1986

The Cypress Network

Douglas E. Comer
Purdue University, comer@cs.purdue.edu

Thomas Narten

Rajendra Yuavatkar

Report Number:

86-653

THE CYPRESS NETWORK: A LOW-COST INTERNET CONNECTION TECHNOLOGY

Douglas Comer
Thomas Narten
Rajendra Yavatkar

CSD TR-653
April 1987

The Cypress Network: A Low-Cost Internet Connection Technology

Douglas Comer
Thomas Narten
Rajendra Yavatkar

TR 653
April 1987

Computer Science Department
Purdue University
West Lafayette, IN 47907

Abstract

An internet consists of multiple interconnected data communication networks that function like a single, large packet switching network. Standard internet communication protocols hide details of the underlying network architectures and technology from users of the internet, and allow computers to communicate across the internet without knowing the structure of the underlying interconnections. This paper restricts attention to the Internet developed by the Defense Advanced Research Projects Agency (DARPA), and actively supported by the National Science Foundation (NSF) as part of NSFnet, the National Aeronautics and Space Agency (NASA), and other groups in government, academia, and industry.

The Cypress project at Purdue University has been exploring ways to provide low-cost Internet connections. This paper reports results of the project, describes a successful prototype network that already supplies production quality service to several sites, presents performance measurements, and outlines plans for expansion. The key ideas behind Cypress include consolidation of functionality, use of off-the-shelf hardware and software, protocols optimized for best-effort delivery of Internet datagrams, and an incrementally expandable hub-site technology.

We gratefully acknowledge support for the Cypress project from AT&T, CSNET, Digital Equipment Corporation, National Science Foundation (ASC-8518369), and Purdue University. Participants in the experimental prototype have purchased some of the implet hardware and paid some of the leased line costs.

1. Introduction

Packet-switched data communication networks allow communicating computers to transfer files of data and programs, to exchange electronic mail, to support remote interactive login, and to participate in remote job submission. No single network technology can fill all data communication needs, however, because no one technology spans the full ranges of functionality, performance, and cost. To permit computers to communicate independent of the network hardware they use, researchers have developed an abstraction called an *internet* [1]. An *internet* consists of multiple interconnected packet switching networks that function like a single, large packet-switching network as Figure 1 shows.

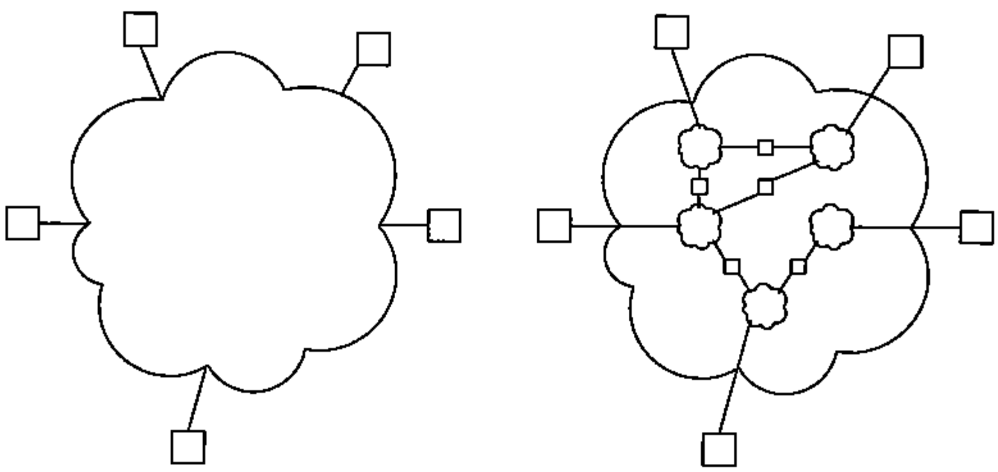

Figure 1. (a) The conceptual view of an internet in which all host computers connect to a single, virtual network, and (b) the reality in which each host connects to a physical network and gateways interconnect networks. The illusion of a single network is achieved by using standard communication protocols and having computers called *gateways* forward packets among the networks to which they connect.

Standard internet communication protocols hide details of the underlying network architectures and technology from users of the internet, and allow computers to communicate anywhere within the internet without knowing the structure of the underlying interconnections.

One of the largest and most popular internets now in use was built by the Defense Advanced Research Projects Agency (DARPA) in the early 1980s [2] and will be referred to as the *Internet* in the remainder of this paper. Computers attached to the Internet can communicate because, in addition to having gateway connections among their various physical networks, they all use standard communication protocols. In particular, they all use the Internet Protocol (IP), for packet delivery, and the Transmission Control Protocol (TCP) for reliable data-stream transport. IP defines the format of packets passing through the Internet, including the details of source and destination addresses, packet size, and the order of fields within the packet. TCP defines how computers cooperate to be sure that data has passed correctly and reliably across the Internet, including the details of error detection and retransmission, buffering, and demultiplexing.

Although the DARPA Internet protocol suite includes many other protocols, IP and TCP are so fundamental that the entire set is commonly referred to as *TCP/IP* [3, 4]. In addition, the Internet encompasses such diverse network technologies as a leased-line, packet-switched network [5], local area networks like an Ethernet [6], a packet radio net [7], and two satellite-based networks [8].

Many colleges and universities, Computer Science Network (CSNET), government and commercial laboratories, and agencies like the National Science Foundation (NSF) and the National Aeronautics and Space Administration (NASA) actively support and use the Internet.

Cross-country networks like the ARPANET [5] and NSFnet [9] form the major arteries of the Internet. In addition, NSF has funded several groups to form regional nets that connect to the main arteries. However, the regional networks cover only a small percentage of potential Internet sites, and high-speed connections are often too expensive for the remaining sites, which are usually small. Thus, both NSFnet and CSNET seek ways to provide Internet service to users who cannot afford high-cost connections. A capillary system is clearly needed.

The Cypress project at Purdue University has been exploring low-cost Internet access. We have developed a technology that delivers all the functionality of IP using slow, inexpensive leased lines. Our protocols, which are optimized for best-effort delivery of IP packets, make efficient use of the line bandwidth. Our technology makes it possible to connect one or more end-user sites to most of the existing Internet sites, and even allows sites to chain themselves together in a vine.

The remainder of this paper discusses Cypress technology, and explains the service it provides. It reviews the successful prototype, and outlines plans for future expansion. The next section describes the basic architecture and gives technical details. Later sections describe the prototype network and draw conclusions from our experiences.

2. Cypress Technology

Project Goals and Consequences

The main goal of Cypress is to explore ways to provide Internet connections at a fraction of the cost of conventional connections. We have insisted from the beginning that connecting to the Internet means having an IP-based network connection, and not just having some of the services that use IP (e.g., electronic mail). The reason for insisting on IP is that it provides the foundation on which many applications rest, and makes it possible for users to create their own applications that use the Internet. Limiting a site to only a few high-level services prohibits that site from participating fully in the Internet community, and makes it a "second class citizen" of the Internet.

One subtle consequence of our decision to support IP is that we need a technology that supplies connection to the Internet at any time. In particular, whenever a program wishes to communicate, it manufactures and transmits IP packets. Furthermore, processes that supply services (called *servers*) must accept incoming packets from other sites asynchronously. Although previous experience using X.25 to send IP packets shows that it is possible to make connections on demand [10], current tariffs make public data networks unattractive compared to dedicated leased lines.

A secondary goal of Cypress is to develop a technology, and not merely a network. Although the current prototype described in this paper consists of a single, fully-connected network, our goal is to develop a variety of techniques that will make it possible to establish multiple Cypress networks, each connecting to the Internet at a different point. Thus, organizations like NSF or CSNET will be able to provide service to their member sites by establishing Cypress networks centered at existing ARPANET or NSFnet sites.

Cypress Basics

A Cypress network consists of a set of packet switches, called *implets*[1], connected together by dedicated, leased, data circuits. Implets can exchange packets with adjacent implets by sending them over the interconnecting leased lines. Each implet functions as a store-and-forward packet switch, accepting incoming packets and routing them on toward their destination, based on a destination address found in each packet. Like most packet switches, implets enqueue packets for transmission while waiting for idle time on one of the leased lines.

One implet resides at each subscriber's site, where it attaches to the user's local area network. One of the most important ideas in Cypress is that implets communicate with the subscriber's machines over a local area network like an Ethernet, making the system loosely coupled. The subscriber's machines think of the implet as their pathway to the rest of the world, and route outgoing packets to it over the local net. In technical terms, the subscriber machines use the implet like an IP gateway that connects to the Internet. The implet accepts IP packets from subscriber's machines and routes them over the leased lines toward their final destination. It accepts packets arriving over the leased lines and routes those destined for the subscriber's machine over the local area network.

One implet on a Cypress network must connect to a main artery of the Internet (e.g., NSFnet or ARPANET). The implet hardware at such a site need not differ from ordinary implet hardware if one uses a local area network to interconnect the implet with some machine that reaches the Internet. Usually, major Internet sites have a local area network that permits such a connection.

Design Decisions and Technical Details

Speed. The desires for high functionality and low-cost guided most design decisions. We realized early that low cost means slow speed. The difference in cost, for example, between leasing a 56 kbps line and a 9.6 kbps line across the country is close to an order of magnitude. Furthermore, modems and computer interfaces for 56 kbps lines are substantially more expensive than those for 9.6 kbps. Thus, of the two most commonly available speeds, we chose 9.6 kbps. In addition, because most computers supply 9.6 kbps asynchronous ports for terminal connections, we chose to use the asynchronous interfaces and use inexpensive external hardware to convert into the synchronous interface used by modems. Recently, however, expansion of the prototype has forced us to add 56 kbps capability, making it possible to have lines that accommodate higher volumes of traffic.

Consolidation. Cypress also achieves low cost through consolidation of functionality. The Cypress implet provides the primary example: it is a multi-function packet switch that consolidates link-level packet switching, IP-level gateway routing, and high-level network monitoring and control into a single machine. At the lowest level, the implet switches packets among the Cypress leased lines that lead to other implets. At the second level, each implet functions as an IP gateway, routing IP packets between the local area network at a subscriber's site and the rest of the Internet. Indeed, an implet performs all of the tasks a gateway does: it recognizes and routes packets to any Internet address; it advertises routes to networks at the subscriber's site; and it propagates routing information to *core* gateways in the Internet, allowing arbitrary Internet sites to reach the subscriber's machines.

The advantages of consolidating IP-level gateway routing and link-level packet switching becomes most evident when considering a subscriber's site. Instead of having two computers, one for packet switching and one for IP gateway routing, the subscriber needs only one. Instead

[1] The term *implet* comes from the ARPANET packet switch, originally called an *Interface Message Processor* or *IMP*.

of having a special-purpose interface to connect the packet switch and gateway, the connection is made in software inside the implet. Furthermore, using a standard local area network to keep the implet loosely coupled to the subscriber's machines allows arbitrary machines to communicate with the Internet, and makes it possible to add new machines without changing the implet.

Hardware and Software. Keeping costs low implies using standard, available technology wherever possible. In particular, we decided to restrict ourselves to conventional computer systems and communication interfaces. While it is tempting to imagine that an implet could be built from scratch for less than the cost of a commercially available processor, the economy of scale obtained with commercial products makes them less expensive. Thus, we chose to use commercially available computers for implets, and adopted conventional communication interfaces.

Of more importance, we chose to begin with an existing, supported operating system instead of starting from scratch with implet software. The major advantage of modifying an existing operating system is that it provides powerful abstractions like processes and a framework for building device and network interfaces. Another advantage is that by using an existing system that understands Internet protocols, we have avoided much of the time and expense involved in implementing and debugging standard protocol software.

Protocol Efficiency. From the start, Cypress was designed to operate over low bandwidth connections, and to not depend on large amounts of processing power. The latter decision was made because implets needed to perform multiple functions, and we wanted to make sure that sufficient processing power was available for all of them. One of the consequences of minimizing processing power was that we decided not to use data compression in our prototype.

To accommodate slow speed lines, we designed the link level protocol to have minimal overhead. First, the protocol is optimized for transport of IP packets because the network was designed for exactly that purpose. In the current protocol, only two octets (8-bit data bytes) of header accompany each link-level packet. There are no checksums or parity bits on link-level packets that carry IP traffic. Second, the protocol uses best-effort delivery with no attempt to detect or recover from transmission errors. The major advantage of best-effort delivery is that the link-level protocols require no checksums and introduce no additional delays. If a packet is dropped, higher-level protocols like TCP detect the problem and retransmit. Third, the protocol uses encapsulation. When an IP packet enters Cypress, the implet computes its destination and encapsulates the entire IP packet in a Cypress packet, for transmission from one implet to another. When the packet reaches the last implet along its path, that implet extracts the IP packet and routes it outside Cypress. The major advantage of encapsulation is that it eliminates recomputing IP header checksums at each packet switch, thereby reducing delay and taking less processing power from the implet.

Flexibility. To keep Cypress flexible and amenable to change, the protocols have separate *packet type* and *packet handling* fields. As with most protocols, the type field specifies what the packet contains. The most commonly used packet type specifies that the packet carries an IP packet encapsulated in it. Other packet types are used to carry Cypress control messages that implets exchange for network maintenance and control. For example, one type of packet carries a message that requests an implet to divulge its current routing table. Another packet type requests an implet to reboot (used in emergencies by the network manager).

The packet handling field specifies whether the packet should be routed directly (used for conventional transmission), sent using reverse-path forwarding or flooding (used mainly to propagate control messages across the network), or sent across one link to an adjacent implet (used to build routing tables). Cypress uses reverse-path forwarding, flooding, and neighbor handling only for special packets. For example, neighbor handling allows an implet to send a "who are you request" to each of its neighbors before it knows their addresses. To guarantee that an implet never misinterprets scrambled packets as control messages, the Cypress protocol specifies that all control packets carry a checksum.

In most networks, packet handling is determined by the packet type. By contrast, the Cypress protocol specifies that an implet should act on the handling field first, and only examine the type field if the packet is destined for the implet itself. When it determines that a packet is destined for itself, the implet examines the packet type field, discarding the packet if it does not understand the specified type.

Separating the type and handling fields, and following the algorithm described above gives Cypress flexibility and allows experimentation with new packet types and protocols not normally possible. To conduct an experiment, one invents a new packet type, *P*, and arranges for an arbitrary pair of implets, *A* and *B*, to use the new type. *A* and *B* can exchange packets of type *P*, even if the packets must pass through other implets that do not understand type *P*.

Topology. Initially, we expected Cypress networks to be vine-like with a single node attached to the Internet and other nodes attached in a long chain[3]. The origin of the growing-vine topology comes from BITNET [11]. The idea is simple: each site leases a line to the nearest existing site, keeping costs to a minimum.

Two things have changed since the early days of Cypress, however, that make the long vine approach less appealing. First, as existing vine-structured networks grew and became more heavily used, their performance grew worse. Second, with expansion of the ARPANET and the advent of NSFnet and NSF regional networks, it became apparent that instead of a single Cypress network, it would be more economical to have many, small Cypress nets each covering a geographical area. We refer to this hub-and-spoke topology as a *cluster*.

Incremental Expandability. Shifting from vine topology to cluster topology changed one assumption we made early in the project and stimulated work on ways to build an incrementally expandable hub site connection. Because each implet can support only a small number of leased lines before the CPU becomes saturated, we looked for a mechanism that could be used to cheaply connect additional lines to the hub site.

The mechanism Cypress uses to provide an expandable hub site consists of a conventional gateway that handles IP routing plus a set of small, inexpensive machines called *X-boxes*[4], that handle data transfer [12]. Each X-box connects to one leased line; all X-boxes connect to the gateway over a short Ethernet as Figure 2 shows.

An X-box assumes responsibility for handling the transfer of packets between a serial line and the gateway. It accepts packets from the gateway destined for the Internet, and transmits them over the serial line to an implet. It accepts packets over the serial line and sends them across the local area network to the gateway. As usual, the gateway routes packets on toward their final destination.

When an X-box starts running, it does not understand any routes. It sends all traffic to the routing gateway. Using standard Internet protocols, the gateway will send *redirect* messages to an X-box to inform it about shorter routes. In particular, the gateway will tell a given X-box which destinations can be reached through some other X-box. Each X-box listens for redirect messages from the gateway and builds up a cache of routes, so that, in the steady state, only traffic destined for the Internet goes through the router; X-boxes send all other traffic among themselves.

---

[3] In fact, the name *Cypress* was chosen because it refers to a fast-growing American vine.
[4] The name *X-box* comes from the Xinu operating system used in the first prototype.

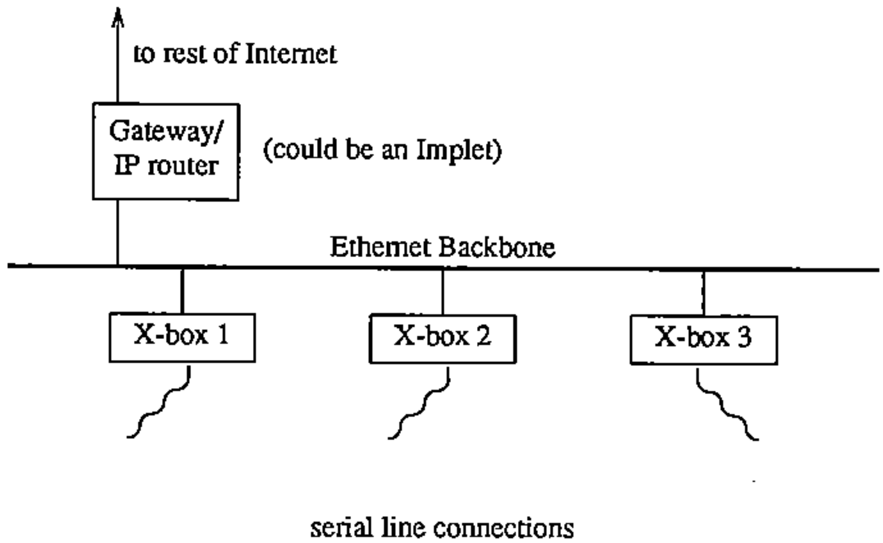

Figure 2. The incrementally expandable interconnection of multiple leased lines to the Internet at a cluster hub site.

The chief advantage of separating the routing and data transfer functions is economic. First, low-cost commercial hardware can be used for X-boxes. Second, because X-boxes are independent of the router hardware, they can be upgraded independently, making it possible to replace part of the hub site hardware without purchasing an entirely new machine.

Leaf-Node Connections. While designing X-box software for cluster hub sites, it became apparent that the same technology could be used for small subscriber sites as well. The main purpose of an X-box is to transfer packets between Cypress (i.e., a serial line) and an Ethernet. Consider a small site with a single Ethernet. By connecting an X-box between the subscriber's Ethernet and a Cypress leased line, that Ethernet can be added to the Internet. The subscriber's computers think of the X-box as an IP gateway; the X-box can send packets to any machine on the Ethernet.

Using X-boxes to connect subscriber sites makes the initial hardware costs extremely small, but limits expandability. Unlike an implet, the X-box is not a real IP gateway. It does not propagate routes, nor does it advertise them to the Internet core gateway system. Most important, the X-box can only reach machines on a given physical network; it does not understand how to route to multiple networks at the subscriber's site.

3. Experimental Prototype

To test our ideas, we built and operated a prototype Cypress network with a hub in the Computer Science Department of Purdue University in West Lafayette, Indiana. The prototype has been operational since November, 1985 with traffic between the prototype and the rest of the Internet passing to the ARPANET through an Internet core gateway, also at Purdue. From West Lafayette, lines run east to Massachusetts, west to California, southwest to Arizona, and north to Chicago, connecting the following sites:

CSNET Coordination and Information Center	BB&N Inc., Cambridge, Massachusetts
Digital Equipment Western Research Laboratory	Palo Alto, California
Williams College	Williamstown, Massachusetts
Boston University	Boston, Massachusetts
University of Chicago	Chicago, Illinois
University of Arizona	Tucson, Arizona
Geosciences Department at Purdue University	West Lafayette, Indiana
SUN Microsystems	Chicago, Illinois

The Chicago area office of SUN Microsystems will be connected as soon as the leased line becomes operational. They are participating in the Cypress project, and will be porting the software to new SUN operating systems and machines.

Several of these sites rely on Cypress as their only connection to the Internet. For example, the University of Arizona's Computer Science Department has used Cypress exclusively for over a year. They routinely send mail and login to remote machines.

Hardware and Software. Early in the design we decided to use off-the-shelf hardware and software systems. The original implets used for our prototype network were VAX 11/725 computers donated by Digital (the 11/725 uses a VAX 11/730 processor which performs at approximately 0.3 MIPS). These are being replaced by newer machines that are much faster. In particular, we have implet software running on Digital Equipment Corporation Microvax II hardware and SUN Microsystem SUN 3/50 or SUN 3/75 systems. Only two 725s remains in production.

We have experimented with a variety of serial line interface hardware. The 725 and Microvax systems now use standard Direct Memory Access (DMA) hardware that transmits an entire packet with only one interrupt. Hardware used to drive ascii terminals can be used at 9.6 kbps, but a special interface board is needed for 56 kbps lines. On the SUN systems, Cypress uses the standard serial ports that require an interrupt-per-character, but the CPU is fast enough to handle the additional overhead. In fact, the SUN implets operate at speeds from 9.6K through 56 kbps using the same hardware.

We started with UNIX† as our base operating system because we had access to sources and considerable local expertise. Our implets now run 4.3BSD UNIX, ULTRIX (Digital Equipment Corporation's BSD-based commercial product), or SUNOS 3.2 (SUN Microsystem's version of UNIX). To add Cypress code to the operating system, we introduced a new network interface for the existing IP software, and added driver code to handle the link-level processing.

Network Monitor and Control Software. Building on a stable, well-understood base made it possible to get an early version of Cypress working quickly, and allowed us to take advantage of others' work on standard protocols like TCP. As improved versions of the operating systems have emerged, we have migrated Cypress to them.

Cypress network monitoring and control programs execute as user-level processes, using the *ioctl* system call to pass data between user processes and the operating system kernel. The idea of separating monitoring from the link-level protocols has worked extremely well, and turns out to be an important ingredient in Cypress. It allows us to change the network monitor without touching the link-level protocols. It allows us to change the network in dramatic ways without rebooting the operating system at all. More important, our monitor and control processes have

†UNIX is a registered trademark of AT&T Bell Laboratories.

access to TCP/IP, so they can communicate with any machine on the Internet. For example, it is possible to contact an implet monitor server from an arbitrary machine and request periodic reports of traffic statistics. Thus, one need not be directly connected to a physical Cypress network to monitor it.

Performance. Performance of the Cypress prototype has been extremely good. Because the protocols are optimized for transport of IP datagrams, the link-level introduces little overhead. During file transfers, for example, user data accounts for approximately 85% of the raw hardware line speed. Typical throughput for a file transfer across a 9.6 kbps line averages 800 Bytes/second (user data). FTP, TCP, and IP overhead accounts for most of the rest, with less than 4% attributable to Cypress. More details can be found in [13, 14, 15]

Most important, user reaction has been fairly positive. Atmospheric scientists in the Geosciences Department at Purdue use Cypress without interest in, or understanding of the underlying technology. They routinely transfer files of over 200,000 bytes from the National Center for Atmospheric Research (NCAR) in Boulder Colorado, going through at least 4 Internet gateways and Cypress. Other sites also use Cypress for file transfer, remote login, and mail delivery. For example, they are easily able to establish a connection and deliver mail using the standard SMTP mail delivery protocol without a break in the connection.

4. Conclusions and Future Directions

When the Cypress project started several years ago, it was not obvious that a best-effort style of link-level delivery over slow speed leased lines would provide reliable service, or that a single CPU would be capable of handling the processing required for a multi-function packet switch like a Cypress implet. The success of the experimental Cypress prototype demonstrates the viability of these ideas, and leads to the inescapable conclusion that it is possible to provide reliable, low-cost Internet access. In fact, advances in processor technology make it clear to us that conventional operating systems should be used in all IP gateway machines, and that monitor and control software should be written to use transport protocols like TCP/IP to communicate.

We also learned that a hub-site topology is important. Cypress started as a way to provide CSNET with a low-cost alternative to ARPANET or X25NET [10], so we looked for a topology that incurred least cost. Part of the motivation for moving to a hub-site approach came from potential subscriber sites, who were willing to pay a little more for a line connected "closer" to the Internet. The expansion of the Internet by NSF introduced many more potential connection points, and forced us to consider alternatives to the original vine topology. With main arteries and regional networks in place, it is clear that Cypress fits in as a technology that provides capillary connections. We find it ironic that, while Cypress was started to help find ways to provide IP connections, the original topology resulted from underestimating how widespread Internet connections would become.

The notion of having a hub-site forced us to find new ways to accommodate many serial line connections. Although we conclude that implet technology is a good idea, the economics of commercially available systems make it less expensive to build a hub from small, independent processors that handle only data transfer. Our hub technology allows incremental expansion, as well as the freedom to upgrade parts of the system.

Another lesson we learned is that there are still many potential Internet sites with only one, sparsely populated network. In fact, many sites have only a few small machines like IBM PC or Apple MacIntosh computers. We realized that such sites could not afford to purchase an implet (and did not need one). In fact, as long as a site has only a single network, the X-box technology originally developed to make hub sites expandable can provide a connection to Cypress. To connect, the site purchases an X-box and runs our software, with the only apparent difference being that the X-box cannot handle multiple networks at the user's site. We now have a version of the

X-box software in production at one of the Cypress sites using LSI-11 hardware. In the future, we expect to have the X-box software itself working on the IBM PC and on Apple MacIntosh. The reason for choosing IBM PCs is that their popularity has driven the price down; the reason for choosing the MacIntosh is that it will allow us to connect sites that only have an Appletalk local network.

Besides experimenting with new machines and networks, we plan to help CSNET establish a higher speed Cypress technology, one that can be used to provide 56 kbps service to many sites. The plan calls for pushing the hub-site technology down one level in the tree, and using an incrementally expandable cluster. CSNET expects, for example, to establish a cluster on the west coast by using Cypress technology to connect an Ethernet to the Internet and then having subscriber sites connect from that Ethernet to their sites using IP routers. The advantage of such a scheme is that it allows an industrial subscriber to use their own hardware for the connection, with Cypress as one possibility. Naturally, we encourage industrial partners to work with us in such experiments.

In addition to activities that will help move Cypress out of the lab and make it more widely available, we continue to tackle new research questions. This year, we have been exploring how TCP retransmission algorithms work over slow networks like Cypress. We have found, for example, that some algorithms respond poorly to long delays by retransmitting frequently. In one case, a long delay caused TCP to retransmit but then to misinterpret the first acknowledgement it received as belonging to the second transmission. As a result, when used over Cypress the algorithm transmitted exactly twice as many packets as needed.

In other research, we have begun to consider routing between Cypress and the Internet. The problem lies in the conceptual model behind the Internet, which does not easily permit multiple interconnections among networks. In the case of the current Cypress prototype, for example, all traffic between Cypress and the Internet must pass through Purdue even though one of the implets resides at BB&N. Thus, traffic from Boston University to MIT travels from Boston to Purdue across Cypress, and then back to MIT across the ARPANET. Our research is looking for ways to route packets in the presence of multiple interconnections between networks. The problem is not unique to Cypress; results will apply to any long-haul network in the Internet, including NSFnet.

5. For More Information

For information about joining CSNET as a Cypress member contact:

Dick Edmiston
CSNET Coordination and Information Center
(617) 497-2777

Cypress software is available without license fee for use by NSFnet sites. For more information, contact the authors.

6. Acknowledgements

We gratefully acknowledge suggestions from many sources. Mary Catherine Privette and Gregory Smith contributed much to the software.

References

[1] V. G. Cerf, P. T. Kirstein, "Issues in Packet-Network Interconnection," *Proceedings IEEE*, 66:11, November 1978, 1386–1408.

[2] R. Hinden, J. Haverty, and A. Sheltzer, "The DARPA internet: Interconnecting heterogeneous computer networks with gateways," *Computer 16*, 1983, 38–48.

[3] J. Postel, "Transmission Control Protocol," RFC 793, DARPA Networking Information Center, September 1981.

[4] J. Postel, "Internet Protocol," RFC 791, DARPA Networking Information Center, September 1981.

[5] J. M. McQuillen and D. C. Walden, "The ARPANET Design Decisions," *Computer Networks*, 1:5, September 1977, 243–289.

[6] R. M. Metcalf and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Comm. ACM*, 19:7, July 1976, 395–404.

[7] R. E. Kahn, "The Organization of Computer Resources into a Packet Radio Net," *IEEE Transactions on Communications*, com-2J, January 1977, 169–178.

[8] I. Jacobs, R. Binder, and E. Hoversten, "General Purpose Packet Satellite Networks," *Proceedings IEEE*, 66:11, November 1978, 1448–1467.

[9] D. M. Jennings, L. H. Landweber, I. H. Fuchs, D. J. Farber, and W. R. Adrion, "Computer Networking for Scientists," *Science*, 231:9, February 1986, 943–950.

[10] D. Comer and J. T. Korb, "CSNET Protocol Software: the IP-to-X.25 Interface," *Proceedings ACM SIGCOMM '83 Symposium*, March 1983, 154–159.

[11] I. H. Fuchs, "BITNET: Because it's time," *Perspectives in Computers*, 3:1, March 1983.

[12] D. Comer, T. Narten, and R. Yavatkar, "The Cypress Coaxial Backbone Packet Switch," *Proc. of the International Conference on Communication and Data Communication, Nivelles-Belgium, May 1987.*

[13] D. Comer and T. Narten, "The Cypress Multifunction Packet Switch," Technical Report CSD-TR-575, Computer Science Department, Purdue University, March 1986.

[14] D. Comer and G. Smith "Early Cypress Performance Experiments," Technical Report CSD-TR-581, Computer Science Department, Purdue University, March 1986.

[15] D. Comer and G. Smith, "Sun Workstations as Cypress Implets," Technical Report CSD-TR-662, Computer Science Department, Purdue University, March 1987.