Adding Packet Radio to the Ultrix Kernel

Clifford Neuman
Wayne Yamamoto

Department of Computer Science, FR-35
University of Washington
Seattle, WA 98195

ABSTRACT

This paper describes the results of a project in which the standard Amateur Packet Radio network link layer protocol, AX.25 (a modified version of X.25), was added to the Ultrix kernel. By implementing AX.25 under Ultrix, and by taking advantage of the IP implementations that already exist for PCs, it is possible for packet radio users with PCs to access IP-based services running on our server and on the Internet. A MicroVAX is being used as an IP gateway for an Amateur Packet Radio network that stretches from Seattle to Tacoma.

1. Introduction

Packet Radio is an increasingly active area of experimentation among amateur radio operators. Stations consist of a radio transceiver connected to a terminal or a computer by means of a device known as a Terminal Node Controller (TNC). The TNC is essentially a modem. It "packetizes" data in a manner conforming to the AX.25 link layer protocol, provides a command interpreter, and has a primitive network layer protocol for use with terminals unable to support this layer on their own. Users with computers rather than terminals have the option of disabling the TNC's network layer protocol and providing their own.

Amateur Packet Radio has evolved somewhat chaotically. Initially, most packet radio stations consisted of terminals instead of computers. Once users had established communication with one another, they simply typed streams of data at each other. The TNC was functionally equivalent to a telephone modem.

One early development arose because users wanted to communicate with stations that couldn't be contacted directly. Relay stations were set up in strategic locations so that messages could be received and passed along to their destination. These relays are known as digipeaters. The standard amateur packet radio link layer protocol[1] allows the specification of up to eight digipeaters through which a packet is to pass. This type of routing is known as source routing.

Another development was that some users connected their TNCs to computers on which they ran packet bulletin board software. This allowed others to access their computer in a manner similar to the way one accesses a BBS by phone using a modem. Users with terminals were able to leave messages and read messages. Other users who connected their TNCs to computers were able to upload and download files. The BBSs would forward mail to other BBSs for non-local users using packet radio.[2]

This work took place while the first author was supported by NSF Grants DCR-8420945 and CCR-8619663 and the second author was employed by AT&T Bell Laboratories and attending the University of Washington. Computing equipment was provided by Digital Equipment Corporation's External Research Program.

[1] This is actually network layer functionality, but because packets are digipeated on the same frequency, and thus within the same subnet, digipeating is often considered a link layer function instead of a network layer function within the amateur packet radio community. True network layer functionality is provided through other mechanisms, some of which are discussed in this paper.

[2] Usually one or two BBSs in each area would connect to

As the number of users of the amateur packet radio network increased, the demand for better connectivity between different parts of the country increased as well. As a result, network layer protocols became a topic of much discussion. A number of network layer protocols were proposed, and work has proceeded on many of them in parallel. The approach taken by COSI [1] and NET/ROM involved the establishment of networks of servers within which routing was taken care of automatically. With NET/ROM, users would connect to a node on the network. They would then connect to the NET/ROM node nearest their destination. Finally, they would connect to their destination. A similar approach is used for COSI, the main difference being that instead of actually connecting to a COSI node, one connects through it as if it were a digipeater. Users still had to know the name of their local node and the name of the node closest to their destination.

At the same time that development was proceeding with NET/ROM and COSI, work was proceeding on supporting a packet radio implementation of TCP/IP for the IBM PC. This work was being done primarily by Karn [4]. One advantage of TCP/IP over the other approaches is that the user's computer becomes part of the network: one connects to the ultimate destination, rather than connecting to a network node and from there connecting to the destination. Another advantage is that users gain the ability to access IP services which are more varied than services accessible through the other approaches.

Although packet radio implementations of IP exist for several computers, the IP suite of protocols is not yet widely used in Amateur Packet Radio. One reason is that many users are still using terminals instead of PCs. These users can't run IP since they only have access to the limited software that runs in the TNC itself. There are also many users with computers for which there currently is no implementation of TCP/IP. For many of the users that do have computers capable of running IP, a major concern is that they will be isolating themselves from the users that can't run IP.

Another reason that the acceptance of TCP/IP is not greater is that much of the value of the Internet protocols is felt when one is using services that aren't available using other protocols or accessing systems that would not be accessible otherwise. Unfortunately, the existing implementation of TCP/IP for PCs only supports telnet (Remote login), SMTP (mail), and FTP (file transfer). These services are already available to users without using TCP/IP by connecting to a BBS and either reading or leaving mail there, or uploading and downloading files. Further, despite the fact that packet radio users of IP speak the same protocol as other systems on the Internet, and despite the fact that they have a block of addresses assigned to them by the Network Information Center, there did not exist a path by which they could connect to conventional Internet sites.

One of the primary objectives of our project was to provide a gateway between packet radio users (or at least, those that speak IP) and the Internet. This allows those users to access many of the network services that we, as Internet users, are used to. It is hoped that access to such services will stimulate the development of services specifically suited to the amateur packet radio community. The availability of such services will provide additional incentive for further stations to begin using IP. Another goal was to provide a gateway between users speaking other protocols over packet radio, and systems running IP. Such a gateway would allow stations to run IP without isolating themselves from the existing amateur packet radio network.

station in different parts of the country (or the world) in order to forward messages from one packet network to another. In this way, connectivity for electronic mail was achieved on a world wide level.

2. Implementation

We achieved the goals outlined in the preceding section by adding support for packet radio to a system running Ultrix that was already on our department's Ethernet and part of the Internet. The code we used to encapsulate and decapsulate packets on our MicroVAX is based on the existing code for the PC.

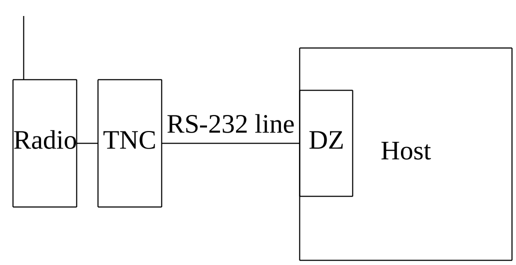

Figure 1. Physical hardware

2.1. The Hardware

As shown in figure 1, the special hardware used by our system to send packets by radio includes a radio and a TNC. The radio corresponds to an Ethernet tranceiver, and the TNC to the Ethernet controller. One difference, though, is that the TNC does not sit on the bus. Instead, one communicates with it though a serial line. Since we did not require the higher software layers of the TNC, we used a stripped down version of the software for it known as the KISS[3] [2] TNC code. This code, which may be downloaded into the TNC, sends and receives data and calculates the necessary checksums. Unlike the normal code that resides in the ROM of the TNC, the KISS TNC code does not worry about the packet format at all.

2.2. The Driver

In adding packet radio support to the Ultrix kernel, a pseudo-device driver for the packet radio controller was implemented. This driver supports the same calls as the drivers for other network devices such as the DEQNA. Since the packet controller does not sit on the bus, communication with it is through a serial line, and hence the driver is a pseudo-driver. Figure 2 shows the function of the hardware and software with respect to the ISO/OSI reference model.

In order to get the kernel to recognize the packet radio interface, we had to create and initialize a structure of the type if_net. The if_net structure contains pointers to the procedures used to initialize the interface, send packets, change parameters, and perform other operations. Kernel procedures to perform each of these operations were created. The most difficult routine to write was one which handled incoming packets from the TNC. When a packet is received by the TNC, the TNC sends the packet as a stream of bytes to the tty line. For each character in the packet, the tty driver calls the packet radio interrupt handler to process the character. Characters are buffered by the interrupt handler until all characters in the packet have been received.

As each character is read by the interrupt handler, some processing of characters is done on the fly. In particular, escaped frame end characters that are embedded in the packet are decoded. When the final frame end is read, meaning that the entire packet has been received, the interrupt handler checks the header of the packet. It verifies that the recipient's amateur radio callsign (which is used as a link address) is either its own, or the broadcast address. It also checks the protocol ID field. If the packet type is IP, the driver then adds the encapsulated IP packet to the queue of incoming IP packets so that it can be dealt with by the existing Ultrix software. This approach to handling incoming packets allows other layer three protocols to be handled in an interesting manner, described later in this paper.

ISO/OSI model	Protocol	Implementation
Application [7] Presentation [6] Session [5]	SMTP Telnet FTP	Existing Ultrix Network Support
Transport[4]	TCP or UDP	
Network[3]	IP	
Link[2]	AX.25	Packet Radio Driver
Physical [1]	Radio	TNC/KISS
		Radio

Figure 2. Comparison to ISO/OSI reference model

2.3. Setup and Testing

Once the packet radio driver was running, our final task was to translate Internet addresses into AX.25 addresses. This is done using the address resolution protocol (ARP) [8] in a manner similar to the way that IP addresses are translated into Ethernet addresses. AX.25 addresses look like amateur radio callsigns followed by a 4 bit system ID. Things are complicated by the fact that some entries may contain additional callsigns for digipeaters. Thus, a different set of ARP routines is needed for packet radio. Karn's IBM-PC code [5] includes an ARP implementation that supports both AX.25 and Ethernet addresses. Because we did not want to modify the code for our system that is used on the Ethernet side of the gateway, this code was not taken. ARP lookup occurs at layer two, and thus, gets called inside either the Ethernet driver, or the AX.25 driver.

[3] The name is from the acronym ''Keep It Simple, Stupid''.

The routing tables at the IP layer determine which driver is called. Since the ARP lookup occurs inside our code, a separate routine that deals specifically with AX.25 addresses can be called.

When the implementation was complete the packet radio interface was enabled at the Internet address of 44.24.0.28.[4] The routing table of another system on our Ethernet was modified so it knew that 44.24.0.28 was the address of a gateway to net 44. After a few rounds of debugging, we were able to telnet from an isolated IBM PC[5] to a system that was on our Ethernet by way of the new gateway. Since then we have used the gateway for file transfer, electronic mail, and remote login in both directions.

2.4. Future Work

In addition to providing a gateway between the packet radio network and the rest of the Internet, we would like our gateway to be able to serve as a gateway between applications running on top of other protocols. Such a gateway would be at the application layer, and specific to remote login and electronic mail. The way AX.25 was implemented in the kernel, such applications do not require kernel support, even though they extend down to layer three of the ISO reference model. Packets that are received from the TNC that are not of type IP can be placed on the input queue for the appropriate tty line. A user program can then read from this line, and maintain the state required to keep track of AX.25 level 3[6] connections. Data can then be passed to a pseudo terminal to support remote login, and to a separate program to support electronic mail.

Work is also proceeding on using another layer three protocol known as NET/ROM to pass IP traffic between gateways. Doing this would allow the use of an existing, and growing, point-to-point backbone in the same way Internet subnets are connected via the ARPANET.

3. Performance

Because the link speed is only 1200 bits per second, the transmission time is the dominant factor in determining throughput and latency. Higher bandwidth links are available, but, at the moment, the hardware is not readily available at a reasonable price.

[4] Net 44 is assigned to Amateur Packet Radio by the Network Information Center.

[5] Connected to only a power outlet and a radio.

[6] AX.25 level 3 is the connection protocol supported inside the TNC.

One performance problem that we noticed is that the gateway slows considerably as traffic on the packet radio subnet climbs. Part of the reason for this is that the present code running inside the TNC passes every packet it receives to the packet radio driver regardless of the destination address of the packet. We are considering changing the TNC code so that it can selectively pass only those packets destined for the broadcast or local AX.25 addresses.

4. Issues

The ability to interconnect amateur packet radio networks and non-radio networks introduces a few problems that have not been completely resolved as of this time. The three main problems are timeouts, routing and access control. In this section, we present these problems and suggest some possible solutions.

4.1. Timeouts

The first problem arises from the difference in the latency for the two networks. Hosts on the Ethernet side expect fast response. If they don't get a response quickly, they time out and retry their transmission. We have found that when the gateway is used to communicate between Ethernet stations and packet radio stations, the system on the Ethernet side initially retransmits packets several times before a response makes it back. This results in wasted bandwidth as packets are needlessly retransmitted. Since these retransmissions are queued at the gateway, they delay other packets. Fortunately, many implementations of TCP dynamically adjust their timeout values. Hence, when the system on the Ethernet side learns the correct timeout value, the frequency of unnecessary packet retransmissions is reduced.

4.2. Internet routing

The second problem arises when we want to allow communication with Internet hosts beyond our Ethernet. Since AMPRnet[7] has been allocated a class 'A' network, most systems will maintain only a single route for it. All packets destined for AMPRnet originating from another internet host, must pass through a single gateway. This is not desirable since a packet destined for 44.24.0.5 should be sent to a West Coast gateway and introduced to the packet radio network there, whereas a

[7] AMPRnet (AMateur Packet Radio NETwork) refers to the subnetwork of the Internet consisting of amateur packet radio hosts, and with Internet addresses of the form 44.*.*.*.

packet destined for 44.56.0.5 should be sent to an East Coast gateway. It is conceivable that something like this could be handled using the Internet Control Message Protocol (ICMP), but at this time, no mechanism is in place.

4.3. Access Control

The third problem is access control. Since operation is on frequencies assigned to the amateur radio service, any communication must be initiated by licensed amateurs. This is, in fact, an instance of a more general problem of trying to control access to a subnet so that hosts on the subnet can use services beyond the subnet, but, at the same time, hosts on the subnet are protected from adversaries beyond the gateway. One way to solve this problem is to maintain a table of authorized addresses on the non-amateur side of the gateway. Associated with each of these addresses is a list of hosts on the amateur side of the gateway with which that host can communicate. Initially the table starts off empty. Whenever a packet is received on the amateur side destined for a non-amateur host, an entry is made in the table, enabling the non-amateur host to send packets in the other direction. After a certain period of time, these entries are removed if packets have not been received from the amateur side of the gateway.

This scheme can be augmented with a few new ICMP messages. One message can force an entry to be removed from the table of authorized non-amateur systems. This allows the amateur radio operator that initiated the link to exercise his control operator function to cut off the link if he detects inappropriate use. Another message would allow one to add an authorized non-amateur host to the tables with an appropriately chosen time-to-live. Both these message are allowed to come from either side of the gateway, but if they come from the non-amateur side, they must include a call sign and a password for an authorized control operator for the gateway.

5. Discussion

Packet radio is one of several options available when setting up computer networks. It is not the ideal solution in all situations. There are a number of problems that make non-radio communications more practical. Among the disadvantages are that radio communication is susceptible to eavesdropping, jamming, and impersonation[8]. Another disadvantage is that the radio spectrum is limited. Where wire or fiber-optic communications is an acceptable alternative it should be used.

Despite the disadvantages, there are situations where packet radio would be the technique of choice. Mobile stations cannot be physically connected to one another. Packet radio is also useful for emergency field communications where one doesn't have the time to string wires. Another reason that packet radio is useful for emergency communications is that in a large scale emergency, such as an earthquake, land based communications will often be disrupted.

In a number of years, wide scale computer networking will be available to the public through services such as the Integrated Services Digital Network (ISDN). Until that time, packet radio allows users, over a large geographic area, to set up networks connecting personal computers and larger computer systems. Use of a system running Ultrix as a gateway and server on a packet radio network is desirable since it allows users of PCs to access many of the services that one is used to having available in the Internet community. Telnet, FTP, and SMTP have all been successfully used across the gateway. We would like to support additional services, some of which are beyond those provided on the Internet. The Ultrix operating system can serve as a platform upon which these services will be built.

One useful service for the amateur radio community might be a distributed callbook. Many amateur radio operators purchase a book each year that contains the callsigns, names and addresses of every other amateur radio operator in the world. These books are organized by callsign, and are used to mail QSL cards.[9] With a distributed callbook server, data for a particular country, or part of a country, could be maintained on a system local to that area. Given a call sign, an application running on a PC could determine what area the call sign is from, and then send off a query to the appropriate server.

Allowing users to add information to their own entries (such as geographic coordinates) would allow many possibilities. It would be possible for someone establishing communication with anyone in the world to type in the station's call sign, and have their antennas automatically rotated to the correct bearing. Or perhaps, as a contact is made, one's computer can print out a mailing label for the QSL card. Someone even suggested having a bitmap of the QSL card sent

[8] These can be solved using encryption and spread spectrum, but this increases the cost and might make licensing more difficult.

[9] A QSL card is a card confirming that a contact took place. Amateur radio operators often collect these.

to the other station through the packet radio network itself.

Distributed computation presents further possibilities. By giving the general public access to computer networks, the number of processors accessible on a network greatly increases. At the same time, the power of the processors will typically be less than those currently part of the Internet. This provides added incentive for users to devise ways to utilize the computing resources of systems that aren't being used to their capacity.

In designing and implementing distributed services, it is important to note that there is a difference between the environment inside a university or corporation with central control, and the environment of a network of PCs, where little central control exists. A network such as the one described in this paper can serve as an interesting testbed for applications that must run in such a decentralized environment.

In order for packet radio to reach its full potential for connecting otherwise isolated computers, it would be helpful for a few frequencies to be allocated outside of the amateur radio service. When using packet radio on amateur frequencies, there are lots of restrictions on the type of data that can be passed that makes productive use less practical. If there were a few non-amateur frequencies set aside for packet radio, and if the requirements and restrictions for operation on these frequencies were less stringent, one might see even more done with packet radio.

6. Conclusions

The Ultrix operating system provides a nice base upon which network services can be provided for the amateur packet radio community. At the same time, such a system can serve as a central node in the interconnection of local area networks running IP, and even those that don't run IP. By linking packet radio networks with more established networks, additional services become available. Such services are available in the Seattle area. These services are necessary if we are to interest people in running TCP/IP. Further, interconnection with non-IP packet radio users is necessary if we are to interest users who would like to try IP, but still want to maintain connectivity with those using other protocols.

7. Acknowledgments

A number of people were helpful in getting our implementation running and in discussing some of the ideas presented in this paper. Among them are Bob Albrightson (N7AKR), Mike Chepponis (K3MC), Bob Donnell (KD7NM), Dennis Goodwin (KB7DZ), Robert Henry (WA6BEV), Ron Hoffmann (WA2EYC), Ed Lazowska (KG7K), and Steve Ward (W1GOH).

8. Bibliography

1. Beattie, J. Gordon Jr., and Moulton, Thomas. ''OSI: A Plan Comes Together''. Sixth ARRL Computer Networking Conference, Redondo Beach, CA, August 1987.

2. Chepponis, Mike, and Karn, Phil. ''The KISS TNC: A Simple Host-to-TNC Communications Protocol''. Sixth ARRL Computer Networking Conference, Redondo Beach, CA, August 1987.

3. Fox, Terry L. ''AX.25 Amateur Packet-Radio Link-Layer Protocol. Version 2.0.'' American Radio Relay league, October 1984.

4. Karn, Phil. ''TCP/IP, A Proposal for Amateur Packet Radio Levels 3 and 4''. Fourth ARRL Computer Networking Conference, San Francisco, March 1985.

5. Karn, Phil. ''The KA9Q Internet (TCP/IP) Package: A Progress Report''. Sixth ARRL Computer Networking Conference, Redondo Beach, CA, August 1987.

6. Leffler, S., Joy, W., Fabry, R., and Karels, M. ''Networking Implementation Notes 4.3 BSD Edition''. Computer Systems Research Group, University of California, Berkeley, June 1986.

7. Neuman, Clifford. ''Packet Radio and IP for the Unix Operating System''. Sixth ARRL Computer Networking Conference, Redondo Beach, CA, August 1987.

8. Plummer, David C. ''An Ethernet Address Resolution Protocol''. Network Information Center, RFC826. September 1982.