in User Terminal*	

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.

Fig 6.8 The SLIP Server is the Internet connection to the modem on the User Terminal and routes all IP packets it receives from the User Terminal in to the Internet.

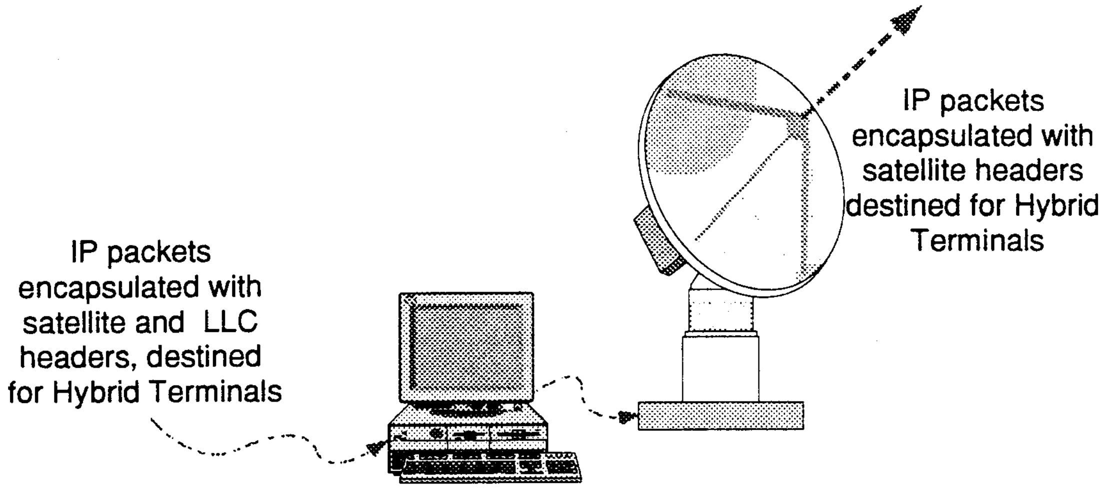


For development, we configured a PC with the Frontier TCP/IP package using two interfaces: a serial port connected to a modem and an ethernet card. The Frontier package acted as a router receiving packets from the SLIP connection and forwarding them to a router available on the ethernet connection with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

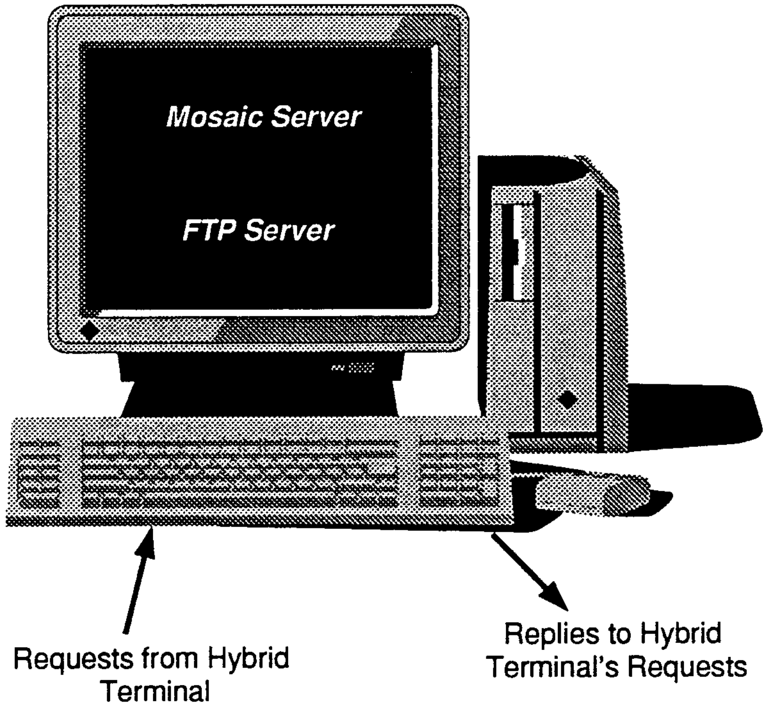
Fig 6.9 The Satellite Gateway sends all packets addressed to it via token ring to the uplink terminal for broadcast.



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.

Fig 6.10 The Application Server can be any host on the Internet running a server version of the client application on the User Terminal.



6.4. The Communication Links

In this section we briefly describe the communication links used in the system. These connections use existing technologies and protocols, are well understood, and undergo no modifications for use in this system.

The SLIP Link

SLIP, or Serial Line IP, is a simple protocol for sending IP packets over a serial line. IP packets are delimited using a special END character (0xC0). If END appears in the data portion of a packet, an ESC char is inserted before a substitute ESC_END character. If

ESC appears in the data portion of the packet, an ESC_ESC character is inserted after it. At the receiving end, incoming data is scanned for END or ESC. If END is received, the protocol knows that a packet has ended. If ESC is received, the next byte is checked for ESC_END or ESC_ESC and the correct byte, either ESC or END is inserted into the packet data, replacing the ESC, ESC_XXX combination.

SLIP contains no provisions for error correction or detection or flow control. It is simple and easy to implement.

The Internet Link

The link between the Internet and the subnet containing the Satellite Gateway and the Hybrid Gateway should be as fast as possible to minimize the delay involved in the tunneling and satellite routing. For the prototype, a fiber optic connection carrying data at T1 rate has been purchased for several months for demonstrating the system. The T1 connects the Gateway network (direcpc.com) with SURAnet, the regional provider of Internet access in the southeastern United States. This is the closest to the Internet backbone that a commercial service can get and so satisfies the maximum bandwidth requirements.

The Satellite Link

The satellite used for the prototype demonstrations was SBS-4, a Hughes geosynchronous satellite. One full Ku band transponder was saturated at 24MHz bandwidth using 20Msymbols BPSK coding. On the downlink, a 0.6 meter receive-only antenna was used to receive HDLC encapsulated LAN packets. Rate 2/3 Viterbi/Reed-Soloman concatenated forward error correction was also used on the satellite frames. The uplink consisted of a 2.4 meter dish at HNS headquarters in Germantown, MD.

7. Guide to Installing a Prototype User Terminal

This section attempts to detail all the specifications necessary to install a prototype User Terminal. First hardware specifications are discussed, then software. The hardware required includes a PC, a DirecPC card, an antenna with a low noise amplifier, and a modem. The software includes the operating system, a TCP/IP package, and the correct drivers.

7.1 Hardware

Because the BIC development is not complete, only certain hardware platforms have assured compatibility with the DirecPC card. For this reason, an IBM PS/2 model 77 EISA bus should be used for the PC. It is important that the machine be equipped with a U16550 UART for the serial port. The development platform had 20M of RAM and this is recommended for operation. It may be possible to run with less, however. Other system parameters aren't critical: VGA monitor and a hard drive with room for Windows and the applications (40Meg is probably enough).

The DirecPC card must be provided by HNS. The card expects an L Band signal from the receive terminal's low noise amplifier (LNB) and any antenna may be used with an appropriate LNB. Hughes has provided a 0.6 meter VSAT for development purposes which accomplishes this.

While the software is configurable for a variety of modem speeds, a modem that is too slow will severely degrade performance and a modem that is too fast may cause problems because the Windows may run too slow to service it. A 9600 baud modem has been tested with the system and is recommended for the current prototype.

7.2 Software

This system has been developed and tested with Microsoft Windows 3.1 running over DOS 5.0. These operating systems are recommended. Windows must be configured with Virtual Memory disabled. This is set in the 386 Enhanced Control Panel document.

The TCP/IP package to use is Frontier's SuperTCP. Install the minimal configuration as in the instructions and include the NDIS software at installation. During the installation a dialog box will query as to which NDIS driver to use. Instead of choosing one of the included drivers, choose "Other..." and install the hyslip.sys driver from floppy disk or wherever it is stored. SuperTCP will create a protocol.ini file in the SuperTCP directory. When installing the driver be sure that the line "TUNNELLING" is included in the protocol.ini file under the portion headed [HYSLIP]. In the driver configuration window within the SuperTCP install or configure application there are dialog boxes to enter the serial COM port, the SLIP IP address, the IP address of Hybrid Gateway, the SLIP MTU, and the communications parameters of the SLIP connections, i.e. 8, N, 1 means 8 data bits, no stop bits, and 1 stop bit. Configure these parameters to match the installed setup. Be sure to look at the protocol.ini file and check that the IP address are surrounded with quotes. The parameters used in the demo were:

SLIP IP address: 128.8.11.237

SLIP MTU: 256

Address of Hybrid Gateway: 198.77.116.42

Com port: 2

Com parameters: 8, N, 1

The BIC driver, WINBIC.DLL, and the executable program that initializes it, HYCTL.EXE, should be loaded onto the hard drive. HYCTL.EXE should be placed in a window and is run from Windows. Also, a Microsoft freeware program, cr3.386, should be placed in the same directory as the other 2 files and requires configuration to the SYSTEM.INI file by adding the line: device=c:\..\cr3.386.

To start the driver, double click on the icon and execute the following three commands:

4 200 2000 1318(This loads the driver with 200 buffers, each one 2048 bytes long with a receive frequency of 1318)

6 45 (This starts the driver which traps all satellite packets with 45 as the lease significant byte of their destination IP address)

12 (This passes all received packets to the Special SLIP)

The window can then be minimized.

To run a session, the following sequence of actions must take place in the correct order.

1. Do a hard reboot of the computer. This resets the BIC which is not reset during Ctl-Alt-Del.

2. Start Windows.

3. Open a DOS window and type "echo d>hyslip$" to enable dialing mode in the SLIP driver. Close the DOS window.

4. Open Terminal. Configure Terminal to talk to the same COM port as the modem and SuperTCP are configured for.

5. Use Terminal to dial up the SLIP server and start a connection by entering the correct commands into the Terminal window. Once the session has been established, quit from Terminal.

6. Open the DOS window again and type "echo s>hyslip$" to re-enable SLIP and disable the dialing mode in the SLIP driver.

7. Open HYCTL and start the BIC driver as described above. Minimize the BIC driver window.

8. Open a SuperTCP application such as Ping. Try pinging a known host in the Internet. If this works, the system is operational.

8. System Costs

The cost per Megabyte is the cost involved in delivering one Megabyte of data. The satellite transponder cost is the largest reoccurring cost of the system. If the transponder on the satellite was 100% utilized all the time it would be possible to get an idea of the costs involved in operating the system by figuring the cost per Megabyte. This cost can be compared to the cost to obtain the same Megabyte from a terrestrial service. Hopefully, a hybrid Internet service price would be based on the system cost per Megabyte.

Hughes charges $700 per hour for the transponder used in the prototype development. The channel can provide 23 Mb/s if every user has a 0.75 meter dish or 11.79 Mb/s if every user has a 0.60 meter dish or 17.7 Mb/s if 60% of the users have 0.60 meter dishes and 40% of the users have 0.75 meter dishes. The system cost per Megabyte is summarized in the following table.

Table 8.1 The transponder cost to deliver one Megabyte of data as compared to a terrestrial T1 line.

Transponder Cost per Megabyte			
Percentage 0.60 meter antenna	Percentage 0.75 meter antenna	Mbps through transponder	$/Megabyte
100%		11.79	0.132
60%	40%	17.7	0.089
	100%	23	0.067
Terrestrial T1@ $874/month =$1.21/hour		1.5	0.0016

When comparing this system to a T1 line one must remember that there is a difference in the design that allows this service to be used by many more people over a larger geographic area than a single T1 line which offers a single, high-bandwidth, point-to-point connection. It can be seen from the table that this service is within an order of magnitude of being competitive with terrestrial services. This is hardly a definitive statement for a company considering offering a service but this statistic is not the only measure of the competitiveness of the service.

Since the transponder cost is the largest cost in the system, it can be examined to give some idea of the per user cost for the system. This can be used to evaluate potential service pricing. Assume that the system is used at 60% of capacity by users with 0.75 meter dishes from 7am Eastern Standard Time until 2am Pacific Standard Time, or 22 hours per day. Further assume that each user is continually receiving an average of 100kbits/sec. Using 0.75 meter dishes, the maximum deliverable bandwidth by the transponder is 23Mbits/sec. Therefore, the maximum number of simultaneous users supported is $23 \times 10^6 / 100 \times 10^3 =$ 230 users. At 60% capacity this is 138 users. Over 23 hours this is 3174 user-hours per day. The transponder costs $700/hour and is dedicated to this service 24 hours/day yielding $16,800/day. By distributing all the cost over all the users this yields $16,800/3174 user-hours = $5.29/user-hour. This rate is somewhat higher then the rates charged by commercial on-line services such as America Online or Prodigy. However, those services do not support data rates over 9600 baud.

9. Opportunities for Future Development & Research

This prototype is meant as a proof-of-concept to show that it is possible to do the asymmetric routing over the Internet without requiring modifications to the hosts. Throughputs of up to 140kbps have been demonstrated using the system as described above. However, the satellite link and the Internet operate at data rates significantly above that. It appears that the bottlenecks are occurring in two places: window size limitations between the applications server and the User Terminal and delays incurred by transmissions over the modem link.

9.1. Spoofing TCP

Data communications over satellite has run into the problem of window sizes before. The difficulty lies in the fact the TCP places a limit, negotiated between the end-points, on the number of packets that can be allowed to be outstanding during transmission, i.e. the number that can be sent before an acknowledgment (ACK) is received. The high bandwidth and long delays incurred in sending packets to an orbiting satellite and back mean that at any given time a lot of packets are "in the pipe" between transmitter and receiver.

The transmitter can send a lot of packets out, up to the window size, and will then stop transmitting and wait to receive an ACK before transmitting any more data. While the purpose of this protocol feature is to limit the number of packets that may need to be re-transmitted in the case of packet loss, the effect in satellite communications is similar to turning a faucet on full blast for a moment, trying to fill a sink, then shutting it off and letting the sink fill the rest of the way by drips. This problem can be alleviated by spoofing the TCP protocol.

Spoofing TCP is the term used to describe fooling the protocol so that more packets are sent and the window limits do not interfere with the maximum bandwidth. In spoofing, a terminal at the uplink side of the satellite connection sends out acknowledgments to the transmitter so that it will continue to send packets. This scheme relies on a reliable satellite link because it disables part of the error detection protocol, i.e. if the Hybrid Gateway sends an ACK for a received packet which is then lost between the Hybrid Gateway and the User Terminal, it will take longer to recover from the error than without the spoofing. So, a tradeoff must be made to optimize when and how much to spoof.

9.2. Dropping ACKs in the hybrid host

One reason for the bottleneck over the modem connection is that SuperTCP wants to send an ACK for every packet it receives. This is not required by the TCP protocol but is an implementation specific issue for Frontier's package. This means that there are a lot of packets competing for the slow link to the SLIP server. For example, an ACK is about 60 bytes long, with 40 bytes for tunneling, 100 bytes. Over a serial link, one byte is about 10 bits long, meaning that each ACK is about 1000 bits. At 9600 baud, only 9 packets can be

ACK'ed per second, no matter what the receive rate is. If each packet is the maximum length, about 1500 bytes, then the absolute maximum receivable data rate is 108kbps.

The TCP ACK scheme is cumulative. This means that when a transmitter receives an ACK stating that the receiver has received the packet with sequence number N, then the receiver has received all the packet ,with sequence numbers up to N, as well. Therefore, there is no reason why every packet needs to be ACK'ed. One way to limit the number of ACKs crossing the modem link is to drop some of them while they are in the driver. The following ones will also acknowledge the ones that were dropped using the cumulative acknowledgment. This is not trivial because dropping the last ACK in a transmission will force retransmission of part of the file and will add more delay to the transaction. However, it is worth exploring.

9.3. Initiating Connections from the Internet

A significant qualitative improvement to this system would be to implement the ability to initiate connections from the Internet. This implies that a user on the Internet could open an FTP session into the user terminal. Upon receipt of a session open request, the Special SLIP driver would dial up the SLIP Server and login automatically. Then the session could proceed. The response time of the session would be pretty poor for FTP because the file transfer would be over the slow modem connection, however, it would allow the User Terminal to have another feature of a typical Internet host allowing for more integration of

Internet applications. Also, Email delivery using SNMP could be allowed without the user being present and without leaving the modem connection open continuously (a significant concern in single phone line households).

Development would focus on scripts which would automate login and would disconnect the modem connection once the TCP connection was down.

9.4 User Capacity of the System

An issue for practical research that may serverly affect the commercial potential of this system as a commercial service is the number of users that can reasonably be supported on a single satellite transponder. While there is a bandwidth bottleneck in the modem connection, there is a resource bottleneck in the satellite link. Only a single user may have the satellite link at a time. The traffic destined for every user on the system must pass over that link. As the number of users grows, so will the amount of simultaneous demands for transmission on the link. Eventually, queuing will have to be implemented at the uplink so that simultaneous arrivals at the uplink can be transmitted. With queueing comes delay and the dynamics of the delay can be studied.

Some questions that should be studied are how much traffic can the system accommodate and have "reasonable" delays? what are reasonable delays? and, most importantly, how many users can the system accommodate within these reasonable delays? The goal of the system is to reduce the delay a customer experiences when he requests a file. If there is too much delay in the satellite link, there is no net benefit to the customer and the system does not fulfill it's goal. If, on the other hand, the number of customers is kept very small to reduce the delay, then the per customer costs will be high and the system will not be affordable, also not fulfilling the system goals.

It will be difficult to do qualitative analysis resulting in the number of customers that can be supported since the nature of traffic generated by customers is difficult to characterize. A possible strategy could be to characterize the traffic capabilities of the link with its queues in an analytic fashion and follow-up with some empirical studies or simulations focussing on the traffic generation of Internet LAN users.

10. 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 120kbit/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.

11. 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.

Appendix A

This file was received by email from Win Treese.

The Internet Index
[Inspired by "Harper's Index"*]
Compiled by Win Treese (treese@crl.dec.com), 7/8/93
Revised: 1/7/94

Annual rate of growth for Gopher traffic: 997%
Tony Rutkowski, "The Internet Is It's Own Revolution", Internet Society
News, 2(3):2, Autumn, 1993.

Annual rate of growth for WWW traffic: 341,634%
#Ibid.

Average time between new networks connecting to the Internet: 10 minutes
#Ibid.

Number of newspaper and magazine articles about the Internet during the
first nine months of 1993: over 2300
Howard L. Funk, "Internet Media Exposure", Internet Society
News, 2(3):28, Autumn, 1993.

Number of on-line coffeehouses in San Francisco: 18
"San Francisco Cafes on the Internet", Internet Society
News, 2(3):14, Autumn, 1993, citing Fortune Magazine (with no date).

Cost for four minutes of Internet time at those coffeehouses: $0.25
#Ibid.

Date of first Internet mail message sent by a US President: 2 March 1993
(Sent by Bill Clinton, President of the United States)
"Heads of State on the Internet", Internet Society News, 2(3):13,
Autumn, 1993.

Date on which first Stephen King short story published via the Internet before
print publication: 19 Sept 1993
#Available through Editorial, Inc./Online BookStore (OBS) (obs@tic.com)
#or file://tic.com/obs

Number of mail messages carried by IBM's Internet gateways
in January, 1993: about 340,000
#A. M. Rutkowski, "Internet Metrics", Internet Society News, 2(1):33-38.

Advertised network numbers in October, 1993: 16,533
Advertised network numbers in October, 1992: 7,505
#Data from Merit. More information available at
#file://nic.merit.edu

Date after which more than half the registered networks were commercial: August, 1991

Number of Internet hosts in Norway, per 1000 population: 5
Number of Internet hosts in United States, per 1000 population: 4
Number of Internet hosts in October, 1993: 2,056,000
#A. M. Rutkowski, "Internet Metrics", Internet Society News, 2(1):33-38.

Round-trip time from MIT to mcmvax.mcmurdo.gov in McMurdo, Antarctica:
640 milliseconds
Number of hops: 18
#Win Treese, suggested by a note in the Internet Society News, 2(1):13,
#Spring, 1993.

Number of USENET articles posted in two weeks during December, 1993: 605,000
Number of megabytes posted: 1450
Number of users posting: 130,000
Number of sites represented: 42,000
#Dan Lawrence (from UUNET data), personal communication.

Number of Silicon Valley real estate agencies advertising with
Internet mail addresses: 1
#Brian Reid, personal communication.
#Also noted in the Internet Society News, 2(1):13, citing an ad
#in the Palo Alto Weekly.

Terabytes carried by the NSFnet backbone in February, 1993: 5
#From NSFnet statistics on file://nic.merit.edu/statistics

Number of countries reachable by electronic mail: 137 (approx.)
Number of countries not reachable by electronic mail: 99 (approx.)
Number of countries on the Internet: 60
#As of 8/1/93. From "International Connectivity" by Larry Landweber.
Internet Society News, 2(3):35, Autumn, 1993. This information is
maintained at file://ftp.cs.wisc.edu/connectivity_table.

Amount of time it takes for Supreme Court decisions to become
available on the Internet: less than one day.
#The decisions are archived on file://ftp.cwru.edu/hermes
and available on the World-Wide Web at
http://www.law.cornell.edu/supct/supct.table.html

Date of first National Public Radio program broadcast simultaneously
on the Internet: 21 May 1993

Percent of Boardwatch Top 100 BBS systems with Internet Connectivity: 21

Number of people on the Internet who know you're a dog: 0
Cartoon in the "New Yorker", July 5, 1993.

*"Harper's Index" is a registered trademark of Harper's Magazine Foundation.

Send updates or interesting statistics to treese@crl.dec.com.