ABSTRACT- ETHERNET AND THE FIFTH GENERATION 820518.doc	2/18/2000 6:18 PM	Microsoft Word 9...	2 KB
another ethernet talk copy 820211.DOC	8/18/1999 9:54 PM	Microsoft Word 9...	20 KB
bob metcalfe ethenet letter DONE 840518.doc	8/19/1999 7:21 AM	Microsoft Word 9...	4 KB
ETHERNET ADVANCED DEVELOPMENT--EMS - BELL,PORTNER 791031.doc	2/18/2000 6:18 PM	Microsoft Word 9...	2 KB
ETHERNET- DEC'S BACKBONE NETWORK AND ET - DENNY BJORK 820609.doc	2/18/2000 6:18 PM	Microsoft Word 9...	4 KB
ETHERNET PERFORMANCE 820218.DOC	8/18/1999 9:54 PM	Microsoft Word 9...	24 KB
ETHERNET PRESENTATION IN NY - THANK YOU 820216.doc	2/18/2000 6:18 PM	Microsoft Word 9...	3 KB
ETHERNET SPEECH 820211.DOC	8/18/1999 9:54 PM	Microsoft Word 9...	28 KB
ETHERNET SPEECH-PRESS CONFERENCE 820208.doc	2/18/2000 6:18 PM	Microsoft Word 9...	29 KB
ETHERNET, ICL PRES WILMOT ON USING ETH. - LACROUTE - 11-81 820226.doc	2/18/2000 6:18 PM	Microsoft Word 9...	5 KB
ETHERNET, UNIBUS OF FIFTH GENERATION 820125.doc	2/18/2000 6:18 PM	Microsoft Word 9...	53 KB
ETHERNET,XEROX-DEC ANNOUNCEMENT OF--EMS - CLAYTON,FULLER 791029.doc	2/18/2000 6:18 PM	Microsoft Word 9...	2 KB
ETHERNET--KEN'S PRES--HELP AND COMMENTS - JOHN ADAMS 820517.doc	2/18/2000 6:18 PM	Microsoft Word 9...	6 KB
ETHERNET--OUT AND STANDARDIZE IT - LACROUTE 810527.doc	11/7/1999 9:36 PM	Microsoft Word 9...	3 KB
ETHERNETS STARS FOR ENG & TYPESETTING REV-ENG STAFF 820517.doc	2/18/2000 6:18 PM	Microsoft Word 9...	4 KB
INTEL--INTERFACE ETHERNET & BUYING FROM THEM - FEDERMAN 810330.doc	2/18/2000 6:18 PM	Microsoft Word 9...	8 KB
XEROX--THE 820, A LOW COST ETHERNET - AVERY 811001.doc	2/18/2000 6:18 PM	Microsoft Word 9...	3 KB
XEROX - ETHERNET - PAKE, DR. GEORGE - CAMPBELL, JAMES 790207.docx	5/7/2017 12:45 PM	Microsoft Word D...	17 KB

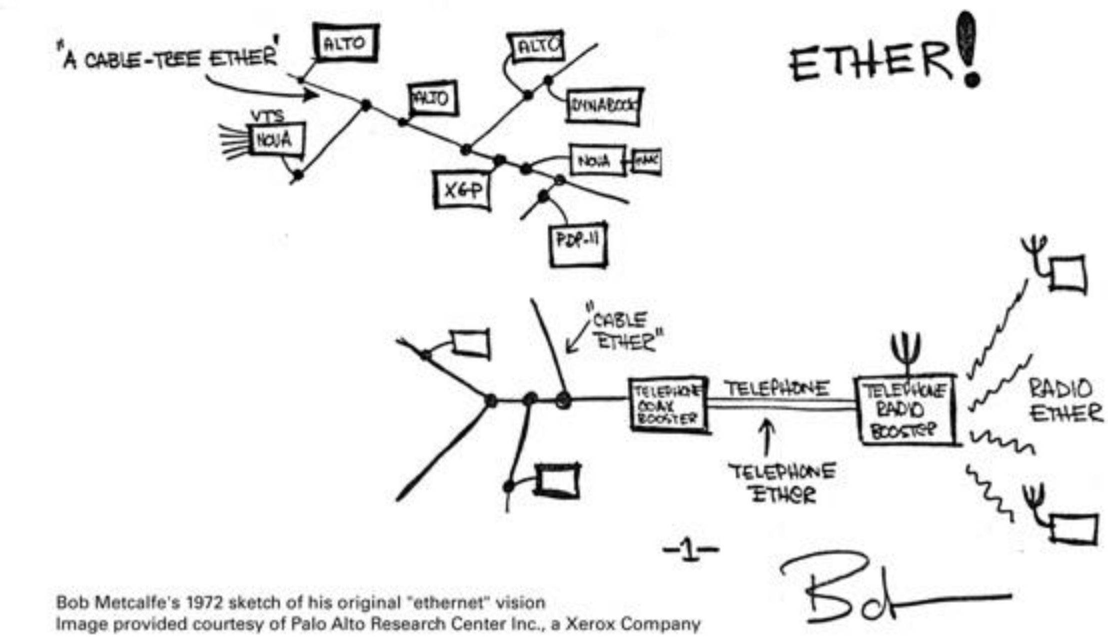
DEC viper ethernet tap

This is all I seem to have on the Ethernet that included an agreement with Xerox and Intel. Bob Metcalfe, under Sam Fuller's care, was the catalyst that made it all happen. Several stories are noteworthy. The agreement with Intel was made using the ATT Picturephone Meeting Service (there were a dozen PMS centers including Boston and San Francisco) I and our team went to Boston, and Phil Kaufman met us in San Francisco. Subsequently, when DEC failed to use the Intel part, Andy Grove, gave me a bit of his personal wrath!

Rob Wilmot, the Chairman of ICL and his team met with us and he brought the European manufacturers into the Ethernet club. We were all competing with the non-existent IBM Token Ring!

David Liddle was the Xerox liaison. They visited us and I sent him back to California with a letter of agreement that he used to secure Xerox's permission. The big deal was NO royalties!

Bob Noyce, Dave Liddle and I made the Ethernet announcement at the World Trade Center and then subsequently in London.



Bob Metcalfe's 1972 sketch of his original "ethernet" vision
Image provided courtesy of Palo Alto Research Center Inc., a Xerox Company

From: Bob Metcalfe [mailto:bob.metcalfe@utexas.edu]
Sent: Sunday, May 7, 2017 6:52 PM
To: Gordon Bell <gbell@outlook.com>
Subject: Re: Ethernet History. Letter to kick it off.

Ahoy! Gordon,

Thanks for the letter. Eventually led to Intel, DIX, and IEEE.

Ethernet's birthday is May 22nd (1973).

Ahoy!

Bob.Metcalfe@UTexas.edu
www.Engr.UTexas.edu/Innovation
Make an innovation grant: http://bit.ly/1SbELZQ

On May 7, 2017, at 3:13 PM, Gordon Bell <gbell@outlook.com> wrote:

David,
Just stumbled across this letter that was sent when you visited DEC with a team.
I asked what do we need to do to get Ethernet as a standard?
You said something like send a letter to our management?
I sat down at a word processor and you guys dictated this letter. I signed it. Your guys took it back with you.
Regards,
G
Later on, I squelched Bob's PARC request or plan to build more 3 Mb Ethernets for universities.

Gordon Bell
611 Washington Street, #2502, San Francisco, CA 94111
Phones: cell 415 640 8255 (preferred); home 415 392 3272
http://gordonbell.azurewebsites.net/
http://TotalRecallBook.com
http://TCM.ComputerHistory.org The Computer Museum (Boston)

<XEROX - ETHERNET - PAKE, DR. GEORGE - CAMPBELL, JAMES 790207.docx>

Network Debate

Published: September 5, 1982

To the Business Editor: There are several inaccuracies in the article, ''Debate Over Office 'Networks,' '' (Aug. 15). Digital Equipment Corporation, the Intel Corporation and the Xerox Corporation, far from ''quibbling,'' have worked together closely for three years in writing and promoting specifications for Ethernet, the best-known local area network on the market.

In fact, the relationship among the three companies could well serve as a model of cooperation in promoting the broad general interests of the marketplace while preserving the integrity of a highly competitive free market.

The three companies did not ''jointly announce Ethernet.'' Ethernet technology is proprietary to Xerox and was patented by them as early as 1976. The announcement in 1980 involved the mutual development of specifications for compatible products employing Ethernet technology.

Competing companies need not agree on standards for local area networks, since they do not set the standards. This is done by professional standards associations which are well along in standardizing around local area networking techniques. In fact, standards groups in the United States and Europe have already adopted standards which are essentially compatible with Ethernet. C. GORDON BELL Vice President, Engineering Digital Equipment Corporation LESLEY VADASZ Senior Vice President Intel Corporation JOHN V. TITSWORTH Executive Vice President Xerox Corporation Aug. 25, 1982

Dear Bob:

Congratulations on enlisting IBM's support to make Ethernet a success. It is truly gratifying to see that persistence (mainly yours) is beginning to payoff. This is going to make LANs possible instead of the continued reinvention of physical links. I can't say that I'm very proud of the rapidity with which I lead the DEC products, but we spent a fair amount of time looking at new cables rather than just building Ethernet products. Unavailability of chips contributed to the slowness too. The newer chips, especially National's should really make it widely available for use with PC's and even terminals. Ethernet will become the base component to build new systems.

I met some folks at Excelan the other day at Electro and they indicated that the IBM cabling announcement was unleashing orders. AT&T's support, along with various vendors such as Prime, is encouraging too.

Maybe it's my inventive mind, but what's the possiblity that IBM has licensed the token ring patent in order to make their own net proprietary and NOT an open standard?

Is it possible to get all (or a few Ethernet vendors) together and make a big cabling announcment? I would lead this if I were at DEC, but you could count on me to be part of a serious spoof if you lead it. It would posit an alternative to their announcement which would solve the same problems, but has been here for two years. I think it needs to be done, and AT&T might be persuaded to join in if the announcement also says don't pull out your old telephones. It was surprising to see that IBM didn't include CATV in the bundle, since there were already 5 other cables.

When I get the full poop on their scheme, I'll try to work this out in more detail, but for now, what do you think of the idea, complete with manuals, cables, etc.?

The dream you inspired in me of "Ethernet is the Unibus of the Fifth Generation" is in sight. Thanks for inventing it, but the real contribution is persistence.

Sincerely,

Gordon Bell
Chief Technical Officer

GB13.19

CC: Ken Olsen, Pat Courtin
FROM: GORDON BELL DATE: WED 31 OCT 1979 4:39 PM EST
DEPT: COD
EXT: 223-2236
TO: LARRY PORTNER
JIM BELL

SUBJECT: ETHERNET ADVANCED DEVELOPMENT

GB0005/52/EMS

I think NI is casting about just like CI and BI. We are busily speccing and we have no experimental basis for our specs, architecture or plans. Could we put on hold the A/D projects in TW (e.g. 1 user Nebula) and possibly elsewhere (e.g. R+D) and get this essential work done now? The priorities and needs, I hope, are clear.

GB:swh
****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION

DATE: SUN 9 MAY 1982 1:24 PM EDT
FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: DIGITAL'S BACKBONE NETWORK AND ETHERNETS

Two months ago I gave presentations to the US and European Press on: WHY ETHERNET IS THE KEY TO THE 5TH GENERATION!
The paper is available and suggested reading for two reasons:

1. I believe this is what computer systems will look like over the next few years. The talk is conservative and doesn't present the more radical view that systems will evolve into clusters of personal computers instead of being central and shared as in today's timeshared minis and mainframes.
2. We (DIS and all organizations) should be aggressively installing both Global and Local Area Networks and evolving to Ethernet throughout the company. I'm appalled at how we talk about distributed systems, yet we still build centralized systems (eg. order processing) that run in batch mode and don't communicate with other batch systems.

In May, sales is presenting their plans for a distributed order processing, clustered around several regional sites. Here's what I believe we must demand from this presentation:

1. DIS will take responsiblity for a main, backbone corporate store and forward network, DBN, capable of carrying the traffic between site Local Area Networks. In essence, this is an ARPAnet or Telenet, formed from interconnected Plutos.
2. DIS will provide links to the backbone network, DBN, for terminals, individual computers and LAN's at the extremities.
3. Each major organization (Engineering, Field Administrative Centers, Manufacturing Plant) or site (eg. Hudson, Colorado

Springs) will install single LAN's so that by Jan. 1984 all DEC's computers can communicate directly rather than via RJE protocols.

It is imperative that we get these plans in place now, because we can not get the responsiveness and error free order processing we need without direct link of the sales machine to the factory!

Similarly, we can not build a CAD/CAM system without a method of linking today's engineering CAD systems to the factory!

We must lead the world in building and using systems like this, because this is what our customers are buying. Furthermore, this is the key to our adminstrative productivity; the batteries of clerks we have transshipping paper to one another will smother us. The current, open-loop batch systems (eg. order processing) insures errors and the inability to make an integrated system

Let's give our support and demand excellence in this project!

"TO" DISTRIBUTION:

DENNY BJORK	AL CRAWFORD	BOB DALEY
JIM FRIEL	BILL HANSON	JIM MILLER AND BOB DALEY
DAVE KNOLL	AVRAM MILLER	OPERATIONS COMMITTEE:
PEG:	TERRY POTTER	

GB3.S5.26

Performance Of A Simulated Ethernet Environment

00 BURT DECGRAM ACCEPTED S 9630 O 75 14-NOV-81 17:51:00
****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION   DATE: SAT 14 NOV 1981
5:48 PM EST

FROM: GORDON BELL
cc: see "CC" DISTRIBUTION   DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP:
ML12-1/A51

SUBJECT: DISCUSSION WITH ICL PRES. WILMOT ON USING ETHERNET

Just finished talking (4:30 EST, Saturday) to Wilmot (011-44-1-949-5903 London, Telex at Putney England) regarding their standardization of Ethernet. They are being driven by their customers (eg. Citibank, Barclay's) to do this because of the proliferation of minis and because of the Xerox PR pressure. He commented: "Everyone is buying it but no one has the slightest idea as to what it is. There is incredible commercial support."

They are collaborating with 3 Rivers via building the PERQs and with MITEL on PABX's; ALL are committed to use Ethernet and will be defining a product there in the next few months. He said: "They have looked at the Xerox Level 3 and 4 protocols and promptly went out and got drunk! These protocols are like SNA squared." They like Ethernet and want to support it because of the commercial momentum, the Level 1 and 2 standard and the need, but are really saddened with the technical merits of the Xerox work. As British engineers, they demand technical elegance. They have also been working with the Ungermann Bass boards and wll probably start using them. They want the following:

1. Use EN for connecting between ICL (largest European Computer Co.) and DEC (for minis), MITEL (for PABX) and 3 Rivers (their professional workstation). Have the standard actually ECMA based (their customer base) which is a superset of OSI and x.25, but goes up to the applications level.

2. Get all these folks together to get a standard. Also get the

customers in to review it.

He suggested a massive technical effort to get these standards set in the next few months with persons from DEC, Mitel, 3 Rivers, DEC, some customers, etc. I said we do not do this sort of thing! I did suggest we have a technical exchange as to where we are and plan to be. The goal would be to see if we have any basis for doing any task force work in this regard.

I stated our position: We are evolving DECnet to be used with Ethernet as a low level transport and we are evolving DECnet to converge with OSI and x.25 as they become standard. We are providing gateways to the other networks such as SNA, x.25 and Xerox servers as necessary.

WE HAD AMAZING CONCURRENCE:

1. Technical standards are becoming a major market stategy and force. We are not moving aggressively enough in recognizing this and acting on it.

2. If we don't hurry, it's all academic. We'll be implementing a
variant of SNA!

THE NEXT STEP: Bernie, please send him a Telex inviting them to give a technical presentation here in exchange for our presenting the protocols we plan to use on EN. I would like to have Xerox and Intel be part of the same meeting. The goal would be to get this done this next week.

"TO" DISTRIBUTION:

SAM FULLER	BERNIE LACROUTE	TONY LAUCK
BILL STRECKER		

"CC" DISTRIBUTION:

BILL DEMMER	MARY JANE FORBES	GVPC:
WIN HINDLE	BILL KIESEWETTER	DAVE RODGERS

GB3.S2.39

****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION
EST

cc: KEN OLSEN

DATE: TUE 6 APR 1982 4:31 PM

FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: KEN'S PRESENTATION ON ETHERNET: HELP AND COMMENTS FOR HIM

Ken's talking to a financial group on the same day as the May announcement and would like to discuss Ethernet. I have him my talk to read. In addition he might want one of you to get more details on competitive technology like Wang, Datapoint IBM and the phone company.

He clearly should hand out the new Tutorial Handbook on LANs, and he could hand out the clean copy of my talk too. Also, he might want some slides.

Could you get other poop together like the Q&A on Ethernet, but without swamping him in paper?

Answering the why nots
Why not wait for the phone company and PABX's? Have you ever tried to use a terminal at 1200 baud.... we're talking about a system 10,000 times faster. It can transmit a high resolution black and white image in 0.1 sec or a color image in 1 sec. We don't see the wide scale availability of even 56Kbits in the foreseeable future from the phone companies on any kind of wide scale.

Why not put in a non ATT data and voice pabx? Why bother with the expense for the extra wiring. It still doesn't have adequate bandwidth between computers, or terminals or personal computers.

Why not use broadband
1. There's not standard for either data or data and catv.
2. Broadband is like a new piping material that can be used to distribute physical goods like gas, sewage, water, oil and steam. The duct can carry anyone of them, it's the sorting it all out that's a bitch.
3. Many users want broadband because they assume some other user is going to pay for the installation.
4. It's hard to believe that broadband is going to be very pervasive in industrial environements. It is not adequate for two way videophones because of the limited bandwidth.
5. Systems like Wangnet use a second cable for return. Why not put in a second yellow wire and keep the two independent.
6. Baseband is simple to install. The users often do it.

7. We don't see broadband as being suitable for voice based on cost of modems and the cost of throwing out an existing plant using a central office type pabx with all its wiring and phones.
8. Ethernet will also carry a reasonable amount of voice, although we probably won't push it to evolve this way. It will mainly be carrying voice mail packets. One of the nice things is that the systems we are talking about are built so that there's less voice traffic. The personal computer will be used to help be less intrusive than the telephone.
9. The ultimate single media system will have much more bandwidth than broadband. This would allow videophones and images and certainly satisfy intercomputer needs. We don't see this as being practical until fiber optics and central switches that support them are available. We see no reason why this couldn't be done within 10 years. However, we've seen no reasonable laboratory demonstration of this yet.
10. If all else fails, then why fight it, we'll use broadband cables and put an Ethernet transceiver to encode our system into broadband. WWe have no real hangup with not using broadband. It's just that it's very nebulous, undefined and unstandardized now.
11. The users are demanding an OPEN standard. We're the only one who's proposed it.

How are you coming on the standardization?
The ieee 802 standard is progressing nicely. Various ecma companies including ICL, olivetti, cii and Siemens have joined in the standard. There are dozens of companies building products to the standard now. You can buy interfaces and components and put Ethernets together. Xerox has installed about 100 of them. We have about 10 Ethernets ourselves within the engineering organization.

What about IBM?
Ask em. They are doing their usual bit to find some standard that everyone else will have to meet. They were clearly in the dark about the need for LANs. We clearly understand them and are predicating all our products on them!

EVERY DEC PRODUCT WILL CONNECT TO ETHERNET EITHER DIRECTLY OR VIA A CONCENTRATOR. All our multiterminal systems will connect to them directly. The stand alone systems like the cpm and decmate will interface to other systems as terminals and file transmission.

What about Wang
We're waiting to see. They typically announce products 3 years before they're ready. The whole world changes in a half a computer generation.

"TO" DISTRIBUTION:

JOHN ADAMS MARION DANCY BERNIE LACROUTE

GB3.S4.40

ETHERNET--OUT AND STANDARDIZE IT/LACROUTE/GB2.S6
TO: BERNIE, CC:SAM, GVPC, DAVE RODGERS, BILL STRECKER, RALPH DEMENT
SUBJ: LET'S GET ETHERNET OUT AND STANDARDIZE IT

I described that we intended to make Local Area Networks based on Ethernet at DECUS and I asked them to plan on wiring their buildings. Several of our customers thanked me for being specific.

Our current laissez faire position on standards is a disaster i.e. say we support standards, fight standards within a committee and then argue after the fact that it doesn't apply to us.

Let's be much more proactive on Ethernet.

0. Let's overtly sell it.

1. Hurry and get a product. Intel has one on 2 Multibus boards using less real estate and with higher performance! Why can't we?? What's wrong with our design team?

2. Let's hold regular meetings to standardize it and array those signed up.

3. Let's call specific people in other companies e.g. HP to get them committed!

4. Let's go all out at the IEEE Committee, ANSI and at ISO!

We need a plan!

GB:swh
GB2.S6.24

```
+---------------------------+
|   |   |   |   |   |   |   |
| D | I | G | I | T | A | L |
|   |   |   |   |   |   |   |
+---------------------------+
```

GB3.S2.18

I N T E R O F F I C E
M E M O R A N D U M

TO: DISTRIBUTION

Date: February 19, 1982
From: Gordon Bell
Dept: Engineering
MS: ML12-1/A51 Ext: 2236
EMS: @Core

SUBJ: DIX Ethernet NY Presentation. Install them now!

PRESENTAION AND PAPER

Let me thank all of you for the work that went into the seminar.

It may have been somewhat painful, but I think it was worthwhile. It had a strong effect on my own thinking. I look forward to being able to circulate the paper:

Why Digital Believes Ethernet is the Unifying Key To The Fifth Generation

for comment. We clearly need the set of written documents that further our customer and internal understanding.

OUR NEXT MOVE: INSTALL THEM WITHIN ENGINEERING
Now, let's get the products! Both Intel and Xerox are shipping.

They are great and we need them for product development and use. Historically, we never believe in or make products work until they are a part of our own use... therefore, let's install them
The MR one should be up soon for testing Pluto on a 10/20. Also, we need it for interconnecting the plethora of machines.

Let's get Ethernets operating in HU, MR, TW, ZK, DECwest, ML by July 1!

DISTRIBUTION:

MARION DANCY
BERNIE LACROUTE
ED CANTY
PAT MURPHY
GRETCHEN WICHTERMAN

CC:

DICK BERUBE
GVPC
ENGINEERING STAFF

WHY DIGITAL BELIEVES ETHERNET IS A UNIFYING KEY TO THE 5TH GENERATION

Gordon Bell
Vice President, Engineering
Digital Equipment Corporation

at Ethernet Press Seminar
City
10, 1982

FROM: GORDON BELL DATE: MON 29 OCT 1979 1:19 PM
EST
DEPT: OOD
EXT: 223-2236
TO: DICK CLAYTON
BERNIE LACROUTE
BILL STRECKER
DAVE RODGERS
GEORGE PLOWMAN
SAM FULLER
WAYNE ROSING

SUBJECT: XEROX/DEC ANNOUNCEMENT OF ETHERNET FOLLOW
UP:11/9/79

GB0005/37/EMS

Why don't we stop screwing around and adopt Ethernet AS IS? Then we can get a product quick, use it, and evolve.

The way we're headed it'll be 2 years to chips (if we're lucky) and another year or two to product. Meanwhile, IBM'll have the whole world wired with SDLC loops and we'll have to interface to them. This way, we get a compatible network with Xerox's printers and WP's. We're losing valuable time. Why not?

GB:swh

SLIDE 1

In the Fifth Computer Generation, a wide variety of computers will communicate with one another. No one argues about this. All the shouting is about how to do it and what form the computers will take.

SLIDE 2

A standard communications language is the key. I believe Ethernet is this unifying key to the 5th computer generation because it interconnects all sizes and types of computers in a passive, tightly-coupled, high performance fashion, permitting the formation of local-area networks. Ethernet is the standard that can hush the argument and let everyone get to work on the computing nodes.

SLIDE 3

Standardization is necessary because no one vendor has it all, or can provide the full spectrum of information processing nodes that are emerging. Most organizations have computers built by different vendors. Although computer data and processes (that is the work) are interdependent, no easy and inexpensive way to send data among machines exists. Everyone's customers are demanding a network standard. Ethernet can do it for everyone.

SLIDE 4

I'm going to tell you four stories that illustrate the different facets of Ethernet. The first is about the UNIBUS and why I think Ethernet is the UNIBUS of the Fifth Generation.

SLIDE 5

In 1970 Digital introduced the UNIBUS to interconnect parts of a computer. The UNIBUS is just a simple ribbon-like cable with 56 conductors as shown in this old ad. With UNIBUS people could easily assemble their own computers and did so in many different ways, and it became a standard.

Virtually all computers built today utilize a UNIBUS-type architecture, including Intel's Multibus, and Motorola's Versabus. Both of these busses are standards too.

SLIDE 6

This bus is a high-speed data path that links all system components within a single computer -- the processor, primary memory, secondary disk memory, communications interfaces, realtime equipment interfaces, interfaces to special customer equipment.

The complete UNIBUS specification is contained in a manual about 1/2 inch thick, roughly the size of the Ethernet blue book specification. From this, users have designed 10's of thousands of machines to match the computer to their application in an almost open-ended fashion. Small dedicated controllers, personal computers, pedagogical machines and large timeshared computers are all built this way. Any kind of computer can be built easily from a common set of components.

What started as a good scheme for interconnecting components that Digital supplied, became a lovely standard for starting a whole plug-compatible business. The unexpected result: an industry with 100's of vendors and lots of new competitors. The plug compatible parts mean lower prices. The non-mundane user designed connections to television cameras, robots and other devices act to stimulate the whole next computer generation, based on need.

SLIDE 7

Ethernet is only an extended unifying bus, like UNIBUS, that interconnects many computer based information processing systems but in a 2.5 by 2 kilometer area.

UNIBUS has a single processor for one computer. Ethernet can support many different computers in all sizes and places doing all types of work.

A UNIBUS system has local data storage; an Ethernet supports databases distributed throughout the network. The latter has evolved to be called the file server.

A UNIBUS system interfaces to other computers via slow communication links and tightly coupled parallel links. An Ethernet always interfaces to other computers directly. Components that are not computers are just not built today. Interconnection occurs directly and via special computers called gateways.

SLIDE 8

Ethernets couple host processors, people using their own special terminals, personal computers and workstations, as well as particular functions like file servers, print servers, communications servers, and realtime equipment in the laboratory and factory.

Gateways to other computers and networks can be provided by these communications servers.

Computer systems decomposed into separate, functional units on an Ethernet will be significantly easier to build.

SLIDE 9

Then users will participate more than ever in the design and building of their own systems and not be limited by the vision of a single supplier.

SLIDE 10

In the Fifth Generation, every computer on the Ethernet, will be both contributing to and sharing in the total resources of the network. The network will be the system

SLIDE 11

Having demonstrated that Ethernet is the UNIBUS of the 5th generation because it provides a passive standard to interconnect all sizes and types of computers into a high speed network, I will turn to the issue of Ethernet's role.

The second story is about the evolution of the computer generations - driven by the semiconductor evolution.

SLIDE 12

The Fifth Computer generation, like its predecessors, will only occur when there are new technologies and needs that converge to create a new computing structure.

Three technologies are fueling the 5th generation: the understanding of how to build a reliable Carrier Sense Multiple Access with Collision Detection (CSMA/CD) type network, in effect the Ether; Very Large Scale Integrated Circuits or VLSI permitting all logic to be computer based, but more importantly permitting a simple, low cost connection to the Ethernet cable, essential for a standard; and finally technologies such as high resolution graphics that accelerate the creation of computing nodes that are a pleasure to use.

More computer use results in increasing human potential and hence an increasing need or demand. GNP grows with the absorption of new technologies that allow higher productivity. Every person's productivity is limited by the rate computers communicate with one another. In effect, we have evolved the quadruped to a thoroughbred but not changed the track. The only paths that they can travel are muddy, rocky and random time-worn paths. We need a fast race track.

Our computers often wait at the gate while users physically carry data between them in what is becoming an inverted society -- the computers do the fun thinking parts and the users carry trivia from machine to machine, or become simple machine to people translators. Ethernet breaks this communication bottleneck. Furthermore, Ethernets can carry voice, graphs and pictures as well as simple messages and data files. They'll restructure use. It won't be a straightforward extrapolation of simple terminal to computer, and computer, to computer networking we know today.

In 1990, we can probably look back and identify trends that are not clear today. So I won't speculate about 1990, but I know the future will be more interesting than the simplistic, evolutionary view I'm presenting today.

SLIDE 13

The development can only happen if we provide the creative environment in which to invent. I think the Ethernet based open Local Area Network is this environment. "A local-area network is a set of information processing nodes, distributed in a single area and fully interconnected via high-speed data links." An open local area network is one in which any vendor or user can supply nodes for the network.

SLIDE 14

The user should be able to communicate over a local area network with the same nonchalance as the telephone, not knowing or caring how the network works or how the message is transmitted.

SLIDE 15

It's amazing that the front end user portion of the telephone and the computer really haven't changed much. The oscilloscope of the Whirlwind (the first real time interactive computer built in 1950) is just a bit bigger and more graphic than the ones on computers today. Jay Forrester and his associates used it as a personal computer. The user walked into a building that was the computer, and into a room that was the console, and sat down at the cathode ray tube. The computer spent most of its time waiting for the user to interact. This wasn't the best use of the world's only interactive personal computer.

SLIDE 16

Other early machines, such as the first one, EDSAC built by Maurice Wilkes in Cambridge, England, sought to be more efficient by keeping the users away from the machine. The programmers worked off-line and then handed programs on paper tape to people who put them on a clothes line and eventually fed them into the computer. This maximized the machine's use.

SLIDE 17

But isolated users quickly grew to hate and to be intimidated by the batch computers. People would prepare their programs on punch cards, submit them to a clerk and the program would be put in the queue. As often as not, errors were found in the program or data so instead of getting an answer to an immediate business problem the user had to rekey his program and and go back to the end of the line. It's no wonder that users wanted a different way of doing things.

SLIDE 18

With the introduction of transistor technology, computers started to get smaller. In 1960, Digital introduced the PDP-1, the first commercial computer with an interactive video display that played Space War, the granddaddy of all computer space games. In 1961, two typewriters were connected to a PDP-1 at Bolt, Bernanek and Newman and the timesharing idea was born.

SLIDE 19

In 1963, Just two years after the first experiment, Digital introduced the first commercial timesharing system, PDP-6, for 8 to 16 users.

SLIDE 20

Then the computer's time, wasted waiting for one user, was used by another. Throughout the sixties, the evolution of batch, personal and time-shared computers continued. Batch mainframes were developed with remote job entry terminals so a few lucky users could enter data from their offices.

Minicomputers, like the PDP-8, were small and inexpensive enough so they could be dedicated to particular applications. Many of these minicomputers were used to prepare data for batch processing on a mainframe.

Other mainframes became specialized timesharing machines. But computing was still very expensive and impersonal.

SLIDE 21

The real breakthrough came in 1972 when we learned how to provide timesharing on a minicomputer. For the first time, low cost, interactive, **personal** computing capabilities could be provided at a cost that most users could justify. Computers came out of the computer rooms and started working with users.

SLIDE 22

After the initial honeymoon, a need developed to interconnect the machines to each other and to the large batch machines which by now could be controlled from terminals. As a result, engineers did what came naturally and started to string wire between them.

SLIDE 23

In the late 70's the interconnection problem was exacerbated by the baby computer boom, known as personal computers.

Like children everyone wants a limited number for their very own. Personal computers give that one-on-one relationship. There's no longer anyone watching you work, not even an accounting program. You can do your own thing in a non-threatening way. No one need know if you use the machine or even if you turn it on... or it turns you on.

But then there are times that you and your personal computer want to be connected with another machine to get programs, transmit messages, look at a picture, or send a non-intrusive voicegram message.

And so many more wires have to somehow be added between the centralized, shared remote batch mainframe; the departmental timeshared minicomputers; and the individual personal computers. If there aren't lines running between all the machines then there probably should be. Otherwise, information that is on one node and needed elsewhere has to be re-entered.

SLIDE 24

Ethernet will provide the structure needed to manage distributed computing. It's coherent structure is capable of handling an ever-growing volume of traffic among all machines.

SLIDE 25

The last two stories address Ethernet user needs.

First, they provide high-speed interconnection among dispersed computers. Creative programmers are kept happy and work effectively when connected to high speed systems. They want to be able to call all the machines in their network and communicate with others in the nework independent of where they are. When we're working with a machine, we have less patience than a 2 year old waiting for a cookie.

Second, Ethernets provide simple interconnections of terminals and personal computers to host processors. New users starting with simple personal computers will be able to improve their performance by accessing larger machines as their needs increase. Clearly, history has shown that the more computer power anyone has, the more he wants. It is an insatiable hunger like none known before with the immediate reward of greater individual productivity.

Third, Ethernets interconnect all kinds of computer controlled equipment. For example, links between computer controlled equipment in the laboratory or on the factory floor, and data processing equipment in the office.

SLIDE 26

Every organization wants open ended, flexible links between personal computers and terminals and larger, more central computers.

SLIDE 27

Today, most users have simple block mode, fixed function terminals. Nearly all of these are evolving into complete personal computer systems.

It makes little difference whether the user has a simple terminal, or a full-fledged personal computer. For simple terminals, high bandwidth is needed for character-at-a-time interaction. For effective use of personal computers, high bandwidth is needed to transfer messages, files, images and voicegram messages.

Today the typical user is most likely linked to a single host computer, and communication with other computers is through this host.

User demanded local area networks develop by users wiring various hosts and terminals together.

SLIDE 28

The most common answer to the problem is to use telephone lines, putting all terminal traffic onto a telephone system which may not be capable of handling it. Then modems have to be installed to convert the digital signal generated by a terminal or computer to an analog signal that can be carried over a telephone line. The biggest problem is that our users want to communicate at least at 9600 bits per second, and this just can't be done economically with these switches.

A second answer to the problem is the data switch. By installing a switch between the user and the computer network it is possible for any user to connect to any computer.

But connecting terminals to the switch involves a lot of wiring. These diagrams are simple enough to draw, building them is complicated. Furthermore, terminal wiring is a never-ending business that requires much planning, results in much inflexibility, and is fueled with much money.

So even if -- at first glance -- both telephone lines and the data switch look like solutions, they aren't. They're part of the problem as anyone who has many terminals and computers will tell you.

SLIDE 29

I know personally, because Digital continually faces this problem like in our facility in Nashua New Hampshire where we have 30 computers and 700 user terminals.

A pristine view shows a number of computers and a big room used for switching the links between terminals and different computers.

SLIDE 30

You don't see the problem until you open the door. Every terminal line is wired to a board in this room. And every time an unplanned terminal is added someone is called to run more wires.

SLIDE 31

Wires are run from the board to the wireroom to a switch computer.

SLIDE 32

This switch computer is now bound and it doesn't grow very gracefully, particularly when the number of lines is multiplied. In three years we plan (Reganomics willing) to triple the number of computers from 30 to 90 and double the number of users from 700 to 1400. And we probably should have planned for 2800 users. Without a solution that grows easily and dynamically, we are going to be strangled by the inertia of the wire and switches and our inability to plan and install them. How can we do this?

SLIDE 33

The users will have recognized the problem and installed a local area network long before any planner. It won't be part of a grand plan that I as head of the organization have to legislate or even worry about. With an Ethernet, direct connection is made between all user terminals via terminal concentrators and the myriad of computers.

SLIDE 34

Ethernet solves a number of problems. By solving the computer to computer interconnection problem, the user interconnection problem is resolved. Any mainframe, minicomputer, or personal computer can access the high-speed network while it is in operation. At 10 million bits per second, users don't complain because the connections are 100 times faster than direct wiring and 1000 times faster than telepone lines.

The biggest gain is open-ended network growth. Direct cable access to the network, often directly by the users, allows adding equipment while the system is in operation. No additional computers or wiring are needed. In many cases the users will have installed their own networks or network segments, as simply as checking out pencils from office supplies so that they can build their own networks by making their own connection.

In this way the network can evolve on need rather than being limited by some planner's limited view of the future or some salesman's ability to get the wrong equipment into a site.

Detailed planning is one of the hardest jobs in evolving and changing organizations, whether it's adding a new department or product line, or whether people are just moving their desks every day. In many organizations planning is done Russian style: a highly centralized top-down affair that includes the range from a new building to a box of pencils. For the dynamic growth and change that can be expected for computing, centralized planning often creates more problems than it solves.

Ethernet technology solves the problem of the dynamic change, allowing tradeoffs in the number and kind of connections, the number of terminals, the number of computers on a day to day basis. The intermediary planners and doer organizations aren't needed: everyone is free to get more work done.

The result: higher productivity by eliminating a function and the interface to that function. Workers can just do the work without begging and negotiating to do work.

SLIDE 35

The last story.

Interconnecting numerous species of computers is somewhat different than connecting terminals or personal computers to shared computers.

SLIDE 36

Again, I would like to turn to a homely example. Our Engineering Network at Digital includes over 200 computer systems serving several thousand terminal users. It looks like a bunch of interconnected links and nodes.

SLIDE 37

But a network is more than just lines and nodes despite the fact that I've been trying to show how simple one can be.

Higher level protocols are needed to support the interconnection of dissimiliar computers, to implement complex network functions such as file and data transfer of all types and terminal-to-terminal communications, and to provide network management.

The protocols are complex. But they are a prerequisite for building a network that includes different computer systems. That's why it is critical that local area network communications are completely compatible with high-level networking protocols.

For the Ethernet Standard, we chose the Open Systems Architecture of the International Standards Organization. In addition, our own DECnet architecture is compatible with this standard.

SLIDE 38

Digital's Engineering Network has over 200 computers in 10 different locations.

SLIDE 39

There are sites in Massachusetts, New Hampshire, Colorado, New Mexico and England connected by special 52,000 bit per second lines and satellite links.

SLIDE 40

The number of sites is increasing more slowly, while the number of computers at each site is increasing very rapidly, and their rate of increase will accelerate as personal computers replace the simple terminals.

At least 80 percent of all network traffic is local traffic and that percentage will increase.

SLIDE 41

Nine links tie the 30 Computers at the Spitbrook site to other network sites.

Notice that what we are trying to achieve is full interconnectivity on a democratic, non-hierarchical basis. If we did this by running wires, 435 wires would be required to interconnect the 30 computers.

SLIDE 42

With 90 computers, 4005 links would be required for total interconnectivity. Also, over 8000 terminating controllers would be required. As you can see, interconnecting these computers on a point-to-point basis results in a topology that's so complex (not to mention so expensive) that it's bound to be ineffective and undesireable.

SLIDE 43

Now see what happens when we install Ethernet. Only one wire and only one terminating control unit is needed per machine. And anyone can make the connection to the cable at any time. Everything is interconnected in a very simple and orderly way. We now have an understandable and workable structure that will provide a number of benefits.

SLIDE 44

Ethernet not only solves the connection problem, but also provides four additional benefits.

One. Systems can be connected and disconnected while the network is in operation.

Two. Communications are a thousand times faster than via direct wire or phone line. Radically new use and applications will follow.

Three. Although costs are reduced we're also getting more computing for each dollar by reducing the switch load on computing nodes. If you look closely at our current network, it turns out that many nodes are primarily switching computers. Ethernet will eliminate the need to use computers as switches. In this way the computers that are doing an overhead function switching messages for their friends can go back to real computing and have fun too. Everyone's productivity is raised.

Four. The last point is the most important one. We can't have orderly open-ended growth without having a structure. With Ethernet there is only one connection per node. In traditional network structures there are many connections and equipment must be provided to switch messages. Ethernet provides a fully distributed switch without the pain and limitations of intensive and erroneous planning.

SLIDE 45

It's also been shown that Ethernet works with a variety of computers. In May of last year Digital, Xerox, and Intel had an Ethernet running at the National Computer Conference. Since each of these companies followed the same standard we were able to transfer print files and send messages back and forth between the Digital, Xerox, and Intel booths.

SLIDE 46

Ethernet is installed in our Central Engineering Department where we're in transition from the data switch to switching concentrators on Ethernet. Here, we also see three generations of switches: the telephone, the data switch and Ethernet.

SLIDE 47

A VAX computer connected to Ethernet.

SLIDE 48

A terminal concentrator manufacturered by one of our competitors is plugged into our Ethernet.

SLIDE 49

Finally I'd like to show you an ad produced by another manufacturer. Note how they feature Ethernet, and listen to what they have to say. Let me read.

"Ethernet...gives you instantaneous access to all resources on the network, such as files, printers, other I/O devices -- even other mainframes -- plus all the speed of a dedicated single-user computer.

"In real terms, what this means is this. Instead of taking as long as 44 seconds to transmit ten pages of data, the transfer takes place in .042 second. In the 4.4 seconds it would take a conventional network to send one page of **War and Peace**, with ... Ethernet, you could send the entire 1000 page novel."

All the people in Xerox's Advertising Department couldn't say it any better. Neither could ours or Intel's. With this performance, with the ease with which you can connect systems to Ethernet, and with the number of different manufacturers lining up behind the Ethernet standard, you're going to see a growing interest in local-area networks.

SLIDE 50

Ethernet provides the needed structure for the Fifth Generation of computers.

It provides for many current needs. The actual use is likely to be quite different.

We use Ethernet and are committed to Ethernet.

Ethernet conforms to the Open Systems Architecture of the International standards Organization, and we believe that because of it's simplicity Ethernet will become the Local Area Network standard.

Digital will certainly be introducing products within the next few months.

Moreover for the future...

SLIDE 51

Since we believe Ethernet is the UNIBUS of the fifth generation,

SLIDE 52

We, therefore, believe Ethernet is the unifying key to the 5th computer generation because it is the right standard to interconnect computers and for Open Local Area Networks.

Unifying Key to the 5th Generation

Gordon Bell
Vice President, Engineering
Digital Equipment Corporation
2/7/82 Sun 17:12:26

ey In the Fifth Computer Generation, a wide variety of computers will communicate with one another. No one argues about this. All the shouting is about how to do it and what form the computers will take.

A standard communications language is the key. I believe Ethernet is this unifying key to the 5th computer generation because it interconnects all sizes and types of computers in a passive, tightly-coupled, high performance fashion, permiting the formation of local-area networks. Ethernet is the standard that can hush the argument and let everyone get to work on the computing nodes.

Standardization is necessary because no one vendor has it all, or can provide the full spectrum of information processing nodes that are emerging. Most organizations have computers built by different vendors. Although computer data and processes (that is the work) are interdependent, no easy and inexpensive way to send data among machines exists. Everyone's customers are demanding a network standard. Ethernet can do it for everyone.

I'm going to tell you four stories that illustrate the different facets of Ethernet. The first is about the Unibus and why I think Ethernet is the Unibus of the Fifth Generation.

In 1970 Digital introduced the Unibus to interconnect parts of a computer. The Unibus is just a simple ribbon-like cable with 56 conductors as shown in this old ad. With Unibus people could easily assemble their own computers and did so in many different ways, and it became a standard.

Virtually all computers built today utilize a Unibus-type architecture, including Intel's Multibus, and Motorola's Versabus. Both of these busses are standards too.

This bus is a high-speed data path that links all system components within a single computer. The processor. Primary memory. Secondary disk memory. Communications interfaces. Realtime equipment interfaces. Interfaces to special customer equipment.

The complete Unibus specification is contained in a manual about 1/2 inch thick, roughly the size of the Ethernet blue book specification. From this, users have designed 10's of thousands of machines to match the computer to their application in an almost open-ended fashion. Small dedicated controllers, personal computers, pedagogical machines and large timeshared computers are all built this way. Any kind of computer can be built easily from a common set of components.

What started as a good scheme for interconnecting components that Digital supplied, became a lovely standard for starting a whole plug-compatible business. The unexpected result, an industry with 100's of vendors and lots of new competitors. The plug compatible parts mean lower prices. The non-mundane user designed connections to television cameras, robots and other devices act to stimulate the whole next computer generation, based on need.

Ethernet is only an extended unifying bus, like Unibus, that interconnects many computer based information processing systems but in a 2.5 by 2 kilometer area.

Unibus has a single processor for one computer, Ethernet can support many different computers in all sizes and places doing all types of work.

A Unibus system has local data storage, an Ethernet supports databases distributed throughout the network. The later has evolved to be called the file server.

A Unibus system interfaces to other computers via slow, communication links and tightly coupled parallel links, an Ethernet always interfaces to other computers directly. Components that are not computers are just not built today. Interconnection occus directly and via special computers called gateways.

e Ethernets couple host processors, people using their own special terminals, personal computers and workstations, as well as particular functions like file servers, print servers, communications servers, and realtime equipment in the laboratory and factory.

Gateways to other computers and networks can be provided by these communications servers.

Computer systems decomposed into seperate, functional units on an Ethernet, will be significantly easier to build.

Then users will participate more than ever in the design and building of their own systems and not be limited by the vision of a single supplier.

In the Fifth Generation, every computer on the Ethernet, will be both contributing to and sharing in the total resources of the network. The network will be the system

Having demonstrated that Ethernet is the Unibus of the 5th generation because it provides a passive standard to interconnect all sizes and types of computers into a high speed network, I will turn to the issue of Ethernet's role.

The second story is about the evolution of the computer generations - driven by the semiconductor evolution.

The Fifth Computer generation, like its predecessors, will only occur when there new technologies and needs that converge to create a new computing structure.

Three technologies are fueling the 5th generation: the understanding of how to build a reliable Carrier Sense Multiple Access with Collision Detection (CSMA-CD) type network, in effect the Ether; Very Large Scale Integrated Circuits or VLSI permitting all logic to be computer based, but more importantly permitting a simple, low cost connection to the Ethernet cable, essential for a standard; and finally technologyies such as high resolution graphics that accelerate the creation of computing nodes that are a pleasure to use.

More computer use results in increasing human potential and hence an increasing need or demand. GNP grows with the absorption of new technologies that allow higher productivity. Every person's productivity is limited by the rate computers communicate with one another. In effect, we have evolved the quadruped to a thoroughbred but not changed the track. The only paths that they can travel are muddy, rocky and random time-worn paths. We need a fast race track.

Our computers often wait at the gate while users physically carry data between them in what is becoming an inverted society the computers do the fun thinking parts and the users carry trivia from machine to machine, or become simple machine to people translators. Ethernet breaks this communication bottleneck. Furthermore, Ethernets can carry voice, graphs and pictures as well as simple messages and data files. They'll resutructure use. It won't be a straightforward extrapolation of simple terminal to computer and computer to computer networking we know today.

In 1990, we can probably look back and identify trends that are not clear today. So I won't speculate about 1990, but I know the future will be more interesting than the simplistic, evolutionary view I'm presenting today.

The development can only happen if we provide the creative environment in which to invent. I think the Ethernet based open Local Area Network is this environment. "A local-area network is a set of information processing nodes, distributed in a single area and fully interconnected via high-speed data links." An open local area network is one in which any vendor or user can supply nodes for the network.

The user should be able to communicate over a local area network with the same nonchalance as the telephone, not knowing or caring how the network works or how the message is transmitted.

It's amazing that the front end user portion of the telephone and the computer really haven't changed much. The oscilloscope of the Whirlwind, the first real time interactive computer built in 1950 is just a bit bigger and more graphic than the ones on computers today. Jay Forrester and his associates used it as a personal computer. The user walked into a building that was the computer and into a room that was the console and sat down at the cathode ray tube and the computer spent most of its time waiting for the user to interact. This wasn't the best use of the world's only interactive personal computer.

Other early machines, such as the first one, EDSAC built by Maurice Wilkes in Cambridge, England, sought to be more efficient by keeping the users away from the machine. The programmers worked off-line and then handed programs on paper tape to people who put them on a clothes line and eventually fed them into the computer. This maximized the machine's use.

But isolated users quickly grew to hate and to be intimidated by the batch computers. People would prepare their programs on punch cards, submit them to a clerk and the program would be put in the queue. As often as not, errors were found in the program or data so instead of getting an answer to an immediate business problem the user had to rekey his program and and go back to the end of the line. It's no wonder that users wanted a different way of doing things.

With the introduction of transistor technology, computers started to get smaller. In 1960, Digital introduced the PDP-1, the first commercial computer with an interactive video display that played Space War, the grand daddy of all computer space games. In 1961, two typewriters were connected to a PDP-1 at Bolt, Bernanek and Newman and the timesharing idea was born.

In 1963, Just two years after the first experiment, Digital introduced the first commercial timesharing system, PDP-6, for 8 to 16 users.

Then the computer's time, wasted waiting for one user, was used by another. Throughout the sixties, the evolution of batch, personal and time-shared computers continued. Batch mainframes were developed with remote job entry terminals so a few lucky users could enter data from their offices.

Minicomputers, like the PDP-8, were small and inexpensive enough so they could be dedicated to particular applications. Many of these minicomputers were used to prepare data for batch processing on a mainframe.

Other mainframes became specialized timesharing machines. But computing was still very expensive and impersonal.

11 The real breakthrough came in 1972 when we learned how to provide timesharing on a minicomputer. For the first time, low cost, interactive, **personal** computing capabilities could be provided at a cost that most users could justify. Computers came out of the computer rooms and started with users.

After the initial honeymoon, a need developed to interconnect the machines to each other and to the large batch machines which by now could be controlled from terminals. As a result, engineers did what came naturally and started to string wire between them

In the late 70's the interconnection problem was exacerbated by the baby computer boom, known as personal computers.

Like children everyone wants a limited number for their very own.

Personal computers give that one on one realtionship. There's no longer anyone watching you work, not even an accounting program. You can do your own thing in a non-threating way. No one need know if you use the machine or even if you turn it on... or it turns you on.

But then there are times that you and your personal computer want to be connected with another machine to get programs, transmit messages, look at a picture, or send a non-intrusive voicegram message.

And so many more wires have to somehow be added between the centralized, shared remote batch mainframe; the departmental timeshared minicomputers; and the individual personal computers. If there aren't lines running between all the machines then there probably should be; otherwise, information on one node needed elsewhere and has to be re-entered.

Ethernet will provide the structure needed to manage distributed computing. It's coherent structure is capable is of handling an ever-growing volume of traffic among all machines.

The last two stories address Ethernet user needs.

First, they provide high-speed interconnection among dispersed computers. (Creative programmers are kept happy and work effectively when connected to high speed systems. They want to be able to call all the machines in their network and communicate with others in the nework independent of where they are. When we're working with a machine, we have less patience than a 2 year old waiting for a cookie.

Second, Ethernets provide simple interconnections of terminals and personal computers to host processors. New users starting with simple personal computers will be able to improve their performance by accessing larger machines as their needs increase. Clearly history has shown that the more computer power anyone has the more he wants. It is an insatiable hunger like none known before with the immediate reward of greater individual productivity.

Third, Ethernets interconnect all kinds of computer controlled equipment. For example, links between computer controlled equipment in the laboratory on the factory floor and data processing equipment in the office.

onal
uter
d

Every organization wants open ended, flexible links between personal computers and terminals and larger, more central computers.

on Today, most users have simple block mode, fixed function terminals. Nearly all of these are evolving into complete personal computer systems.

It makes little difference whether the user has a simple terminal, or a full-fledged personal computer. For simple terminals, high bandwidth is needed for character at a time interaction, and for effective use of personal computers, high bandwidth is needed to transfer messages, files, images and voicegram messages.

Today the typical user is most likely linked to a single host computer, and communication with other computers is through this host.

User demanded local area networks develop by wiring various hosts and terminals together by the users.

The most common answer to the problem is to use telephone lines, putting all terminal traffic onto a telephone system which may not be capable of handling it. Then modems have to be installed to convert the digital signal generated by a terminal or computer to an analog signal that can be carried over a telephone line. The biggest problem is that our users want to communicate at least at 9600 bits per second, and this just can't be done economically with these switches.

A second answer to the problem is the data switch. By installing a switch between the user and the computer network it is possible for any user to connect to any computer.

But connecting terminals to the switch involves a lot of wiring. These diagrams are simple enough to draw, building them is complicated. Furthermore, terminal wiring is a never ending business requiring much planning, resulting in much inflexibility and all fueled with much money.

So even if -- at first glance -- both telephone lines and the data switch look like solutions they aren't. They're part of the problem as anyone who has many terminals and computers will tell you.

I know personally, because Digital continually faces this problem like our facility in Nashua New Hampshire where we have 30 computers and 700 user terminals.

A pristine view shows a number of computers and a big room used for switching the links between terminals and different computers.

You don't see the problem until you open the door. Every terminal line is wired to a board in this room And every time an unplanned terminal is added someone is called to run more wires.

Wires are run from board to the wireroom to a switch computer.

o

This switch computer is now bound and it doesn't grow very gracefully, particularly when the number of lines is multiplied. In three years we plan, Reganomics willing, to triple the number of computers from 30 to 90 and double the number of users from 700 to 1400. And we probably should have planned for 2800 users. Without a solution that grows easily and dynamically, we are going to be strangled by the inertia of the wire and switches and our inability to plan and istall them How can we do this?

The users will have recognized the problem and installed a local area network long before any planner. It won't be part of a grand plan that I as head of the organization have to legislate or even worry about. With an Ethernet direct connection is made between all user terminals via terminal concentrators and the myriad of computers.

the
nal)
em

Ethernet solves a number of problems. By solving the computer to computer interconnection problem, the computer interconnection problem is resolved. Any mainframe, minicomputer, or personal computer can access the high-speed network while it is in operation. At 10 million bits per second, users don't complain because the connections are 100 times faster than direct wiring and 1000 times faster than telepone lines.

The biggest gain is open-ended network growth. Direct cable access to the network, often directly by the users allows adding equipment while the system is in operation. No additional computers or wiring are needed. In many cases the users will have installed their own networks or network segments, as simply as checking out pencils from office supplies so that the can build their own networks by making their own connection.

In this way the network can evolve on need rather than being limited by some planner's limited view of the future or some salesman's ability to get the wrong equipment into a site.

Detailed planning is one of the hardest jobs in evolving and changing organizations. Whether it's adding a new department or product line, or whether people are just moving their desks every day. In many organizations planning is done Russian style: a highly centralized top-down affair that includes the range from a new building to a box of pencils. For the dynamic growth and change that can be expected for computing, centralized planning often creates more problems than it solves.

Ethernet technology solves the problem of the dynamic change, allowing tradeoffs in the number and kind of connections, the number of terminals, the number of computers on a day to day basis. The intermediary planners and doer organizations aren't needed: everyone is free to get more work done.

The result: higher productivity by eliminating a function and the interface to that function. Workers can just do the work without begging and negotiating to do work.

The last story.
Interconnecting numerous species of computers is somewhat different than connecting terminals or personal computers to shared computers.

Again, I would like to turn to a homely example. Our Engineering Network at Digital includes over 200 computer systems serving several thousand terminal users. It looks like a bunch of interconnected links and nodes.

ed But a network is more than just lines and nodes despite the fact that I've been trying to show how simple one can be.

Higher level protocols are needed to support the interconnection of dissimiliar computers; to implement complex network functions such as file and data transfer of all types and terminal-to-terminal communications; and to provide network management.

The protocols are complex. But they are a prerequisite for building a network that includes different computer systems. That's why it is critical that local-area network communications are completely compatible with high-level networking protocols.

For the Ethernet Standard, we chose the Open Systems Architecture of the International Standards Organization. In addition, our own DECnet architecture is compatible with this standard.

Digital's Engineering Network has over 200 computers in 10 different locations.

There are sites in Massachusetts, New Hampshire, Colorado, New Mexico and England connected by special 52,000 bit per second lines and satellite links.

The number of sites is increasing slowly, while the number of computers at each site sites is increasing very rapidly and their rate of increase will accelerate as personal computers replace the simple terminals.

At least 80 percent of all network traffic is local traffic and that percentage will increase.

s Nine links tie the 30 Computers at the Spitbrook site to other network sites.

Notice that what we are trying to achieve is full interconnectivity on a democratci, non-hierarchical basis. If we did this by running wires, 435 wires would be required to interconnect the 30 computers.

es With 90 computers, 4005 links would be required for total interconnectivity. Also, over 8000 terminating controllers would be required. As you can see, interconnecting these computers on a point-to-point basis results in a topology that's so complex-not to mention so expensive-that it's bound to be ineffective and undesireable.

Now see what happens when we install Ethernet. Only one wire and only one terminating control unit is needed per machine. And anyone can make the connection to the cable at any time. Everything is interconnected in a very simple and orderly way. We now have an understandable and workable structure that will provide a number of benefits.

the
nnect

Ethernet not only solves the connection problem, but provides four additional benefits.

One. Systems can be connected and disconnected while the network is in operation.

Two. Communications are a thousand times faster than via direct wire or phone line. Radically new use and applications will follow.

Three. Although costs are reduced we're also getting more computing for each dollar by reducing the switch load on computing nodes. If you look closely at our current network, it turns out that many nodes are primarily switching computers. Ethernet will eliminate the need to use computers as switches. In this way the computers , who are doing an overhead function switching messages for their friends can go back to real computing and have fun too. Everyone's productivity is raised.

Four. The last point is the most important one. We can't have orderly open-ended growth without having a structure. With Ethernet there is only one connection per node. In traditional network structures there are many connections and equipment must be provided to switch messages. Ethernet provides a fully distributed switch without the pain and limitations of intensive and erroneous planning.

It's also been shown that Ethernet works with a variety of computers. In May of last year Digital, Xerox, and Intel and had an Ethernet running at the National Computer Conference. Since each of these companies followed the same standard we were able to transfer print files and send messages back and forth between the Digital, Xerox, and Intel booths.

Ethernet installed in our Central Engineering Department where we're in transition from the data switch to switching concentrators on Ethernet. Here, we also see three generations of switches: the telephone, the data switch and Ethernet.

A VAX computer connected to Ethernet.

A terminal concentrator manufacturered by one of our competitors is plugged into our Ethernet.

Finally I'd like to show you an ad produced by another manufacturer Note how they feature Ethernet and listen to what they have to say. Let me read.

"Ethernet...gives you instantaneous access to all resources on the network, such as files, printers, other I/O devices -- even other mainframes -- plus all the speed of a dedicated single-user computer.

"In real terms, what this means is this. Instead of taking as long as 44 seconds to transmit ten pages of data, the transfer takes place in .042 second. In the 4.4 seconds it would take a conventional network to send one page of **War and Peace**, with PERQ and Ethernet, you could send the entire 1000 page novel."

All the people in Xerox's Advertising Department couldn't say it any better. Neither could ours or Intel's. With this performance and with the ease with which you can connect systems to Ethernet, and with the number of different manufacturers lining up behind the Ethernet standard you're going to see a growing interest in local-area networks.

f

Ethernet provides the needed structure for the Fifth Generation of computers.

It provides for many current needs. The actual use is likely to be quite different.

We use Ethernet and are committed to Ethernet.

Ethernet conforms to the Open Systems Architecture of the International standards Organization, and we believe that because of it's simplicity Ethernet will become the Local Area Network standard.

Digital will certainly be introducing products within the next few months.

Moreover for the future...

Since we believe Ethernet is the UNIBUS of the fifth generation,

We, therefore, believe Ethernet is the unifying key
to the 5th computer generation because it is the right standard to
interconnect computers and for Open, Local Area Networks.

00 BURT DECGRAM ACCEPTED S 9630 O 75 14-NOV-81 17:51:00

****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION
cc: see "CC" DISTRIBUTION

DATE: SAT 14 NOV 1981 5:48 PM EST
FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: DISCUSSION WITH ICL PRES. WILMOT ON USING ETHERNET

Just finished talking (4:30 EST, Saturday) to Wilmot (011-44-1-949-5903 London, Telex at Putney England) regarding their standardization of Ethernet. They are being driven by their customers (eg. Citibank, Barclay's) to do this because of the proliferation of minis and because of the Xerox PR pressure. He commented: "Everyone is buying it but no one has the slightest idea as to what it is. There is incredible commercial support."

They are collaborating with 3 Rivers via building the PERQs and with MITEL on PABX's; ALL are committed to use Ethernet and will be defining a product there in the next few months. He said: "They have looked at the Xerox Level 3 and 4 protocols and promptly went out and got drunk! These protocols are like SNA squared." They like Ethernet and want to support it because of the commercial momentum, the Level 1 and 2 standard and the need, but are really saddened with the technical merits of the Xerox work. As British engineers, they demand technical elegance. They have also been working with the Ungermann Bass boards and wil probably start using them. They want the following:

1. Use EN for connecting between ICL (largest European Computer Co.) and DEC (for minis), MITEL (for PABX) and 3 Rivers (their professional workstation). Have the standard actually ECMA based (their customer base) which is a superset of OSI and x.25, but goes up to the applications level.

2. Get all these folks together to get a standard. Also get the customers in to review it.

He suggested a massive technical effort to get these standards set in the next few months with persons from DEC, Mitel, 3 Rivers, DEC, some customers, etc. I said we do not do this sort of thing! I did suggest we have a technical exchange as to where we are and plan to be. The goal would be to see if we have any basis for doing any task force work in this regard.

I stated our position: We are evolving DECnet to be used with Ethernet as a low level transport and we are evolving DECnet to converge with OSI and x.25 as they become standard. We are providing gateways to the other networks such as SNA, x.25 and Xerox servers as necessary.

WE HAD AMAZING CONCURRENCE:
1. Technical standards are becoming a major market stategy and force. We are not moving aggressively enough in recognizing this and acting on it.

2. If we don't hurry, it's all academic. We'll be implementing a variant of SNA!

THE NEXT STEP: Bernie, please send him a Telex inviting them to give a technical presentation here in exchange for our presenting the protocols we plan to use on EN. I would like to have Xerox and Intel be part of the same meeting. The goal would be to get this done this next week.

"TO" DISTRIBUTION:

SAM FULLER	BERNIE LACROUTE	TONY LAUCK
BILL STRECKER		

"CC" DISTRIBUTION:

BILL DEMMER	MARY JANE FORBES	GVPC:
WIN HINDLE	BILL KIESEWETTER	DAVE RODGERS

GB3.S2.39
Unifying Key to the 5th Generation

Gordon Bell
Vice President, Engineering
Digital Equipment Corporation

of The Fifth Generation

In 1970 Digital introduced the Unibus to interconnect computer parts together so that a user could create his own computer in an open ended, flexible fashion to match his needs. The Unibus is just a simple ribbon-like cable or printed wires along a backplane with 56 conductors. But most important, it became a standard.

I believe Ethernet is one of the keys to the development of the Fifth Generation of computers because...

Ethernet provides a **standard** to interconnect all sizes and types of computers in a passive, tightly-coupled, high performance fashion, forming a local-area network.

The "standard" must be emphasized because no one vendor has it all, nor is likely to be able to provide the variety of information processing nodes that are possible and will emerge during this generation. Most organizations have computers built by different vendors and the fifth generation requires that these machines communicate with one another. Although computer data and processes (that is the work) is interdependent, there is no easy way to send data among machines today easily and cheaply. Everyone's customers are demanding a standard that will make it easy to interconnect systems built by different manufacturers.

Interconnection standards are not a new idea. The standards make it possible to build computers families that shared common components. Lasting investments can be preserved and people can plan.

One such standard is the Digital Unibus, introduced in 1970 designed to solve the ease of building a computer problem. It's very heart is nothing more than wires. But more important it is a set of conventions, a stanard that defines the way a computer's components can be interconnected.

This bus is a high-speed data path that links all system components within a single computer. The processor. Primary memory. Disk memory. Communications interfaces. Realtime interfaces.

Virtually all computers built today utilize a Unibus-type architecture, including Intel's Multibus, and Motorola's Versabus. Both companies have extended the idea.

Although the simple bus with its 56 wires and several hundred pages of specifications that define it are very straightforward, the results are not obvious, and it is these results that we can learn from. Also, it is why I think we have a perfect analogy.

With the Unibus, we wanted computer users to build any kind of system they wanted by buying our components and by designing and adding components that they built. It worked, they bought the simplicity of the idea and 10's of thousands of machines have equipment that users designed to match the computer to their application in an almost open-ended fashion. Small dedicated controllers, personal computers, pedagogical machines and large timeshared computers are all built this way. It works to allow about any kind of computer to be built easily from a common set of components.

Unfortunately, there was a side-effect that we view is a double edged sword. What was a good scheme for interconnecting components that we supplied, was a lovely standard for starting a whole plug-compatible business. The result, an industry with 100's of vendors many of which compete with us.

Everyone who has this type of bus experiences the same phenomenon! Users love it because it is so easy to build the computer they want, and a whole competitive, plug-compatible industry forms supplying everything from plug-compatible memory to television and robot interfaces. The later types, stimulated by the users, really fuels the new generation.

Ethernet is an extended bus that interconnects many information processing systems in many locations. But the basic components remain the same, independent of where the connection is made.

A Unibus system has a single processor for one computer, Ethernet can support many different computers in all sizes and places doing all types of work.

A Unibus system has local data storage, an Ethernet supports databases distributed throughout the network within other computer nodes and as special function nodes. The later has evolved to be called the file server.

A Unibus system interfaces to other computers via slow, communication links and tightly coupled parallel links, an Ethernet always interfaces to other computers directly, because we simply do not build components that are not computers any longer. The interconnection is both directly and via special computers we call gateways.

The UNIBUS is limited to 15 meters and is used to build a computer, Ethernet covers an area 2.5 by 5 kilometers and is used to build a high speed network.

Because of the rapid development of semiconductor technology, everything connected to an Ethernet is computer based.

Ethernets link host processors, particular functions like file servers, print servers, and realtime equipment such as laboratory instruments and machine tools with embedded computer intelligence, and finally people coupled via intelligent workstations and personal computers of every type and description.

We'll see gateways to other computers and networks. Thus, it should be significantly easier to build computer systems because they are decomposed into seperate, functional boxes.

Ethernet has the simplicity to provide the standard that allows all equipment to work together, the standard that will allow the evolution of the next computer generation. Computer evolution will not be limited by the vision of a single supplier. The users will participate more than ever in the design and building of their own systems.

In the Fifth generation, all the information processing nodes that is, every terminal, every workstation, and every computer on the Ethernet will be part of a larger system and share the total resources of the network. Thus, the network is truely the system

For example any user on the network -- provided they have the right clearances -- will be able to access data stored anywhere within the network.

The Fifth Computer generation will only occur when there is a convergence of new technologies, needs, permitting a new computing structure to be built and followed by significant use.

Three technologies will fuel the 5th generation: the understanding of how to build a reliable Carrier Sense Multiple Access with Collision Detection

(CSMA-CD) type network, in effect the Ether; Very Large Scale Integrated Circuits or VLSI permitting all logic to be computer based, but more importantly permitting the connection to the Ethernet cable in a trivial and cost-effective fashion that is essential in it becoming a useful standard; and finally other technology such as high resolution graphics to accelerate the building of more worthwhile computing nodes that people will enjoy working with.

A new set of needs have resulted from the growing computer population. GNP growth only occurs from the absorption of new technology giving higher productivity. Now every person involved with computing is limited by the communication with other systems. In effect, we have evolved the quadruped to a thoroughbred and the only paths that they can travel are muddy, rocky and random time-worn paths. We want a fast race track.

Ethernet breaks this communication bottleneck by providing a local area network instead of one where people carry most of the information between machines. In addition, these networks can carry voice, graphs and pictures as well as simple messages and data files. This will totally restructure the applications, going far beyond the simple terminal to computer and computer to computer networking we know today.

This is the final part of the story of a generation. It must be solidly based on a current need, but after the fact, say in 1990, we can look back and find that the actual use and the resulting networks were quite different than we thought. I won't speculate about 1990 but I know it will be radically different from the simplistic, evolutionary view I'm presenting today.

Before we proceed, I'll summarize, "A local-area network is a set of information processing nodes, distributed in a single area and fully interconnected via high-speed data links.

When local networks are installed people will no longer have to function as switches and trivial information carriers. I don't think that the way to become president is by starting as a messenger person. Too many jobs for intelligent men and women are still just as switches, such as interoffice mail carriers or order entry clerks.

The user should be able to communicate over a local area network with the same nonchalance as the telephone, not knowing or caring how the network works or how the message is transmitted.

It's amazing that the front end user potion of the telephone and the computer really havent changed much. The oscilloscope of the Whirlwind, the first real time interactive computer built in 1950 is only a little bigger than the ones on computers today. Jay Forrester and his associates used it as a personal computer -- the problem was that there was only one Whirlwind. The user walked into a building that was the computer and into a room that was the console and sat down at the cathode ray tube and the computer spent most of its time waiting for the user to interact. This wasn't the best use of a scarce resource.

Other early machines, such as the first one, EDSAC built by Maurice Wilkes in Cambridge, England, sought to make this more efficient by having the programmers work off-line and then batch process the work through the machine, by handing programs and data to an operator, thus maximizing machine efficiency.

Computers weren't exactly easy to use. They even looked intimidating as you can see from this picture of an early batch computer.

People would prepare their programs on punch cards or paper tape, submit them to the manager of the facility, and the program would be put in the queue. As often as not, errors were found in the program or data so instead of getting an answer to an immediate business problem the user had to rekey his program and and go back to the end of the line. It's no wonder that users wanted a different way of doing things.

With the introduction of transistor technology, computers started to get smaller. In 1960, Digital Equipment Corporation introduced the PDP-1, the first commercial computer to be sold with an interactive video display that played Space War, the grand daddy of all computer space games. In 1961, two typewriters were connected to a PDP-1 at Bolt, Bernanek and Newman and the idea for timesharing was born.

The computer's time, wasted waiting for one user was put to good use for the other. Just two years after the first experiment, in 1963, Digital introduced the first commercial timesharing system, the PDP-6, for 8 to 16 users.

Throughout the sixties, the evolution of batch, personal and time-shared computers continued. Batch mainframes were developed with remote job entry terminals so a few lucky users could enter data from their offices.

Minicomputers, like the PDP-8, were small and inexpensive enough so they could be dedicated to particular applications. Many of these minicomputers were used to prepare data for batch processing on a mainframe.

Other mainframes became specialized timesharing machines. But computing was still a very expensive and impersonal business.

11 The real breakthrough came in 1972 when we learned how to provide timesharing on a minicomputer. For the first time, interactive, **personal** computing capabilities could be provided at a cost that most users could justify. People got them as another member of their group or department, and computers came out of the computer rooms and started to cohabit.

We saw the proliferation of these computers in a dispersed fashion, and a need developed to interconnect the machines to each other and to the large batch machines which could now be controlled from terminals.

The proliferating personal computers springing from the microprocessor in the mid-70's also exacerbate dispersed computing. The need for communications of the data created and stored at the plethora of machines is clear.

Everyone wants their own computer for the one on one realtionship because it changes the way we think about machines. There's no longer anyone watching you work. You can do your own thing in a non-threating way. No one need know if you use the machine or even if you turn it on... or it turns you on.

Now we have lines running everyway between all three levels of computers: the centralized, shared remote batch mainframe; the departmental timeshared minicomputers; and the individual personal computers. If there aren't lines running between all the machines then there should be because there's information on one node that's needed somewhere else and is likely to be re-entered by a person.

Ethernet will provide the structure needed to manage distributed computing. An Ethernet is a coherent structure capable of handling an ever-growing volume of traffic among all machines.

It can replace incremental hodge podge connections that are difficult and messy to maintain, and besides don't have any systematic logic. (Maybe the cows understood the paths to the Boston Common but I don't and I surely wish someone would have replaced them with the simple Manhattan-type grid.

An Ethernet addresses three critical user needs.

One. for high-speed interconnection among dispersed computers; (Sophisticated programmers are only kept happy and working effectively if they are connected to the appropriate high speed system They want tight coupling with the machine for fast response time and don't want to wait.)

Two. for simple interconnections of terminals and personal computers to host processors; (The new users starting with simple personal computers will be able to improve their performance by accessing the appropriate machines as their needs increase and change. And clearly history has shown that the more computer power anyone has the more he wants. It is an insatiable hunger like none known before, because it can lead to greater and not less productivity.)

Three. for the interconnection of computer controlled equipment with other information processors. In other words, a local-area network can provide a link between computer controlled equipment on the factory floor and data processing equipment in the office.

Finally, there'll be quite a different use. But we'll have to wait 10 years to observe this.

onal
uter
d

The need to link personal computers and terminals to file servers, print servers, and to communication gateways as well as to computers systems on the network is very real. Today's terminals are connected to Ethernet in clusters via communications concentrators.

on Today, most users have simple terminals. But a rapidly growing number have smart terminals and nearly all of these terminals are evolving into complete personal computer systems.

It makes little difference whether the user has a simple terminal, a smart terminal, or a full-fledged personal computer. For simple terminals bandwidth is needed for character at a time interaction, and for powerful personal computers bandwidth is needed to transfer files, images and voicegrams.

Today the typical user is most likely linked to a single computer. The only way he can communicate with other computers is through this host.

What we have is a sort of local-area network based on pieces of wire strung from here to there.

The issue can be illustrated by a case study of the local-area networking problem at Digital's Spitbrook, New Hampshire facility where we have 30 computers and 700 user terminals. By 1985 we expect a population of 90 computers and 1400 users.

One answer to the problem is the data switch. By installing a switch between the user and the computer network it is possible for any user to connect to any computer.

But connecting terminals to the switch involves a lot of wiring. The diagram was simple enough to draw, building it was complicated.

Another alternative is to use telephone lines, putting all terminal traffic onto your telephone system Then modems have to be installed to convert the digital signal generated by a terminal or computer to an analog signal that can be carried over a telephone line.

And you have to remember that telephone wiring is a never ending business.

So even if -- at first glance -- both telephone lines and the data switch look like solutions they aren't. They're part of the problem as anyone who has a lot of terminals and computers will tell you.

Ethernet minimizes this interconnect complexity.

A pristine view shows a number of computers and a big room used for switching the links between
terminals and different computers.

You don't see the problem until you open the door to the room Here it is. Every terminal line is wired to a board in this room And everytime you add a new terminal you have to run more wires.

The wires run from board to the wireroom and then to a switch computer.

Here's the front of that switching computer. The problem is that this computer is really bound and it doesn't grow very gracefully, particularly if you are going to double the number of lines. We have 700 users now. We'll soon have 1400 and we probably should be planning for 2800. We're trying to solve a dynamic problem that is, we want to interconnect a growing number of users, and at the same time, we want to simplify the interconnection of new terminals. How can we do this?

on

hernet

We install an Ethernet. With Ethernet we can make a direct connection between all user terminals via terminal concentrators and all shared and special function computers.

the
nal)
em

We solve two problems at once. By solving the computer to computer interconnection problem we also solve the terminal to computer interconnection problem. With Ethernet, any mainframe, any minicomputer, any personal computer can access the network. And we can add to the network while it is in operation. This is a high-speed network that can handle 10 million bits per second, instead of 9600 bits per second when directly connected or 1200 bits per second limited by telephone lines. Many interconnection systems used today are limited to 300 bits per second. Hard to believe in the 1980's.

The biggest thing that we gain is open-ended network growth. We have direct cable access to the network without extra equipment so that one can avoid all the inefficiencies that come with planning large facilites.

I don't know about you, but I feel that one of our hardest jobs in an evolving and changing organization whether it's adding a new department or product line, or whether people are just moving their desks every day is simply the problem of planning, I don't think people plan very well. In most organizations planning is done Russian style around a capital budget down to the pencil. This is a highly centralized top-down affair that often creates more problems than it solves. Five year plans permeate networking. It's too bad because I don't beieve we have the understanding about machines and their useage to look that far. What we need is the ability to interconnect any machine to any other one and the whole problem of design and caring for networks is minimized. In this way the network can evolve on need rather than being limited by some planner's limited view of the future.

Technology can solve the problem of the massive organization. Finally, technology allows tradeoffs on a day to day basis in the number and kind of connections, the number of terminals, the number of computers without the need of intermediary organizations.

The problem of interconnecting the growing number of all kinds of computers is different than the terminal-to-computer interconnection problem.

Again, I would like to turn to a homely example. Our Engineering Network at Digital Equipment Corporation includes over 200 computer systems serving about eight thousand terminal users.

The computers are concentrated in about a dozen sites. At first glance this network, like most networks, is nothing more than a bunch of links and nodes.

ed

It is important that we understand that a network is more than just lines and nodes despite the fact that I've been trying to convince us all how simple it is.

Higher level protocols are needed to support the interconnection of dissimiliar computers; to implement complex network functions such as file and data transfer of all types and terminal-to-terminal communications; and to provide network management.

The protocols are complex. But they are a prerequisite for building a network that includes different computer systems. That's why it is critical that local-area network communications are completely compatible with high-level networking protocols.

We chose the Open Systems Architecture of the International Standards Organization for our standard. In addition, our own DECnet architecture is compatible with this standard.

A closer look at Digital's Engineering Network shows over 200 computers in 13 different locations.

Nine of the sites are in Massachusetts. Two are in New Hampshire. One is in Colorado Springs. One in England. Traffic between these sites is carried by satelite and by dedicated 52,000 bit-per-second lines.

The number of sites is not increasing at a particularly high rate but the number of computrers at each site sites is increasing very rapidly and that rate of increase will accelerate as personal computers replace the terminals.

At least 80 percent of all network traffic is local traffic and that percentage will increase.

s

Focussing on the site at Spitbrook, there are 30 computers and nine links to other network sites. It looks pretty complicated.

Notice that what we are trying to achieve is full interconnectivity. If we did this by running wires, 435 wires would be required to interconnect the 30 computers and 4005 when we install 90 machines. Also, over 8000 terminating controllers would be required. Ethernet is a simple way to do it with one wire and only one terminating control unit per machine. And anyone can make the connection to the cable at any time.

And it's going to get a lot more complicated. Within five years when we have 90 computers, the interconnect will be far more complex. As you can see, interconnecting these computers on a point-to-point basis results in a topology that's so complex-not to mention so expensive-tht its bound to be ineffective and undesireable.

Now see what happens when we install Ethernet. Everything is interconnected in a very simple and orderly way. We now have an understandable and workable structure that will provide a number of benefits.

the

nnect

Ethernet solves the connection problem four ways.

One. Systems can be connected and disconnected while the network is in operation.

Two. Communications are a thousand times faster than we have today. Radically new use and applications will follow.

Three. Although costs are reduced we're also getting more computing for each dollar by reducing the load on computing nodes. Let me explain -- if you look closely at our current network, it turns out that many nodes are primarily switching nodes. Ethernet will eliminate the need to use computers as switches.

Four. The last point is the most important one. We can't have orderly open-ended growth without having a structure. With Ethernet there is only one connection per node. In traditional network structures there are many connections and equipment must be provided to switch messages. Ethernet provides a fully distributed switch without the pain of intensive planning.

It's also been shown that Ethernet works with a variety of computers. In May of last year Digital, Xerox, and Intel announced the Ethernet Specification and had an Ethernet running at the National Computer Conference. Since each of these companies followed the same standard we were able to transfer print files and send messages back and forth between the Digital, Xerox, and Intel booths.

Ethernet installed in our Central Engineering Department.

A VAX computer connected to Ethernet.

A terminal concentrator manufacturered by one of our competitors is plugged into our Ethernet.

Finally I'd like to show you an ad produced by another competitor. Note how they feature Ethernet and listen to what they have to say. Let me read.

"Ethernet...gives you instantaneous access to all resources on the network, such as files, printers, other I/O devices -- even other mainframes -- plus all the speed of a dedicated single-user computer.

"In real terms, what this means is this. Instead of taking as long as 44 seconds to transmit ten pages of data, the transfer takes place in .042 second. In the 4.4 seconds it would take a conventional network to send one page of **War and Peace**, with PERQ and Ethernet, you could send the entire 1000 page novel."

Xerox couldn't say it any better. Neither could Digital or Intel. With this performance and with the ease with which you can connect systems to Ethernet, and with the number of different manufacturers lining up behind the Ethernet standard you're going to see a growing interest in local-area networks.

f

Ethernet provides the needed structure for the Fifth Generation of computers.

It provides for many current needs. The actual use is likely to be quite different.

We use Ethernet and are committed to Ethernet.

Ethernet conforms to the Open Systems Architecture of the International standards Organization, and we believe that because of it's simplicity Ethernet will become the Local Area Network standard.

Digital will certainly be introducing products within the next few months.

Moreover for the future...

We believe Ethernet is the UNIBUS of the fifth generation.

et is going to be the UNIBUS of the fifth generation because it provides a passive standard to interconnect all kinds of computers into tightly coupled high performance networks. It's important that we emphasize the word standard because I don't see that any one vendor has it all and we need ways of being able to communicate with all the various confunctional computers as this next generation evolves.

d UNIBUS, I want to draw a little bit on history here to show how this kind of structure which we introduced in 1970 with the BB11 has really speed up the evolution of computers these last ten years. I really should say UNIBUS type interconnection because virtually all computers introduced today up through minis are structured this way certainly my <?> are built this way. In essence the UNIBUS is really just a piece of cable with 56 wires on it, but what is does is permits anyone any kind of computer option or interface to be connected to it such that one can easily form a very tightly coupled computer in that usually all this is placed in a cabinet. In the case of micros, this is just placed in a backplane, but the usual components of a computer with processor, primary memory, disks, interfaces to other terminals, if it's a time shared computer and certainly the communication to other computers by a phone line, and then for real time computing interfaces to the various real time equipment. In essence the importance of the UNIBUS was its simplicity for forming computers both at the manufactured side and at the users side anyone could help in building the machine and this is why there are so many vendors for equipment that's organized in this way whether it's a UNIBUS, a MULTIBUS or an S100 bus or any of the other four or five standards that exist. I think this is the absolute ideal analogy to what we're providing with Ethernet, and we expect computer networks to form just as a computer evolves, or has evolved so rapidly these last ten years.

rtance of Ethernet is that it is all passive, there is no inherent investment in a central facility or inherent investment in a controller, in essence it doesn't have to be managed, an Ethernet can grow from the bottom up, it can grow by havng a dozen independent Ethernets formed independently and then finally connected together if one wants to. Whereas the UNIBUS was limited to 15 meters, the ethernet goes over two stanza radius of 2.5 kilometers. Given the VLSI drive of the fifth generation, we note that everything connected to the Ethernet will be basically a computer, we'll see our host processors, we see file computers, we see work stations, that is computers tightly coupled computer terminals, we see gateways to other computers, we see real time equipment where virtually all real time equipment at this point has an embedded computer. So this standard will let this evolution take place very rapidly. And now, I think the important thing is that in the fifth generation as were talking about here, the network becomes the system. Every note on that every terminal, every system, every comptuer is part of the total system, and we we'll see much wider use.

, I've talked about the fifth generation let me say what I believe the fifth generation is going to look like. A generation generally is a convergence of technology and need permitting a structure to be built, and then finally after the generation is over you can look at the use. This has been true over the last four generations. So for technology we have Ethernet as one of the components, we have VLSI for higher density, for more powerful comptuers, and highler denisty memories, and then we have a number of other components which I'll not speak to today. Need is certainly being driven because we have so much information we want to interchange, we have networks, every network, basically thats formed evolves incredibly rapidly and every network that I see is out of interconnect speed. Every site that has a number of dispersed computers, whether they be mainframes, whether they be minis, whether they be personal computers, whether they be control computers, are in desperate need of communication with their fellow computers unless, of course, we want an inverted society where all the comptuers are seperate and we have people just carriying information from computer to computer like robots. But I think there may be better things for people to do. This structure that we believe will form from that is this thing we call the local area network, and the ultimate use of such a thing I don't think we can totally predict at this point, I think it is only observable after the generation is formed, but certainly we expect pictures, graphics voice to be communicated as well as files and printing images and terminal traffic and other computer to computer traffic.

let me give my definition of a local area network. Basically it's a high speed data network limited to a local geography or site and permitting direct intercommunication among all of the information processing nodes that connect to it. I think it's important, again to reiterate that computing has evoled to require complete interconnectivity in order to avoid this problem of the people becoming the switching network as well as our needs for transmitting images among machines. In this next series I'm going to take you thorugh a very brief tour of computer history and show computer evolution and show how I think what the effect has been in the early 50's machine, which is really the first generation, we evolved rapidly to the notion of batch processing because computers were so expensive and people would put their cards in hopper and be read and data would be printed in a batch like basis. At the same time I think it's important to note that large number of computers were used interactively by their users. There were in the late fifties, one could sign up for different machines. Interactive computing began with computers and it was really only for efficiency that batch was introduced because of cost. If you looked at, here's a typical 650 operation, in the late fifties, has a ramac <sp> **(find out what computer it is. Not sure it it's a 650).** At the same time the background that I'm familiar with was the whirlwind computer which was by any standards a personal computer, it's just that the first personal computer that you walked into because it was housed in a complete building that we're not looking at the computer here at all, were looking at just a console, on that computer, the cathode ray tube was in the light pin <?> and the first operating system were all born.

d and our first computer which was introduced in 1960, of course, was a personal computer and interactive, one person at a time used it, it had a CRT and its lightpen if you want to see it playing space war, it's available today in the Digital Computer Museum in Marlboro, MA. In the 60's because of the expense of that personal computer that we introduced, we then introduced the notion of time sharing on the PDP6. At the same time the classical mainframes evoled in a couple of directions, they wanted more efficiency and they did that be introducing an offline computer, for getting their preparing cards and printer and toward late part of the 60's the notion of remote job entry was introduced where the cards were so great that everybody wanted them in their own place so one could have remote job entry and in a few cases toward the last part of the 60's a lucky few could have a terminal to prepare their own cards on and edit and submit them directly.

icture of the first commercial time sharing sytsem which is really just a console of it which we introduced for personal computing. In the 70's really time sharing got going mean while the batch mainframe began to work alot better and it worked so well that we started doing time sharing on the time shared on our mini computers, and they time share just as well as mainframes, and in essence that let's computerers be brought in for a group or a department or a small number of users instead of having it be brought in for a collection or whole organization where it has to be run for the user by someone else. Note in these cases the users are connected by the terminal, very simple terminals,directly to the computer, and in fact toward the last part of the 70's or the mid part of the 70's we introduced DECnet because we have so many minicomputers and people wanted acess from machine to machine and to the mainframe because there was so much information being dispersed that it really had to be fed forward to the classical mainframe and this is a shot of our RSTS time sharing system that was introduced in 72.

ition I'd like to get over very rapidly in terms of this notion of interactive computing and personal computing that might be helpful, I think first let's look at a time shared computer or a batch computer for that. A computer has a classical processor and primary memory, has a file memory for programming programs, for storing programs and data and then communication links to input/output. In particular to terminals and also to other sytems it is important to note that it's use is interactive and several persons use it in a shared fashion as they have files to share, they have work to share and so what could be a liability of being shared is actually and asset because there is so much that has to be communicated among the various machines, whereas a personal computer again a processor primary memory, it's own file memory, communications links this time to other personal computers and to other things for sharing it has transducers for human IO. in essence the terminal is just a shared computer that operates interactively by one person at a time and may be owned by that individual. **Put in text - full definition he had on origianal slide he wants to put all the clauses**.

ing the 80's. With the personal computer, because they are so easy to buy they are so easy to use, they are nonthreatening, nobody needs to know whether you ever turn your machine on or whether you actually use it, or anything, or whether you can use it, or what you do with it. Their just nice to have around. So as we enter the 80's what we see is now a large number of personal computers that are bigger globs. We still have a large number of the simple terminals connected to minis and now we see a very tight interconnection forming between all of the machines. That really is the setting for why we thing Ethernet is so important. In essence what we want to do is simplify the interconnection during this time. In essence all that's happened here is that we removed the links which were beginning to encumber us because everything was beinginning to have to be connected to everything else and that's one basic problem that Ethernet solves, but that's only the beginning. That only allows the revolution to form based on a quiet evolution from what we know now.

k at the network in detail in terms of, I've put down, three really important needs that we're in trouble with in the current generation: First off we've got a lot of computers and we're finding that there's more need for high speed interconnection among those networks. Second is that we had bound all of the connection of terminals to each of those computers and we're finding because there's different kinds of computers around different facilites no one terminal wants to be connected to a particular computer. Finally there is the personal computers that all want to be connected to host computers because they find that a personal computer is a good way to learn about computing, it's fine for doing certian kinds of tasks but once you've gone that far, one then needs to have a real computer.

s an evolution there's alot of computer control equipment within other nodes that would like to be connected to this in a very direct fashion. First look at this problem of we've got a tremendous number of terminals and personal computers and I want to interconnect that to some set of central computers, so what's our goal here? Fundamentally it's really an open ended interconnection of terminals to these host processors and other special information processing facilities let's just file, central files and central printing and also gateways to other computers and networks. Now look at the goal.

tuation today. Users are connected to very simple terminals, the smart terminals are becoming personal computers, although that's in essence really kind of an irrelevant thing, we have those machines generally bound to particular computers, we have the computers at the site all connected to each other in some kind of a local network basically on a hard wired length bases. Because of the problem of wanting acess to any computer what most installations do is connect, install some kind of a data switch so that their users, any person that a terminal connect can communicate with any computer. Again it doesn't matter whether thats a terminal or a personal computer on the left and in this series of slides I'm gong to use our own experience in 1981, we've got about 30 computers at a particular site. Basically a large number of them are simply dumb terminals, we have 700 users so there are 700 terminals and by 85 we expect that there probably will be 90 computers that will be linked together and probably 1400 users.

e view of this is we have a number of computers and their is some kind of a telephone switch that allows terminals to connect to that, that's prestine view if you look at the telephone switching line once these tremendous switching room where somebody has to come in and make up what the terminal to connection is and then again those get translated to the back of a switching computer and here's the front of a switching computer where we're now starting to switch some of the 700 terminals into anyone of the comptuers. The problem is that that switching computer is really bound and it doesn't grow very gracefully, particularly as you want to go from 700 to 1400 or for that matter, I doubt if it will be 1400 it will probably be more like 2500 and furthermore there is a dynamic problem here that we are trying to solve and then there is basically an ease of connection flexibility point that we are trying to solve, and so how do we do that? Basically, it really quite a simple way. Ethernet does provide this where we now are able to make a direct connection of these user terminals to the network on one side and then all of our computers are connected to the net on the other side, so we solve two problems and once, the computer to computer problem, but basically now any terminal or personal comptuer can acess the network. So in the process it solves this user at a terminal or personal computer to host or serving computer connection by first making a very simplified connection, a network is still on operation. It has high speed communication whereas in many of the switches we're often limited to 1200 baud and occasionally we're actually limited to 300 baud. Hard to believe in the 1980's.

est thing that we see is the open ended network growth by having this direct access, without extra equipment so that one can avoid all the inefficiencies that come with planning large facilites. Don't know about you, but one of our hardest things in an evolving organization is whether it's evolivng in a room, or whether people are just moving their desks every day, is simply the problem of planning, I don't think people plan very well and we've sort of evolved to centralized Russian type planning environment by the way that we've gone in much of the network. What we want is basically a bottom up planning. This planning shouldn't be the limit for this kind of thing, let's let technology solve our problem not have to have massive organization. Then finally in conjunction with the planning it really allows a tradeoff in the number and kind of connections, the number of terminals, the number of computers and that turns out to be the big problem with all of these switches, because it is a fully distributed switch.

t this other problem, we've got computers that we want to interconnect and so we've got to form a network out of all these dispersed computers, again here's the network we have today, we have over 200 systems and several thousand users on our computers weve got all in about a dozen sites. So it's at first glance a networking is really just a bunch of links and nodes, but on the other hand, I want to put in a plug here for the need for all the high level protocols that we have. Ethernet certainly solves the two low level interconnect problems of interconnecting dissimilar computers. Ethernet solves the basic interconnect need which is really the two levels two lowest international standards organization ISO opens some architecture levels <?>(256) but it's important to get to the next five layers here to address these other important needs, namely how does one connect dissimilar computers how does one provide these high level functions as file transfers, terminal to terminal use, document control voice and then on to a voice and finally how do you manage a network? These are all issues that the high level protocols address.

detail at this particular network that is currently over 250 nodes. Here's a topology of the whole thing, actually it's structured a bit more than that. Here are 13 of the sites, note that nine of them are in Massachusetss, a couple in New Hampshire, one in Colorado Springs, one in England, using a satelite link, and these are connected in general by a number fo 56 kilibit <?> links. However, what we are finding is that the number global sites is not increasing at a particularly high rate and we're finding that the number of local sites is increasing very rapidly and that will increase even more as personal computers replace the terminals. Now if we look at a particular node we find here's a node where there are about 30 computers and there are nine links to external machines to external sites. Another node where there are 31 computers and nine links to other sites. This is what happens when we install Ethernet in that site, fundamentally we expect to be connected in a very simple and orderly way and in doing that we expect a number of benefits.
see it solving this connection problem by really simplifying the connection.
Again while the network operates, one can put the machines on and off the network. We see certainly the high speed communication, the factor of at least a thousand times improvement over today's network. At, I might add, a much reduced cost, but I'm really not so concerned about the cost, I'm more concerned about what we get for what we pay. Then in the structure that we've got now, which is really the nodes are doing active

switching of network traffic, so we reduce the node load. One might say, I could do that by adding another network as the upernet structure has done, an internal network, which does only switching and in essence, that's what we do already, part of that network, if you look at in detail turns out to be many of the nodes are only traffic communication nodes and the others are real computing nodes. I believe item four is the most important one, which is; we can't have an orderly open ended growth without having any equipment and only one connection is required per node versus all these other structures that are inherently many connections and much equipment for the switch. The fully distributed switch that Ethernet provides is really the important part of the system. This is just a schematic representation.

t some real Ethernet. Here in May last year we announced Xerox, Intel and DEC announced the Ethernet and had it running and communicating among the various standards, various companies we were able to transfer print files and send messages among each other. Here it is in operation on one of our buildings finally, here we see the connection to a VAX and here we see another component produced by one of our competitors that has Ethernet to it which is where we are using a terminal concentrator to a VAX and now finally here's an ad by a competitor. Note how they feature Ethernet in the ad, but really the important part of this ad is the text which, let me read.

ou instantaneous access to all resources on the networks such as files, printers and other devices and other mainframes, plus all the need <?> single user computer. In real terms, what this means is this: Instead of taking as long as 44 seconds to translate ten pages of text, the transfer takes place in .042 seconds. In the 4.4 seconds it would take a conventional network to send one page of War and Peace, with Perk and Ethernet you could send the entire 1000 page novel. With that kind of performance and with that ease of interconnects, I think we are going to see a rapid change to a really different kind of computing versus the evolution, so I think it's an important component of the fifth generation, it provides for many current needs, we use it and are committed to it, it conforms to the ISO standards, and we believe that because of it's simplicity it will become a standard. We will certainly be introducing products within the next few months, and moreover for the future, we believe it's the real UNIBUS for the fifth generation.

FROM: GORDON BELL DATE: MON 29 OCT 1979 1:19 PM EST
DEPT: OOD
EXT: 223-2236
TO: DICK CLAYTON
BERNIE LACROUTE
BILL STRECKER
DAVE RODGERS
GEORGE PLOWMAN
SAM FULLER
WAYNE ROSING

SUBJECT: XEROX/DEC ANNOUNCEMENT OF ETHERNET FOLLOW UP:11/9/79

GB0005/37/EMS

Why don't we stop screwing around and adopt Ethernet AS IS? Then we can get a product quick, use it, and evolve.

The way we're headed it'll be 2 years to chips (if we're lucky) and another year or two to product. Meanwhile, IBM'll have the whole world wired with SDLC loops and we'll have to interface to them. This way, we get a compatible network with Xerox's printers and WP's. We're losing valuable time. Why not?

GB:swh

****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION

cc: KEN OLSEN

DATE: TUE 6 APR 1982 4:31 PM EST
FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: KEN'S PRESENTATION ON ETHERNET: HELP AND COMMENTS FOR HIM

Ken's talking to a financial group on the same day as the May announcement and would like to discuss Ethernet. I have him my talk to read. In addition he might want one of you to get more details on competitive technology like Wang, Datapoint IBM and the phone company.

He clearly should hand out the new Tutorial Handbook on LANs, and he could hand out the clean copy of my talk too. Also, he might want some slides.

Could you get other poop together like the Q&A on Ethernet, but without swamping him in paper?

Answering the why nots
Why not wait for the phone company and PABX's? Have you ever tried to use a terminal at 1200 baud.... we're talking about a system 10,000 times faster. It can transmit a high resolution black and white image in 0.1 sec or a color image in 1 sec. We don't see the wide scale availability of even 56Kbits in the foreseeable future from the phone companies on any kind of wide scale.

Why not put in a non ATT data and voice pabx? Why bother with the expense for the extra wiring. It still doesn't have adequate bandwidth between computers, or terminals or personal computers.

Why not use broadband
1. There's not standard for either data or data and catv.
2. Broadband is like a new piping material that can be used

to distribute physical goods like gas, sewage, water, oil and steam. The duct can carry anyone of them, it's the sorting it all out that's a bitch.
3. Many users want broadband because they assume some other user is going to pay for the installation.
4. It's hard to believe that broadband is going to be very pervasive in industrial environements. It is not adequate for two way videophones because of the limited bandwidth.
5. Systems like Wangnet use a second cable for return. Why not put in a second yellow wire and keep the two independent.
6. Baseband is simple to install. The users often do it.
7. We don't see broadband as being suitable for voice based on cost of modems and the cost of throwing out an existing plant using a central office type pabx with all its wiring and phones.
8. Ethernet will also carry a reasonable amount of voice, although we probably won't push it to evolve this way. It will mainly be carrying voice mail packets. One of the nice things is that the systems we are talking about are built so that there's less voice traffic. The personal computer will be used to help be less intrusive than the telephone.
9. The ultimate single media system will have much more bandwidth than broadband. This would allow videophones and images and certainly satisfy intercomputer needs. We don't see this as being practical until fiber optics and central switches that support them are available. We see no reason why this couldn't be done within 10 years. However, we've seen no reasonable laboratory demonstration of this yet.
10. If all else fails, then why fight it, we'll use broadband cables and put an Ethernet transceiver to encode our system into broadband. WWe have no real hangup with not using broadband. It's just that it's very nebulous, undefined and unstandardized now.
11. The users are demanding an OPEN standard. We're the only one who's proposed it.

How are you coming on the standardization?
The ieee 802 standard is progressing nicely. Various ecma companies including ICL, olivetti, cii and Siemens have joined in the standard. There are dozens of companies building products to the standard now. You can buy interfaces and components and put Ethernets together. Xerox has installed about 100 of them. We have about 10 Ethernets ourselves within the engineering organization.

What about IBM?
Ask em. They are doing their usual bit to find some standard that everyone else will have to meet. They were clearly in the dark about the need for LANs. We clearly understand them and are predicating all our products on them!

EVERY DEC PRODUCT WILL CONNECT TO ETHERNET EITHER DIRECTLY OR VIA A CONCENTRATOR. All our multiterminal systems will connect to them directly. The stand alone systems like the cpm and decmate will interface to other systems as terminals and file transmission.

What about Wang
We're waiting to see. They typically announce products 3 years before they're ready. The whole world changes in a half a computer generation.

"TO" DISTRIBUTION:

JOHN ADAMS MARION DANCY BERNIE LACROUTE

GB3.S4.40
I described that we intended to make Local Area Networks based on Ethernet at DECUS and I asked them to plan on wiring their buildings. Several of our customers thanked me for being specific.

Our current laissez faire position on standards is a disaster i.e. say we support standards, fight standards within a committee and then argue after the fact that it doesn't apply to us.

Let's be much more proactive on Ethernet.

0. Let's overtly sell it.

1. Hurry and get a product. Intel has one on 2 Multibus boards using less real estate and with higher performance! Why can't we?? What's wrong with our design team?

2. Let's hold regular meetings to standardize it and array those signed up.

3. Let's call specific people in other companies e.g. HP to get them committed!

4. Let's go all out at the IEEE Committee, ANSI and at ISO!

need a plan!

swh
.S6.24

```
*****************
* d i g i t a l *
*****************
```

TO: ENG STAFF:
DON METZGER
cc: see "CC" DISTRIBUTION

DATE: WED 7 APR 1982 6:21 PM EST
FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: STARS/ETHERNETS FOR ENGINEERING&TYPESETTING REVOLUTION

STARS ON ETHERNETS. Let's get a plan in place to get print/plot service on all the Ethernets when they come into operation by July 1. The current Xerox print/plotter service is about 1000 lines per minute, but substantially faster than all plotters, especially when the plots involve reproduction of vellum where we measure output in minutes per page, not pages per minute.

The highest priority places would be MR, TW, ML and HU so we can begin to revolutionize the way we engineer. We can eliminate the whole concept of waiting for plotting or print production as engineers have done for the last few thousand years.

ENGINEERING REVOLUTION. We really need to get the plotting going this way because it's going to impact the way we do engineering, given that a page is 8-1/2 x 11. Based on the experience of others, I'm certain that this is the best thing for us too. The resolution of the plotter of 300 lines per inch means that we can put still put high quality D size prints on the A size paper if we want to. This is the key to getting rid of the whole kludgy microreproductions system and going to total on-line design, storage, retrieval and transmission of all drawings.

The goal for a complete electrical cad/cam system is thus: NO MANUAL INTERVENTION OR RETRANSFORMATION OF DESIGN INFORMATION FROM THE ENTRY OF SCHEMATICS TILL THE FULLY TESTED PARTS ARE PRODUCED. THIS IMPLIES CAD, NETWORKING, ON LINE STORAGE, AND NO MANUAL STEPS (eg. print production).

TYPESETTING. STAR also offers high quality typesetting together with graphics that we all should know, need and love... and get into products asap! We need it to produce Camera Ready copy for ALL of our hardware and software manuals. Somehow, we need to interface our typesetting and wp systems to it. I was impressed with STAR's function, not its performance (but that's really a trivial problem that Xerox can fix). We have a competitor in Xerox! Let's learn from them and use their products to boot strap our way into providing quality printing and plotting.

If you haven't...; see and use the Xerox system in Tewksbury.

We can start this revolution now with STARs at all sites. The key research universities have operated this way for the last 10 years!!! We have to do this for productivity!

How can we get this total program going faster?

"CC" DISTRIBUTION:

DAVE COPELAND	AL CRAWFORD	BOB DALEY
BOB GLORIOSO	GOLDFEIN AND TEICHER	RICHARD GONZALES
BILL HEFFNER	BOB KUSIK	DEL LIPPERT
BOB MARSHALL	JULIUS MARCUS	ROY REZAC
JOHN RING	DAVE RODGERS	PETER SMITH
JACK SMITH	BOB TRAVIS	ARMEN VARTERESSIAN

GB3.S4.38

****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION　　　　DATE: MON 30 MAR 1981 9:14 EST
FROM: GORDON BELL

cc: see "CC" DISTRIBUTION

DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: INTEL INTERFACE INCLUDING ETHERNET AND BUYING FROM THEM

On Friday, Andy Grove called me in extreme distress that we had mistreated them, as one of the partners of Ethernet. He believes we had a joint development and then taking what we had learned during the development of the standard, went about getting a commodity part, based on their knowledge of communications chips and in the process taught two competitors about communications and got them into the business. Mitch Federman prepared an excellent post mortem on how this happened. Dick interpreted the specifics of this in his memos.

I'll not attempt to interpret this here, but rather outline how I hope we can work with Intel. The specifics have to be really worked everywhere within our organization, because it's clear I and the top management of Intel have inadequate understanding of how we actually work together. I am making the assumption that it is to our joint advantage to co-operate more closely. Specifically:

. Jim Cudmore will take over the interface since many of his people interface with Intel. His first act is going to be to write a charter which we will get agreed to with them on what are the areas we might work together, and those where we view ourselves as competitive. It would include all issues and interfaces:
    + commodity parts, use of production capacity, cad tools, process information, use of VAX's there, Ethernet, parts specifically developed for us, parts they are developing that we might influence and use; and
    - add on memories to DEC, the architectures and systems we view as competitive to DEC, future systems, etc.

. Dave Rodgers will work to get us back, to be a co-partner in the development of Ethernet. I believe they'll be the first.
    . Avram should aggressively start to buy the chips for use within the CT, because it has immediate needs. Also, I believe they will have better surrounding chips because they will try to use the part in their systems too.
    . Dave's product management folks will work with either Micros or AS&G to make the transceiver (what is traditionally called a modem) a product so that we can become a supplier. For now, we must figure out some way to seed samples to potential users, while this interminable red tape and miscommunication is straighted out.

. Pat Buffet should identify a group who might use the 8086 or 8088 as a high performance processor to be used within a product. It would seem that both the communications protocol boards and disks and controllers like RSO, Pinon and Aztec would be candidates. In both cases, the electronics costs are uncompetitive and this is the way to go for these products. We should bring in the part, evaluate it by looking at the software systems we would use to support it, and build a breadboard of something. In no way should we bring thes parts in with the intent of using assembly language programming except where we run into timing problems. I would hope that either Pascal, or their PL/M would be used. Also, I would hope that their operating systems would be useful as components in these products. Therefore, I'd like something from Pat on who, what, and then how (to be reviewed with Sam Fuller) soon.

. Mitch and Pat should identify the parts we are working with them to use that are in their development cycle so that we might use them. Specifically, there is a video chip that looks interesting, there is rumblings about a very high speed programmable I/O processor suitable for use within a disk data path, and Carsten described a one chip computer for use within a computing terminal. At all times, I'd like someone to have a list of these chips for all vendors we are working with! WHO'll volunteer?

. Jim should have a clear policy about what we intend to make versus buy with respect to VLSI, in general. I can not emphasize strongly enough that we must buy commodity VLSI whenever possible. We are not using enough in our products. Our posture has to improve! It should distinguish the cases:

  . clearly commodity like memories, uarts, microprocessors;
  . we have significant knowledge and technology and it is worthwhile to be proprietary and make our own (presumably this is the case with the VT200 video display controller); the only parts that we should worry about being proprietary about would be "user-visible". That is, a display screen controller if it gave us marketing uniqueness would be proprietary, and a floppy controller should be considered to be a commodity because it gives no user uniqueness.
  . we have to do our own because of vanity architectures (VAX, PDP-11); and
  . parts that can give us time to market and really competitive products if we are able to work with a vendor to get what we want. It is clear that others will have the chip too either at the same time, or somewhat later. Here, we should use their one chip computing terminal, even though others are too...I don't expect us to look at it, say we can do better and start a competitive chip that comes out 1-2 years later at several times the cost (our standard behavior in the past). Most of the time these are initiated by the vendor and we should find out and use them, although I can think of no cases where we're effectively doing or have done this now. Also, we should get parts built for us, even though they will be industry standard.

Overall, I want to state our implicit and explicit policies, written down and then to start behaving according to this. This means real change. I'm convinced we have it. Jim will lead us through this.

GB2.S4.41

"TO" DISTRIBUTION:

JIM CUDMORE AVRAM MILLER MITCH FEDERMAN @LSIG
PATRICK BUFFET @MLXX DAVE RODGERS

"CC" DISTRIBUTION:

DICK CLAYTON HENRY CROUSE DAN HAMEL
ANDY KNOWLES BERNIE LACROUTE SI LYLE
JACK MACKEEN DON METZGER ROY MOFFA
BILL PICOTT BOB SAVELL STOCK STOECKELMANN @LSIG
STEVE TEICHER JOE ZEH
00 BURT DECGRAM ACCEPTED S 27626 O 64 13-SEP-81 23:11:29

****
* d i g i t a l *
****

TO: see "TO" DISTRIBUTION
cc: see "CC" DISTRIBUTION

DATE: SUN 13 SEP 1981 23:09 EST
FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: XEROX, THE 820, A LOW COST ETHERNET, ETC.

I listened to a description of this product and learned:
.Xerox did it in 6 months, and hardly designed any of it
.It changed the mentality of Xerox engineering who had built itself a bureuacracy based on 6 year product intros
.They are making 7,000 per month
.The product will be profitable in the first year, even with Xerox overheads, costs, etc.
.Xerox used to believe they should do Custom Circuits and Bipolar designs, Now they are sold on Industry Standard Parts (probably so is IBM)!
.They will follow it with another product in 6-9 months which will be at same cost, but more power
.The 860 isn't doing all that well
.Don Massaro is a real Hero within Xerox
.There is concern that Ethernet is too late! Some insiders would like to have a low cost, 1 megabit link, NOW! (One such possibility is the IEEE 488 that has been used in this configuration ... it might be the best for us too especially since we need the 488 for CT and it exists on a number of our current systems. What you think????)
.Xerox believes they must CAPTURE THE DESK! (The competition is Apple, Tandy and now IBM too.)
.The are going to set up a strong field organization in order to be able to sell and support the complexities of Ethernet.
.They see the phone as an important connection to Ethernet and are working on it at PARC.

It's clear we can identify with some of these points and concerns.

"TO" DISTRIBUTION:

BILL AVERY GEORGE CHAMPINE BARRY JAMES FOLSOM
BOB GLORIOSO SI LYLE AVRAM MILLER

BRUCE STEWART

"CC" DISTRIBUTION:

ENG STAFF: GVPC: KEN OLSEN
DAVE RODGERS

GB2.S8.24

February 9, 1979

Mr. James S. Campbell
President
Xerox Business Systems
701 S. Aviation Blvd.
El Segundo, California 90245

Dr. George E. Pake
Vice President
Xerox Research
3333 Coyote Hill Road
Palo Alto, California 94304

Dear Sirs:

I am writing at the suggestion of Bob Metcalfe. We believe that Ethernet might serve as the basis for a productive arrangement between Xerox and Digital. I would like to explore with you or the appropriate persons within Xerox the establishment of Ethernet as a standard for communication among computers.

I recently learned that Xerox may be placing some of its Altos and Dovers (connected by Ethernets) in universities at which there are major DARPA contracts. We too have a number of computers in these universities and would like to use the Ethernet scheme for interconnection. The first opportunity that I see in a possible Xerox-DEC agreement about Ethernet would be that of providing DARPA with a powerful local networking capability among their DEC computers and Xerox workstations and printers.

DEC has been working in the local computer networking area for some time, looking for the proper approach. One of the problems in networking is that there are so many alternatives for local computer interconnection and no widely used system with which to attain the critical mass required. We have been following your work on Ethernet and now believe that it would be a very good candidate for a communication standard around which a critical mass could be formed. Imagine, for example, the mutual advantage in our being able to offer customers direct interconnection of DEC computers with Xerox products like the 9700.

We would be interested in learning whether Xerox would consider licensing its Ethernet patent to Digital. Further, we would like to discuss whether and how Digital might become compatible with Xerox's Ethernet as it now exists and is about to appear at DARPA sites. Might we take over responsibility for manufacturing and servicing Ethernet equipment for DEC computers? Does Xerox have longer term plans for Ethernet?

As a separate issue we would also like to explore the availability of a low cost Xerographic printer with you. We would especially be interested in a low cost adapter to a regular, low cost office copier.

I look forward to our discussing these possibilities with the appropriate people at Xerox.

Sincerely,

Gordon Bell
Vice President, Engineering

GB:ljp

GB0001/7

CC: Jim Conway, Xerox
Andy Knowles, DEC
Joe Meany, DEC
Tom Siekman, DEC

ETHERNET AND THE FIFTH GENERATION

Gordon Bell
Vice President, Engineering
Digital Equipment Corporation

In the Fifth Computer Generation, a wide variety of computers will communicate with one another. No one argues about this. The concern is about how to do it and what form the computers will take.

A standard communications language is the key. I believe Ethernet is this unifying key to the 5th computer generation because it interconnects all sizes and types of computers in a passive, tightly-coupled, high performance fashion, permiting the formation of local-area networks.

May 16, 2003

2003 National Medal of Technology Committee
nmt2003@ta.doc.gov

Dear 2003 National Medal of Technology Committee:

Subject: Recommendation of Dr. Robert M. Metcalfe for the National Medal of Technology

I am very pleased to write this recommendation for Bob Metcalfe to receive the National Medal of Technology at this propitious time -- just one week before the 30th birthday of Ethernet.
Bob was not only the inventor of the first Ethernet, but was the catalyst to get it adopted as a standard, through a consortium of Digital Equipment, Intel, and Xerox Corporations. Ethernet moved rapidly to adoption as an International standard with the IEEE and ECMA (European Computer Manufacturers Association). Also helpful was the fact that a more complex token ring alternative that IBM and TI were pursuing took many years to develop and was tainted by a bogus patent. However, at the time, the computer industry didn't know it needed LANs (Local Area Networks).

Ethernet has evolved through numerous bandwidth (10, 100, 1,000 and now 10,000 Million bits per second) and technology changes including radio, but equipment built to the first standards still interoperates.

My understanding and belief in this nomination comes from the direct involvement as Vice President of Engineering of Digital Equipment Corporation when we made the Ethernet decision. At that time I was working on the VAX Strategy and DEC architecture for interconnecting computers, and a LAN was at its core. Bob consulted with us, and together we were able to convince Xerox of the overall need, urgency, and benefit of Ethernet technology and a standard.

Finally, it is literally impossible to state the overall impact Ethernet has had on computing and communications *outside of creating a major industry*. It has enabled distributed computing architectures such as our own PCs, to building metropolitan area networks and includes today's wireless hot spots that are springing up at houses and businesses to unwire the planet and provide computing access everywhere.

Sincerely,

C Gordon Bell

C Gordon Bell
Senior Researcher
National Medal of Technology, 1991

e@30

e t h e r n e t @ t h i r t y

PALO ALTO RESEARCH CENTER | 3333 COYOTE HILL ROAD | PALO ALTO, CA 94304 USA

THURSDAY | MAY 22, 2003 | 3:00 - 5:00 PM

hosted by:

COMPUTER HISTORY MUSEUM

parc Palo Alto Research Center

sponsored by:

3com

Cisco Systems

hp invent

intel

special thanks to:

THE DOCUMENT COMPANY XEROX

printed on:

Xerox DocuColor iGen3 Digital Production Press

XEROX

ETHER!

Xerox Sensitive

MEMO

MAY 22, 1973

TO: ALTO ALOHA DISTRIBUTION

FROM: BOB METCALFE

SUBJECT: ETHER ACQUISITION

HERE IS MORE ROUGH STUFF ON THE ALTO ALOHA NETWORK.

I PROPOSE WE STOP CALLING THIS THING "THE ALTO ALOHA NETWORK". FIRST, BECAUSE IT SHOULD SUPPORT ANY NUMBER OF DIFFERENT KINDS OF STATION -- SAY, NOVA, PDP-11, ... SECOND, BECAUSE THE ORGANIZATION IS BEGINNING TO LOOK VERY MUCH MORE BEAUTIFUL THAN THE ALOHA RADIO NETWORK -- TO USE CHARLES'S "BEAUTIFUL".

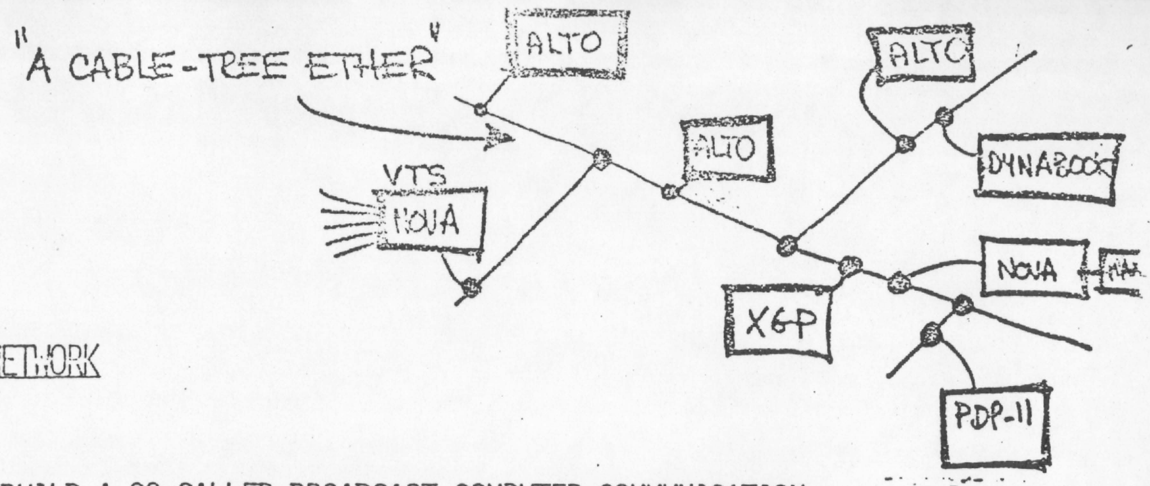
MAYBE: "THE ETHER NETWORK". SUGGESTIONS?

LAZY SUZAN
BULLETIN BOARD
PARLEY
PARLIAMENTARY PROCEDURE

I HOPE TO BE SIMULATING SOON. HELP? INPUTS?

I HOPE YOU WILL NOT BE OFFENDED BY MY ATTEMPTS TO MAKE THIS THINKING AND DESIGN APPEAR THEORETICAL.

Bob

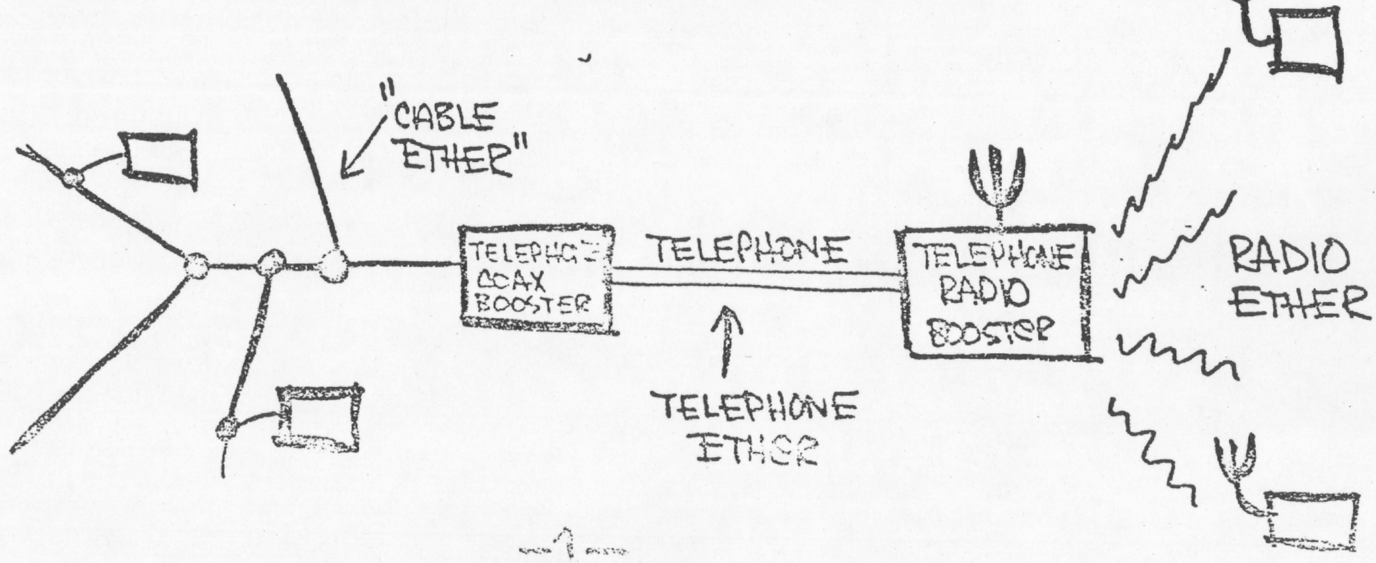


THE ETHER NETWORK

WE PLAN TO BUILD A SO-CALLED BROADCAST COMPUTER COMMUNICATION NETWORK, NOT UNLIKE THE ALOHA SYSTEM'S RADIO NETWORK, BUT SPECIFICALLY FOR IN-BUILDING MINICOMPUTER COMMUNICATION. WE THINK IN TERMS OF NOVA'S AND ALTO'S JOINED BY COAXIAL CABLES.

WHILE WE MAY END UP USING COAXIAL CABLE TREES TO CARRY OUR BROADCAST TRANSMISSIONS, IT SEEMS WISE TO TALK IN TERMS OF AN ETHER, RATHER THAN 'THE CABLE', FOR AS LONG AS POSSIBLE. THIS WILL KEEP THINGS GENERAL AND WHO KNOWS WHAT OTHER MEDIA WILL PROVE BETTER THAN CABLE FOR A BROADCAST NETWORK; MAYBE RADIO OR TELEPHONE CIRCUITS, OR POWER WIRING OR FREQUENCY-MULTI-PLEXED CATV, OR MICROWAVE ENVIRONMENTS, OR EVEN COMBINATIONS THEREOF.

THE ESSENTIAL FEATURE OF OUR MEDIUM -- THE ETHER -- IS THAT IT CARRIES TRANSMISSIONS, PROPAGATES BITS TO ALL STATIONS. WE ARE TO INVESTIGATE THE APPLICABILITY OF ETHER NETWORKS.



XEROX

ETHER ACQUISITION

HOW DOES A STATION'S TRANSMITTER ACQUIRE THE USE OF THE ETHER FOR A PARTICULAR TRANSMISSION? THERE ARE MANY POSSIBLE WAYS.

THE ALOHA RADIO NETWORK USES WHAT WE CALL "DE FACTO" ETHER ACQUISITION. A STATION DESIRING TO TRANSMIT SIMPLY DOES, IT JUMPS RIGHT ON AND USES THE ETHER. IF THE TRANSMISSION GOES THROUGH, THE ETHER HAS BEEN SUCCESSFULLY ACQUIRED, DE FACTO. IF SOME OTHER TRANSMISSION CONFLICTS; THEN BOTH (ALL) ARE LOST AND ARE RETRIED SOME RANDOM TIME LATER; THE ETHER HAS FAILED TO BE ACQUIRED.

AT LEAST TWO FACTS ABOUT THE ALOHA ETHER AND TRANSCEIVERS SUPPORT THE USE OF DE FACTO ETHER ACQUISITION. FIRST, THE ALOHA ETHER IS VERY BIG, IT TAKES A LONG TIME FOR TRANSMISSIONS TO PROPAGATE; AND SECOND, ALOHA TRANSCEIVERS ARE STRICTLY HALF-DUPLEX, THEY CANNOT DETECT INTERFERENCE WHILE TRANSMITTING. NEITHER OF THESE TWO FACTS IS TRUE OF OUR ETHER OR OUR STATIONS AS THEY ARE ENVISIONED.

XEROX

AND NOW, FOUR AXIOMS: **Axioms?**

(1) THE ETHER AXIOM: THE ETHER CARRIES TRANSMISSIONS TO ALL STATIONS.

(2) THE PROXIMITY AXIOM: PROPAGATION TIMES ARE SOMEWHAT SMALL.

(3) THE DETECTION AXIOM: STATIONS CAN DETECT, AT ALL TIMES, TRANSMISSIONS OF OTHER STATIONS, AS THEY PASS, IN ABOUT ONE BIT TIME.

(4) THE DEFERENCE AXIOM: WHILE DETECTING A PASSING TRANSMISSION; NO STATION WILL BEGIN OR CONTINUE ITS OWN TRANSMISSION.

THE ETHER AXIOM FREES US FROM CONSIDERING NETWORK ROUTING. ← NOT THE LOCAL NETWORK! THE PROXIMITY AXIOM ALLOWS US TO CONSIDER SOLUTIONS WHICH WOULD BE TOTALLY IMPRACTICAL OTHERWISE -- SAY AS IN ALOHA RADIO. THE DETECTION AXIOM DOES NOT IMPLY THAT CONFLICTS CAN BE AVOIDED; SEPARATED TRANSCEIVERS CAN BEGIN TRANSMISSION ON FREE ETHER ONLY TO DISCOVER LATER THAT THEIR TRANSMISSIONS HAVE COLLIDED ELSEWHERE. THE DEFERENCE AXIOM FOLLOWS FROM NOTHING MORE THAN OUR BASIC INTUITION -- MAYBE IT SHOULD BE DISCARDED SOMETIME..

XEROX

AND NOW, A DEFINITION:

A STATION IS SAID TO HAVE ACQUIRED THE ETHER WHEN AND ONLY WHEN IT HAS BEGUN TRANSMITTING A PACKET AND ALL OF THE OTHER STATIONS HAVE DETECTED THE TRANSMISSION AND ARE DEFERRING TO IT.

AFTER ACQUIRING THE ETHER, A STATION IS SAID TO HOLD THE ETHER AS LONG AS IT CONTINUES TRANSMITTING.

THE DEFERENCE AXIOM IMPLIES THAT ONCE A STATION HAS ACQUIRED THE ETHER, IT CAN HOLD THE ETHER AS LONG AS IT WANTS, USING IT WITHOUT CONFLICT FOR THE DURATION OF ITS TRANSMISSION. A STATION VIOLATING THE DEFERENCE AXIOM COULD, OF COURSE, BREAK A HOLD ON THE ETHER AND ACQUIRE IT, BUT FOR THE MOMENT WE DISALLOW THIS BEHAVIOR.

IF THE ETHER IS TO BE SHARED IN SOME REASONABLE WAY, THEN FURTHER AGREEMENTS WILL BE REQUIRED TO REGULATE THE MAXIMUM HOLDING TIME. BUT THIS COMES LATER.

XEROX

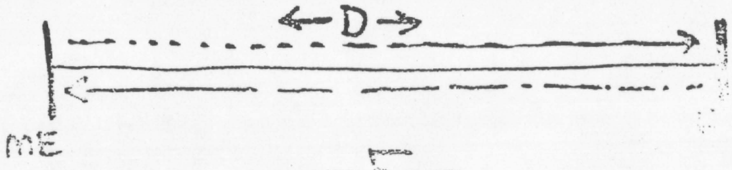
AND NOW, ANOTHER SO-CALLED AXIOM:

(5) THE DIAMETER AXIOM: FOR ANY GIVEN ETHER NETWORK, THERE EXISTS A DIAMETER D, THE PROPAGATION DELAY BETWEEN MOST DISTANT STATIONS, THE MAXIMUM TIME FROM START OF TRANSMISSION TO DETECTION OF TRANSMISSION BY A DISTANT STATION.

BY THE PROXIMITY AXIOM, D IS "SOMEWHAT" SMALL.

AND NOW A FACT:

HOW LONG AFTER BEGINNING TRANSMISSION MUST I DETECT NO CONFLICT BEFORE I CAN BE CERTAIN THAT I HAVE ACQUIRED THE ETHER? THE ANSWER: 2D, ONE ROUND TRIP. SAY THAT THERE IS THIS STATION AT THE FAR END OF THE ETHER, D SECONDS AWAY. AFTER I START TRANSMISSION ON THE OPEN ETHER, IT CAN BE D SECONDS BEFORE HE KNOWS ABOUT IT. BUT IF JUST BEFORE MY TRANSMISSION REACHES HIM HE DECIDES TO TRANSMIT HIMSELF, THEN IT WILL BE D MORE SECONDS BEFORE I FIND OUT ABOUT IT -- IT CAN BE 2D SECONDS BEFORE I SENSE CONFLICT AND THEREFORE FAILURE TO ACQUIRE. HE WILL HAVE SENT A BIT OR TWO BEFORE DETECTING MY TRANSMISSION AND WILL DEFER, BUT IT'S TOO LATE. HIS BRIEF TRANSMISSION WILL CAUSE ME TO LET GO OF THE ETHER ACCORDING TO THE AXIOM OF DEFERENCE. IT TAKES 2D SECONDS OF ETHER TIME TO ACQUIRE.



XEROX

DEFINITION: A TRANSMISSION IS SAID TO BE <u>CONFLICT-FREE</u> WITH RESPECT TO ITS TRANSMITTER AND A SPECIFIED RECEIVER (DISREGARDING ETHER NOISE) IF AND ONLY IF THE TRANSMISSION PLACED ON THE ETHER BY THE TRANSMITTER IS LATER CORRECTLY RECEIVED (I.E., WITHOUT INTERFERENCE) AT THE RECEIVER.

FACT: IF THE ETHER IS ACQUIRED FOR A TRANSMISSION, THEN THE TRANSMISSION IS CONFLICT-FREE FOR ALL RECEIVERS.

FACT: ETHER ACQUISITION IS NOT NECCESSARY FOR CONFLICT-FREE TRANSMISSIONS, EVIDENCE ALOHA SUB-ACQUISITION TRANSMISSIONS.

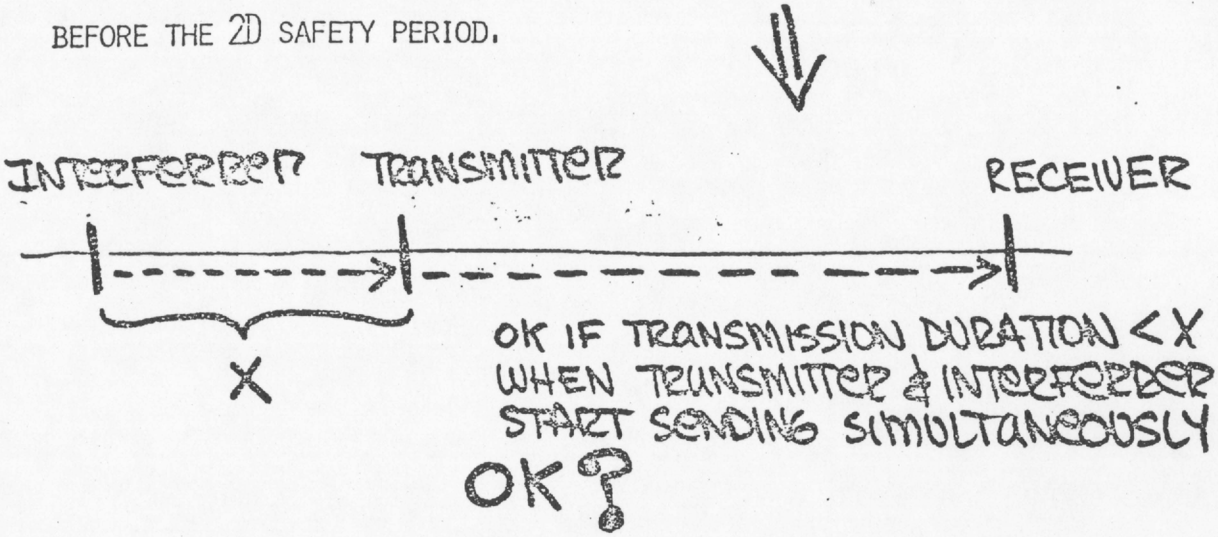
FACT: THE LONGEST CONFLICT-FREE SUB-ACQUISITION TRANSMISSION IS D SECONDS LONG.

OK?

ƎROX

FACT: A TRANSMISSION OF ANY LENGTH D (EVEN LESS THAN D) CAN BE DETERMINED TO BE CONFLICT-FREE FOR ALL RECEIVERS BY ITS TRANSMITTER IF NO CONFLICTING TRANSMISSIONS ARE DETECTED FOR A PERIOD OF 2D SECONDS AFTER THE START OF TRANSMISSION.

FACT: A TRANSMISSION MAY BE CONFLICT-FREE WITH RESPECT TO ITS INTENDED RECEIVER EVEN IF AN OTHER TRANSMISSION IS DETECTED BEFORE THE 2D SAFETY PERIOD.



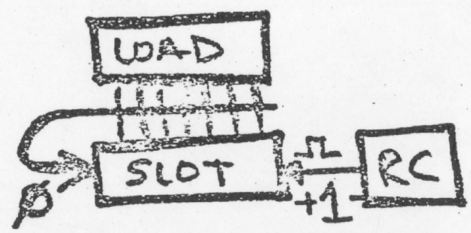
XEROX

ETHER BARGAINING LOGIC

WE PRESUME WE KNOW THE ETHER'S DIAMETER AND THAT IT IS SMALL. WE PROPOSE THE FOLLOWING LOGIC FOR A STATION'S BARGAINING WITH THE ETHER.

FIRST, A CLOCK; CALL IT THE ROUND-TRIP CLOCK (RC). THE RC NEED NOT BE VERY GOOD; AN UGLY MULTI-VIBRATOR PERHAPS. IT SHOULD HAVE A PERIOD OF 2D+EPSILON, FOR SOME SMALL EPSILON.

SECOND, A COUNTER; CALL IT THE SLOT COUNTER (SC). THE SC IS ALWAYS COUNTING UP, INCREMENTED BY THE ROUND-TRIP CLOCK.



THIRD, A REGISTER; CALL IT THE LOAD REGISTER (LR). THE LOAD REGISTER TELLS THE SLOT COUNTER WHEN TO RETURN TO ZERO. THE LR HOLDS A NUMBER WHICH IS A MEASURE OF ETHER TRAFFIC LOAD. IN COUNTING UP FROM ZERO, THE SLOT COUNTER RETURNS TO ZERO WHEN ITS CONTENTS ARE EQUAL TO THAT OF THE LOAD REGISTER. THE LOAD REGISTER DEFINES THE LENGTH OF THE SLOT COUNTERS CYCLE.

FOURTH, OTHER-DRIVE DETECTOR, OD. THE OD LOOKS AT THE ETHER TO DETECT WHEN THE ETHER IS BEING DRIVEN BY SOME TRANSMITTER OTHER THAN ITS OWN, AT THE POINT OF THE TRANSMITTER.

The Ethernet

A Local Area Network

Data Link Layer
and
Physical Layer
Specifications

Digital Equipment Corporation
Maynard, MA

Intel Corporation
Santa Clara, CA

Xerox Corporation
Stamford, CT

Version 1.0

September 30, 1980

IMPORTANT INFORMATION AND DISCLAIMERS

Preface

This document contains the specification of the Ethernet, a local area network developed jointly by Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. The Ethernet specification arises from an extensive collaborative effort of the three corporations, and several years of work at Xerox on an earlier prototype Ethernet.

This specification is intended as a design reference document, rather than an introduction or tutorial. Readers seeking introductory material are directed to the reference list in Section 2, which cites several papers describing the intent, theory, and history of the Ethernet.

This document contains 7 sections, falling into three main groups:

Sections 1, 2, and 3 provide an overall description of the Ethernet, including its goals, and the scope of the specification.

Sections 4 and 5 describe the architectural structure of the Ethernet in terms of a functional model consisting of two layers, the Data Link Layer and the Physical Layer.

Sections 6 and 7 specify the two layers in detail, providing the primary technical specification of the Ethernet.

Readers wishing to obtain an initial grasp of the organization and content of the specification will be best served by reading Sections 1, 3, and 4. Readers involved in actual implementation of the Ethernet will find Sections 5, 6, and 7 to contain the central material of the specification. Section 2 provides references, and the appendices provide supplementary material.

The approach taken in the specification of the Data Link Layer in Section 6 is a procedural one; in addition to describing the necessary algorithms in English and control flow charts, the specification presents these algorithms in the language Pascal. This approach makes clear the required behavior of Data Link Layer, while leaving individual implementations free to exploit any appropriate technology.

Because the procedural approach is not suitable for specifying the details of the Physical Layer, Section 7 uses carefully worded English prose and numerous figures and tables to specify the necessary parameters of this layer.

Some aspects of the Ethernet are necessarily discussed in more than one place in this specification. Whenever any doubt arises concerning the official definition in such a case, the reader should utilize the Pascal procedural specification of the Data Link Layer in Section 6.5, and the detailed prose specification of the Physical Layer in Sections 7.2 through 7.9.

One aspect of an overall network architecture which is not addressed by this specification is network management. The network management facility performs operation, maintenence, and planning functions for the network:

- Operation functions include parameter setting, such as address selection.
- Maintenance functions provide for fault detection, isolation, and repair.
- Planning functions include collection of statisical and usage information, necessary for planned network growth.

While network management itself is properly performed outside the Ethernet Data Link and Physical Layers, it requires appropriate additional interfaces to those layers, which will be defined in a subsequent version of this specification.

Table of Contents

Preface i

1. INTRODUCTION 1

2. REFERENCES 3

3. GOALS AND NON-GOALS 4

    3.1 Goals 4

    3.2 Non-Goals 5

4. FUNCTIONAL MODEL OF THE ETHERNET ARCHITECTURE 6

    4.1 Layering 8

    4.2 Data Link Layer 9

    4.3 Physical Layer 10

    4.4 Ethernet Operation and the Functional Model 12

        4.4.1 Transmission Without Contention 12

        4.4.2 Reception Without Contention 12

        4.4.3 Collisions: Handling of Contention 13

5. INTER-LAYER INTERFACES 15

    5.1 Client Layer to Data Link Layer 15

    5.2 Data Link Layer to Physical Layer 16

6. ETHERNET DATA LINK LAYER SPECIFICATION 19

    6.1 Data link Layer Overview and Model 19

    6.2 Frame format 19

        6.2.1 Address Fields 21

            6.2.1.1 Destination Address Field 21

            6.2.1.2 Source Address Field 21

        6.2.2 Type Field 21

        6.2.3 Data Field 22

        6.2.4 Frame Check Sequence Field 22

        6.2.5 Frame Size Limitations 22

    6.3 Frame Transmission 23

        6.3.1 Transmit Data Encapsulation 23

            6.3.1.1 Frame Assembly 23

            6.3.1.2 Frame Check Sequence Generation 23

        6.3.2 Transmit Link Management 23

6.3.2.1 Carrier Deference 23
6.3.2.2 Interframe Spacing 24
6.3.2.3 Collision Handling 24
6.3.2.3.1 Collision Detection and Enforcement 24
6.3.2.3.2 Collision Backoff and Retransmission 25

6.4 Frame Reception 25
6.4.1 Receive Data Decapsulation 25
6.4.1.1 Framing 25
6.4.1.1.1 Maximum Frame Size 26
6.4.1.1.2 Integral Number of Octets in Frame 26
6.4.1.2 Address Recognition 26
6.4.1.2.1 Physical Addresses 26
6.4.1.2.2 Multicast Addresses 26
6.4.1.3 Frame Check Sequence Validation 26
6.4.1.4 Frame Disassembly 27
6.4.2 Receive Link Management 27
6.4.2.1 Collision Filtering 27

6.5 The Data Link Layer Procedural Model 27

6.5.1 Overview of the Procedural Model 27

6.5.1.1 Ground Rules for the Procedural Model 27

6.5.1.2 Use of Pascal in the Procedural Model 29

6.5.2 The Procedural Model 31
6.5.2.1 Global Declarations 34
6.5.2.1.1 Common Constants and Types 34
6.5.2.1.2 Transmit State Variables 35
6.5.2.1.3 Receive State Variables 35
6.5.2.1.4 Summary of Interlayer Interfaces 36
6.5.2.1.5 State Variable Initialization 37
6.5.2.2 Frame Transmission 38
6.5.2.3 Frame Reception 42
6.5.2.4 Common Procedures 44

7. ETHERNET PHYSICAL LAYER SPECIFICATION 45

7.1 Physical Channel Overview and Model 45

7.1.1 Channel Goals and Non-Goals 45
7.1.1.1 Goals 45
7.1.1.2 Non-Goals 45

7.1.2 Characteristics of the Channel 46

7.1.3 Functions Provided by the Channel 46

7.1.4 Implementation of the Channel 46
7.1.4.1 General Overview of Channel Hardware 47
7.1.4.2 Compatibility Interfaces 48

7.1.5 Channel Configuration Model 49

7.1.6 Channel Interfaces 53

7.2 Transceiver Cable Compatibility Interface Specifications 53

7.2.1 Transceiver Cable Signals 53
7.2.1.1 Transmit Signal 53
7.2.1.2 Receive Signal 54
7.2.1.3 Collision Presence Signal 54
7.2.1.4 Power 54

7.2.2 Transceiver Cable Parameters 54
7.2.2.1 Mechanical Configuration 54
7.2.2.2 Characteristic Impedance 55
7.2.2.3 Attenuation 55
7.2.2.4 Velocity of Propagation 55
7.2.2.5 Pulse Distortion 55
7.2.2.6 Resistance 55
7.2.2.7 Transfer Impedance 55

7.2.3 Transceiver Cable Connectors 56

7.2.4 Transceiver Cable Drive 56

7.2.5 Transceiver Cable Receive 57
7.2.5.1 Load Impedance and Termination 57
7.2.5.2 Common Mode and CMRR 57

7.3 Coaxial Cable Compatibility Interface Specifications 58

7.3.1 Coaxial Cable Component Specifications 58
7.3.1.1 Coaxial Cable Parameters 58
7.3.1.1.1 Characteristic Impedance 58
7.3.1.1.2 Attenuation 58
7.3.1.1.3 Velocity of Propagation 58
7.3.1.1.4 Mechanical Requirements 59
7.3.1.1.5 Pulse Distortion 59
7.3.1.1.6 Jacket Marking 59
7.3.1.1.7 Transfer Impedance 59
7.3.1.2 Coaxial Cable Connectors 60
7.3.1.3 Coaxial Cable Terminators 60
7.3.1.4 Transceiver-to-Coaxial Cable Connections 61

7.3.2 Coaxial Cable Signaling 61

7.4 Transceiver Specifications 62

7.4.1 Transceiver-to-Coaxial Cable Interface 62
7.4.1.1 Input Impedance 62
7.4.1.2 Bias Current 62
7.4.1.3 Transmit Output Levels 63

7.4.2 Transceiver-to-Transceiver Cable Interface 63
7.4.2.1 Transmit Pair 63
7.4.2.2 Receive Pair 63
7.4.2.3 Collision Presence Pair 64
7.4.2.4 Power Pair 64

7.4.3 Electrical Isolation 64

7.4.4 Reliability 64
7.5 Channel Logic 64
7.5.1 Channel Encoding 64
7.5.1.1 Encoder 65
7.5.1.2 Decoder 65
7.5.1.3 Preamble Generation 66
7.5.2 Collision Detect Signal 67
7.5.3 Carrier Sense Signal 67
7.5.4 Channel Framing 68
7.5.4.1 Beginning-of-Frame Sequence 68
7.5.4.2 End-of-Frame Sequence 69
7.6 Channel Configuration Requirements 69
7.6.1 Cable Sectioning 69
7.6.2 Transceiver Placement 70
7.6.3 System Grounding 70
7.6.4 Repeaters 71
7.6.4.1 Carrier Detect and Transmit Repeat 71
7.6.4.2 Collision Detect and Collision Repeat 71
7.6.4.3 Repeater Signal Generation 71
7.6.4.3.1 Signal Amplification 71
7.6.4.3.2 Signal Timing 71
7.7 Environment Specifications 72
7.7.1 Electromagnetic Environment 72
7.7.2 Temperature and Humidity 72

Appendices

APPENDIX A: GLOSSARY 73
APPENDIX B: ASSIGNMENT OF ADDRESS AND TYPE VALUES 76
APPENDIX C: CRC IMPLEMENTATION 77
APPENDIX D: IMPLEMENTATION OF TRANSCEIVER CABLE DRIVER AND RECEIVER 80
APPENDIX E: INTERFRAME RECOVERY 82

Figures and Tables

Figure 4-1: Ethernet Architecture and Typical Implementation 7
Figure 4-2: Architectural Layering 8
Figure 4-3: Data Link Layer Functions 10
Figure 4-4: Physical Layer Functions 11
Figure 6-1: Data Link Layer Frame Format 20
Figure 6-2: Structure of the Data Link Procedural Model 31
Figure 6-3: Control Flow Summary -- Client Layer Processes 32
Figure 6-4: Control Flow Summary -- Data Link Layer Processes 33
Figure 7-1: Physical Channel Configurations 50
Table 7-1: Physical Channel Propagation Delay Budget 52
Figure 7-2: Maximum Transceiver Cable Transfer Impedance 55
Figure 7-3: Typical Transceiver Cable Waveform 57
Figure 7-4: Maximum Coaxial Cable Transfer Impedance 60
Figure 7-5: Typical Coaxial Cable Waveform 62
Figure 7-6: Manchester Encoding 65
Figure 7-7: Preamble Encoding 66
Figure 7-8: Functional Logic of **collisionDetect** Signal 67
Figure 7-9: Functional Logic of **carrierSense** Signal 68
Figure C-1: CRC Implementation 78
Figure D-1: Typical Transceiver Cable Driver 80
Figure D-2: Typical Transceiver Cable Receiver 81

1. INTRODUCTION

The Ethernet local area network provides a communication facility for high speed data exchange among computers and other digital devices located within a moderate-sized geographic area. Its primary characteristics include:

Physical Layer:

Data rate: 10 Million bits/sec

Maximum station separation: 2.5 Kilometers

Maximum number of stations: 1024

Medium: Shielded coaxial cable, base-band signalling

Topology: Branching non-rooted tree

Data Link Layer:

Link control procedure: Fully distributed peer protocol, with statistical contention resolution (CSMA/CD)

Message protocol: Variable size frames, "best-effort" delivery

The Ethernet, like other local area networks, falls in a middle ground between long distance, low speed networks which carry data for hundreds or thousands of kilometers, and specialized, very high speed interconnections which are generally limited to tens of meters. The Ethernet is intended primarily for use in such areas as office automation, distributed data processing, terminal access, and other situations requiring economical connection to a local communication medium carrying bursty traffic at high peak data rates. Use in situations demanding resistance to hostile environments, real-time response guarantees, and so on, while not specifically excluded, do not constitute the primary environment for which the Ethernet is designed.

The precursor to the Ethernet specified in this document was the "Experimental Ethernet", designed and implemented by Xerox in 1975, and used continually since that time by thousands of stations. The Ethernet defined here builds on that experience, and on the larger base of the combined experience of Digital, Intel, and Xerox in many forms of networking and computer interconnection.

In specifying the Ethernet, this document provides precise detailed definitions of the lowest two layers of an overall network architecture. It thus defines what is generally referred to as a *link-level* facility. It does not specify the higher level protocols needed to provide a complete network architecture. Such higher level protocols would generally include such functions as internetwork communication, error recovery, flow control, security measures (e.g. encryption), and other higher level functions that increase the power of the communication facility and/or tailor it to specific applications. In particular, it should be noted that all error recovery functions have been relegated to higher level protocols, in keeping with the low error rates that characterize local networks.

One of the main objectives of this specification is *compatibility.* As stated in Section 3, it is intended that *every implementation of the Ethernet be able to exchange data with every other implementation.* It should be noted that higher level protocols raise their own issues of compatibility over and above those addressed by the Ethernet and other link-level facilities. This does not eliminate the importance of link-level compatibility, however. While the compatibility provided by the Ethernet does not guarantee solutions to higher level compatibility problems, it does provide a context within which such problems can be addressed, by avoiding low level incompatibilities that would make direct communication impossible.

2. REFERENCES

The following three papers describe the Experimental Ethernet, and are reprinted in: "The Ethernet Local Network: Three Reports," Xerox Palo Alto Research Center Technical Report CSL-80-2. (February, 1980.)

[1] Metcalfe, R. M. and Boggs, D. R., "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM 19* 7 (July 1976).

[2] Crane, R. C. and Taft, E. A. "Practical Considerations in Ethernet Local Network Design," Presented at *Hawaii International Conference on System Sciences* (January, 1980).

[3] Shoch, J. F. and Hupp, J. A. "Measured Performance of an Ethernet Local Network," Presented at *Local Area Communications Network Symposium* Boston (May 1979).

The following references describe the ISO Open Systems Model:

[4] Zimmermann, H., "OSI Reference Model -- The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communication COM-28* 4 (April 1980).

[5] International Organization for Standardization (ISO), "Reference Model of Open Systems Interconnection," *Document no. ISO/TC97/SC16 N227* (June 1979).

The following references describe the Pascal language (used in the Data Link Layer procedural model) and its derivative Concurrent Pascal:

[6] Jensen, K. and Wirth, N., *Pascal User Manual and Report, 2nd Edition.* Springer-Verlag (1974).

[7] Brinch Hansen, P., *Concurrent Pascal Report.* Technical Report CIT-IS-TR 17, California Institute of Technology (1975).

The following references discuss the CRC code used for the frame check sequence:

[8] Hammond, J. L., Brown, J. E. and Liu, S. S., "Development of a Transmission Error Model and an Error Control Model," Technical Report RADC-TR-75-138, Rome Air Development Center (1975).

[9] Bittel, R., "On Frame Check Sequence (FCS) Generation and Checking.," ANSI working paper X3-S34-77-43, (1977).

3. GOALS AND NON-GOALS

This section states the assumptions underlying the design of the Ethernet.

3.1 Goals

The goals of the Ethernet design are:

Simplicity: Features which would complicate the design without substantially contributing to the meeting of the other goals have been excluded.

Low cost: Since technological improvements will continue to reduce the overall cost of stations wishing to connect to the Ethernet, the cost of the connection itself should be minimized.

Compatibility: All implementations of the Ethernet should be capable of exchanging data at the data link level. For this reason, the specification avoids optional features, to eliminate the possibility of incompatible variants of the Ethernet.

Addressing flexibility: The addressing mechanisms should provide the capability to target frames to a single node, a group of nodes, or to all nodes on the network.

Fairness: All nodes should have equal access to the network when averaged over time.

Progress: No single node operating in accordance with the protocol should be able to prevent the progress of other nodes.

High speed: The network should operate efficiently at a data rate of 10 Megabits per second.

Low delay: At any given level of offered traffic, the network should introduce as little delay as possible in the transfer of a frame.

Stability: The network should be stable under all load conditions, in the sense that the delivered traffic should be a monotonically non-decreasing function of the total offered traffic.

Maintainability: The Ethernet design should allow for network maintenance, operation, and planning.

Layered Architecture: The Ethernet design should be specified in layered terms to separate the logical aspects of the data link protocol from the physical details of the communication medium.

3.2 Non-Goals

The following are *not* goals of the Ethernet design:

Full duplex: At any given instant, the Ethernet can transfer data from one source station to one or more destination stations. Bi-directional communication is provided by rapid exhange of frames, rather than full duplex operation.

Error control: Error handling at the data link level is limited to detection of bit errors in the physical channel, and the detection and recovery from collisions. Provision of a complete error control facility to handle detected errors is relegated to higher layers of the network architecture.

Security: The data link protocol does not employ encryption or other mechanisms to provide security. Higher layers of the network architecture may provide such facilities as appropriate.

Speed flexibility: This specification defines a physical channel operating at a single fixed data rate of 10 Megabits per second.

Priority: The data link protocol provides no support of priority station operation.

Hostile user: There is no attempt to protect the network from a malicious user at the data link level.

4. FUNCTIONAL MODEL OF THE ETHERNET ARCHITECTURE

There are two important ways to view the Ethernet design, corresponding to:

Architecture, emphasizing the logical divisions of the system, and how they fit together.

Implementation, emphasizing the actual components, and their packaging and interconnection.

Figure 4-1 illustrates these two views as they apply to a typical implementation, showing how each view groups the various functions.

This document is organized along *architectural* lines, emphasizing the large-scale separation of the Ethernet system into two parts: the *Data Link Layer* and the *Physical Layer*. These layers are intended to correspond closely to the lowest layers of the ISO Model for Open Systems Interconnection [4,5]. Architectural organization of the specification has two main advantages:

Clarity: A clean overall division of the design along architectural lines makes the specification clearer.

Flexibility: Segregation of medium-dependent aspects in the Physical Layer allows the Data Link Layer to apply to transmission media other than the specified coaxial cable.

As is evident in Figure 4-1, the architectural model is based on a set of interfaces different from those emphasized in the implementations. One crucial aspect of the design, however, must be addressed largely in terms of the implementation interfaces: *compatibility*. Two important compatibility interfaces are defined within what is architecturally the Physical Layer:

Coaxial cable interface: To communicate via the Ethernet, all stations must adhere rigidly to the exact specification of coaxial cable signals defined in this document, and to the procedures which define correct behavior of a station. The medium-independent aspects of the Data Link Layer should not be taken as detracting from this point: *communication via the Ethernet requires complete compatibility at the coaxial cable interface.*

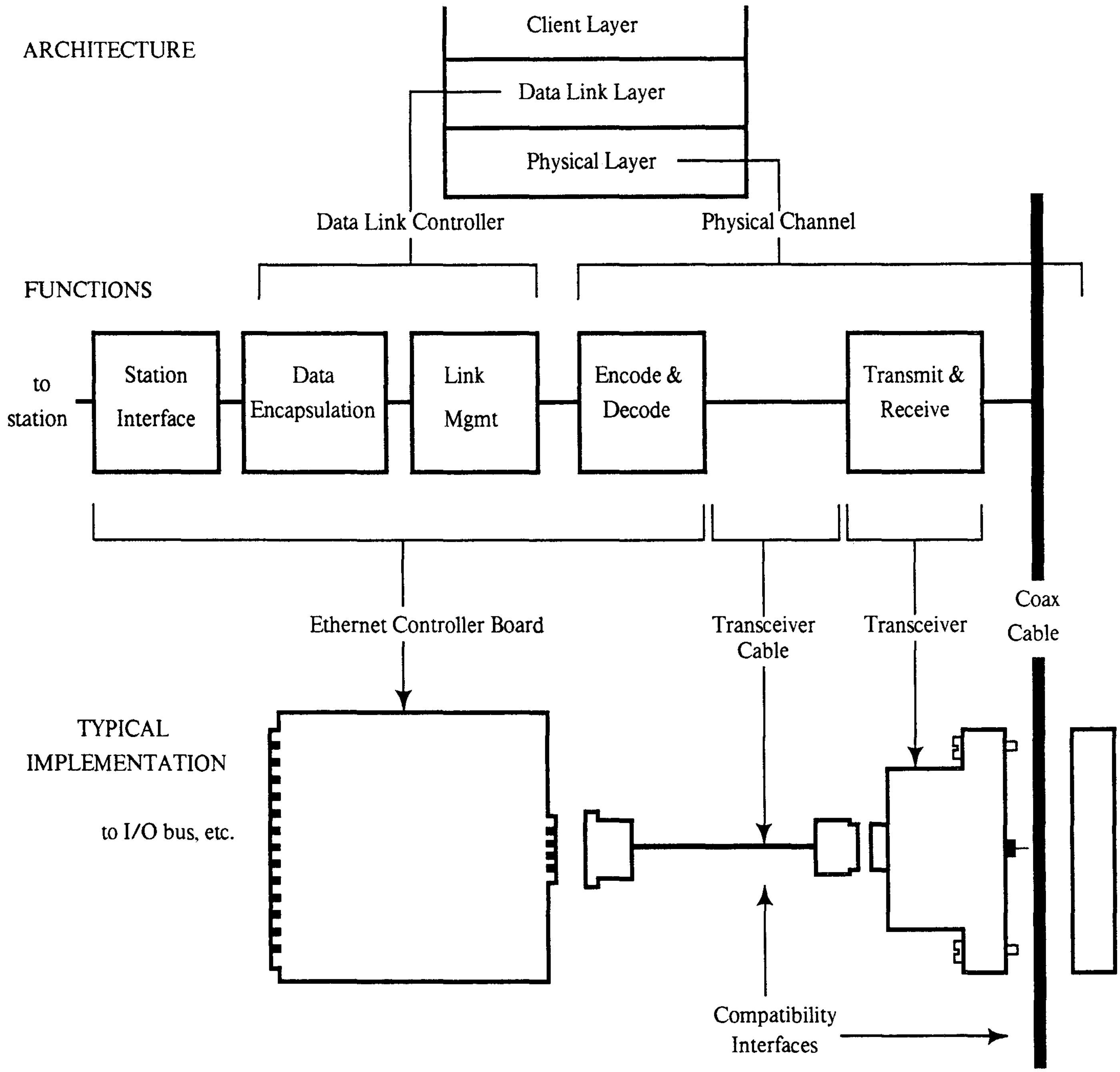


Figure 4-1: Ethernet Architecture and Typical Implementation

Transceiver cable interface: It is anticipated that most stations will be located some distance away from their connection to the coaxial cable. While it is necessary to place a small amount of circuitry (the *transceiver*) directly adjacent to the coaxial cable, the majority of the electronics (the *controller*) can and should be placed with the station. Since it is desirable for the same transceiver to be usable with a wide variety of stations, a second compatibility interface, the *transceiver cable interface*, is defined. While conformance with this interface is not strictly necessary to insure communication, it is highly recommended, since it allows maximum flexibility in intermixing transceivers and stations.

4.1 Layering

The major division in the Ethernet Architecture is between the Physical Layer and the Data Link Layer, corresponding to the lowest two levels in the ISO model. The higher levels of the overall network architecture, which use the Data Link Layer, will be collectively referred to in this document as the "Client Layer" since, strictly speaking, the identity and function of higher level facilities are outside the scope of this specification. The intent, however, is that the Ethernet Physical and Data Link Layers support the higher layers of the ISO model (Network Layer, Transport Layer, etc.).

The overall structure of the layered architecture is shown in Figure 4-2.

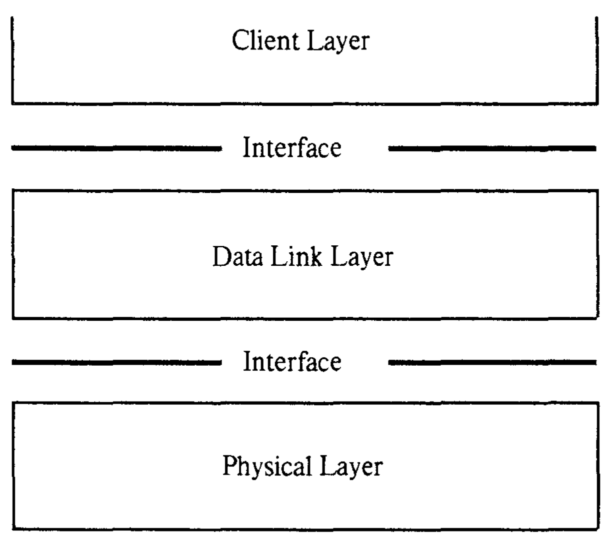


Figure 4-2: Architectural Layering

In the architectural model used here, the layers interact via well defined interfaces.

> The interface between the Client Layer and the Data Link Layer includes facilities for transmitting and receiving frames, and provides per-operation status information for use by higher-level error recovery procedures.
>
> The interface between the Data Link Layer and the Physical Layer includes signals for framing (carrier sense, transmit initiation) and contention resolution (collision detect), facilities for passing a pair of serial bit streams (transmit, receive) between the two layers, and a wait function for timing.

These interfaces are described more precisely in Section 5.

As mentioned in the preface, additional interfaces are necessary to allow a higher level network management facility to interact with the Data Link Layer and Physical Layer to perform operation, maintenance and planning functions.

4.2 Data Link Layer

The Data Link Layer defines a medium-independent link level communication facility, built on the medium-dependent physical channel provided by the Physical Layer. It is applicable to a general class of local area broadcast media suitable for use with the channel access discipline known as carrier-sense multiple-access with collision-detection (CSMA-CD). Compatibility with non-contention media (e.g., switched lines, token-passing rings, etc.), while a worthwhile topic for further research, is not addressed in this specification.

The Data Link Layer specified here is intended to be as similar as possible to that described in the ISO model. In a broadcast network like the Ethernet, the notion of a data link between two network entities does not correspond directly to a distinct physical connection. Nevertheless, the two main functions generally associated with a data link control procedure are present:

Data encapsulation

- framing (frame boundary delimitation)
- addressing (handling of source and destination addresses)
- error detection (detection of physical channel transmission errors)

Link management

- channel allocation (collision avoidance)
- contention resolution (collision handling)

This split is reflected in the division of the Data Link Layer into the Data Encapsulation sub-layer and the Link Management sub-layer, as shown in Figure 4-3.

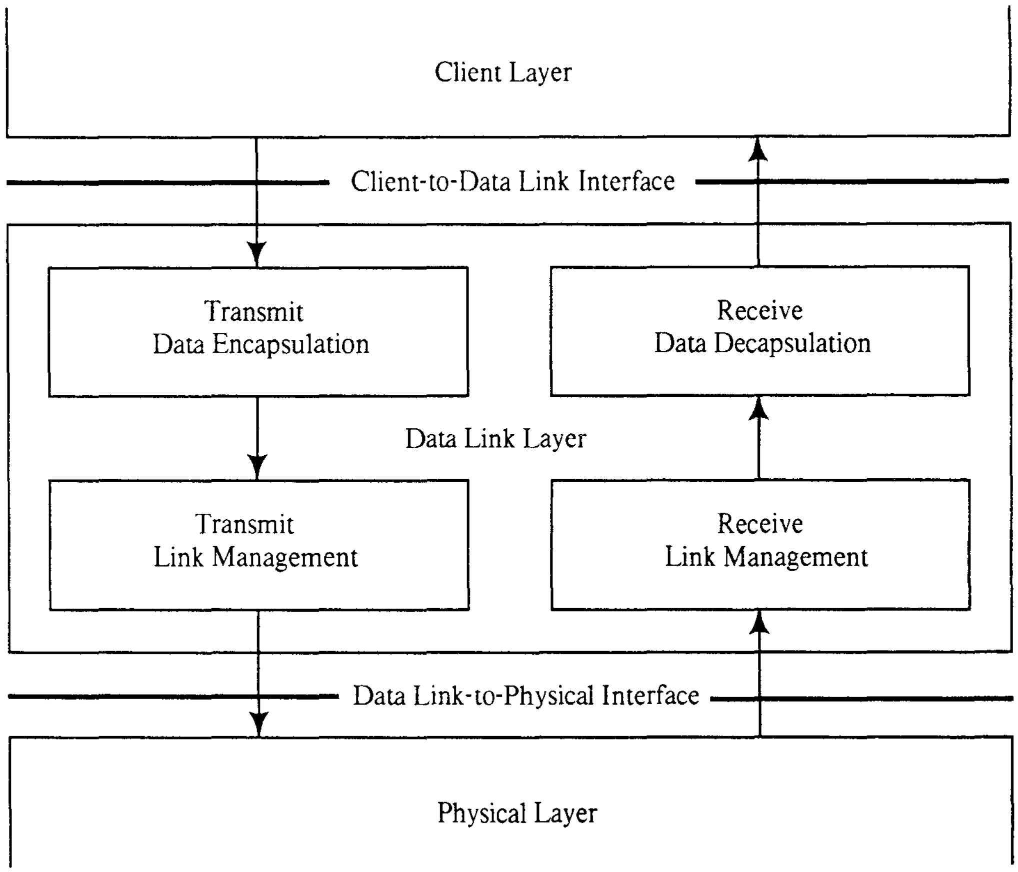


Figure 4-3: Data Link Layer Functions

In terms of the ISO model, the Ethernet Data Link Layer provides a multi-endpoint connection between higher-layer entities wishing to communicate. The connection provided is called a *data link*, and is implemented between two or more Data Link Layer entities called *data link controllers* via a Physical Layer connection called the *physical channel.*

4.3 Physical Layer

The Physical Layer specified in this document provides a 10 MBit/sec physical channel through a coaxial cable medium. Because one purpose of the layered architecture is to insulate the Data Link Layer from the medium-specific aspects of the channel, the Physical Layer completely specifies the essential physical characteristics of the Ethernet, such as data encoding, timing, voltage levels, etc. Implementation details are left unspecified, to retain maximum flexibility for the implementor. In all cases, the criterion applied in distinguishing between essential characteristics and implementation details is *guaranteed compatibility:* any two correct implementations of the Physical Layer specified here will be capable of exchanging data over the coaxial cable, enabling communication between their respective stations at the Data Link Layer.

The Physical Layer defined in this specification performs two main functions generally associated with physical channel control:

Data encoding

- preamble generation/removal (for synchronization)
- bit encoding/decoding (between binary and phase-encoded form)

Channel access

- bit transmission/reception (of encoded data)
- carrier sense (indicating traffic on the channel)
- collision detection (indicating contention on the channel)

This split is reflected in the division of the Physical Layer into the Data Encoding sub-layer and the Channel Access sub-layer, as shown in Figure 4-4.

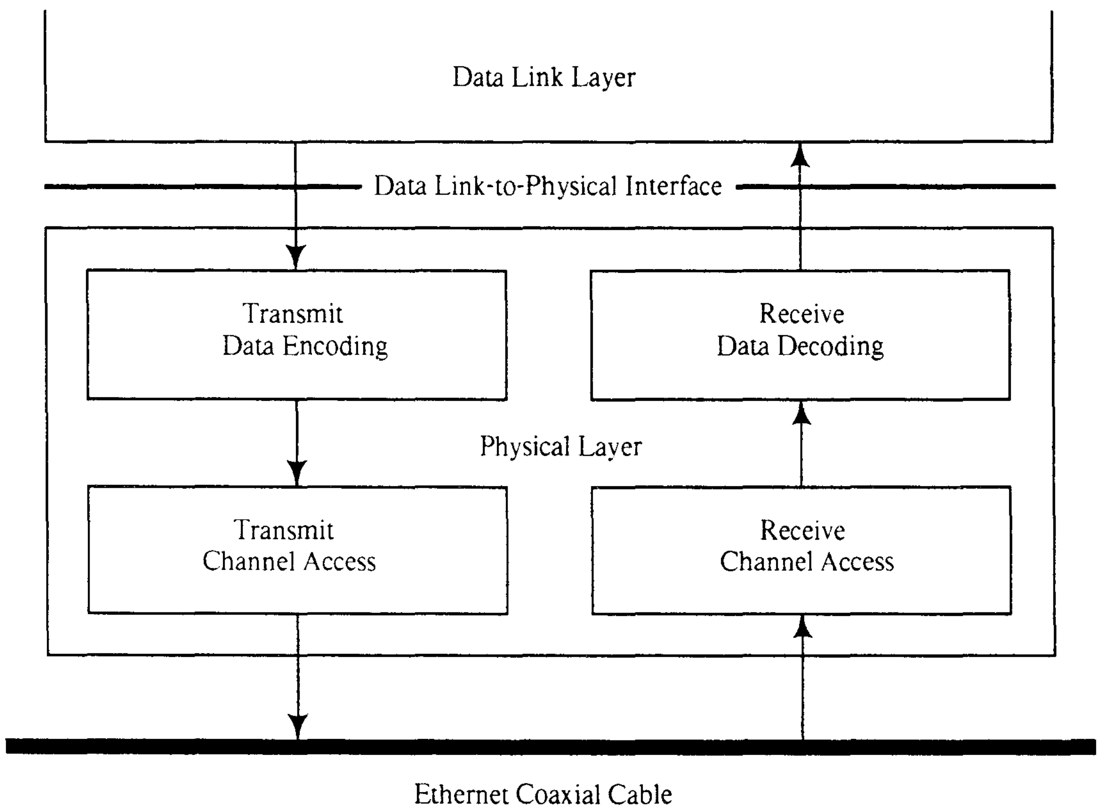


Figure 4-4: Physical Layer Functions

4.4 Ethernet Operation and the Functional Model

This section provides an overview of frame transmission and reception in terms of the functional model of the architecture. This overview is descriptive, rather than definitional; the formal specifications of the operations described here are given in Sections 6 and 7.

4.4.1 Transmission Without Contention

When the Client Layer requests the transmission of a frame, the Transmit Data Encapsulation component of the Data Link Layer constructs the frame from the client-supplied data and appends a frame check sequence to provide for error detection. The frame is then handed to the Transmit Link Management component for transmission.

Transmit Link Management attempts to avoid contention with other traffic on the channel by monitoring the *carrier sense* signal and *deferring* to passing traffic. When the channel is clear, frame transmission is initiated (after a brief interframe delay to provide recovery time for other data link controllers and for the physical channel). The Data Link Layer then provides a serial stream of bits to the Physical Layer for transmission.

The Data Encoding component of the Physical Layer, before sending the actual bits of the frame, sends an encoded preamble to allow the receivers and repeaters along the channel to synchronize their clocks and other circuitry. It then begins translating the bits of the frame into encoded form and passes them to the Channel Access component for actual transmission over the medium.

The Channel Access component performs the task of actually generating the electrical signals on the medium which represent the bits of the frame. Simultaneously, it monitors the medium and generates the *collison detect* signal, which, in the contention-free case under discussion, remains off for the duration of the frame.

When transmission has completed without contention, the Data Link Layer so informs the Client Layer and awaits the next request for frame transmission.

4.4.2 Reception Without Contention

At the receiving station, the arrival of a frame is first detected by the Receive Channel Access component of the Physical Layer, which responds by synchronizing with the incoming preamble, and by turning on the carrier sense signal. As the encoded bits arrive from the medium, they are passed to the Receive Data Decoding component.

Receive Data Decoding translates the encoded signal back into binary data and discards the leading bits, up to and including the end of the preamble. It then passes subsequent bits up to the Data Link Layer.

Meanwhile, the Receive Link Management component of the Data Link Layer, having seen carrier sense go on, has been waiting for the incoming bits to be delivered. Receive Link Management collects bits from the Physical Layer as long as the carrier sense signal remains on. When the carrier sense signal goes off, the frame is passed to Receive Data Decapsulation for processing.

Receive Data Decapsulation checks the frame's destination address field to decide whether the frame should be received by this station. If so, it passes the contents of the frame to the Client Layer along with an appropriate status code. The status code is generated by inspecting the frame check sequence to detect any damage to the frame enroute, and by checking for proper octet-boundary alignment of the end of the frame.

4.4.3 Collisions: Handling of Contention

If multiple stations attempt to transmit at the same time, it is possible for their transmitting data link controllers to interfere with each others' transmissions, in spite of their attempts to avoid this by deferring. When two stations' transmissions overlap, the resulting contention is called a *collision*. A given station can experience a collision during the initial part of its transmission (the "collision window"), before its transmitted signal has had time to propagate to all parts of the Ethernet channel. Once the collision window has passed, the station is said to have *acquired* the channel; subsequent collisions are avoided, since all other (properly functioning) stations can be assumed to have noticed the signal (via carrier sense) and to be deferring to it. The time to acquire the channel is thus based on the *round-trip propagation time* of the physical channel.

In the event of a collision, the Transmit Channel Access component of a transmitting station's Physical Layer first notices the interference on the channel and turns on the collision detect signal. This is noticed in turn by the Transmit Link Management component of the Data Link Layer, and collision handling begins. First, Transmit Link Management *enforces* the collision by transmitting a bit sequence called the *jam*. This insures that the duration of the collision is sufficient to be noticed by the other transmitting station(s) involved in the collision. After the jam is sent, Transmit Link Management terminates the transmission and schedules a retransmission attempt for a randomly selected time in the near future. Retransmission is attempted repeatedly in the face of repeated collisions. Since repeated collisions indicate a busy channel, however, Transmit Link Management attempts to adjust to the channel load by *backing off* (voluntarily delaying its own retransmissions to reduce its load on the channel). This is accomplished by expanding the interval from which the random retransmission time is selected on each retransmission attempt. Eventually, either the transmission succeeds, or the attempt is abandoned on the assumption that the channel has failed or has become overloaded.

At the receiving end, the bits resulting from a collision are received and decoded by the Physical Layer just as are the bits of a valid frame. In particular, collisions do *not* turn on the receiving station's collision detect signal, which is generated only during transmission. Instead, the fragmentary frames received during collisions are distinguished from valid frames by the Data Link's Receive Link Management component, by noting that a collision fragment is always smaller than the shortest valid frame. Such fragments are discarded by Receive Link Management.

5. INTER-LAYER INTERFACES

The purpose of this section is to provide precise definitions of the interfaces between the architectural layers defined in Section 4. In order to provide such a definition, some precise notation must be adopted. The notation used here is the Pascal language, in keeping with the procedural nature of the formal Data Link Layer specification (see 6.5). Each interface is thus described as a set of procedures and/or shared variables which collectively provide the only valid interactions between layers. The accompanying text describes the meaning of each procedure or variable and points out any implicit interactions among them.

Note that the description of the interfaces in Pascal is a notational technique, and in no way implies that they can or should be implemented in software. This point is discussed more fully in 6.5, which provides complete Pascal declarations for the data types used in the remainder of this section. Note also that the "synchronous" (one frame at a time) nature of the frame transmission and reception operations is a property of the architectural interface between the Client Layer and the Data Link Layer, and need not be reflected in the implementation interface between a station and its controller.

5.1 Client Layer to Data Link Layer

The two primary services provided to the Client Layer by the Data Link Layer are transmission and reception of frames. The interface through which the Client Layer uses the facilities of the Data Link Layer therefore consists of a pair of functions.

Functions:
TransmitFrame
ReceiveFrame

Each of these functions has the components of a frame as its parameters (input or output), and returns a status code as its result.

The Client Layer transmits a frame by invoking *TransmitFrame*:

```
function TransmitFrame (
  destinationParam: AddressValue;
  sourceParam: AddressValue;
  typeParam: TypeValue;
  dataParam: DataValue): TransmitStatus;
```

The *TransmitFrame* operation is synchronous, in the sense that its duration is the entire attempt to transmit the frame, so that when the operation completes, transmission has either succeeded or failed, as indicated by the resulting status code:

```
type TransmitStatus = (transmitOK, excessiveCollisionError);
```

Successful transmission is indicated by the status code *transmitOK*; the code *excessiveCollisionError* indicates that the transmission attempt was aborted due to excessive collisions, because of heavy traffic or a network failure. Implementations may define additional implementation-dependent status codes if necessary.

The Client Layer accepts incoming frames by invoking *ReceiveFrame*:

```
function ReceiveFrame (
  var destinationParam: AddressValue;
  var sourceParam: AddressValue;
  var typeParam: TypeValue;
  var dataParam: DataValue): ReceiveStatus;
```

The *ReceiveFrame* operation is synchronous, in the sense that the operation does not complete until a frame has been received. The fields of the frame are delivered via the output parameters, along with a status code:

```
type ReceiveStatus = (receiveOK, frameCheckError, alignmentError);
```

Successful reception is indicated by the status code *receiveOK*. The code *frameCheckError* indicates that the frame received was damaged by a transmission error in the physical channel. The code *alignmentError* indicates that the frame received was damaged, and that in addition, its length was not an integral number of octets. Implementations may define additional implementation-dependent status codes if necessary.

5.2 Data Link Layer to Physical Layer

The interface through which the Data Link Layer uses the facilities of the Physical Layer consists of a function, a pair of procedures and three Boolean variables.

Function:	Variables:
ReceiveBit	*collisionDetect*
Procedures:	*carrierSense*
TransmitBit	*transmitting*
Wait	

During transmission, the contents of an outgoing frame are passed from the Data Link Layer to the Physical Layer via repeated use of the *TransmitBit* operation:

```
procedure TransmitBit (bitParam: Bit);
```

Each invocation of *TransmitBit* passes one new bit of the outgoing frame to the Physical Layer. The *TransmitBit* operation is synchronous, in the sense that the duration of the operation is the entire transmission of the bit, so that when the operation completes, the Physical Layer is ready to accept the next bit immediately. (Note: this does not imply that all invocations of *TransmitBit* are of exactly equal duration; for example, if the Physical Layer must perform some initial processing -- e.g., preamble generation -- before transmitting the first bit of a frame, the first invocation of *TransmitBit* may take significantly longer.)

The overall event of data being transmitted is signaled to the Physical Layer via the variable *transmitting*:

```
var transmitting: Boolean;
```

Before sending the first bit of a frame, the Data Link Layer sets *transmitting* to *true*, to inform the Physical Link that a stream of bits will be presented via the *TransmitBit* operation. After the last bit of the frame has been presented, the Data Link Layer sets *transmitting* to *false* to indicate the end of the frame.

The presence of a collision in the physical channel is signaled to the Data Link Layer via the variable *collisionDetect*:

```
var collisionDetect: Boolean;
```

The *collisionDetect* signal remains *true* during the duration of the collision. (Note: Since an entire collision may occur during the first invocation of *TransmitBit* -- e.g., during preamble removal -- the Data Link Layer must handle this possibility by monitoring *collisionDetect* concurrently with its transmission of outgoing bits. See 6.5 for details.)

The *collisionDetect* signal is generated only during transmission and is never *true* at any other time; in particular, it cannot be used during frame reception to detect collisions between overlapping transmissions from two or more other stations.

During reception, the contents of an incoming frame are retrieved from the Physical Layer by the Data Link Layer via repeated use of the *ReceiveBit* operation:

```
function ReceiveBit: Bit;
```

Each invocation of *ReceiveBit* retrieves one new bit of the incoming frame (i.e., not including any preamble bits) from the Physical Layer. The *ReceiveBit* operation is synchronous, in the sense that its duration is the entire reception of a single bit. (As with *TransmitBit*, the first invocation of *ReceiveBit* make take significantly longer -- e.g., due to preamble removal). Upon receiving a bit, the Data Link Layer must immediately request the next bit until all bits of the frame have have been received. (See 6.5 for details.)

The overall event of data being received is signaled to the Data Link Layer via the variable *carrierSense*:

```
var carrierSense: Boolean;
```

When the Physical Layer sets *carrierSense* to *true*, the Data Link Layer must immediately begin retrieving the incoming bits via the *ReceiveBit* operation. When *carrierSense* subsequently becomes *false*, the Data Link Layer can begin processing the received bits as a completed frame. Note that the *true/false* transitions of *carrierSense* are not defined to be precisely synchronized with the beginning and end of the frame, but may precede the beginning and lag the end, respectively. If an invocation of *ReceiveBit* is pending when *carrierSense* becomes *false*, *ReceiveBit* returns an undefined value, which should be discarded by the Data Link Layer. (See 6.5 for details.)

The Data Link Layer must also monitor the value of *carrierSense* to defer its own transmissions when the channel is busy.

The Physical Layer also provides the procedure *Wait*:

```
procedure Wait (bitTimes: integer);
```

This procedure waits for the specified number of bit times. This allows the Data Link Layer to measure time intervals in units of the (physical-channel-dependent) bit time.

Another important property of the Physical Layer which is an implicit part of the interface presented to the Data Link Layer is the *round-trip propagation time* of the physical channel. This figure represents the maximum time required for a signal to propagate from one end of the network to the other, and for a collision to propagate back. The round-trip propagation time is primarily (but not entirely) a function of the physical size of the network. The round-trip propagation time of the Physical Layer is defined to be at most 450 bit times (see 7.1.2).

6. ETHERNET DATA LINK LAYER SPECIFICATION

6.1 Data Link Layer Overview and Model

As defined in Section 4, the Ethernet Architecture consists of the Data Link Layer, and below it, the Physical Layer. Furthermore, the Data Link Layer is divided into two sub-layers (see Figure 4-3).

Data encapsulation
- framing
- addressing
- error detection

Link management
- channel allocation
- contention resolution

This model is used throughout this section to structure the detailed specification of the Data Link Layer. An English description of the Data Link Layer is given in 6.2, 6.3, and 6.4. A more precise algorithmic definition is given in 6.5, which provides a procedural model for the Data Link Layer in the form of a program in the language Pascal. Note that whenever there is any apparent ambiguity concerning the definition of some aspect of the Data Link Layer, it is the Pascal procedural specification in 6.5 which should be consulted for the definitive statement.

6.2 Frame Format

The data encapsulation function of the Data Link Layer comprises the construction and processing of frames. The subfunctions of framing, addressing, and error detection are reflected in the frame format as follows:

Framing: No explicit framing information is needed, since the necessary framing cues (*carrierSense* and *transmitting*) are present in the interface to the Physical Layer.

Addressing: Two address fields are provided to identify the source and destination stations for the frame.

Error detection: A Frame Check Sequence field is provided for detection of transmission errors.

Figure 6-1 shows the five fields of a frame: the addresses of the frame's source and destination, a type field for use by higher layers (see 6.2.2), a data field containing the transmitted data, and the frame check sequence field containing a cyclic redundancy check value to detect transmission errors. Of these five fields, all are of fixed size except the data field, which may contain any integral number of octets between the minimum and maximum values specified below (see 6.2.5).

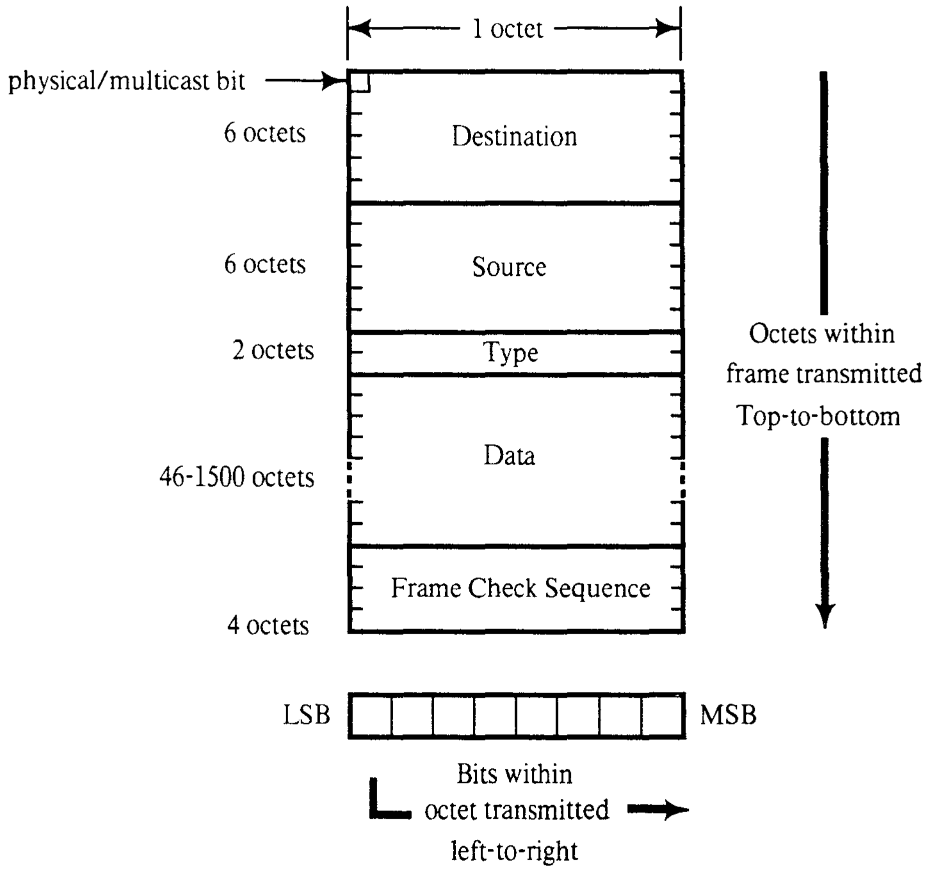


Figure 6-1: Data Link Layer Frame Format

Relative to Figure 6-1, the octets of a frame are transmitted from top to bottom, and the bits of each octet are transmitted from left to right.

NOTE

This document does *not* define an order of transmission for the octets of standard multi-octet data types (strings, integers, etc), since no values of such data types appear in the data link frame format. The order in which implementations of the Ethernet store the octets of a frame in computer memory, and the manner in which higher level protocols interpret the contents of the data field as values of various multi-octet data types, are beyond the scope of this specification.

The Ethernet itself is also totally insensitive to the interpretation of bits within a octet as constituting the digits of an 8-digit binary numeric value. Since some uniform convention is helpful, however, in avoiding needless incompatibility among different station types, the interpretation is arbitrarily defined to be that the left-most bit (first transmitted) is the low-order (2^0) digit and the right-most bit (last transmitted) is the high-order (2^7) digit.

6.2.1 Address Fields

Data link addresses are 6 octets (48 bits) in length. A data link address is of one of two types:

Physical address: The unique address associated with a particular station on the Ethernet. A station's physical address should be distinct from the physical address of any other station on *any* Ethernet.

Multicast address: A multi-destination address, associated with one or more stations on a given Ethernet. There are two kinds of multicast address:

- *Multicast-group address:* An address associated by higher-level convention with a group of logically related stations.
- *Broadcast address:* A distinguished, predefined multicast address which always denotes the set of *all* stations on a given Ethernet.

The first bit of a data link address distinguishes physical from multicast addresses:

$0 \Rightarrow$ physical address

$1 \Rightarrow$ multicast address

In either case, the remainder of the first octet and all of the subsequent octets form a 47-bit pattern. In the case of the broadcast address, this pattern consists of 47 one-bits. There is no standard "null" address value.

The procedures for assigning suitably unique values for physical and multicast addresses are discussed in Appendix B.

6.2.1.1 Destination Address Field

The destination address field specifies the station(s) for which the frame is intended. It may be a physical or multicast (including broadcast) address. For details of address recognition by the receiving station(s), see 6.4.1.2.

6.2.1.2 Source Address Field

The source address field specifies the station sending the frame. The source address field is not interpreted at the Data Link Layer. It is specified at the data link level because a uniform convention for the placement of this field is crucial for most higher level protocols.

6.2.2 Type Field

The type field consists of a two-octet value reserved for use by higher levels (in particular, to identify the Client Layer protocol associated with the frame). The type field is uninterpreted at the Data Link Layer. It is specified at this level because a uniform convention for the placement and value assignment of this field is crucial if multiple higher level protocols are to be able to share the same Ethernet network without conflict. Appendix B discusses the assignment of type field values.

6.2.3 Data Field

The data field contains a sequence of n octets, where $46 \leq n \leq 1500$ Within this range, full data transparency is provided, in the sense that any arbitrary sequence of octet values may appear in the data field.

6.2.4 Frame Check Sequence Field

The frame check sequence (FCS) field contains a 4-octet (32-bit) cyclic redundancy check (CRC) value. This value is computed as a function of the contents of the source, destination, type and data fields (i.e., all fields except the frame check sequence field itself). The encoding is defined by the generating polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

(This polynomial is also used in the Autodin-II network; its properties are investigated in [8].)

Mathematically, the CRC value corresponding to a given frame is defined by the following procedure:

1. The first 32 bits of the frame are complemented.
2. The n bits of the frame are then considered to be the coefficients of a polynomial M(x) of degree n-1. (The first bit of the destination address field corresponds to the x^{n-1} term and the last bit of the data field corresponds to the x^0 term.)
3. M(x) is multiplied by x^{32} and divided by G(x), producing a remainder R(x) of degree ≤ 31.
4. The coefficients of R(x) are considered to be a 32-bit sequence.
5. The bit sequence is complemented and the result is the CRC.

The 32 bits of the CRC value are placed in the frame check sequence field so that the x^{31} term is the leftmost bit of the first octet, and the x^0 term is the rightmost bit of the last octet. (The bits of the CRC are thus transmitted in the order $x^{31}, x^{30},...,x^1, x^0$)

Appendix C discusses CRC implementation issues.

6.2.5 Frame Size Limitations

Given the limitations on the size of the data field specified in 6.2.3 and the 18 octet total size for the other four fields, the smallest valid frame contains 64 octets and the largest valid frame contains 1518 octets.

6.3 Frame Transmission

The Data Link frame transmission and reception are as follows:

Frame transmission includes data encapsulation and link management aspects:

Transmit Data Encapsulation includes the assembly of the outgoing frame (from the values provided by the Client Layer) and frame check sequence generation.

Transmit Link Management includes carrier deference, interframe spacing, collision detection and enforcement, and collision backoff and retransmission.

The performance of these functions by a transmitting data link controller interacts with corresponding actions by other data link controllers to jointly implement the Ethernet data link protocol.

6.3.1 Transmit Data Encapsulation

6.3.1.1 Frame Assembly

The fields of the data link frame are set to the values provided by the Client Layer as arguments to the *TransmitFrame* operation (see 5.1), with the exception of the frame check sequence, which is set to the CRC value generated by the data link controller.

6.3.1.2 Frame Check Sequence Generation

The CRC value defined in 6.2.4 is generated and inserted in the frame check sequence field, following the fields supplied by the Client Layer. Appendix C discusses CRC implementation.

6.3.2 Transmit Link Management

6.3.2.1 Carrier Deference

Even when it has nothing to transmit, the data link controller monitors the physical channel for traffic by watching the *carrierSense* signal provided by the Physical Layer. Whenever the channel is busy, the data link controller *defers* to the passing frame by delaying any pending transmission of its own. After the last bit of the passing frame (i.e., when *carrierSense* changes from true to false), the data link controller continues to defer for 9.6 μsec to provide proper interframe spacing (see 6.3.2.2). At the end of that time, if it has a frame waiting to be transmitted, transmission is initiated independent of the value of *carrierSense*. When transmission has completed (or immediately, if there was nothing to transmit) the data link controller resumes its original monitoring of *carrierSense*.

When a frame is submitted by the Client Layer for transmission, the transmission is initiated as soon as possible, but in conformance with the rules of deference stated above.

6.3.2.2 Interframe Spacing

As defined in 6.3.2.1, the rules for deferring to passing frames insure a minimum interframe spacing of 9.6 μsec. This is intended to provide interframe recovery time for other data link controllers and for the physical channel.

Note that 9.6 μsec is the minimum value of the interframe spacing. If necessary for implementation reasons, a transmitting controller may use a larger value with a resulting decrease in its throughput. The value should not exceed 10.6 μsec.

6.3.2.3 Collision Handling

Once a data link controller has finished deferring and has started transmission, it is still possible for it to experience contention for the channel. As discussed in 4.4.3, collisions can occur until acquisition of the network has been accomplished through the deference of all other stations' data link controllers.

The dynamics of collision handling are largely determined by a single parameter called the *slot time*. This single parameter describes three important aspects of collision handling:

- It is an upper bound on the acquisition time of the network.
- It is an upper bound on the length of a frame fragment generated by a collision. (See 6.4.2.1)
- It is the scheduling quantum for retransmission. (See 6.3.2.3.2)

In order to fulfill all three functions, the slot time must be larger than the sum of the Physical Layer round-trip propagation time (450 bit times; see 7.1.2) and the Data Link Layer maximum jam time (48 bit times, see 6.3.2.3.1). The slot time is defined to be 512 bit times.

6.3.2.3.1 Collision Detection and Enforcement

Collisions are detected by monitoring the *collisionDetect* signal provided by the Physical Layer. When a collision is detected during a frame transmission, the transmission is not terminated immediately. Instead, the transmission continues until at least 32 (but not more than 48) additional bits have been transmitted (counting from the time *collisionDetect* went on). This collision enforcement or "jam" guarantees that the duration of the collision is sufficient to insure its detection by all transmitting stations on the network. The content of the jam is unspecified; it may be any fixed or variable pattern convenient to the data link controller implementation, but should not be the 32-bit CRC value corresponding to the (partial) frame transmitted prior to the jam.

6.3.2.3.2 Collision Backoff and Retransmission

When a transmission attempt has terminated due to a collision, it is retried by the transmitting data link controller until either it is successful, or 16 attempts (the original attempt plus 15 retries) have been made and all have terminated due to collisions. Note that all attempts to transmit a given frame are completed before any subsequent outgoing frames are transmitted. The scheduling of the retransmissions is determined by a controlled randomization process called "truncated binary exponential backoff". At the end of enforcing a collision (jamming), the data link controller delays before attempting to retransmit the frame. The delay is an integral multiple of the slot time. (See 6.3.2.3). The number of slot times to delay before the n^{th} retransmission attempt is chosen as a uniformly distributed random integer r in the range $0 \leq r < 2^k$ where $k = \min(n, 10)$. If all 16 attempts fail, this event is reported as an error.

Note that the values given above define the most aggressive behavior that a station may exhibit in attempting to retransmit after a collision. In the course of implementing the retransmission scheduling procedure, a station may introduce extra delays which will degrade its own throughput, but in no case may a station's retransmission scheduling result in a lower average delay between retransmission attempts than the procedure defined above.

6.4 Frame Reception

Frame reception includes both data decapsulation and link management aspects:

> *Receive Data Decapsulation* comprises framing, address recognition, frame check sequence validation, and frame disassembly to pass the fields of the received frame to the Client Layer.
>
> *Receive Link Management's* main function is the filtering of collision fragments from complete incoming frames.

The performance of these functions by a receiving data link controller interacts with corresponding actions by other data link controllers to jointly implement the Ethernet data link protocol.

6.4.1 Receive Data Decapsulation

6.4.1.1 Framing

The data link controller recognizes the boundaries of an incoming frame by monitoring the *carrierSense* signal provided by the Physical Layer. There are two possible length errors that can occur, which indicate ill-framed data: the frame may be too long, or its length may not be an integral number of octets.

6.4.1.1.1 Maximum Frame Size

The receiving data link controller is not required to enforce the frame size limit specified in 6.2.5, but it is allowed to truncate frames longer than 1518 octets and report this event as an (implementation-dependent) error.

6.4.1.1.2 Integral Number of Octets in Frame

Since the format of a valid frame specifies an integral number of octets, only a collision or an error can produce a frame with a length that is not an integral multiple of 8. Complete frames (i.e., not rejected as collision fragments; see 6.4.2.1) that do not contain an integral number of octets are truncated to the nearest octet boundary. If frame check sequence validation (see 6.4.1.3) detects an error in such a frame, the status code *alignmentError* is reported.

6.4.1.2 Address Recognition

The Ethernet data link controller is capable of recognizing physical and multicast addresses, as defined in 6.2.1.

6.4.1.2.1 Physical Addresses

The data link controller recognizes and accepts any frame whose destination field contains the physical address of the station.

The physical address of each station is set by network management to a unique value associated with the station, and distinct from the address of any other station on any Ethernet. The setting of the station's physical address by network management allows multiple data link controllers connected to single station all to respond to the same physical address. The procedures for allocating unique addresses are discussed in Appendix B.

6.4.1.2.2 Multicast Addresses

The data link controller recognizes and accepts any frame whose destination field contains the broadcast address.

The data link controller is capable of activating some number of multicast-group addresses as specified by higher layers. The data link controller recognizes and accepts any frame whose destination field contains an active multicast-group address. An active multicast-group address may be deactivated.

6.4.1.3 Frame Check Sequence Validation

FCS validation is essentially identical to FCS generation. If the bits of the incoming frame (exclusive of the FCS field itself) do not generate a CRC value identical to the one received, an error has occurred and is reported as such. Implementation issues are discussed in Appendix C.

6.4.1.4 Frame Disassembly

The frame is disassembled and the fields are passed to the Client Layer via the output parameters of the *ReceiveFrame* operation (see 5.1).

6.4.2 Receive Link Management

6.4.2.1 Collision Filtering

As specified in 6.2.5, the smallest valid frame must contain at least 64 octets. Any frame containing less than 64 octets is presumed to be a fragment resulting from a collision and is discarded by the receiving data link controller. Since occasional collisions are a normal part of the link management procedure, the discarding of such a fragment is not reported as an error to the Client Layer.

6.5 The Data Link Layer Procedural Model

6.5.1 Overview of the Procedural Model

The functions of the Ethernet Data Link Layer are presented below, modeled as a program written in the language Pascal [6]. This procedural model is intended as the primary specification of the functions to be provided in any Ethernet Data Link Layer implementation. It is important to distinguish, however, between the model and a real implementation. The model is optimized for simplicity and clarity of presentation, while any realistic implementation must place heavier emphasis on such constraints as efficiency and suitability to a particular implementation technology or computer architecture. In this context, several important properties of the procedural model must be considered.

6.5.1.1 Ground Rules for the Procedural Model

a) First, it must be emphasized that the description of the Data Link Layer in a programming language is in no way intended to imply that a data link controller must be implemented as a program executed by a computer. The implementation may consist of any appropriate technology including hardware, firmware, software, or any combination.

b) Similarly, it must be emphasized that it is the *behavior* of Data Link Layer implementations that must match the specification, *not* their internal structure. The internal details of the procedural model are useful only to the extent that they help specify that behavior clearly and precisely.

c) The handling of incoming and outgoing frames is rather stylized in the procedural model, in the sense that frames are handled as single entities by most of the Data Link Layer and are only serialized for presentation to the Physical Layer. In reality, many data link controller implementations will instead handle frames serially on a bit, octet or word basis. A serial implementation would typically perform the required functions (address recognition, frame check sequence generation/validation, etc.) in an overlapped, pipelined fashion. This approach has not been reflected in the procedural model, since this would only complicate the description of the functions without changing them in any way.

d) The model consists of algorithms designed to be executed by a number of concurrent processes; these algorithms collectively implement the Ethernet data link control procedure. The timing dependencies introduced by the need for concurrent activity are resolved in two ways:

- *Processes vs. External events:* It is assumed that the algorithms are executed "very fast" relative to external events, in the sense that a process never falls behind in its work and fails to respond to an external event in a timely manner. For example, when a frame is to be received, it is assumed that the data link procedure *ReceiveFrame* is always called well before the frame in question has started to arrive.

- *Processes vs. Processes:* Among processes, no assumptions are made about relative speeds of execution. This means that each interaction between two processes must be structured to work correctly independent of their respective speeds. Note, however, that the timing of interactions among processes is often, in part, an indirect reflection of the timing of external events, in which case appropriate timing assumptions may still be made.

It is intended that the concurrency in the model reflect the parallelism intrinsic to the task of implementing the Ethernet data link, although the actual parallel structure of the implementations is likely to vary.

6.5.1.2 Use of Pascal in the Procedural Model

Pascal was chosen for the procedural model because of its relative simplicity and clarity, and its general acceptance.

Several observations need to be made about the way in which Pascal is used for the model, including:

a) Some limitations of the language have been circumvented in order to simplify the specification:

1) The elements of the program (variables, procedures, etc) are presented in logical groupings, in top-down order. Certain Pascal ordering restrictions have thus been circumvented to improve readability.

2) The **process** and **cycle** constructs of the Pascal derivative Concurrent Pascal [7] have been introduced to indicate the sites of autonomous concurrent activity. As used here, a process is simply a parameterless procedure that begins execution at "the beginning of time" rather than being invoked by a procedure call. A cycle statement represents the main body of a process and is executed repeatedly forever.

3) The lack of variable array bounds in the language has been circumvented by treating frames as if they are always of a single fixed size (which is never actually specified). In fact, of course, the size of a frame depends on the size of its data field, hence the value of the "pseudo-constant" *frameSize* should be thought of as varying in the long-term, even though it is fixed for any given frame.

4) The use of a variant record to represent a frame (both as fields and as bits) follows the letter but not the spirit of the Pascal Report, since it allows the underlying representation to be viewed as two different data types. (It also assumes that this representation is as shown in Figure 6-1.)

b) The model makes no use of any explicit interprocess synchronization primitives. Instead, all interprocess interaction is done via carefully stylized manipulation of shared variables. For example, some variables are set by only one process and inspected by another process in such a manner that the net result is independent of their execution speeds. While such techniques are not generally suitable for the construction of large concurrent programs, they simplify the model and more nearly resemble the methods appropriate to the most likely implementation technologies (e.g. microcode, hardware state-machines, etc.)

6.5.2 Procedural Model

The procedural model used here is based on five cooperating concurrent processes. Of these, three are actually defined in the Data Link Layer. The remaining two processes are provided by the Client Layer and utilize the interface operations provided by the Data Link Layer. The five processes are thus:

Client Layer:

Frame Transmitter Process **Frame Receiver Process**

Data Link Layer:

Bit Transmitter Process **Bit Receiver Process**

Deference Process

This organization of the model is illustrated in Figure 6-2, and reflects the fact that the communication of entire frames is initiated by the Client Layer, while the timing of collision backoff and of individual bit transfers is based on interactions between the Data Link Layer and the Physical-Layer-dependent bit-time.

Figure 6-2 depicts the static structure of the procedural model, showing how the various processes and procedures interact by invoking each other. Figures 6-3 and 6-4 summarize the dynamic behavior of the model during transmission and reception, focusing on the steps that must be performed, rather than the procedural structure which performs them. The usage of the shared state variables is not depicted in the figures, but is described in the comments in 6.5.2.1.

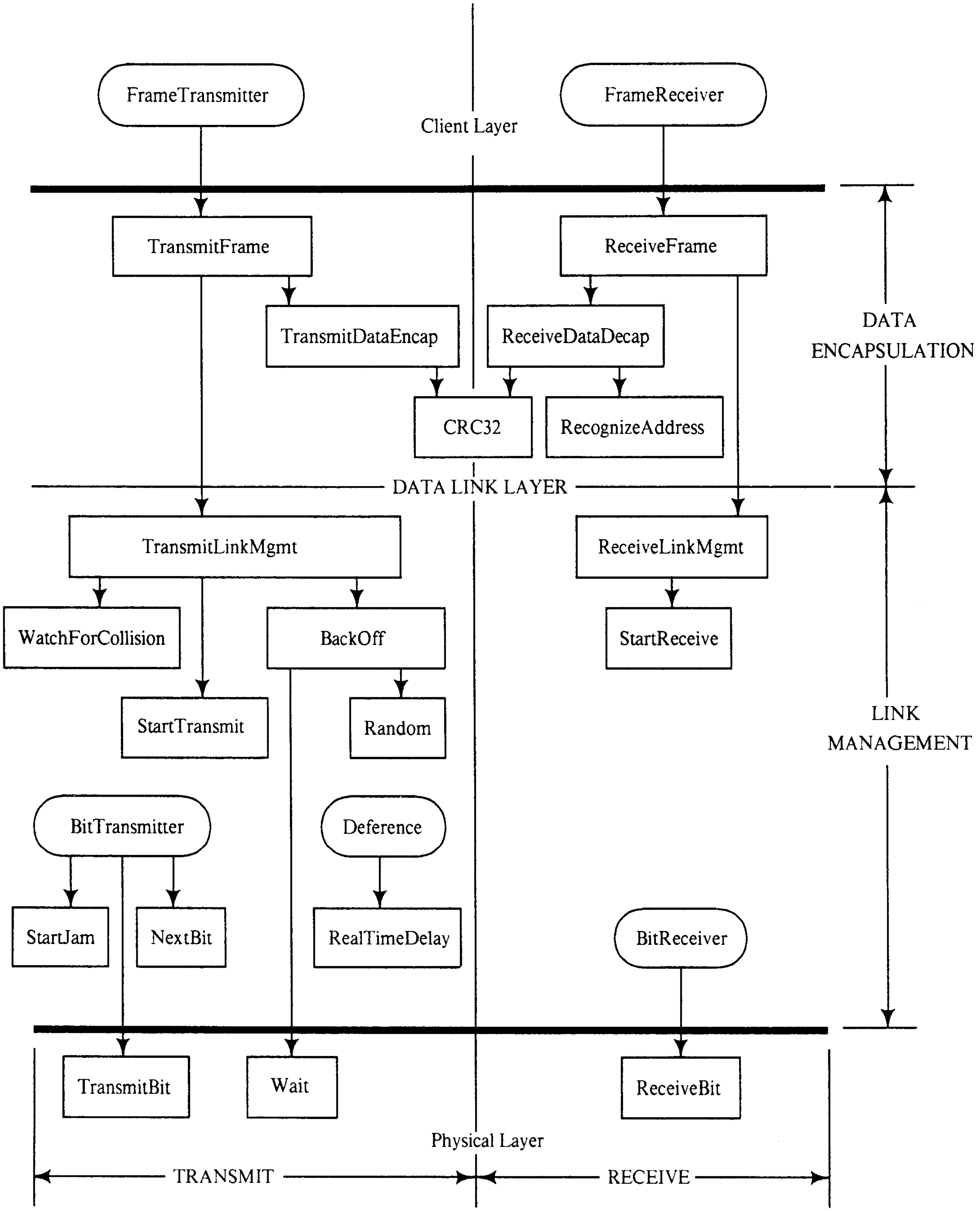


Figure 6-2: Structure of the Data Link Procedural Model

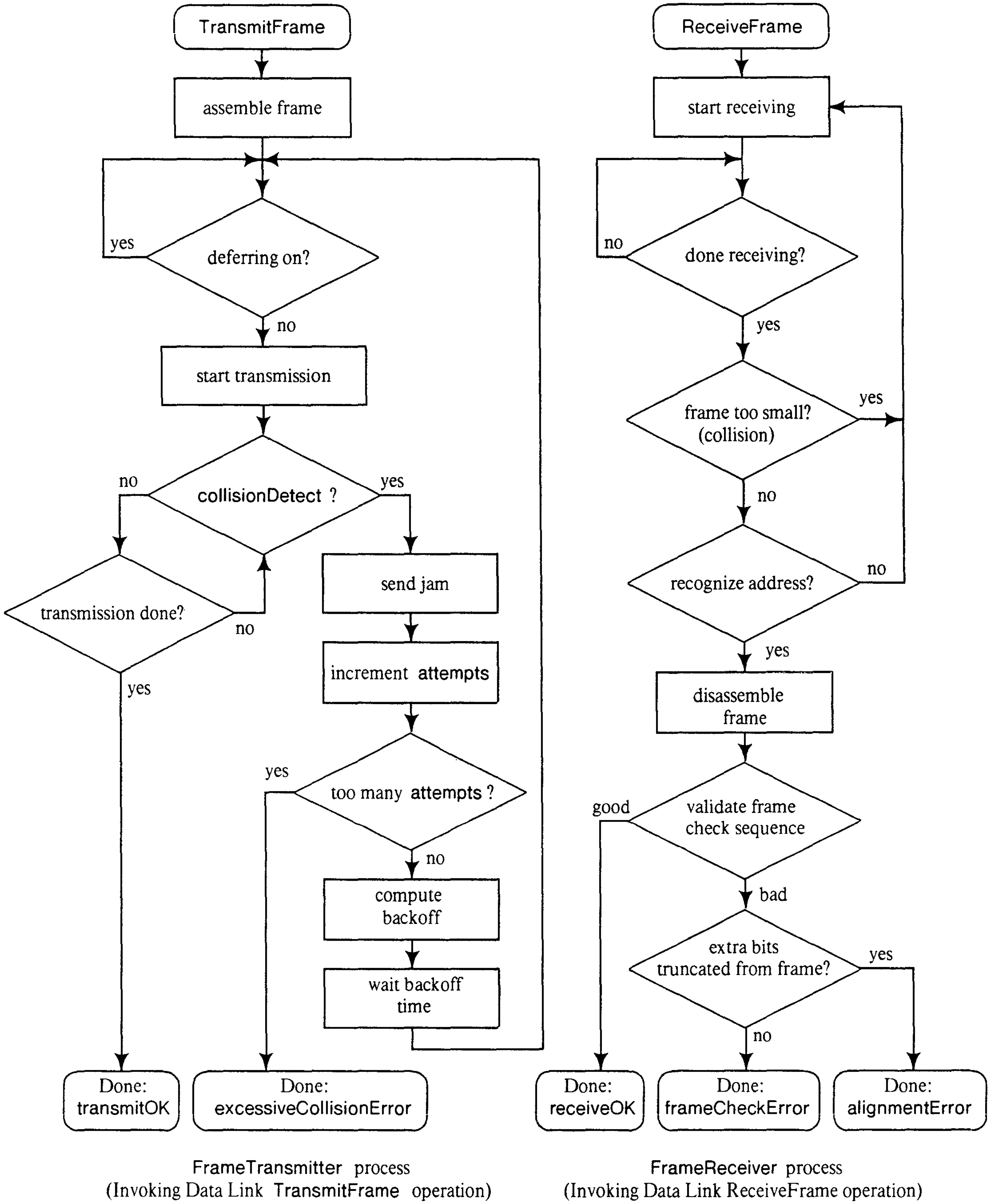


Figure 6-3: Control Flow Summary -- Client Layer Processes

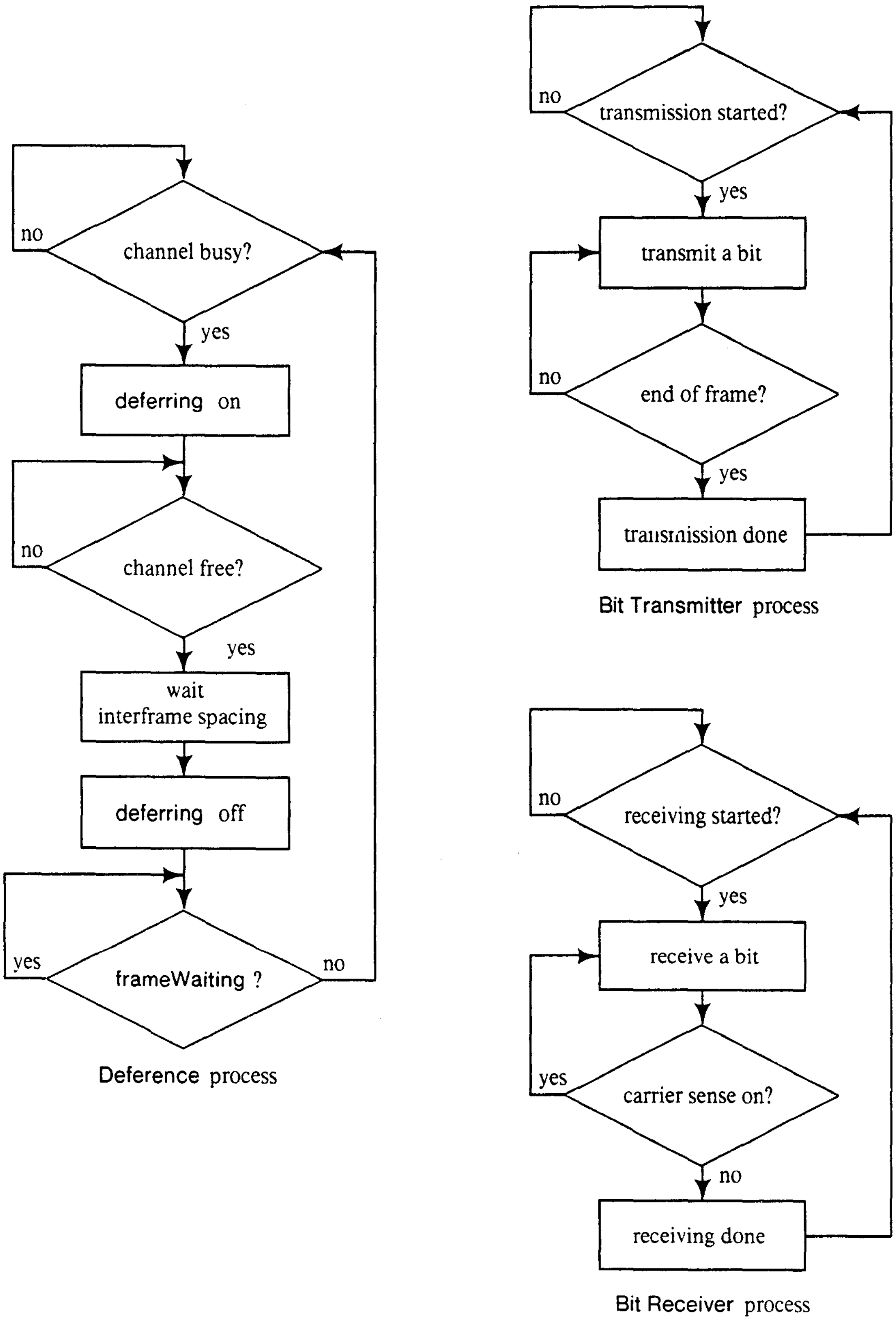


Figure 6-4: Control Flow Summary -- Data Link Layer Processes

6.5.2.1 Global Declarations

6.5.2.1.1 Common Constants and Types

The following declarations of constants and types are used by the frame transmission and reception sections of each data link controller:

```
const
    addressSize = 48; {48 bit address = 6 octets}
    typeSize = 16; {16 bit protocol type = 2 octets}
    dataSize = ...; {see 6.5.1.2, note 3}
    crcSize = 32; {32 bit CRC = 4 octets}
    frameSize = ...; { = 2*addressSize + typeSize + dataSize + crcSize...see 6.5.1.2,
    note 3}

    slotTime = 512; {unit of time for collision handling}

type
    Bit = 0..1;
    AddressValue = array [1..addressSize] of Bit;
    TypeValue = array [1..typeSize] of Bit;
    DataValue = array [1..dataSize] of Bit;
    CRCValue = array [1..crcSize] of Bit;

    ViewPoint = (fields, bits); {Two ways to view the contents of a frame}

    Frame = record {Format of data link frame}
      case view: ViewPoint of
        fields: (
            destinationField: AddressValue;
            sourceField: AddressValue;
            typeField: TypeValue;
            dataField: DataValue;
            fcsField: CRCValue);
        bits: (
            contents: array [1..frameSize] of Bit)
      end; {Frame}
```

6.5.2.1.2 Transmit State Variables

The following items are specific to frame transmission. (See also 6.5.2.1.4 on interfaces)

```
const
    interFrameSpacing = 9.6; {minimum time between frames, in microseconds}
    attemptLimit = 16; {Max number of times attempt transmission}
    backOffLimit = 10; {Limit on number of times to back off}
    jamSize = 32; {jam may be 32 to 48 bits long}

var
    outgoingFrame: Frame; {The frame to be transmitted}
    currentTransmitBit, lastTransmitBit: 1..frameSize; {Positions of current and
      last outgoing bits in outgoingFrame}
    deferring: Boolean; {True implies any pending transmission must wait for the
      channel to clear}
    frameWaiting: Boolean; {Indicates that outgoingFrame is deferring}
    attempts: 0..attemptLimit; {Number of transmission attempts on outgoingFrame}
    newCollision: Boolean; {Indicates that a collision has occurred but has not yet
      been jammed}
    transmitSucceeding: Boolean; {Running indicator of whether transmission is
      succeeding}
```

6.5.2.1.3 Receive State Variables

The following items are specific to frame reception. (See also 6.5.2.1.4 on interfaces)

```
var
    incomingFrame: Frame; {The frame being received}
    currentReceiveBit: 1..frameSize; {Position of current bit in incomingFrame}
    receiving: Boolean; {Indicates that a frame reception is in progress}
    excessBits: 0..7; {Count of excess trailing bits beyond octet boundary}
    receiveSucceeding: Boolean; {Running indicator of whether reception is
      succeeding}
```

6.5.2.1.4 Summary of Interlayer Interfaces

The interface to the Client Layer, defined in 5.1, is summarized below:

```
type
  TransmitStatus = (transmitOK, excessiveCollisionError); {Result of
    TransmitFrame operation}
  ReceiveStatus = (receiveOK, frameCheckError, alignmentError); {Result
    of ReceiveFrame operation}

function TransmitFrame (
  destinationParam: AddressValue;
  sourceParam: AddressValue;
  typeParam: typeValue;
  dataParam: DataValue): TransmitStatus; {Transmits one frame}

function ReceiveFrame (
  var destinationParam: AddressValue;
  var sourceParam: AddressValue;
  var typeParam: TypeValue;
  var dataParam: DataValue): ReceiveStatus; {Receives one frame}
```

The interface to the Physical Layer, defined in 5.2, is summarized below:

```
var
    carrierSense: Boolean; {Indicates incoming bits}
    transmitting: Boolean; {Indicates outgoing bits}
    collisionDetect: Boolean; {Indicates channel contention}
procedure TransmitBit (bitParam: Bit); {Transmits one bit}
function ReceiveBit: Bit; {Receives one bit}
procedure Wait (bitTimes: integer); {Waits for indicated number of bit-times}
```

6.5.2.1.5 State Variable Initialization

The procedure *Initialize* must be run when the Data Link Layer begins operation, before any of the processes begin execution. *Initialize* sets certain crucial shared state variables to their initial values. (All other global variables are appropriately reinitialized before each use.) *Initialize* then waits for the channel to be idle, and starts operation of the various processes.

```
procedure Initialize;
begin
  frameWaiting := false;
  deferring := false;
  newCollision := false;
  transmitting := false; {In interface to Physical Layer; see below}
  receiving := false;
  while carrierSense do nothing;
  {Start execution of all processes}
end; {Initialize}
```

6.5.2.2 Frame Transmission

The algorithms in this section define data link frame transmission.

The function *TransmitFrame* implements the frame transmission operation provided to the Client Layer:

```
function TransmitFrame (
    destinationParam: AddressValue;
    sourceParam: AddressValue;
    typeParam: typeValue;
    dataParam: DataValue): TransmitStatus;
  procedure TransmitDataEncap; ... {nested procedure; see body below}
begin
  TransmitDataEncap;
  TransmitFrame := TransmitLinkMgmt
end; {TransmitFrame}
```

First, *TransmitFrame* calls the internal procedure *TransmitDataEncap* to construct the frame. It then calls *TransmitLinkMgmt* to perform the actual transmission. The *TransmitStatus* returned indicates the success or failure of the transmission attempt.

TransmitDataEncap builds the frame and places the 32-bit CRC in the frame check sequence field:

```
procedure TransmitDataEncap;
begin
  with outgoingFrame do
  begin {assemble frame}
    view := fields;
    destinationField := destinationParam;
    sourceField := sourceParam;
    typeField := typeParam;
    dataField := dataParam;
    fcsField := CRC32(outgoingFrame);
    view := bits
  end {assemble frame}
end; {TransmitDataEncap}
```

TransmitLinkMgmt attempts to transmit the frame, deferring first to any passing traffic. If a collision occurs, transmission is terminated properly and retransmission is scheduled following a suitable backoff interval:

```
function TransmitLinkMgmt: TransmitStatus;
begin
  attempts := 0; transmitSucceeding := false;
  while attempts < attemptLimit and not transmitSucceeding do
  begin {loop}
    if attempts > 0 then BackOff;
    frameWaiting := true;
    while deferring do nothing;  {defer to passing frame, if any}
    frameWaiting := false;
    StartTransmit;
    while transmitting do WatchForCollision;
    attempts := attempts + 1
  end; {loop}
  if transmitSucceeding then TransmitLinkMgmt := transmitOK
  else TransmitLinkMgmt := excessiveCollisionError
end; {TransmitLinkMgmt}
```

Each time a frame transmission attempt is initiated, *StartTransmit* is called to alert the *BitTransmitter* process that bit transmission should begin:

```
procedure StartTransmit;
begin
  currentTransmitBit := 1;
  lastTransmitBit := frameSize;
  transmitSucceeding := true;
  transmitting := true
end; {StartTransmit}
```

Once frame transmission has been initiated, *TransmitLinkMgmt* monitors the channel for contention by repeatedly calling *WatchForCollision:*

```
procedure WatchForCollision;
begin
  if transmitSucceeding and collisionDetect then
  begin
    newCollision := true;
    transmitSucceeding := false
  end
end; {WatchForCollision}
```

WatchForCollision, upon detecting a collision, updates *newCollision* to insure proper jamming by the *BitTransmitter* process.

After transmission of the jam has completed, if *TransmitLinkMgmt* determines that another attempt should be made, *BackOff* is called to schedule the next attempt to retransmit the frame.

```
var maxBackOff: 2..1024; {Working variable of BackOff}

procedure BackOff;
begin
  if attempts = 1 then maxBackOff := 2 else if attempts ≤ backOffLimit
    then maxBackOff := maxBackOff*2;
  Wait(slotTime*Random(0, maxBackOff))
end; {BackOff}

function Random (low, high: integer): integer;
begin
  Random := ...{uniformly distributed random integer r such that low ≤ r <
  high}
end; {Random}
```

BackOff performs the truncated binary exponential backoff computation and then waits for the selected multiple of the slot time.

The *Deference* process runs asynchronously to continuously compute the proper value for the variable *deferring*.

```
process Deference;
begin
  cycle {main loop}
    while not carrierSense do nothing;  {watch for carrier to appear}
    deferring := true;  {delay start of new transmissions}
    while carrierSense do nothing;  {wait for carrier to disappear}
    RealTimeDelay(interFrameSpacing);
    deferring := false;  {allow new transmissions to proceed}
    while frameWaiting do nothing  {allow waiting transmission (if any)}
  end {main loop}
end; {Deference}

procedure RealTimeDelay (usec: real);
begin
  {Wait for the specified number of microseconds}
end; {RealTimeDelay}
```

The *BitTransmitter* process runs asynchronously, transmitting bits at a rate determined by the Physical Layer's *TransmitBit* operation:

```
process BitTransmitter;
begin
  cycle {outer loop}
    while transmitting do
    begin {inner loop}
      TransmitBit(outgoingFrame[currentTransmitBit]); {send next bit to
      Physical Layer}
      if newCollision then StartJam else NextBit
    end {inner loop}
  end {outer loop}
end; {BitTransmitter}

procedure NextBit;
begin
  currentTransmit3it := currentTransmitBit + 1;
  transmitting := (currentTransmitBit ≤ lastTransmitBit)
end; {NextBit}

procedure StartJam;
begin
  currentTransmitBit := 1;
  lastTransmitBit := jamSize;
  newCollision := false
end; {StartJam}
```

BitTransmitter, upon detecting a new collision, immediately enforces it by calling *StartJam* to initiate the transmission of the jam. The jam may contain 32 to 48 bits of arbitrary data. (*StartJam* uses the first 32 bits of the frame, merely to simplify this program).

6.5.2.3 Frame Reception

The algorithms in this section define data link frame reception:

The procedure *ReceiveFrame* implements the frame reception operation provided to the Client Layer:

```
function ReceiveFrame (
    var destinationParam: AddressValue;
    var sourceParam: AddressValue;
    var typeParam: TypeValue;
    var dataParam: DataValue): ReceiveStatus;
  function ReceiveDataDecap: ReceiveStatus; ... {nested function; see body
  below}
begin
  repeat
    ReceiveLinkMgmt;
    ReceiveFrame := ReceiveDataDecap;
  until receiveSucceeding
end; {ReceiveFrame}
```

ReceiveFrame calls *ReceiveLinkMgmt* to receive the next valid frame, and then calls the internal procedure *ReceiveDataDecap* to return the frame's fields to the Client Layer if the frame's address indicates that it should do so. The returned *ReceiveStatus* indicates the presence or absence of detected transmission errors in the frame.

```
function ReceiveDataDecap: ReceiveStatus;
begin
  receiveSucceeding := RecognizeAddress
    (incomingFrame.destinationField);
  if receiveSucceeding then with incomingFrame do
  begin {disassemble frame}
    view := fields;
    destinationParam := destinationField;
    sourceParam := sourceField;
    typeParam := typeField;
    dataParam := dataField;
    if fcsField = CRC32(incomingFrame) then ReceiveDataDecap := receiveOK
    else if excessBits = 0 then ReceiveDataDecap := frameCheckError
    else ReceiveDataDecap := alignmentError;
    view := bits
  end {disassemble frame}
end; {ReceiveDataDecap}
```

```
function RecognizeAddress (address: AddressValue): Boolean;
begin
  RecognizeAddress := ... {Returns true for the set of physical, broadcast, and
      multicast-group addresses corresponding to this station}
end; {RecognizeAddress}
```

ReceiveLinkMgmt attempts repeatedly to receive the bits of a frame, discarding any fragments from collisions by comparing them to the minimum valid frame size:

```
procedure ReceiveLinkMgmt;
begin
  repeat
    StartReceive;
    while receiving do nothing; {wait for frame to finish arriving}
    excessBits := frameSize mod 8;
    frameSize := frameSize - excessBits; {truncate to octet boundary}
    receiveSucceeding := (frameSize ≥ slotTime); {reject collision fragments}
  until receiveSucceeding
end; {ReceiveLinkMgmt}

procedure StartReceive;
begin
  currentReceiveBit := 1;
  receiving := true
end; {StartReceive}
```

The *BitReceiver* process run asynchronously, receiving bits from the channel at the rate determined by the Physical Layer's *ReceiveBit* operation:

```
process BitReceiver;
  var b: Bit;
begin
  cycle {outer loop}
    while receiving do
    begin {inner loop}
      b := ReceiveBit; {Get next bit from physical link}
      if carrierSense then
      begin{append bit to packet}
        incomingFrame[currentReceiveBit] := b;
        currentReceiveBit := currentReceiveBit + 1
      end; {append bit to packet}
      receiving := carrierSense
    end {inner loop}
  end {outer loop}
end; {BitReceiver}
```

6.5.2.4 Common procedures

The function *CRC32* is used by both the transmit and receive algorithms to generate a 32 bit CRC value:

```
function CRC32 (f: Frame): CRCValue;
begin
  CRC32 := {The 32-bit CRC as defined in 6.2.4}
end; {CRC32}
```

Purely to enhance readability, the following procedure is also defined:

```
procedure nothing; begin end;
```

The idle state of a process (i.e., while waiting for some event) is cast as repeated calls on this procedure.

7. ETHERNET PHYSICAL LAYER SPECIFICATION: Baseband Coaxial System

7.1 Physical Channel Overview and Model

The Ethernet physical channel (henceforth referred to as the channel) provides the lowest layer in the Ethernet architecture. It performs all the functions needed to transmit and receive data at the physical level, while supporting the Data Link to Physical Layer Interface described in 5.2.

This section describes the requirements for interface and compatibility with a baseband coaxial implementation of the channel.

7.1.1 Channel Goals and Non-goals

This section states the objectives underlying the design of the channel.

7.1.1.1 Goals

The following are the goals of the channel:

1. Provide a means for communication between Ethernet Data Link Entities.
2. Define physical interfaces which can be implemented compatibly among different manufacturers of hardware.
3. Provide all clocks, synchronization, and timing required for both itself and the Ethernet Data Link.
4. Provide high bandwidth and low bit error rates.
5. Provide for ease of installability and serviceability.
6. Provide for high network availability.
7. Support the Ethernet Data Link to Physical Link interface.
8. Low cost.

7.1.1.2 Non-Goals

The following are not goals of the baseband coaxial channel design:

1. Operation at data rates other than 10 megabits per second.
2. Operation with media other than the specified coaxial cable.
3. Simultaneous use of the channel by transmitters using signals not specified in this document.
4. Protection against a malicious user or a malfunctioning Data Link Entity is not provided by the channel as specified. However, higher layers (above the Data Link) and/or physical security means may be employed to acheive this.

7.1.2 Characteristics of the Channel

The channel provides (and the data link assumes) the following characteristics:

1. The ability to send and receive information (non-simultaneously) between any two or more data link entities on the same network.
2. The ability to detect the presence of another station's transmission while not transmitting (carrier sense).
3. The ability to detect the presence of another station's transmission while transmitting (collision detect).
4. A total worst-case round trip signal propagation delay (including actual propagation time, synchronization time for all intervening electronics, and signal rise time degradation) of 450 bit times (equal to 45 μs for this 10 Mbit channel).

7.1.3 Functions Provided by the Channel

The channel hardware provides the following functions in the performance of its role:

1. Means for transmitting and receiving serial bit streams between the data link layer and the media.
2. Generation of clock for synchronization and timing.
3. Means for detecting carrier (non-idle channel).
4. Means for detecting collisions (simultaneous transmission attempts by multiple stations).
5. Coding and decoding of the data link bit stream into a self-synchronizable sequence of electrical signals suitable for transmission on the media provided by the channel.
6. Generation and removal of coding-specific preamble information (a synchronizing header sequence inserted before the first bit of the frame) to ensure that all channel electronics are brought to a known steady-state before the data link frame is transmitted.

7.1.4 Implementation of the Channel

The physical channel specification is implementation dependent; most of the channel hardware is fully specified, and little leeway is given to the individual designer. This is done in the interest of compatibility; any system which allows different implementors to use different channel cables, connectors, clock speeds and the like will not be compatible across manufacturer boundaries. Only the design of channel components which are not critical to system compatibility is left to the implementor.

7.1.4.1 General Overview of Channel Hardware

The channel minimally consists of the following functional blocks:

1. The passive broadcast medium (coaxial cable),
2. The transceiver (transmitter-receiver for the coaxial cable),
3. The means for connecting transceivers to a coaxial cable segment and for connecting coaxial cable segments together,
4. The channel clock,
5. The channel data encoder and decoder,
6. The preamble generator and remover,
7. The carrier and collision detect circuits.

The coaxial medium is the only element common to the entire network. A transceiver is required for each station connected to the medium. The transceiver must be located adjacent to the coaxial cable. The latter four components are generally located within, and tightly coupled to, the station hardware implementing the data link function.

It may be useful to be able to physically separate the transceiver from the rest of the channel hardware. This allows topological flexibility, packaging advantages, and improved system availability, as well as allowing for independent manufacture of station hardware and transceivers. To ensure that compatibility is maintained, a physical interface (known as the *transceiver cable*) is identified and specified to connect the transceiver to the station.

Finally, it may be necessary to add *repeaters* to the system, to reach the maximum allowable distance between stations, and to provide additional topological flexibility. Repeaters are implemented using standard transceivers, plus a simple, non-buffered finite state machine.

7.1.4.2 Compatibility Interfaces

There are a number of possibilities for implementing systems or subsystems compatible in whole or in part with this specification. It is important that all implementations be compatible at some point, so that heterogenous systems from different manufacturers' implementations can be interconnected on the same medium. It is not necessary in every case to implement all of the components described herein; e.g., it is possible to design an integrated station/transceiver (without requiring the transceiver cable). The implementor must make the required trade-offs between topological flexibility, system availability, configurability, user needs, and cost when designing the system.

For a device to be considered compatible, it must meet the applicable requirements at either the transceiver cable or the coaxial cable interface, as appropriate, in addition to the Data Link compatibility required for all stations connected to the network.

All Ethernets must be compatible at the coaxial cable.

If a transceiver cable is used, it should be the one specified in this document. This allows device manufacturers to build hardware compatible with the Ethernet at the transceiver cable level, without concerning themselves with the details of transceiver implementation. Devices implementing transceiver cable compatibility should be capable of using transceivers designed and built by another manufacturer, on the specified coaxial cable.

Equipment designed for connection to the specified coaxial cable either without a physically separate transceiver or with a non-standard transceiver cable interface will be capable of communication. However, a sacrifice may have been made with respect to interchangeability with other stations.

This scheme of multiple compatibility interfaces allows individual designers some flexibility in making system tradeoffs, yet allows cable manufacturers, transceiver manufacturers and systems manufacturers to use standard commodity parts to produce a compatible communications system.

7.1.5 Channel Configuration Model

Certain physical limits have been placed on the physical channel. These revolve mostly around maximum cable lengths (or maximum propagation times), as these affect the slot time as defined in the data link. While the precise specification (in later sections) specify these maxima in terms of propagation times, they were derived from the physical configuration model described here.

The maximum configuration is as follows:

1. A coaxial cable, terminated in its characteristic impedance at each end, constitutes a cable *segment*. A segment may contain a maximum of 500 meters of coaxial cable.

2. A maximum of 2 repeaters in the path between any two stations. Repeaters do not have to be located at the ends of segments, nor is the user limited to one repeater per segment. In fact, repeaters can be used not only to extend the length of the channel, but to extend the topology from one to three-dimensional. Repeaters occupy transceiver positions on each cable segment and count towards the maximum number of transceivers on a segment just as do the logically distinguishable stations.

3. A maximum total coaxial cable length along the longest path between any two transceivers of 1500 meters. The propagation velocity of the coaxial cable is assumed to be 0.77 c worst-case. (c is the velocity of light *in vacuo*; 300,000 kilometers per second.) The total round-trip delay for all the coaxial cable in the system is therefore 13 μs worst-case.

4. A maximum of 50 meters of transceiver cable between any station and its associated transceiver. Note that in the worst case the signal must pass through six 50 meter transceiver cables, one at the transmitting station, one at the receiving station, and 2 at each repeater (two repeaters possible). The propagation velocity of the transceiver cable is assumed to be .65 c worst-case. The total round-trip delay for all the transceiver cables is therefore 3.08 μs worst-case.

5. A maximum of 1000 meters of point-to-point link anywhere in the system. This will typically be used as a way of linking cable segments in different buildings. Note that a repeater with this internal point-to-point link can be used to repeat signals between segments many hundreds of meters apart. The worst-case propagation velocity of the link cable is assumed to be .65 c; the round-trip propagation delay for 1000 meters is 10.26 μs.

Table 7-1 summarizes the allocation of the round-trip propagation delay to the individual components in the channel. Figure 7-1 shows a minimum, typical, and large-scale channel configuration.

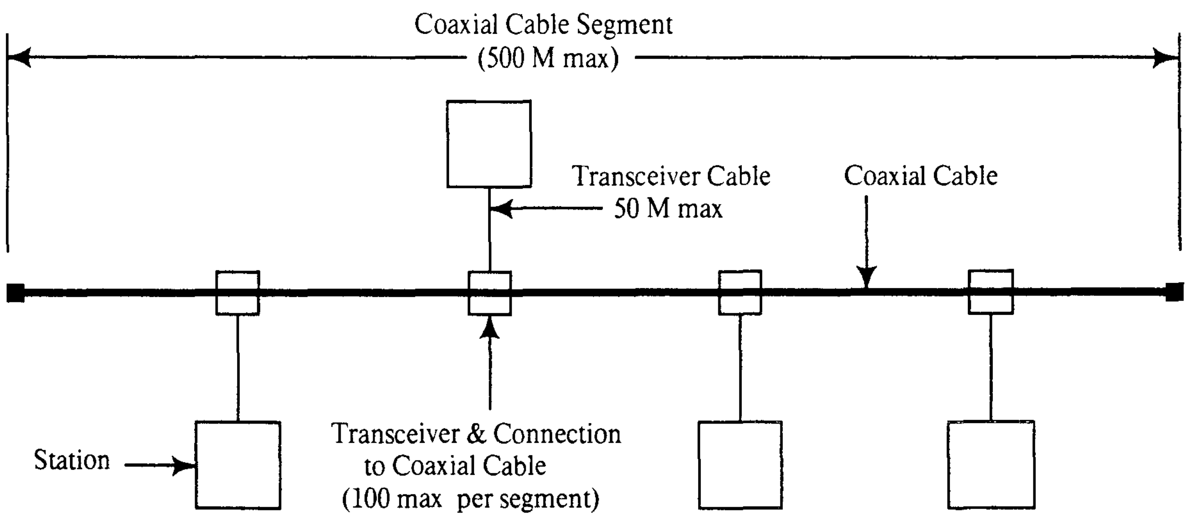


Figure 7-1a: Minimal Configuration

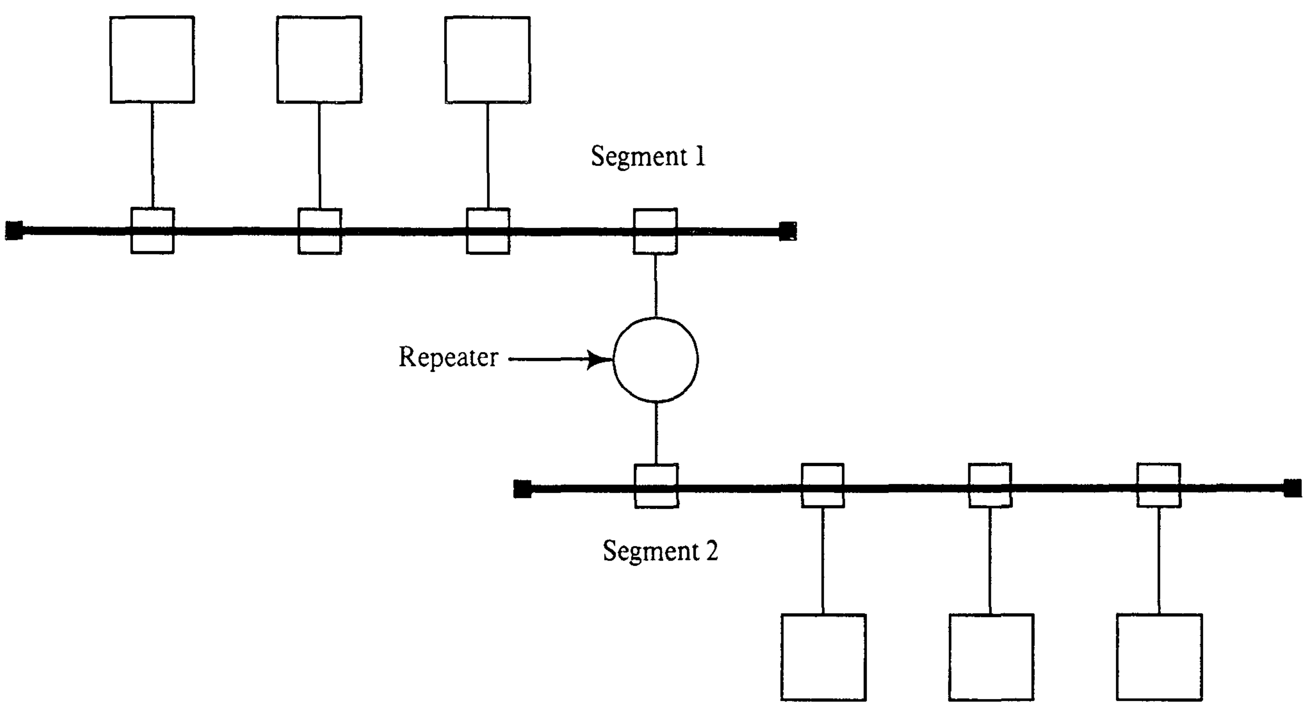


Figure 7-1b: A Typical Medium-scale Configuration

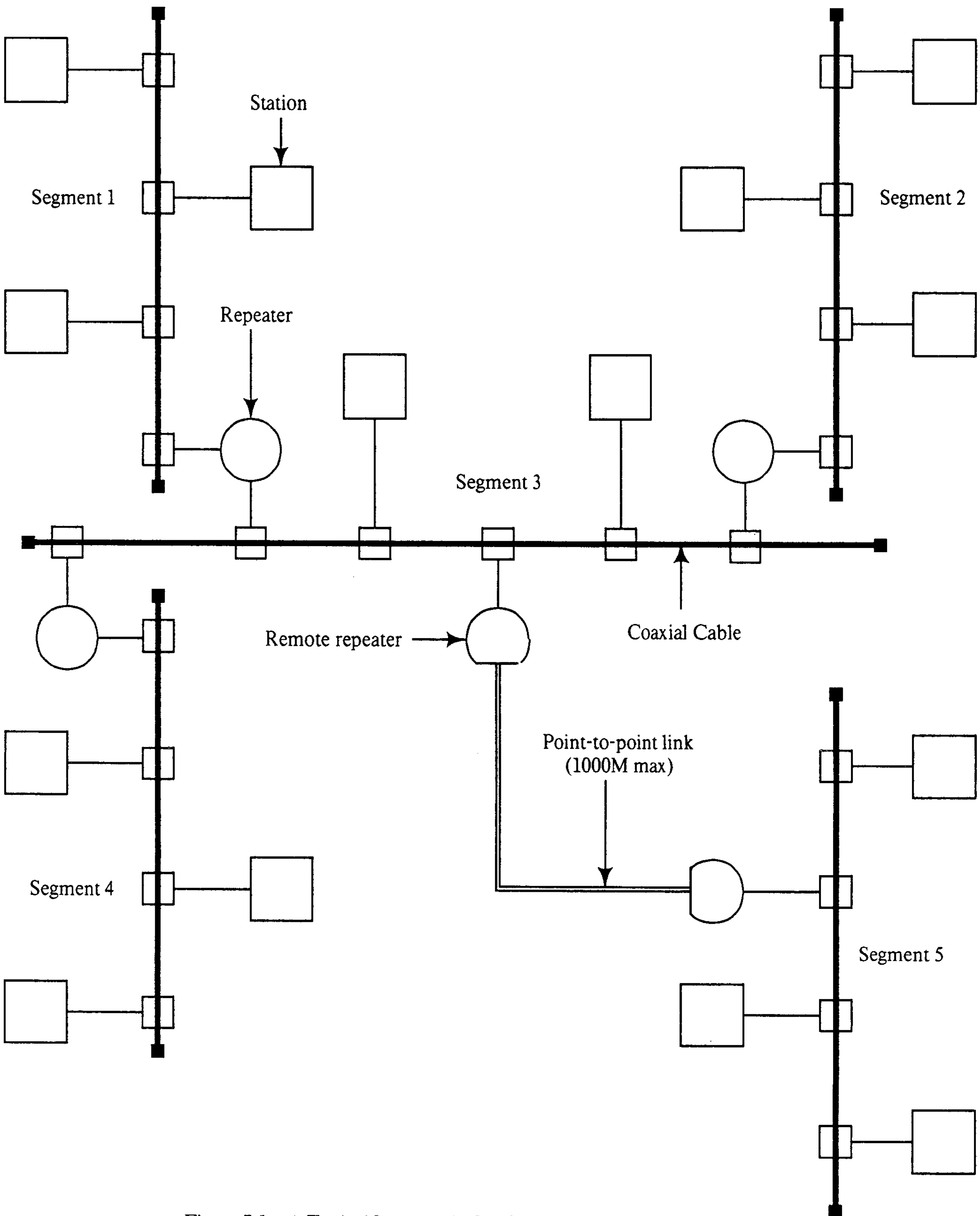


Figure 7-1c: A Typical Large-scale Configuration

Table 7-1: Physical Channel Propagation Delay Budget (Note 1)

Element	Unit Steady-State Delay	Unit Startup Delay	# Units Forward Path (Note 2)	# Units Return Path	Total Delay
Encoder	0.1 μS	0	3	3	0.60 μS
Transceiver.Cable	5.13 nS/M	0	300 M	300 M	3.08 μS
Transceiver (transmit path)	0.50 μS	0.2 μS	3	3	1.50 μS
Transceiver (receive path)	0.50 μS	0.5 μS	3	0	1.65 μS
Transceiver (collision path)	0	0.5 μS	0	3	1.50 μS
Coaxial Cable	4.33 nS/M	0	1500 M	1500 M	13.00 μS
Point-to-Point Link Cable	5.13 nS/M	0	1000 M	1000 M	10.26 μS
Repeater (repeat path)	0.8 μS	0	2	0	1.60 μS
Repeater (collision path)	0.2 μS	0	0	2	0.40 μS
Decoder	0.1 μS	0.8 μS	2	0	1.80 μS
Carrier Sense	0	0.2 μS	3	0	0.60 μS
Collision Detect	0	0.2 μS	0	3	0.60 μS
Signal Rise Time (to 70% in 500 M) (Note 3)	0	0.1 μS	3	0	0.30 μS
Signal Rise Time (50% to 94% in 500 M) (Note 4)	0	2.7 μS	0	3	8.10 μS
		Total Worst-Case Round-Trip Delay			44.99 μS

Note 1: All quantities given are worst-case (both number of units and unit delays per unit).

Note 2: The propagation delay has been separated into "forward-path" and "return path" delay. This is because in one direction it is carrier sense which is being propagated through the channel, and in the return direction it is collision detect which is being propagated. The two signals have different propagation delays.

Note 3: In the worst-case, the propagated signal must reach 70% of its final value to be detected as valid carrier at the end of 500 meters of coaxial cable. This rise time must be included in the propagation delay budget.

Note 4: In the worst-case the propagated collision on the return path must reach 94% of its final value to be detected as a collision at the end of 500 meters of coaxial cable.

7.1.6 Channel Interfaces

The channel specification hinges around three well-defined entities; the transceiver and coaxial cables (shown as compatibility interfaces in Figure 4-1), and the logical interface between the physical channel and the data link controller (shown in Figure 4-4). Note that the former two are physical interfaces specific to the channel, and are specified in the interest of compatibility. The latter is provided as a means by which the data link controller can interact with the physical channel.

The channel access component of the logical interface (discussed in 4.4.1) comprises the collision and carrier detect functions described in 7.5.2 and 7.5.3, as well as the actual transmission of signals on the media. The data encoding and decoding functions described in 4.4.1 comprise the generation and decomposition of encoded signals suitable for transmission (described in 7.5.1), the generation and removal of code-specific preamble (described in 7.5.1.3 and 7.5.4.1), and the serial bit stream interface between the layers.

Section 5 describes the interface between the data link and physical layers as a series of Pascal procedures, functions, and shared variables. The data link specification in section 6 shows how the data link uses this interface to communicate between client layers. However, this specification will not attempt to model the operation of the physical channel in Pascal. The interface between layers is supported by the physical hardware which provides the ability to send and receive bit streams, provide timing, and signal carrier sense and collision detect to the data link.

The remainder of this section specifies the requirements for compatibility at both the transceiver cable and the coaxial cable. In addition, the specifications for the transceiver, which interfaces the transceiver cable to the coaxial cable is given, as well as the specification for the logic required between the transceiver cable and the interface to the data link.

7.2 Transceiver Cable Compatibility Interface Specifications

The transceiver cable is the means by which a physically separate transceiver is connected to a station. It provides one of the compatibility interfaces described in 7.1.4.2.

7.2.1 Transceiver Cable Signals

The transceiver cable carries four signals: Transmit, Receive, Collision Presence, and Power. Each signal is carried on a twisted pair of conductors in the cable.

7.2.1.1 Transmit Signal

The transmit pair carries encoded data for which the data link is requesting transmission on the channel. This signal is generated by the data encoder, with the transceiver cable drive characteristics specified in 7.2.4.

7.2.1.2 Receive Signal

The receive pair carries encoded data from the transceiver to the station. It typically goes to the data decoder and the carrier sense circuitry. In the steady-state, all transitions and lack of transitions on the coaxial cable become transitions and lack of transitions on the receive pair, with the transceiver cable drive characteristics specified in 7.2.4. (During start-up, the first few bits may be absorbed by the transceiver to attain steady-state.)

In the case of a station transmitting without collision interference, the station's own transmit transitions on the coaxial cable will also appear on the receive pair, after a delay due to propagation through the transceiver. During collisions (whether or not that transceiver is involved in the collision) transitions on the receive lead are undefined; they may or may not meet decoder phase requirements, or they may not be present at all for extended periods. Thus the receive signal on the transceiver cable cannot be used alone to deterministically generate the carrier sense signal. This is described in more detail in 7.5.3.

7.2.1.3 Collision Presence Signal

The collision presence pair is used by the transceiver to indicate the presence of multiple transmission attempts on the coaxial cable. This is done by transmitting a square wave with a 10MHz fundamental frequency through the standard transceiver cable driver (described in 7.2.4). An oscillator is used instead of a simple level shift to allow AC coupling at the transceiver. Transceivers use the collision presence signal to indicate one of two conditions: the transceiver is transmitting and there is an attempt by another station to transmit at the same time, or there is a simultaneous transmission attempt by three or more stations regardless of whether the transceiver in question is transmitting.

7.2.1.4 Power

A pair of wires is designated for providing power to the transceiver. When the transceiver cable is implemented, the station end of the cable must supply a voltage between +12 and +15 Vdc ± 5% with at least 0.5 Amperes available to the cable for remotely powering the transceiver. The power source must meet applicable requirements for UL Class 2 wiring devices.

7.2.2 Transceiver Cable Parameters

7.2.2.1 Mechanical Configuration

The transceiver cable consists of four stranded, twisted pair conductors, plus an overall shield and insulating jacket. The conductor and jacket insulating material may be polyethylene or other suitable material. The flammability characteristics of the insulating material must be suitable for the installed environment.

7.2.2.2 Characteristic Impedance

The differential mode characteristic impedance of all pairs shall be 78 Ω, ± 5 Ω, in the configuration.

7.2.2.3 Attenuation

The signal attenuation of any pair shall not exceed 3 dB (measured at 10 MHz) for the total length between the transceiver and the station.

7.2.2.4 Velocity of Propagation

The minimum velocity of propagation of the transceiver cable shall be 0.65 c.

7.2.2.5 Pulse Distortion

Pulse distortion shall not exceed ± 1 nS at the end of 50 meters of cable when driven with random 10 Mbit data encoded in accordance with 7.5.1.

7.2.2.6 Resistance

The resistance of the conductors used for the power pair shall not exceed 40 milliohms per meter.

7.2.2.7 Transfer Impedance

The common mode transfer impedance of the transceiver cable shall not exceed the values shown in Figure 7-2 as a function of frequency. The differential mode transfer impedance of the cable with respect to any pair shall be 20 dB lower than the specified common mode transfer impedance.

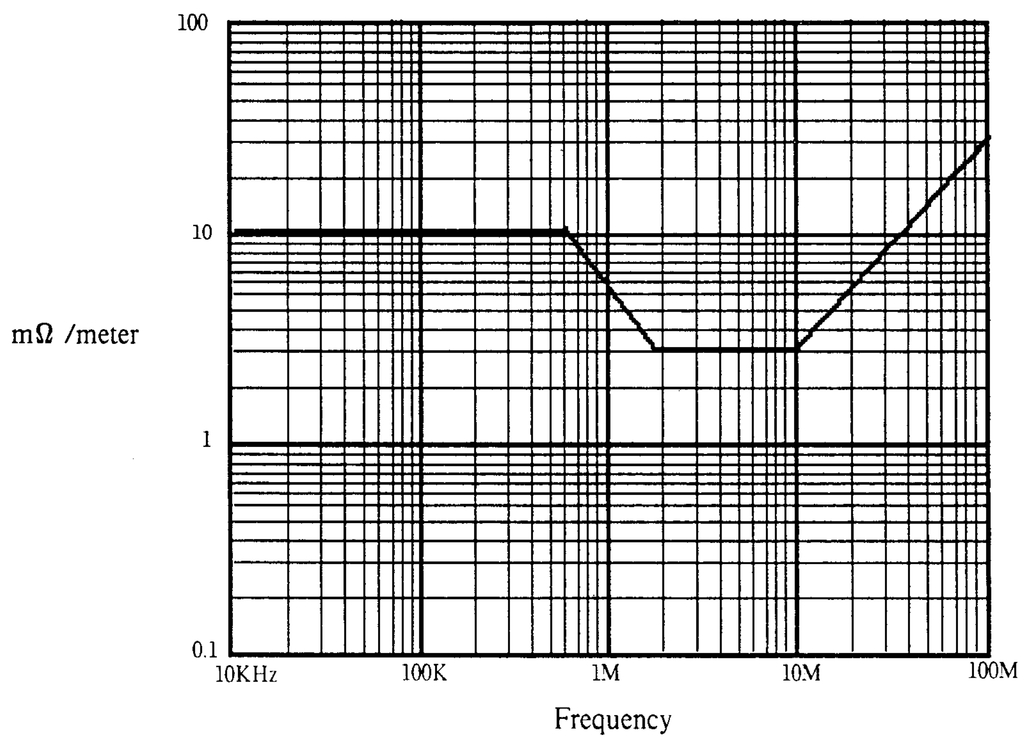


Figure 7-2: Transceiver Cable Transfer Impedance

7.2.3 Transceiver Cable Connectors

The connectors used at the ends of the transceiver cable shall be 15 conductor 'D' subminiature types (Cinch type DASM-15 or equivalent). The end of the cable that mates with the transceiver must use a female connector with a slide lock assembly (Cinch type DA 51220-1 or equivalent). The transceiver must provide a mating male connector with locking posts. The other end of the transceiver cable (which mates with a female connector at the station) must use a male connector with locking posts (Cinch type D 53018 or equivalent). The station must provide a female connector with the slide lock assembly.

Because of the end-to-end matching of the connectors, transceiver cables may be extended by concatenating transceiver cable sections. (The transceiver cable sections function as 'extension cords'.) A cable with multiple sections must still meet the cable loss characteristics of 7.3.1.1.2.

The pin assignment is given in the following table:

Transceiver Cable Connector Pin Assignment

1.	Shield (See note)		
2.	Collision Presence +	9.	Collision Presence -
3.	Transmit +	10.	Transmit -
4.	Reserved	11.	Reserved
5.	Receive +	12.	Receive -
6.	Power return	13.	Power
7.	Reserved	14.	Reserved
8.	Reserved	15.	Reserved

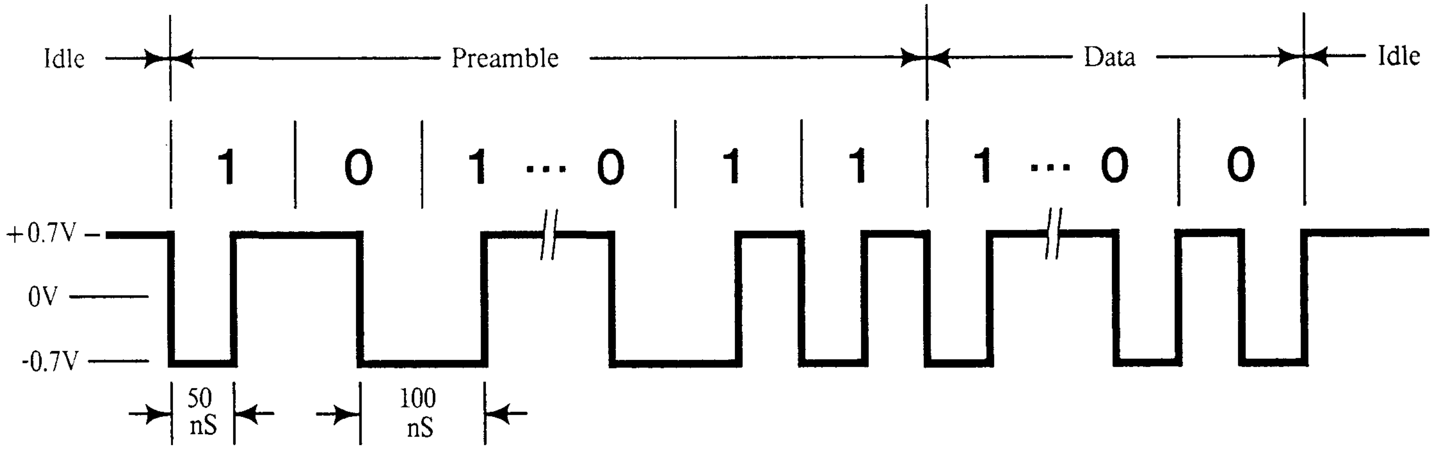
Note: Shield must be terminated to connector shell as well as pin 1.

Metal, metallized plastic, or otherwise shielded connector backshells must be used to ensure shield integrity.

7.2.4 Transceiver Cable Drive

This section describes the requirements for driving any of the signal pairs in the transceiver cable: transmit, receive, and collision presence.

The AC signal levels presented to the transceiver cable shall be ± 700 mV nominal, balanced differential drive into 78 ± 5 Ω. The common mode voltage presented to the transceiver cable shall not exceed that allowed at the receiver, as specified in 7.2.5.2. Signal waveform shall be as shown in Figure 7-3.



1. Voltages are measured differentially at output of transceiver cable driver.
2. Rise and fall times meet 10,000 series ECL requirements.

Figure 7-3: Typical Transceiver Cable Waveform

The transceiver cable driver must be capable of maintaining the specified minimum differential signal into the worst case low cable impedance (73 Ω differential, 18.5 Ω common mode) in the environment specified in section 7-7.

The idle state of the output shall be high (+ 700 mV nominal); the first transition presented is negative-going, the last transition must be positive-going. Note that the presence of AC coupling may cause the voltage as specified at the output of the transceiver cable drive circuit not to appear on the transceiver cable in the idle state.

A typical transceiver cable drive circuit is given in Appendix D.

7.2.5 Transceiver Cable Receive

The following sections specify the requirements for receiving signals from any signal pair in the transceiver cable: transmit, receive, and collision presence. The circuit must be capable of receiving the signals from the transceiver cable driver specified in 7.2.4 through the cable specified in 7.2.2 in the worst case. A typical receive circuit is given in Appendix D.

7.2.5.1 Load Impedance and Termination

The termination impedance shall be 78 Ω ± 1% differential mode, and 18.5 Ω minimum common-mode, over the frequency range of 3-20 MHz.

7.2.5.2 Common Mode and CMRR

The common mode range and the common mode rejection ratio shall be sufficient to maintain a 5:1 signal to noise ratio in the environment specified in 7.7, measured at the input to the transceiver cable receiver. The common mode DC voltage at the input of the receiver shall be in the range of zero to +5 Vdc..

7.3 Coaxial Cable Compatibility Interface Specifications

The coaxial cable is the common, shared broadcast medium through which stations communicate. It provides one of the compatibility interface points described in 7.1.4.2.

7.3.1 Coaxial Cable Component Specifications

The cable is of constant impedance, coaxial construction. It is terminated at each end by a *terminator* (specified in 7.3.1.3), and connection provided for each transceiver. Coaxial cable connectors are used to make the connection from the cable to the terminators, and between cable sections (if needed). The cable has various electrical and mechanical requirements which must be met to ensure proper operation.

7.3.1.1 Coaxial Cable Parameters

7.3.1.1.1 Characteristic Impedance

The average characteristic impedance of the cable shall be 50 ± 2 Ω, measured according to Mil. Std. C17-E. Periodic variations in impedance along a single piece of cable may be up to ± 3 Ω sinusoidal, centered around the average value, with a period < 2 meters. Note that the proper operation of the network is dependent upon the cable characteristic impedance; its value and tolerance are critical.

7.3.1.1.2 Attenuation

The attenuation of a cable segment shall not exceed 8.5 dB measured at 10 MHz, nor 6.0 dB measured at 5 MHz.

7.3.1.1.3 Velocity of Propagation

The minimum acceptable velocity of propagation is 0.77 c..

7.3.1.1.4 Mechanical Requirements

The cable used should be suitable for routing in various environments, including but not limited to, dropped ceilings, raised floors, and cable troughs. The jacket must provide insulation between the cable sheath and any building structural metal. Also, the cable must be capable of accepting coaxial cable connectors, described in 7.3.1.2. The cable must in addition conform to the following requirements:

1. The center conductor must be 0.0855" ± .0005" diameter solid, tinned copper,
2. The core dielectric material must be foamed,
3. The inside diameter of the innermost shield must be .242" minimum,
4. The outside diameter of the outermost shield must be .326" ± .007",
5. The outermost shield must be greater than 90% coverage tinned copper braid,
6. The jacket O.D. must be 0.405" nominal,
7. The cable concentricity must be 90% minimum.

The cable must also meet applicable flammability criteria and local codes for the installed environment. Different (e.g., polyethylene and Teflon dielectric) types of cable sections may be interconnected, while meeting the sectioning requirements of 7.6.1.

7.3.1.1.5 Pulse Distortion

Pulse distortion shall not exceed ± 7 nS at the end of 500 meters of cable when driven with random 10 Mbit data encoded in accordance with 7.5.1.

7.3.1.1.6 Jacket Marking

The cable jacket must be marked with annular rings in a color contrasting with the background color of the jacket. The rings must be spaced at 2.5 meter ± 5 cm regularly along the entire length of the cable. It is permissible for the 2.5 meter spacing to be interrupted at discontinuities between cable sections joined by connectors. (See 7.6.2 for transceiver placement rules which mandate cable markings.)

7.3.1.1.7 Transfer Impedance

The transfer impedance of the cable shall not exceed the values shown in Figure 7-4 as a function of frequency.

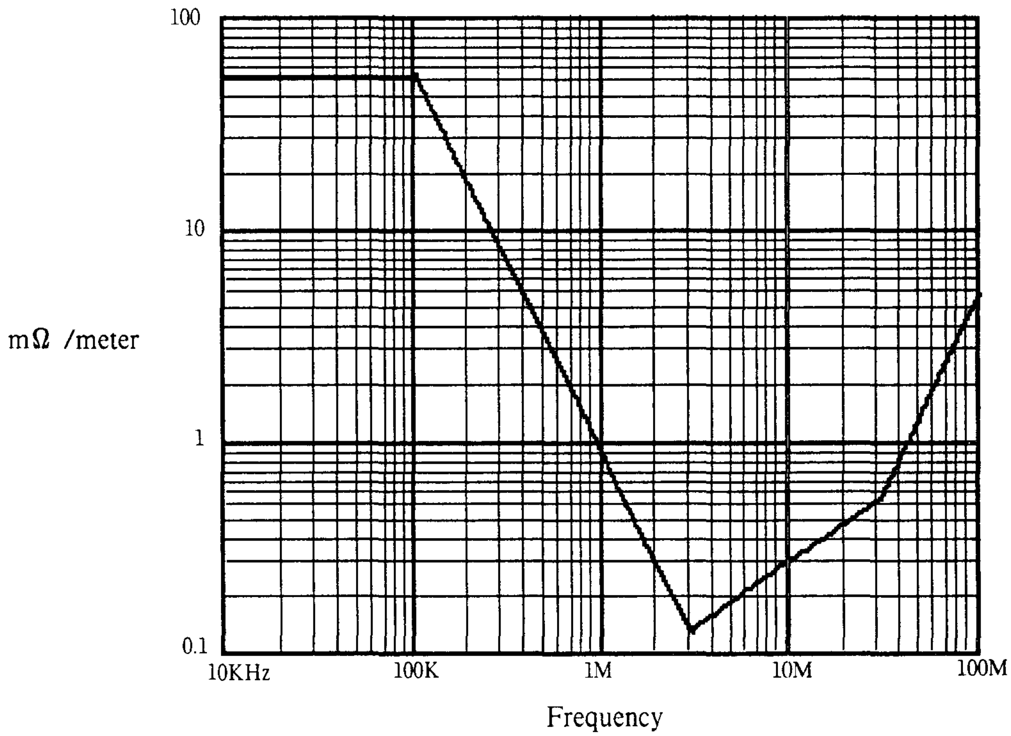


Figure 7-4: Maximum Coaxial Cable Transfer Impedance

7.3.1.2 Coaxial Cable Connectors

Coaxial cable connectors are used to join cable sections and attach terminators. Three types of connectors may be necessary; male plugs, female jacks, and female-to-female barrels. Plugs are used exclusively at the ends of all cable sections. Jacks are used to house cable terminators. Barrels are used to join cable sections.

All connectors are N series, 50 Ω constant impedance types. Since the frequencies present in the transmitted data are well below UHF range (being band-limited to approximately 20 MHz), military versions of the connectors are not required (but are acceptable).

Means must be provided to ensure that the connector shell (which connects to the cable sheath) does not make contact with any building metal, or other unintended conductor. A sleeve or boot to be slid over the connector at installation time is suitable.

7.3.1.3 Coaxial Cable Terminators

Coaxial cable terminators are used to provide a termination impedance for the cable equal in value to its characteristic impedance, thereby eliminating any reflection from the ends of the cables. Terminators shall be packaged within an inline female jack connector. The termination impedance shall be 50 Ω ± 1% measured from 0-50 MHz, with the magnitude of the phase angle of the impedance not to exceed 5 degrees. The terminator power rating shall be 1 watt or greater.

7.3.1.4 Transceiver-to-Coaxial Cable Connections

A means must be provided to allow for attaching a transceiver to the coaxial cable. The connection must disturb the transmission line characteristics of the cable as little as possible; it must present a predictably low shunt capacitance, and therefore a negligibly short stub length. For this reason, the transceiver must be located as close to its cable connection as possible; they are normally considered to be one assembly. Long (greater than 3 cm) connections between the coaxial cable and the input of the transceiver are not acceptable.

The transceiver-to-coaxial cable connection shall present less than 2 picofarads shunt capacitance to the coaxial cable, not including any transceiver electronics. If the design of the connection is such that the coaxial cable must be severed to install the transceiver, the coaxial cable segment must still meet the sectioning requirements of 7.6.1. Any coaxial connectors used on a severed cable must be type N, as specified in 7.3.1.2.

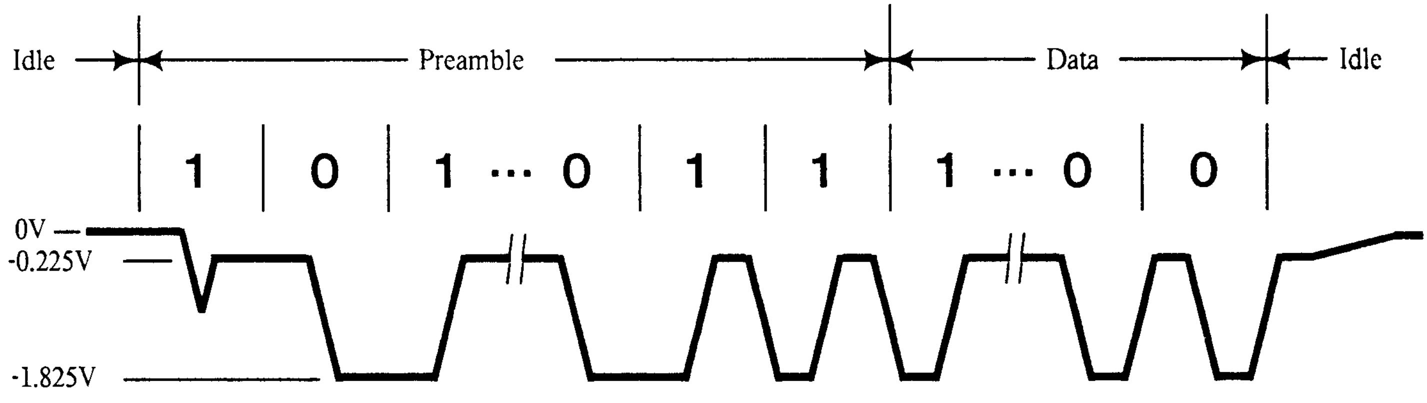
7.3.2 Coaxial Cable Signaling

The AC component of the signal on the coaxial cable due to a single transceiver as measured on the coaxial cable immediately adjacent to the transceiver connection shall be ± 16 mA nominal (14 mA min, 19 mA max). The DC component shall be one-half the AC component, plus 4.5 mA (4 mA min, 5 mA max). The actual current measured at a given point on the cable is a function of the transmitted current and the cable loss to the point of measurement. Positive current is defined as current out of the center conductor of the cable (into the transceiver). Cable loss is specified in 7.3.1.1.2.

The 10%-90% rise and fall times shall be 25 ± 5 nsec. Figure 7-5 shows typical waveforms present on the cable. Harmonic content generated from a 10 MHz fundamental periodic input shall meet the following requirements:

Second and Third Harmonics:	-20 dB min
Fourth and Fifth Harmonics:	-30 dB min
Sixth and Seventh Harmonics:	-40 dB min
All Higher Harmonics:	-50 dB min

The signals as generated from the encoder (described in 7.5.1.1) shall appear on the coaxial cable without any inversions.



1. Voltages given are nominal; worst case is given in text.
2. Rise time is 25 nS nominal.
3. Voltages are measured on coaxial cable adjacent to transceiver.

Figure 7-5: Typical Coaxial Cable Waveform

7.4 Transceiver Specifications

The following sections specify the requirements for a transceiver.

7.4.1 Transceiver-to-Coaxial Cable Interface

The following sections describe the interface between the transceiver and the coaxial cable. Positive current is defined as current into the transceiver (out of the center conductor of the cable).

7.4.1.1 Input Impedance

The shunt capacitance presented to the coaxial cable by the transceiver circuitry (not including the means of attachment to the coaxial cable) shall not exceed 2 picofarads. The shunt resistance presented to the coaxial cable shall be greater than 50 KΩ.

These conditions must be met in both the power off and the power on, not transmitting states.

7.4.1.2 Bias Current

The transceiver must draw between -2 and +50 uA in the power-off and the power-on, not transmitting states.

7.4.1.3 Transmit Output Levels

Signals received from the transceiver cable transmit pair must be transmitted onto the coaxial cable with the characteristics specified in 7.3.2. Note that 7.3.2 specifies the current level on the coaxial cable. Since the coaxial cable proceeds in two directions away from the transceiver, the current into the transceiver is actually twice the current measured on the coaxial cable.

Transmitted output asymmetry shall not exceed 2 ns for a 50/50 duty cycle input on the transceiver cable transmit pair.

7.4.2 Transceiver-to-Transceiver Cable Interface

7.4.2.1 Transmit Pair

The transceiver must present the transceiver cable receive characteristics specified in 7.2.5 to the transmit pair. At the start of a frame transmission, no more than 2 bits (two 100 ns bit cells) of information may be received from the transmit pair and not transmitted onto the coaxial cable. The steady-state propagation delay between the transmit pair input and the coaxial cable output shall not exceed 50 ns. There are no signal inversion between the transceiver cable transmit pair and the coaxial cable.

7.4.2.2 Receive Pair

The transceiver must present the transceiver cable transmit characteristics specified in 7.2.4 to the receive pair. Asymmetry as seen on the receive pair shall not exceed ± 2 nsec for a ± 200 mV peak sinusoidal input from the coaxial cable.

The signal from the coaxial cable shall pass through AC coupling with an appropriate time constant before proceeding to the receive pair. The time constant should compensate for the coaxial cable pulse distortion.

At the start of a frame reception from the coaxial cable, no more than 5 bits (five 100 ns bit cells) of information may be received from the coaxial cable and not transmitted onto the receive pair. In addition, it is permissible for the first bit sent over the receive pair to contain encoding phase violations or invalid data, however all successive bits of the frame shall be valid and meet encoding rules. The steady-state propagation delay between the coaxial cable and the receive pair output shall not exceed 50 ns. There are no signal inversions between the coaxial cable and the transceiver cable receive pair.

7.4.2.3 Collision Presence Pair

The transceiver must present the transmitter characteristics specified in 7.2.4 to the collision presence pair. The signal presented to the collision presence pair shall be a periodic waveform with a 10 MHz ± 15% frequency. This signal shall be presented to the collision presence pair no more than 5 bit times (500 nS) after the average signal on the coaxial cable at the transceiver exceeds either that which could be produced by two transceiver outputs in the worst case (if the transceiver in question is not transmitting), or that which could be produced by that transceiver alone in the worst case (if that transceiver is transmitting).

7.4.2.4 Power Pair

The transceiver cable provides power which may be used for operation of the transceiver electronics. The power available shall be as described in 7.2.1.4. The distribution impedance of the transceiver cable is 4 Ω maximum, for a 50 meter cable with the resistance specified in 7.2.2.6. In order for the transceiver to derive its operating power from the power pair, circuitry must be employed to provide the required electrical isolation specified in 7.4.3.

7.4.3 Electrical Isolation

The transceiver must provide electrical isolation between the transceiver cable and the coaxial cable. The isolation impedance shall be greater than 250 KΩ, measured between any conductor (including shield) of the transceiver cable and either the center conductor or shield of the coaxial cable, at 60 Hz. The breakdown voltage of the isolation means provided shall be at least 250 VAC, rms.

7.4.4 Reliability

No single nor double component failure within the transceiver electronics shall impede communication among other transceivers on the coaxial cable. Connectors and other passive components comprising the means of connecting the transceiver to the coaxial cable shall be designed to minimize the probability of total network failure.

7.5 Channel Logic

The following sections describe the functions that must be performed to properly interface between the data link and the transceiver cable. They are normally implemented as logic, typically within the same device implementing the data link layer.

7.5.1 Channel Encoding

The channel shall use Manchester phase encoding, with a data rate of 10 Mbps, ± .01%, measured at the encoder clock. Thus, each bit cell is 100 ns long.

The following section describes the requirements for encoding and decoding signals to be transmitted on, or received from the coaxial or transceiver cables.

7.5.1.1 Encoder

The encoder is used to translate physically separate signals of clock (synchronization) and data into a single, self-synchronizable serial bit stream, suitable for transmission on the coaxial cable by the transceiver.

During the first half of the bit cell time, the serial signal transmitted is the logical complement of the bit value being encoded during that cell. During the second half of the bit cell time, the uncomplemented value of the bit being encoded is transmitted. Therefore, there is always a signal transition (either positive-going or negative-going, depending on the bit being encoded) in the center of each bit cell. A timing diagram for a typical bit stream is given in Figure 7-6.

The encoder output drives the transmit pair of the transceiver cable, and ultimately, the coaxial cable through the transceiver. The encoder output asymmetry must not exceed 0.5 ns. The encoder shall provide the defined output for the first (and all subsequent) bits presented to its input. All information submitted for encoding shall appear at the output of the encoder.

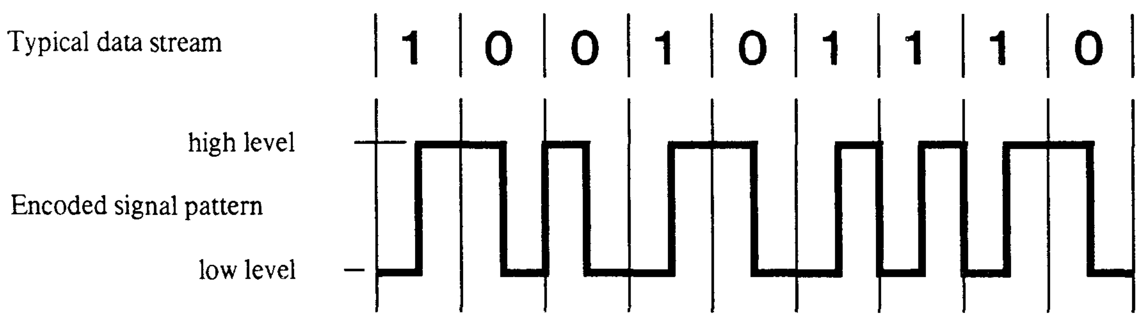


Figure 7-6: Manchester Encoding

7.5.1.2 Decoder

The decoder is used to separate the incoming phase encoded bit stream into a data stream and a clock signal. The decoder must be able to provide data and clock signals usable by the data link under the asymmetry imposed by the worst case system configuration. The decoder must provide usable output (clock and data) after no more than 8 bit cell times after reception of an encoded signal. The first signals received from the transceiver at the beginning of frame reception may not constitute a valid, properly encoded bit; it is possible for the time from the first transition seen to the first true mid-bit cell transition to assume any value from zero to 100 nS.

The decoder input is normally derived from the coaxial cable, through the transceiver cable receive pair. It is not necessary for the decoder to provide usable output when there is a collision on the coaxial cable, regardless of whether the station using that decoder is involved in the collision.

7.5.1.3 Preamble Generation

Because most of the channel circuitry is allowed to provide valid output some number of bit times after being presented valid input, it is necessary for a preamble to be sent before the start of data link information, to allow the channel circuitry to reach its steady-state, with valid outputs throughout the system. Upon request by the data link to transmit the first bit of a new frame, the channel shall first transmit the preamble; a predetermined bit sequence used for channel stabilization and synchronization. If, while transmitting the preamble, the channel logic asserts the collision detect signal as specified in 7.5.2, any remaining preamble bits shall not be sent. The channel should immediately proceed with the transmission of the bit submitted by the data link.

The preamble is a 64 bit pattern to be presented to the channel encoder in the same manner as data link information. The pattern is:

10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101011.

The bits are transmitted in order, from left to right. The nature of the pattern is such that when encoded, it appears as a periodic waveform on the cable, with a 5 MHz frequency. Excepting the final two bits, the only transitions present in the waveform are in the center of the bit cells. This is depicted in Figure 7-7. The last two bits of the preamble contain transitions at both the bit cell centers and the edges, and are used to indicate the end of the preamble, and the beginning of the data link encapsulation portion of the frame. The next bit transmitted is the bit originally submitted by transmission by the data link.

Preamble removal on reception is discussed in 7.5.4.1.

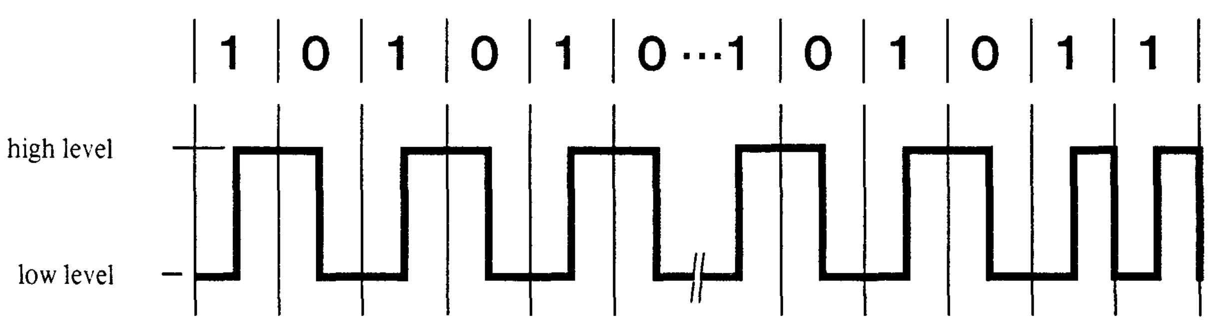


Figure 7-7: Preamble Encoding

7.5.2 Collision Detect Signal

The channel must indicate to the data link when the signals on the coaxial cable imply simultaneous transmission attempts by more than one station. This is normally indicated through the collision presence pair in the transceiver cable, described in 7.2.1.3.

The channel logic must assert the collision detect signal within 2 bit times (200 ns), following the onset of collision presence. This collision detect signal shall be asserted only when the data link is transmitting. A functional logic description of the collision detect signal is shown in Figure 7-8.

Following the loss of collision presence information, the channel must deassert the collision detect signal within 1.6 bit cell times (160 ns).

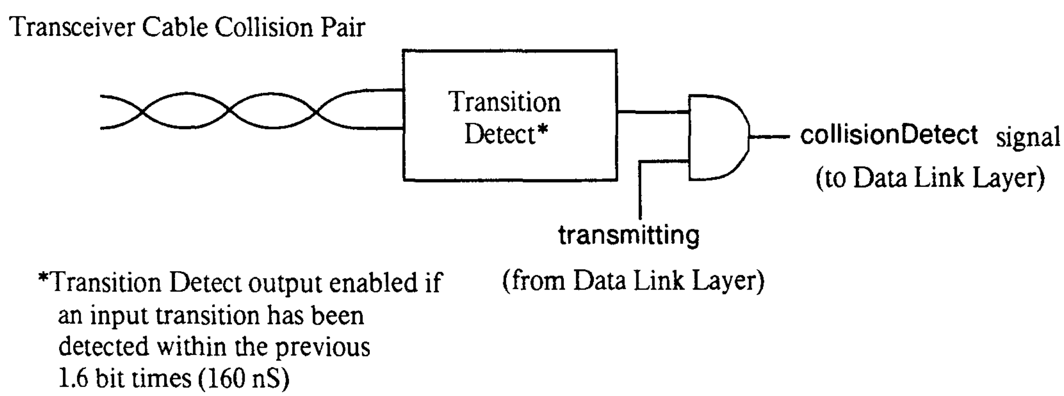


Figure 7-8: Functional Logic of collisionDetect Signal

7.5.3 Carrier Sense Signal

The channel must indicate to the data link the presence of carrier, a signal transmission attempt on the coaxial cable by a station. This is normally indicated through both the receive and collision presence pairs in the transceiver cable, described in 7.2.1.

The carrier sense signal shall be asserted when one or more station is attempting transmission on the cable, regardless of whether the station sensing carrier is transmitting at that time. The channel logic must assert the carrier sense signal within 2 bit times (200 ns) following the onset of carrier presence information. A functional logic description of these signals is shown in Figure 7-9.

Following the loss of carrier presence information (receive transitions and collision presence information) the channel must deassert the carrier sense signal within 1.6 bit cell times (160 ns).

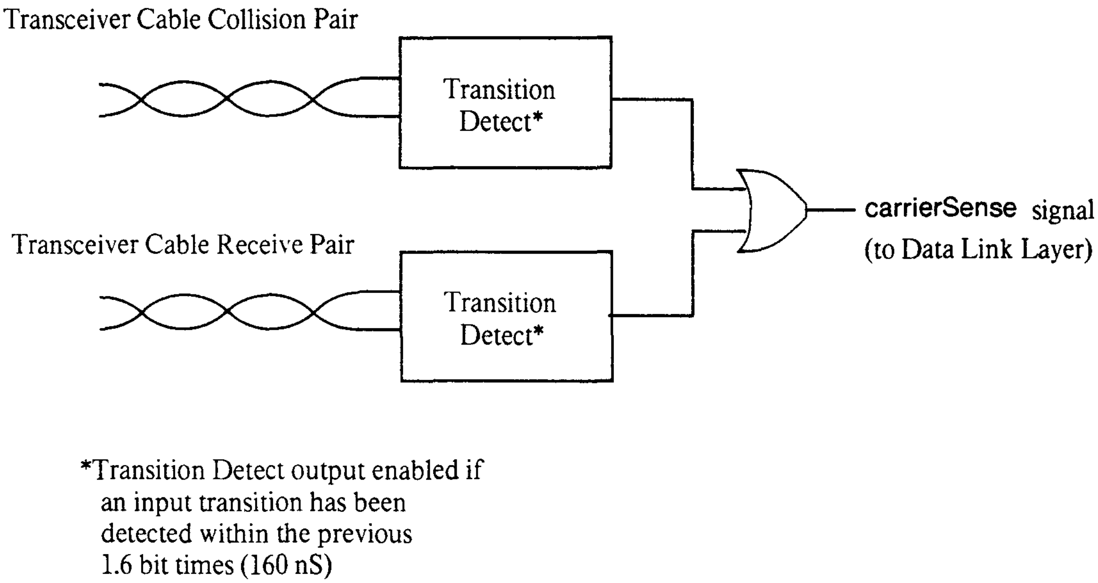


Figure 7-9: Functional Logic of carrierSense Signal

7.5.4 Channel Framing

During reception, the channel must provide the data link with signals to indicate beginning and end of frame.

7.5.4.1 Beginning-of-Frame Sequence

The channel logic recognizes the presence of activity on the medium through the carrier sense signal. This is the first indication that the frame reception process should begin. However, dependent upon the physical configuration of the system, there are some number of preamble bits to be received by the channel before the start of the data link frame as indicated by the double-1 at the end of preamble. In addition, the first signals received from the decoder may be invalid due to the first bit allowance of the transceiver (see 7.4.2.2). The channel must wait no less than 8 bit times (800 nS) before monitoring the output of the decoder for the 'double-1' indicating end of preamble, and beginning of data link frame. Upon reception of the double-1, the channel shall begin passing successive bits to the data link through the defined receive bit stream interface. If, after waiting the required 8 bit times, a 'double-0' is encountered, the physical channel shall not pass any bits of the current frame to the data link. Normal operation of the data link and channel shall resume on the subsequent frame.

7.5.4.2 End-of-Frame Sequence

As specified in 7.5.3, the carrier sense signal must be deasserted no later than 1.6 bit times (160 ns) after the cessation of activity on the coaxial cable as seen by the channel logic. The channel ensures that no extraneous bits will appear at the end of a frame following the last valid bit.

7.6 Channel Configuration Requirements

7.6.1 Cable Sectioning

The 500 meter maximum length coaxial cable segment need not be made from a single, homogeneous length of cable. The boundary between two cable sections (joined by coaxial connectors; two male plugs and a barrel) represents a signal reflection point due to the impedance discontinuity caused by the batch-to-batch impedance tolerance of the cable. Since the worst-case variation from 50 Ω is 2 Ω (see 7.3.1.1.1), a possible worst-case reflection of 4% may result from the joining of two cable sections. The configuration of long cable segments (up to 500 meters) from smaller sections must be made with care. The following recommendations apply, and are given in order of preference:

1. If possible, the total segment should be made from one homogeneous (no breaks) cable. This is feasible for short segments, and results in minimal reflections from cable impedance discontinuities.
2. If cable segments must be built up from smaller sections, it is highly desirable to ensure that all the sections are from the same manufacturer and lot. This is equivalent to using a single cable, since the cable discontinuities are due to extruder limitations, and not extruder-to-extruder tolerances. There are no restrictions in cable sectioning if this method is used. However, if a cable section in such a system is later replaced, it must be replaced either with another cable from the same manufacturer and lot, or with one of the standard lengths described below.
3. If uncontrolled cable sections must be used in building up a longer segment, the lengths should be chosen such that reflections, when they occur, do not have a high probability of adding in phase. This can be accomplished by using lengths which are odd integral multiples of a half-wavelength in the cable at 5 MHz; this corresponds to using lengths of 23.4, 70.2, and 117 meters (± 0.5 meters) for all sections. These are considered to be the standard lengths for all cable sections. Using these lengths exclusively, any mix or match of cable sections may be used to build up a 500 meter segment without incurring excessive reflections.

4. As a last resort, an arbitrary configuration of cable sections may be employed, if it has been confirmed by analysis or measurement that the worst-case signal reflection due to the impedance discontinuities at any point on the cable does not exceed 7% of the incident wave when driven by a transceiver meeting the specifications of 7.4.

7.6.2 Transceiver Placement

Transceivers and their associated connections to the cable cause signal reflections due to their non-infinite bridging impedance. While this impedance must be implemented as specified in 7.3.1.4 and 7.4.1, the placement of transceivers along the coaxial cable must also be controlled to insure that reflections from transceiver do not add in phase to a significant degree.

Coaxial cables marked as specified in 7.3.1.1.6 have marks at regular 2.5 meters spacing; a transceiver may be placed at any mark on the cable. This guarantees both a minimum spacing between transceivers of 2.5 meters, as well as controlling the relative spacing of transceivers to insure non-alignment on fractional wavelength boundaries.

The total number of transceivers on a cable segment shall not exceed 100.

7.6.3 System Grounding

The sheath conductor of the coaxial cable shall not make electrical contact with any earth reference, building structural metal, ducting, plumbing fixture, or other unintentioned conductor. Insulators may be used to cover any coaxial connectors used to join cable sections and terminators, to insure that this requirement is met. A sleeve or boot attached at installation time is acceptable.

The sheath conductor of the transceiver cable shall be connected to the earth reference or chassis of the device housing the station logic.

7.6.4 Repeaters

Repeaters are used to extend the channel length and topology beyond that which could be achieved by a single coaxial cable segment. (See the channel configuration model in 7.1.5.) A repeater requires a transceiver on each of the segments between which it is repeating signals. These transceivers must be as specified in 7.4, and must be counted towards the maximum specified in 7.6.2.

A maximum of two repeaters may be in the signal path between any two transceivers on the channel.

7.6.4.1 Carrier Detect and Transmit Repeat

Repeaters must implement the carrier sense function as specified in 7.5.3 for both segments between which it is connected. Upon detection of carrier from one segment, the repeater must retransmit all received signals from that segment onto the other segment. Signals shall be retimed and amplified as specified in 7.6.4.3. The maximum steady-state propagation delay through the repeater for the repeated signal (not including startup delays, carrier sense delay or retiming delays) shall not exceed 800 nS.

7.6.4.2 Collision Detect and Collision Repeat

Repeaters must implement the collision detect function as specified in 7.5.2 for both segments between which it is connected. If, while repeating signals as specified in 7.6.4.1, collision is detected on either side, the repeater must ensure that all stations involved in the collision recognize the event as a collision, regardless of which side of the repeater the station is on. The maximum time between the recongnition of the collision and the repeating of the collision indication (not including carrier sense of retiming delays) shall not exceed 200 nS.

7.6.4.3 Repeater Signal Regeneration

7.6.4.3.1 Signal Amplification

The repeater (with its associated transceivers) shall ensure that any signals repeated between segments shall have the same amplitude characteristics at the transceiver output of the repeated-to segment as they did at the output of the transmitter on the repeated-from segment, allowing for transceiver output tolerances as specified in 7.4.1.3. Any loss of signal-to-noise ratio due to cable loss and noise pickup is thus regained at the output of the repeater.

7.6.4.3.2 Signal Timing

The repeater must ensure that the symmetry characteristics of the signals at the transceiver output of the repeated-to segment are the same as those at the output of the transmitter on the repeated-from segment, allowing for transceiver and transceiver cable tolerances. Any loss of symmetry due to transceivers and cable distortion is thus regained at the output of the repeater.

7.7 Environment Specifications

The following sections specify the physical environment in which all channel components must operate to be considered compatible.

7.7.1 Electromagnetic Environment

The physical channel hardware shall meet its specifications when operating in the following ambient plane-wave fields:

2 Volts/Meter from 10 KHz through 30 MHz

5 Volts/Meter from 30 MHz through 1 GHz

7.7.2 Temperature and Humidity

All physical channel hardware, with the possible exception of the channel logic components shall operate over the ambient temperature range of 5 to 50 degrees Celsius, and humidity range of 10% to 95% non-condensing. The channel logic components are normally part of the station hardware, and are thus subject to individual station product requirements. Hardware which does not meet the temperature and humidity requirements specified must state so in its published product specification.

APPENDIX A: GLOSSARY

This section defines some of the essential terminology associated with the Ethernet.

baseband coaxial system: A system whereby information is directly encoded and impressed on the coaxial transmission medium. One information signal at a time can be present on the medium without disruption (see collision).

binary exponential backoff: The algorithm used to schedule retransmissions after a collision. So called because the interval from which the retransmission time is selected is expanded exponentially with repeated collisions.

broadcast: Describes the class of media for which the Ethernet is designed, in which all stations are capable of receiving a signal transmitted by any other station. Also, describes the mode of usage of such a medium by the Data Link Layer in which all stations are instructed to receive a given frame.

carrier sense: A signal provided by the Physical Layer to the Data Link Layer to indicate that one or more stations are currently transmitting on the channel.

channel logic: The logical functions provided between the transceiver cable and the Data Link, which support the defined interface between the data link and the physical layers.

Client Layer: Collective term used to describe any layer of a network architecture, which use the Ethernet Data Link and Client interface.

coaxial cable: A two-conductor, concentric, constant impedance transmission line.

coaxial cable interface: The electrical, mechanical, and logical interface to the shared coaxial cable medium. This is a mandatory compatibility interface, which must be correctly implemented by every Ethernet implementation.

coaxial cable section: An unbroken piece of coaxial cable, fitted with coaxial connectors at its ends, used to build up coaxial cable segments.

coaxial cable segment: A length of coaxial cable made up from one or more coaxial cable sections and coaxial connectors, terminated at each end in its characteristic impedance. A 500 meter segment is the longest configuration possible without repeaters.

collision: The result of multiple transmissions overlapping in the physical channel, resulting in garbled data and necessitating retransmission.

collision detect: A signal provided by the Physical Layer to the Data Link Layer to indicate that one or more other stations are contending with the local station's transmission. It can be true only during transmission.

collision enforcement: Transmission of extra, encoded "jam" bits after a collision is detected, to insure that the duration of the collision is sufficient to guarantee its detection by all transmitting stations.

compatibility interfaces: The coaxial cable interface, and the transceiver cable interface, the two points at which hardware compatibility is defined to allow connection of independently designed and manufactured components to the Ethernet.

contention: Interference between colliding transmissions (see collision). Resolution of occasional contention is a normal part of the Ethernet's distributed link management procedure (see CSMA-CD).

controller: The implementation unit which connects a station to the Ethernet, typically comprising part of the Physical Layer, much or all of the Data Link Layer, and appropriate electronics for interfacing to the station.

CSMA-CD: Carrier Sense Multiple Access with Collision Detection, the generic term for the class of link management procedure used by the Ethernet. So called because it a) allows multiple stations to access the broadcast channel at will, b) avoids contention via carrier sense and deference, and c) resolves contention via collision detection and retransmission.

Data Link Layer: The higher of the two layers in the Ethernet design, which implements a medium-independent link level communication facility on top of the physical channel provided by the Physical Layer.

deference: A process by which a data link controller delays its transmission when the channel is busy to avoid contention with ongoing transmissions.

frame check sequence: An encoded value appended to each frame by the Data Link Layer to allow detection of transmission errors in the physical channel.

interframe spacing: An enforced idle time between transmission of successive frames to allow receiving data link controllers and the physical channel to recover.

jam: An encoded bit sequence used for collision enforcement.

Manchester encoding: A means by which separate data and clock signals can be combined into a single, self-synchronizable data stream, suitable for transmission on a serial channel.

multicast: An addressing mode in which a given frame is targeted to a group of logically related stations.

physical address: The unique address value associated with a given station on the network. An Ethernet physical address is defined to be distinct from all other physical addresses on all Ethernets.

Physical Channel: The implementation of the physical layer.

Physical Layer: The lower of the two layers of the Ethernet design, implemented by the physical channel using the specified coaxial cable medium. The Physical Layer insulates the Data Link Layer from medium-dependent physical characteristics.

preamble: A sequence of 64 encoded bits which the Physical Layer transmits before each frame to allow synchronization of clocks and other Physical Layer circuitry at other sites on the channel.

repeater: A device used to extend the length and topology of the physical channel beyond that imposed by a single segment, up to the maximum allowable end-to-end channel length.

round-trip propagation time: In bit times, the time required in the worst-case for a transmitting station's collision detect signal to be asserted due to normal contention for the channel. This delay is the primary component of the slot time.

slot time: A multi-purpose parameter which describes the contention behavior of the Data Link Layer. It serves as a) an upper bound on the collison vulnerability of a given transmission, b) an upper bound on the size of the frame fragment produced by a collision, and c) the scheduling quantum for collision retransmission.

station: A single addressable site on the Ethernet, generally implemented as a computer and appropriate peripherals, and connected to the Ethernet via a controller and a transceiver.

transceiver: The portion of the Physical Layer implementation that connects directly to the coaxial cable and provides both the electronics which send and receive the encoded signals on the cable and the required electrical isolation.

transceiver cable: A four pair, shielded cable used for the transceiver cable interface.

transceiver cable interface: The electrical, mechanical and logical interface which connects the transceiver to the controller. The standard transceiver cable is a recommended compatibility interface.

APPENDIX B: NOTES ON ADDRESS AND TYPE ASSIGNMENT, AND LICENSING

Address and Type Assignment

The address and type fields will be administered by Xerox Corporation.

A block of addresses will be assigned to each licensee of Ethernet patents (see below). Others may obtain an address block or type field assignment by request. A nominal fee to cover administrative costs will be charged.

Submit written requests to:

Xerox Corporation
Ethernet Address Administration Office
3333 Coyote Hill Road
Palo Alto, CA 94304

Licensing

Ethernet incorporates features that are protected by one or more patents assigned to Xerox Corporation. Questions on the need for licensing particular uses of this specification should be directed to:

Xerox Corporation
Director of Licensing
Long Ridge Road
Stamford, CT 06904

APPENDIX C: CRC IMPLEMENTATION

Every frame contains, in its frame check sequence field, a 32-bit cyclic redundancy check (CRC) code. Because the formal mathematical definition of this code (see 6.2.4) is not suggestive of an appropriate implementation, this appendix outlines one possible implementation in terms of a feedback shift register. This type of implementation is likely to be common in practice, but is not a mandatory part of the specification.

The feedback shift register (see Figure C-1) is used to represent division of the pre-scaled message by the generating polynomial. The 32-bit register is accessed via the three signals Input, Output, and Control. When Control = 1, Input bits are shifted into the feedback shift register and also fed directly back to Output. When Control = 0, the feedback paths are disabled and the shift register shifts the complement of its contents to Output.

Before CRC generation at the transmitting end, initialization logic (not shown in Figure C-1) preloads the shift register to all 1's. Control is then held at 1 while the address, type and data fields of the outgoing frame are shifted into Input and the CRC is generated. Meanwhile, the same bits emerging at Output are transmitted over the network. When the last bit of the data field has been processed, Control is set to 0 and the complemented CRC is shifted out for transmission, starting with the x^{31} term (see 6.2.4).

CRC checking at the receiving end also begins with the shift register preloaded to all 1's. Control is then held at 1 while the incoming bits are shifted into Input to regenerate the CRC. When the last bit of the data field has been processed, the shift register should contain the CRC whose binary complement is about to arrive on the network. Since this field boundary cannot be recognized by the receiver, however, Control remains at 1 and the bits of the CRC continue to feed into the the shift register until the end of the entire frame is reached. If the two CRCs match, the final contents of the shift register is the value:

11000111 00000100 11011101 01111011

(where the leftmost bit corresponds to the x^{31} term of the polynomial and the rightmost to the x^0 term). Any other final value indicates a detected error. (The extra logic to test for this value is not shown in Figure C-1).

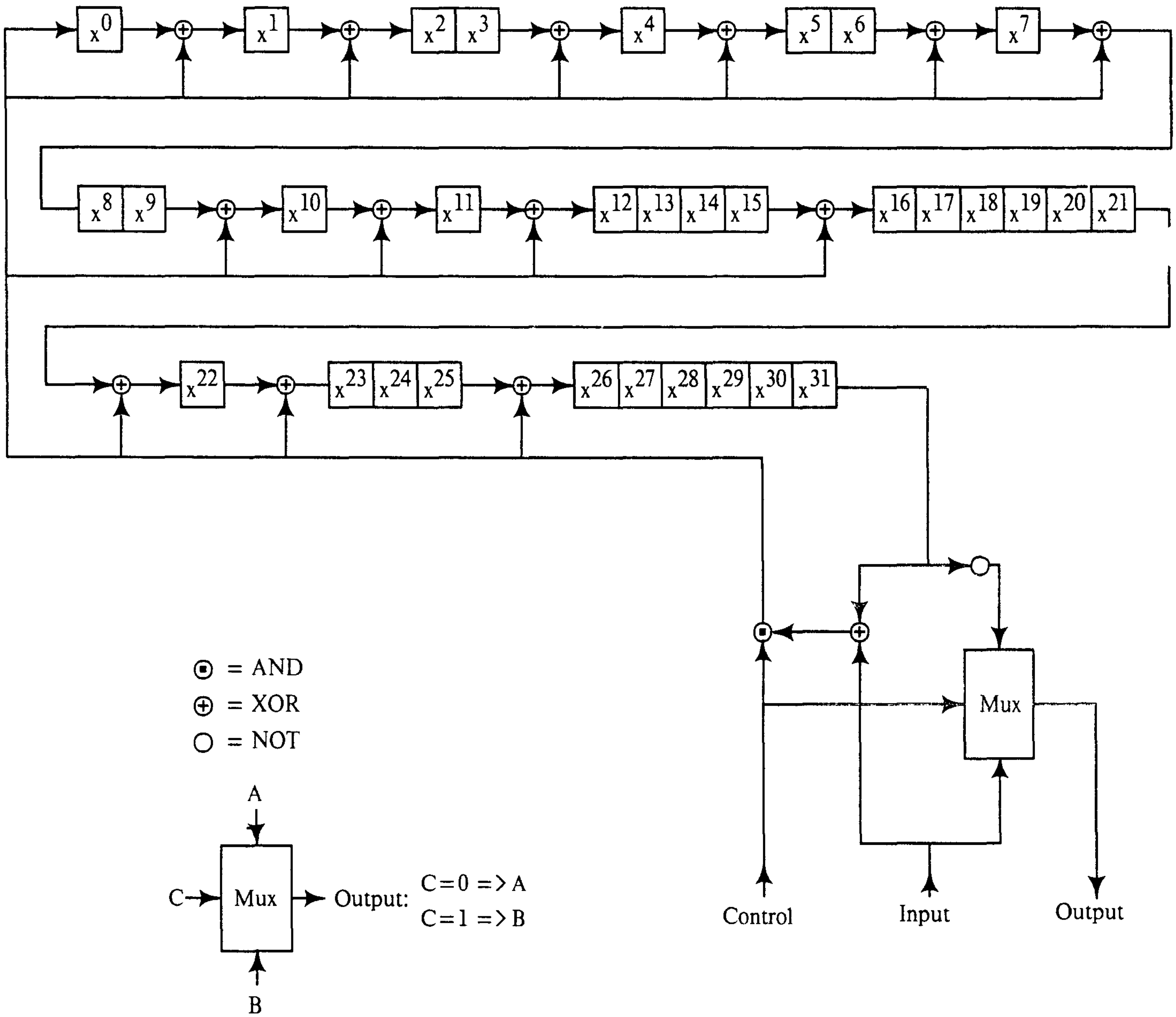


Figure C-1: CRC Implmentation

One potential problem which is avoided in this implementation is insensitivity of the shift register to incoming zero-bits when it is in the all-zero state. Following standard practice, this state is avoided at the beginning and end of the frame by preloading the shift register with all 1-bits, and by inverting each bit of the final CRC. Logically, these correspond, respectively, to the complementing of the first 32 bits of the frame and to the final complementing of the remainder, as specified in the mathematical definition in 6.2.4. See also [9] for further discussion.

APPENDIX D: IMPLEMENTATION OF TRANSCEIVER CABLE DRIVER AND RECEIVER

This appendix presents circuit digrams for typical implementations of the transceiver cable drivers and receivers. The use of these exact circuits is not necessary for conformance to the specification; equivalent circuits may be used as long as the relevant specifications are met.

Figure D-1 depicts an implementation of the transceiver cable driver specified in 7.2.4. It is suitable for use at either end of the transceiver cable, as necessary; i.e., it would be located at the station end to drive the transmit pair, and at the transceiver end to drive the receive and collision presence pairs. In addition, it is capable of driving suitable isolation circuits required to be located within the transceiver.

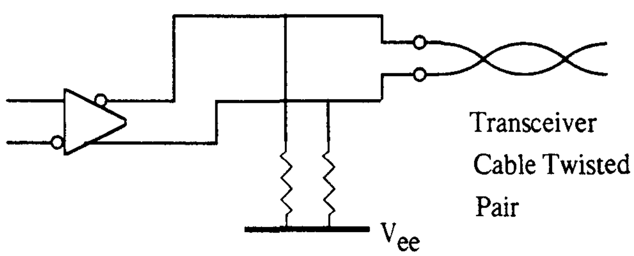


Figure D-1: Typical Transceiver Cable Driver

Figure D-2 depicts an implementation of the transceiver cable receiver specified in 7.2.5. It is suitable for use at either end of the transceiver cable, as necessary; i.e., it would be located at the station end to receive from the receive and collision presence pairs, and at the transceiver to receive from the transmit pair. It is capable of operating through suitable isolation circuits required to be located within the transceiver.

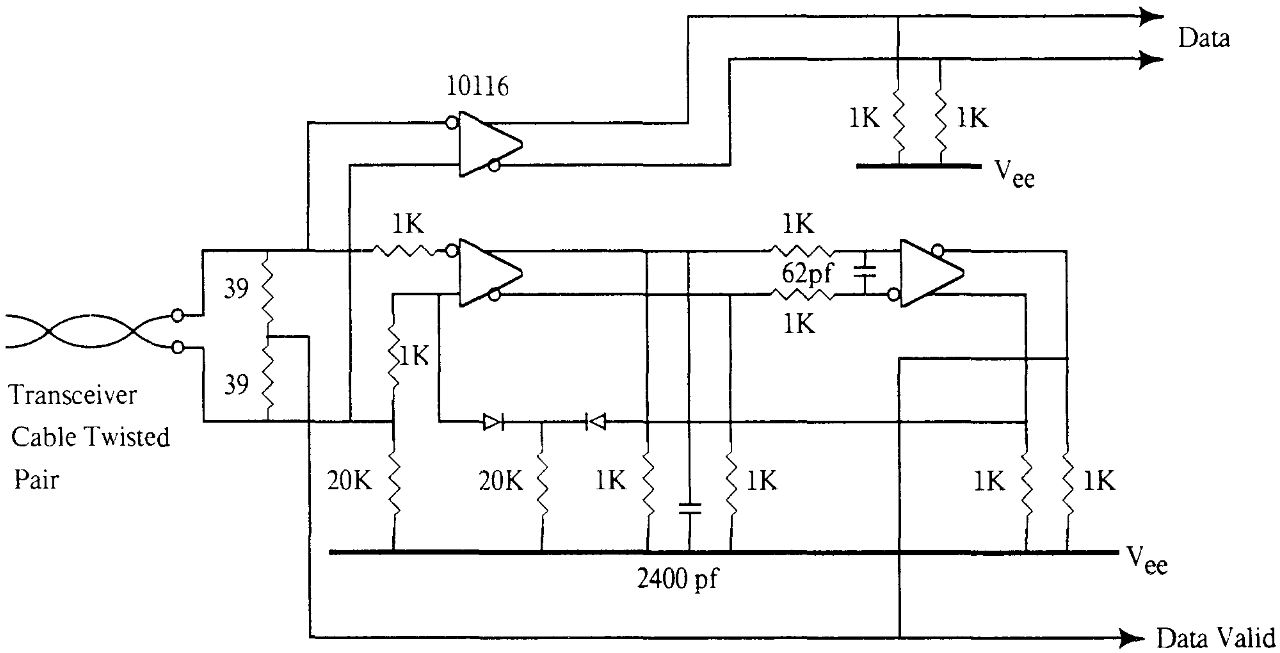


Figure D-2: Typical Transceiver Cable Receiver

APPENDIX E: INTERFRAME RECOVERY

It is important that data link controller implementations be able to receive a frame that arrives immediately after another frame has been transmitted or received. Here, "immediately" means 9.6 μsec, based on the minimum interframe spacing provided as recovery time for the data link. (See 6.3.2.2) It is important that the data link controller be able to resume reception within that time.

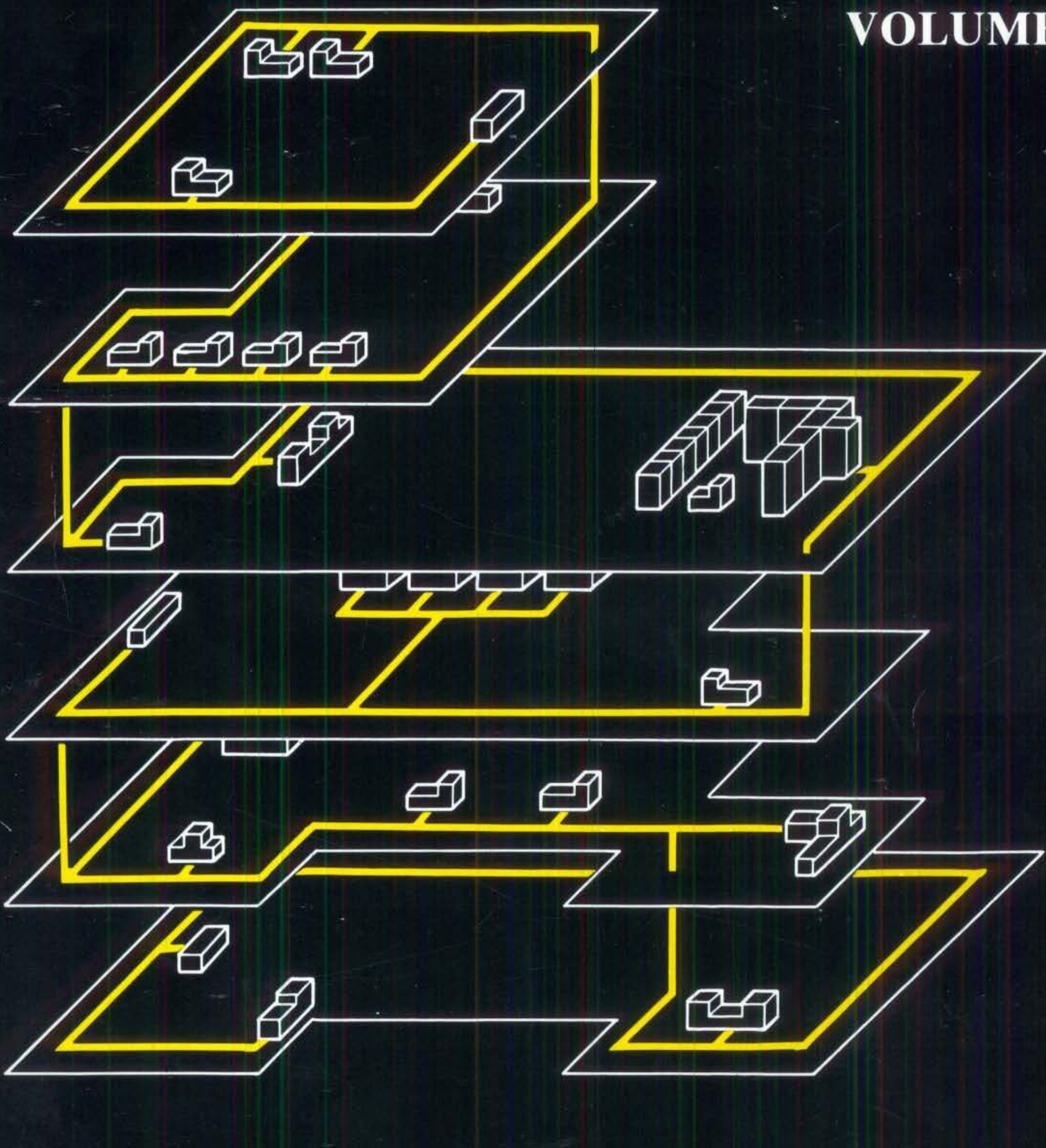
Reception of multiple closely spaced incoming frames is a very desirable capability, and is crucial for stations which tend to communicate with several other stations concurrently. There is one important case in which a data link controller implementation cannot reasonably be expected to receive closely spaced incoming frames: if the station hardware (e.g. I/O bus) is intrinsically unable to accept the bits of a frame at the rate at which they arrive over the network, each incoming frame must be buffered to allow the station to accept it at some lower rate. Assuming limited buffering resources (e.g. a one frame buffer), reception of subsequent frames cannot occur until sufficient buffer space is available. This mode of operation is allowed for low performance stations.

Reception of an incoming frame immediately after transmission of an outgoing frame is a very important capability, even for stations which do not tend to communicate with several other stations concurrently. All stations, low performance to high performance, should allow reception of an incoming frame immediately after transmission of an outgoing frame.

LOCAL AREA NETWORKING:

ETHERNET

VOLUME I



ED-16317-05

LOCAL AREA NETWORKS: ETHERNET

DIGITAL EQUIPMENT CORPORATION
CORPORATE SALES TRAINING DEVELOPMENT
MAYNARD, MASSACHUSETTS

Printed in U.S.A.

1st Printing, March 1982

LOCAL AREA NETWORKS: ETHERNET

CONTENTS

I. INTRODUCTION

II. WHY DIGITAL BELIEVES ETHERNET IS A UNIFYING KEY TO THE 5TH GENERATION - Gordon Bell

III. PERFORMANCE OF A SIMULATED ETHERNET ENVIRONMENT - Bill Hawe

INTRODUCTION

This volume contains two presentations from the Ethernet Press Seminar which was held in New York City on February 10, 1982, and in Amsterdam, The Netherlands on March 2, 1982. The goal of the seminar was to educate the business and technical press about the status of Ethernet.

Presentations were given by Digital, INTEL, and XEROX. The Digital speakers and their presentations are included in this volume. They are:

1. Gordon Bell, Vice President, Engineering: "Why Digital Believes Ethernet Is A Unifying Key to the 5th Generation".

2. Bill Hawe, Principal Engineer, Systems Performance Analysis Group: "Performance Of A Simulated Ethernet Environment".

We are making these presentations available to you for three reasons:

- To provide important information about Local Area Networks and Ethernet

- To reinforce Digital's corporate commitment to Ethernet

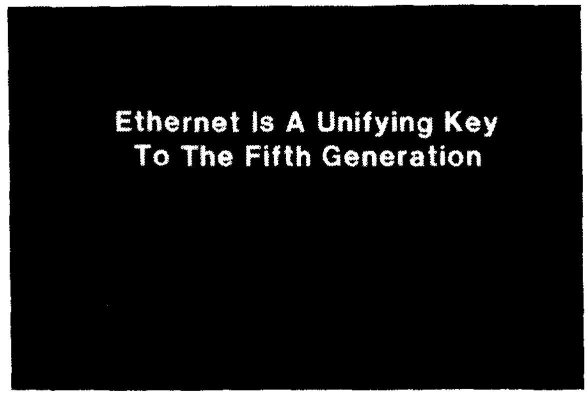
- To provide materials that could be used in customer presentations

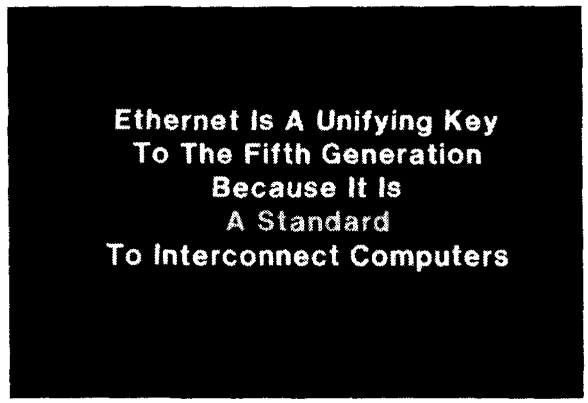
The actual 35mm slides have been distributed to worldwide district offices. You may order additional copies of slides through Corporate Sales Communications, mail code: BG/S51, RCS: BG51. Order # EF-16317-05.

WHY DIGITAL BELIEVES ETHERNET IS A UNIFYING KEY TO THE 5th. GENERATION

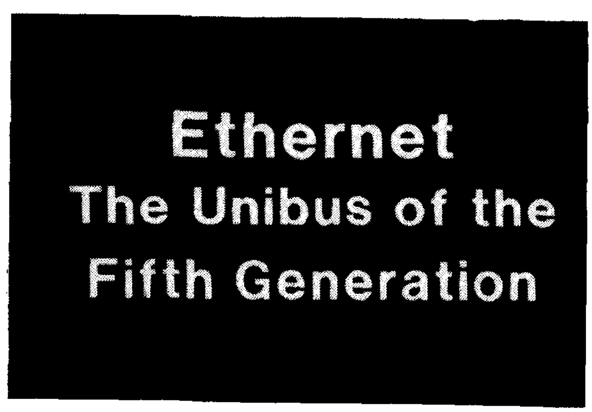
GORDON BELL
VICE PRESIDENT,ENGINEERING
DIGITAL EQUIPMENT CORPORATION

NOTE: Presented at ETHERNET PRESS SEMINAR
New York City, February 10,1982
Amsterdam, The Netherlands, March 2,1982


Ethernet Is A Unifying Key
To The Fifth Generation


Ethernet Is A Unifying Key
To The Fifth Generation
Because It Is
A Standard
To Interconnect Computers


No One
Vendor
Has It All
intel
digital


Ethernet
The Unibus of the
Fifth Generation

SLIDE 1

In the Fifth Computer Generation, a wide variety of computers will communicate with one another. No one argues about this. All the shouting is about how to do it and what form the computers will take.

SLIDE 2

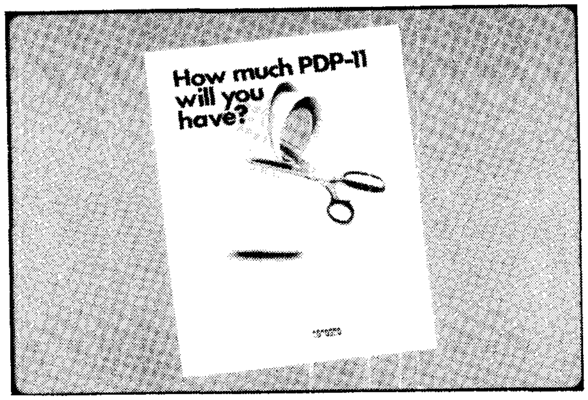
A standard communications language is the key. I believe Ethernet is this unifying key to the 5th computer generation because it interconnects all sizes and types of computers in a passive, tightly-coupled, high performance fashion, permitting the formation of local-area networks. Ethernet is the standard that can hush the argument and let everyone get to work on the computing nodes.

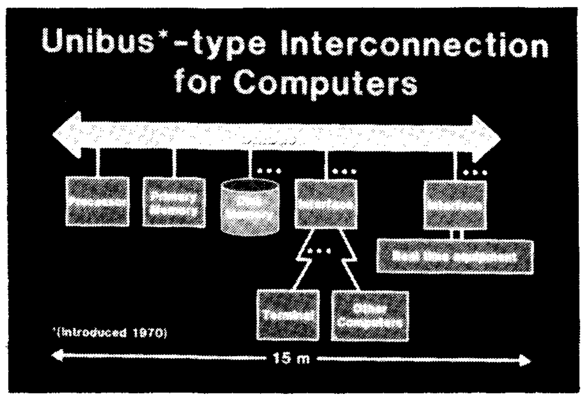
SLIDE 3

Standardization is necessary because no one vendor has it all, or can provide the full spectrum of information processing nodes that are emerging. Most organizations have computers built by different vendors. Although computer data and processes (that is the work) are interdependent, no easy and inexpensive way to send data among machines exists. Everyone's customers are demanding a network standard. Ethernet can do it for everyone.

SLIDE 4

I'm going to tell you four stories that illustrate the different facets of Ethernet. The first is about the UNIBUS and why I think Ethernet is the UNIBUS of the Fifth Generation.


How much PDP-11
will you
have?


Unibus*-type Interconnection
for Computers
*(Introduced 1970)
15 m

SLIDE 5

In 1970 Digital introduced the UNIBUS to interconnect parts of a computer. The UNIBUS is just a simple ribbon-like cable with 56 conductors as shown in this old ad. With UNIBUS people could easily assemble their own computers and did so in many different ways, and it became a standard.

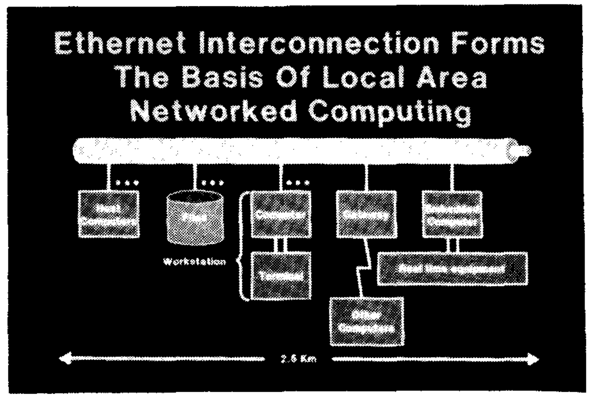
Virtually all computers built today utilize a UNIBUS-type architecture, including Intel's Multibus, and Motorola's Versabus. Both of these busses are standards too.

SLIDE 6

This bus is a high-speed data path that links all system components within a single computer -- the processor, primary memory, secondary disk memory, communications interfaces, realtime equipment interfaces, interfaces to special customer equipment.

The complete UNIBUS specification is contained in a manual about 1/2 inch thick, roughly the size of the Ethernet blue book specification. From this, users have designed 10's of thousands of machines to match the computer to their application in an almost open-ended fashion. Small dedicated controllers, personal computers, pedagogical machines and large timeshared computers are all built this way. Any kind of computer can be built easily from a common set of components.

What started as a good scheme for interconnecting components that Digital supplied, became a lovely standard for starting a whole plug-compatible business. The unexpected result: an industry with 100's of vendors and lots of new competitors. The plug compatible parts mean lower prices. The non-mundane user designed connections to television cameras, robots and other devices act to stimulate the whole next computer generation, based on need.



	Unibus	Ethernet
Purpose:	Interconnect A Computer	Interconnect Computers To Form A Network
Distance:	15 Meters	2,500 Meters
Components Connected On The Bus:	Processor Primary Memory Disks Printers Terminal Interface Communication Interface Real Time Interface	Computer Workstations File Servers Print Servers Terminal Servers Gateways Real Time Equipment

SLIDE 7

Ethernet is only an extended unifying bus, like UNIBUS, that interconnects many computer based information processing systems but in a 2.5 by 2 kilometer area.

UNIBUS has a single processor for one computer. Ethernet can support many different computers in all sizes and places doing all types of work.

A UNIBUS system has local data storage; an Ethernet supports databases distributed throughout the network. The latter has evolved to be called the file server.

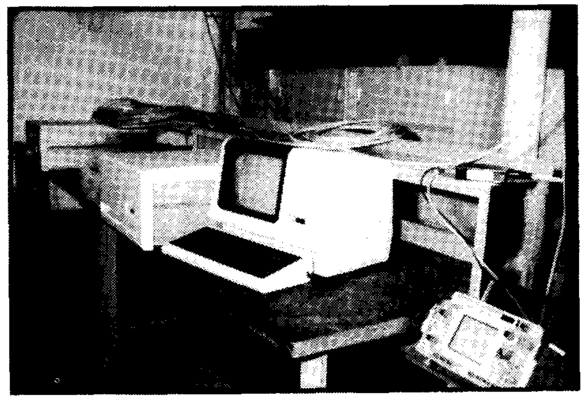
A UNIBUS system interfaces to other computers via slow communication links and tightly coupled parallel links. An Ethernet always interfaces to other computers directly. Components that are not computers are just not built today. Interconnection occurs directly and via special computers called gateways.

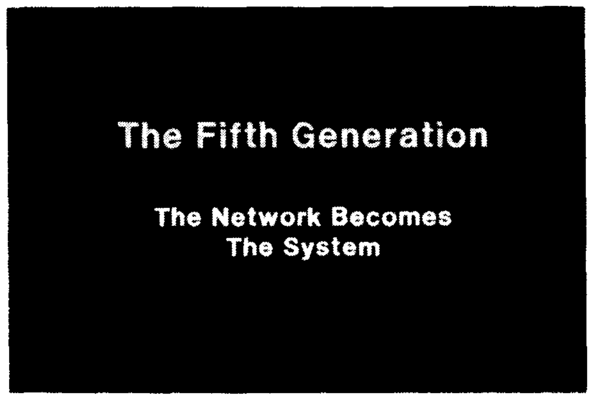
SLIDE 8

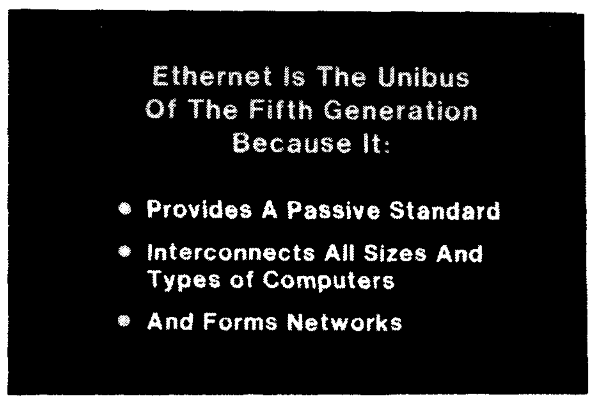
Ethernets couple host processors, people using their own special terminals, personal computers and workstations, as well as particular functions like file servers, print servers, communications servers, and realtime equipment in the laboratory and factory.

Gateways to other computers and networks can be provided by these communications servers.

Computer systems decomposed into separate, functional units on an Ethernet will be significantly easier to build.




The Fifth Generation
The Network Becomes
The System


Ethernet Is The Unibus
Of The Fifth Generation
Because It:
Provides A Passive Standard
Interconnects All Sizes And
Types of Computers
And Forms Networks

SLIDE 9

Then users will participate more than ever in the design and building of their own systems and not be limited by the vision of a single supplier.

SLIDE 10

In the Fifth Generation, every computer on the Ethernet, will be both contributing to and sharing in the total resources of the network. The network will be the system.

SLIDE 11

Having demonstrated that Ethernet is the UNIBUS of the 5th generation because it provides a passive standard to interconnect all sizes and types of computers into a high speed network, I will turn to the issue of Ethernet's role.

The second story is about the evolution of the computer generations - driven by the semiconductor evolution.

The Fifth Computer Generation Is The Convergence Of:

Technology: VLSI (For Computers, Memory)
Advanced Graphics
Ethernet

Need: Higher Speed Intercommunication Among Dispersed Computers

Forming A Much Improved

Structure: Local Area Networks

Resulting In A

Use: Ex Post-Facto Observable

A Local Area Network Is A

Set of Information Processing Nodes, Distributed In A Single Area And Fully Interconnected Via High Speed Links

SLIDE 12

The Fifth Computer generation, like its predecessors, will only occur when there are new technologies and needs that converge to create a new computing structure.

Three technologies are fueling the 5th generation: the understanding of how to build a reliable Carrier Sense Multiple Access with Collision Detection (CSMA/CD) type network, in effect the Ether; Very Large Scale Integrated Circuits or VLSI permitting all logic to be computer based, but more importantly permitting a simple, low cost connection to the Ethernet cable, essential for a standard; and finally technologies such as high resolution graphics that accelerate the creation of computing nodes that are a pleasure to use.

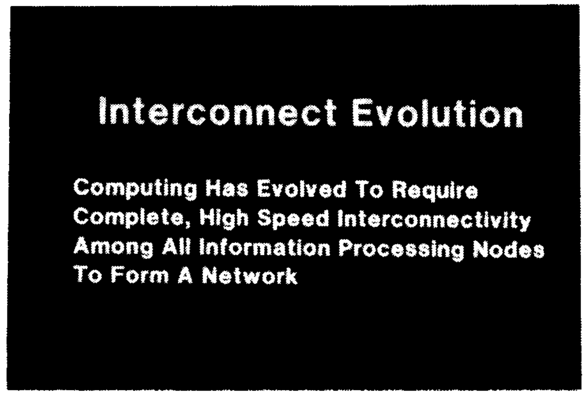
More computer use results in increasing human potential and hence an increasing need or demand. GNP grows with the absorption of new technologies that allow higher productivity. Every person's productivity is limited by the rate computers communicate with one another. In effect, we have evolved the quadruped to a thoroughbred but not changed the track. The only paths that they can travel are muddy, rocky and random time-worn paths. We need a fast race track.

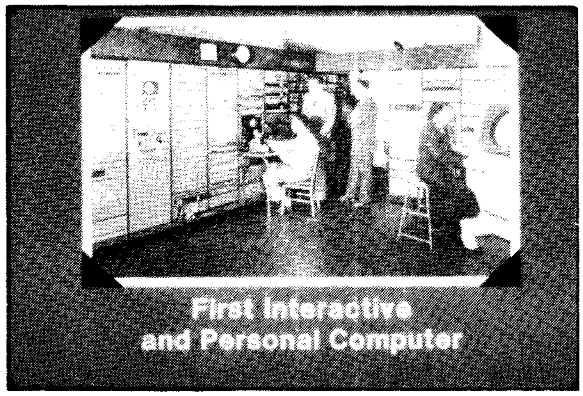
Our computers often wait at the gate while users physically carry data between them in what is becoming an inverted society -- the computers do the fun thinking parts and the users carry trivia from machine to machine, or become simple machine to people translators. Ethernet breaks this communication bottleneck. Furthermore, Ethernets can carry voice, graphs and pictures as well as simple messages and data files. They'll restructure use. It won't be a straightforward extrapolation of simple terminal to computer, and computer, to computer networking we know today.

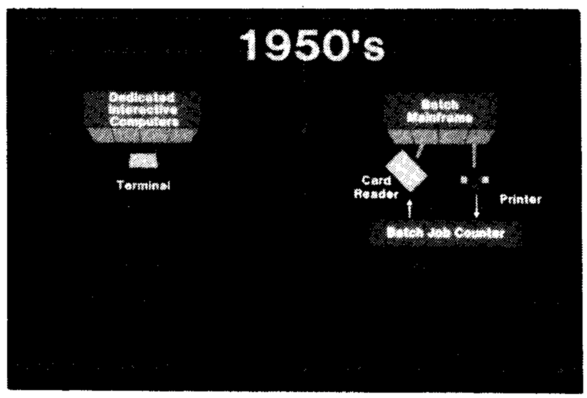
In 1990, we can probably look back and identify trends that are not clear today. So I won't speculate about 1990, but I know the future will be more interesting than the simplistic, evolutionary view I'm presenting today.

SLIDE 13

The development can only happen if we provide the creative environment in which to invent. I think the Ethernet based open Local Area Network is this environment. "A local-area network is a set of information processing nodes, distributed in a single area and fully interconnected via high-speed data links." An open local area network is one in which any vendor or user can supply nodes for the network.


Interconnect Evolution
Computing Has Evolved To Require Complete, High Speed Interconnectivity Among All Information Processing Nodes To Form A Network


First Interactive and Personal Computer


1950's
Terminal
Card Reader
Printer


Batch Computer

SLIDE 14

The user should be able to communicate over a local area network with the same nonchalance as the telephone, not knowing or caring how the network works or how the message is transmitted.

SLIDE 15

It's amazing that the front end user portion of the telephone and the computer really haven't changed much. The oscilloscope of the Whirlwind (the first real time interactive computer built in 1950) is just a bit bigger and more graphic than the ones on computers today. Jay Forrester and his associates used it as a personal computer. The user walked into a building that was the computer, and into a room that was the console, and sat down at the cathode ray tube. The computer spent most of its time waiting for the user to interact. This wasn't the best use of the world's only interactive personal computer.

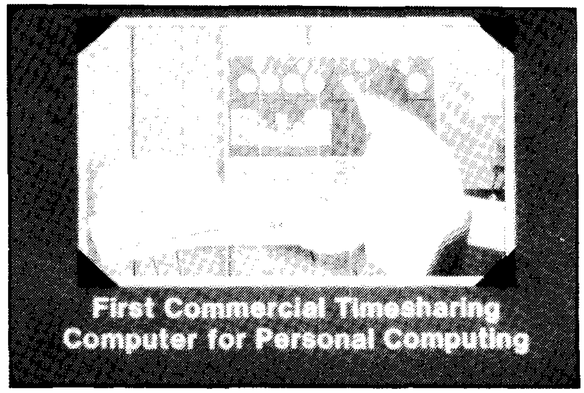
SLIDE 16

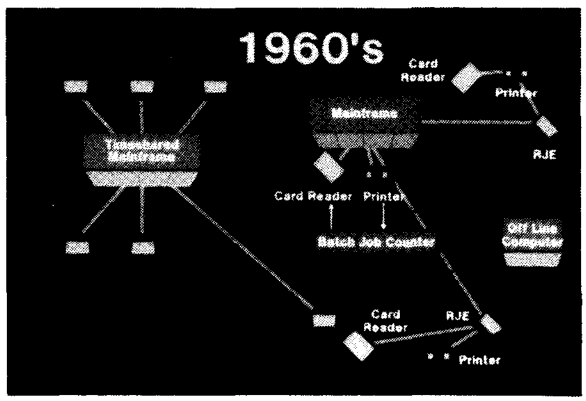
Other early machines, such as the first one, EDSAC built by Maurice Wilkes in Cambridge, England, sought to be more efficient by keeping the users away from the machine. The programmers worked off-line and then handed programs on paper tape to people who put them on a clothes line and eventually fed them into the computer. This maximized the machine's use.

SLIDE 17

But isolated users quickly grew to hate and to be intimidated by the batch computers. People would prepare their programs on punch cards, submit them to a clerk and the program would be put in the queue. As often as not, errors were found in the program or data so instead of getting an answer to an immediate business problem the user had to rekey his program and and go back to the end of the line. It's no wonder that users wanted a different way of doing things.


First Spacewar Game


First Commercial Timesharing
Computer for Personal Computing


1960's
Card Reader
Printer
Mainframe
RJE
Card Reader
Printer
Batch Job Counter
Card Reader
RJE
Printer

SLIDE 18

With the introduction of transistor technology, computers started to get smaller. In 1960, Digital introduced the PDP-1, the first commercial computer with an interactive video display that played Space War, the granddaddy of all computer space games. In 1961, two typewriters were connected to a PDP-1 at Bolt, Bernanek and Newman and the timesharing idea was born.

SLIDE 19

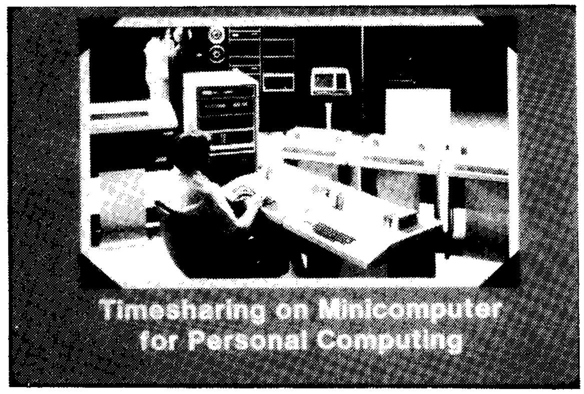
In 1963, Just two years after the first experiment, Digital introduced the first commercial timesharing system, PDP-6, for 8 to 16 users.

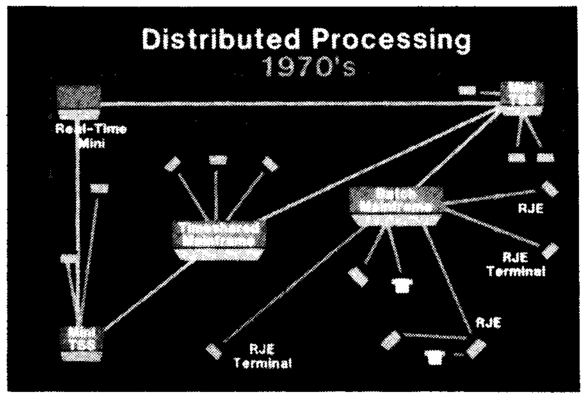
SLIDE 20

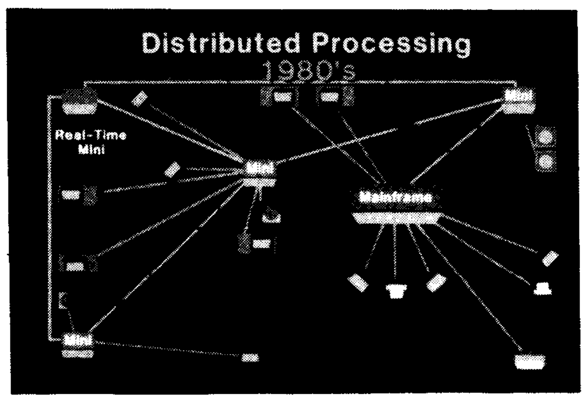
Then the computer's time, wasted waiting for one user, was used by another. Throughout the sixties, the evolution of batch, personal and time-shared computers continued. Batch mainframes were developed with remote job entry terminals so a few lucky users could enter data from their offices.

Minicomputers, like the PDP-8, were small and inexpensive enough so they could be dedicated to particular applications. Many of these minicomputers were used to prepare data for batch processing on a mainframe.

Other mainframes became specialized timesharing machines. But computing was still very expensive and impersonal.


Timesharing on Minicomputer
for Personal Computing


Distributed Processing
1970's
Mini
RJE
RJE
Terminal
RJE
RJE
Terminal


Distributed Processing
1980's
Real-Time
Mini
Mini
Mini
Mainframe
Mini

SLIDE 21

The real breakthrough came in 1972 when we learned how to provide timesharing on a minicomputer. For the first time,low cost, interactive, **personal** computing capabilities could be provided at a cost that most users could justify. Computers came out of the computer rooms and started working with users.

SLIDE 22

After the initial honeymoon, a need developed to interconnect the machines to each other and to the large batch machines which by now could be controlled from terminals. As a result, engineers did what came naturally and started to string wire between them.

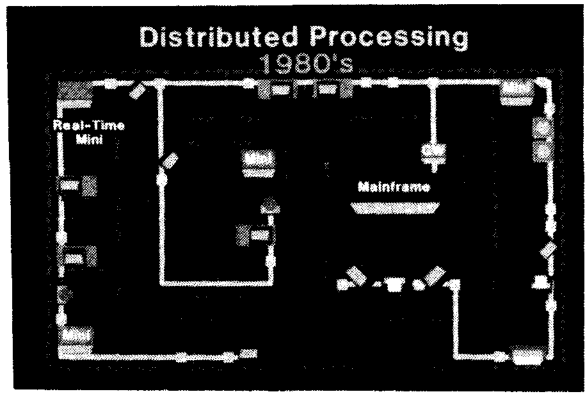
SLIDE 23

In the late 70's the interconnection problem was exacerbated by the baby computer boom, known as personal computers.

Like children everyone wants a limited number for their very own. Personal computers give that one-on-one relationship. There's no longer anyone watching you work, not even an accounting program. You can do your own thing in a non-threatening way. No one need know if you use the machine or even if you turn it on... or it turns you on.

But then there are times that you and your personal computer want to be connected with another machine to get programs, transmit messages, look at a picture, or send a non-intrusive voicegram message.

And so many more wires have to somehow be added between the centralized, shared remote batch mainframe; the departmental timeshared minicomputers; and the individual personal computers. If there aren't lines running between all the machines then there probably should be. Otherwise, information that is on one node and needed elsewhere has to be re-entered.



Local Area Networks Address These Needs:

- High Speed Interconnection Among Dispersed Computers (i.e. High Performance Networks)
- Simplified Interconnection Of Terminals And Personal Computers To Host Processors
- Interconnection Of Evolving Computer Controlled Equipment With Other Information Processors

Needs

Open Ended And Complete Interconnection Of:

- **Terminals And**
- **Personal Computers**

To:

- **Host Processing Facilities And**
- **Special Processing Facilities**

SLIDE 24

Ethernet will provide the structure needed to manage distributed computing. It's coherent structure is capable of handling an ever-growing volume of traffic among all machines.

SLIDE 25

The last two stories address Ethernet user needs.

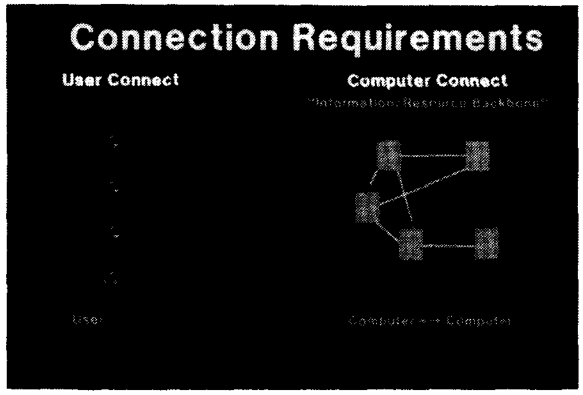
First, they provide high-speed interconnection among dispersed computers. Creative programmers are kept happy and work effectively when connected to high speed systems. They want to be able to call all the machines in their network and communicate with others in the nework independent of where they are. When we're working with a machine, we have less patience than a 2 year old waiting for a cookie.

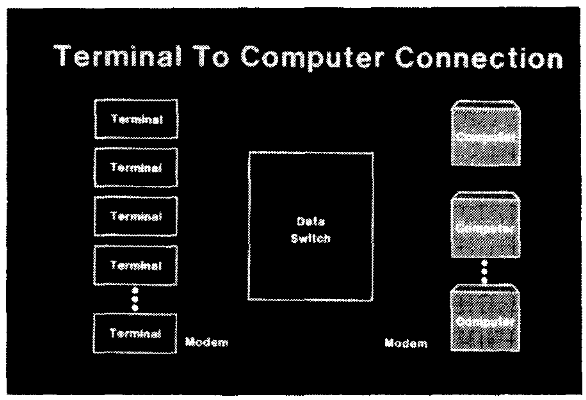
Second, Ethernets provide simple interconnections of terminals and personal computers to host processors. New users starting with simple personal computers will be able to improve their performance by accessing larger machines as their needs increase. Clearly, history has shown that the more computer power anyone has, the more he wants. It is an insatiable hunger like none known before with the immediate reward of greater individual productivity.

Third, Ethernets interconnect all kinds of computer controlled equipment. For example, links between computer controlled equipment in the laboratory or on the factory floor, and data processing equipment in the office.

SLIDE 26

Every organization wants open ended, flexible links between personal computers and terminals and larger, more central computers.


Connection Requirements
User Connect
Computer Connect


Terminal To Computer Connection
Terminal
Terminal
Terminal
Terminal
Terminal
Data Switch
Modem
Modem

SLIDE 27

Today, most users have simple block mode, fixed function terminals. Nearly all of these are evolving into complete personal computer systems.

It makes little difference whether the user has a simple terminal, or a full-fledged personal computer. For simple terminals, high bandwidth is needed for character-at-a-time interaction. For effective use of personal computers, high bandwidth is needed to transfer messages, files, images and voicegram messages.

Today the typical user is most likely linked to a single host computer, and communication with other computers is through this host.

User demanded local area networks develop by users wiring various hosts and terminals together.

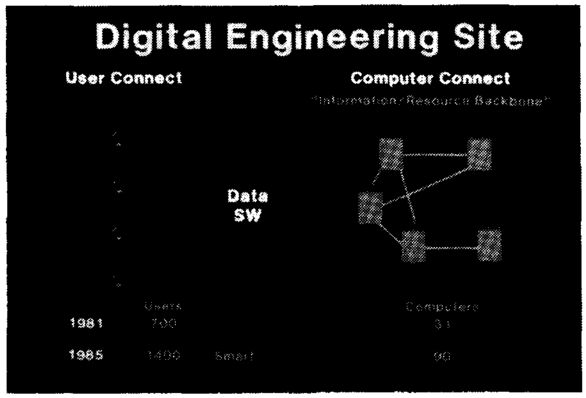
SLIDE 28

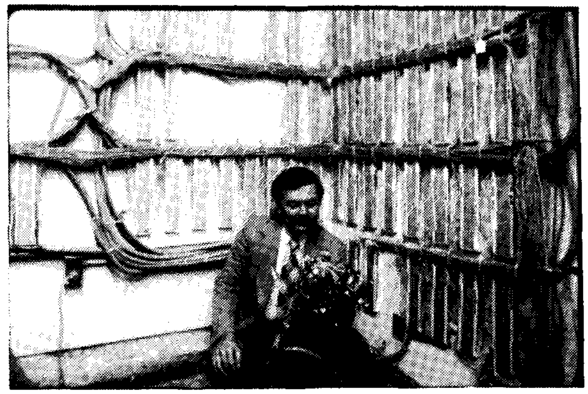
The most common answer to the problem is to use telephone lines, putting all terminal traffic onto a telephone system which may not be capable of handling it. Then modems have to be installed to convert the digital signal generated by a terminal or computer to an analog signal that can be carried over a telephone line. The biggest problem is that our users want to communicate at least at 9600 bits per second, and this just can't be done economically with these switches.

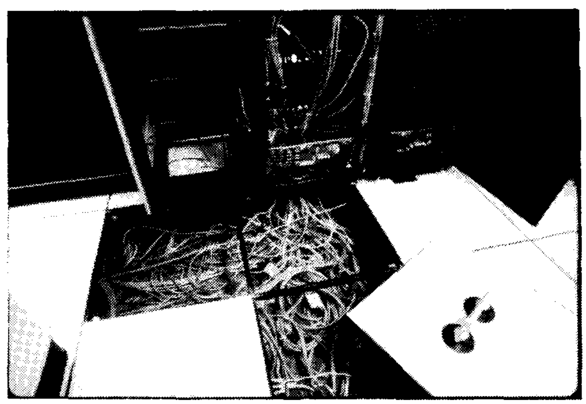
A second answer to the problem is the data switch. By installing a switch between the user and the computer network it is possible for any user to connect to any computer.

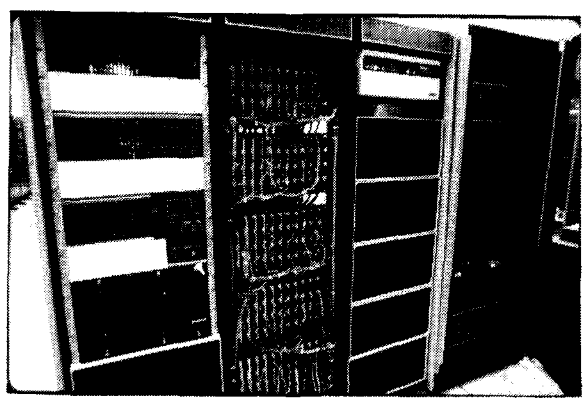
But connecting terminals to the switch involves a lot of wiring. These diagrams are simple enough to draw; building them is complicated. Furthermore, terminal wiring is a never-ending business that requires much planning, results in much inflexibility, and is fueled with much money.

So even if -- at first glance -- both telephone lines and the data switch look like solutions, they aren't. They're part of the problem as anyone who has many terminals and computers will tell you.


Digital Engineering Site
User Connect
Computer Connect
"Information/Resource Backbone"
Data
SW
Users
Computers
1981
700
1985
1400
Smart
90







SLIDE 29

I know personally, because Digital continually faces this problem like in our facility in Nashua New Hampshire where we have 3Ø computers and 7ØØ user terminals.

A pristine view shows a number of computers and a big room used for switching the links between terminals and different computers.

SLIDE 3Ø

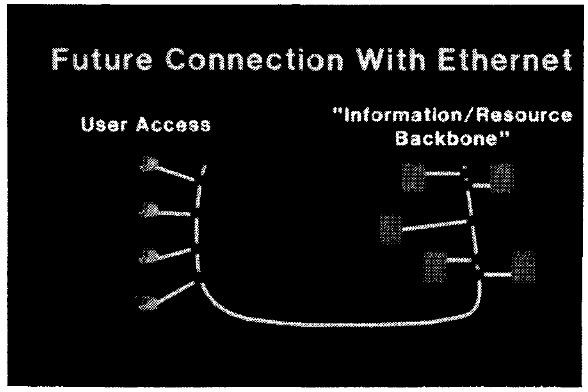
You don't see the problem until you open the door. Every terminal line is wired to a board in this room. And every time an unplanned terminal is added someone is called to run more wires.

SLIDE 31

Wires are run from the board to the wireroom to a switch computer.

SLIDE 32

This switch computer is now bound and it doesn't grow very gracefully, particularly when the number of lines is multiplied. In three years we plan (Reganomics willing) to triple the number of computers from 3Ø to 9Ø and double the number of users from 7ØØ to 14ØØ. And we probably should have planned for 28ØØ users. Without a solution that grows easily and dynamically, we are going to be strangled by the inertia of the wire and switches and our inability to plan and install them. How can we do this?



Ethernet Solves The User To Host Or Computer Connection Problem:

- Simplify Installation, While Net Operates
- Provide High Speed Communication For Interactive Computing
- Permit Open-Ended Network Growth By Direct Cable Access, Without Equipment
- Permit Tradeoff In The Number And Kind Of Connections

SLIDE 33

The users will have recognized the problem and installed a local area network long before any planner. It won't be part of a grand plan that I as head of the organization have to legislate or even worry about. With an Ethernet, direct connection is made between all user terminals via terminal concentrators and the myriad of computers.

SLIDE 34

Ethernet solves a number of problems. By solving the computer to computer interconnection problem, the user interconnection problem is resolved. Any mainframe, minicomputer, or personal computer can access the high-speed network while it is in operation. At 10 million bits per second, users don't complain because the connections are 100 times faster than direct wiring and 1000 times faster than telepone lines.

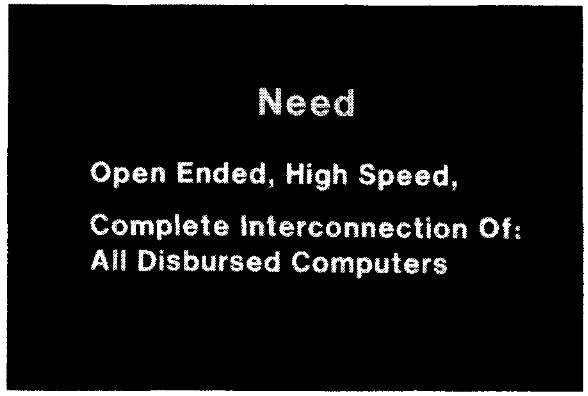
The biggest gain is open-ended network growth. Direct cable access to the network, often directly by the users, allows adding equipment while the system is in operation. No additional computers or wiring are needed. In many cases the users will have installed their own networks or network segments, as simply as checking out pencils from office supplies so that they can build their own networks by making their own connection.

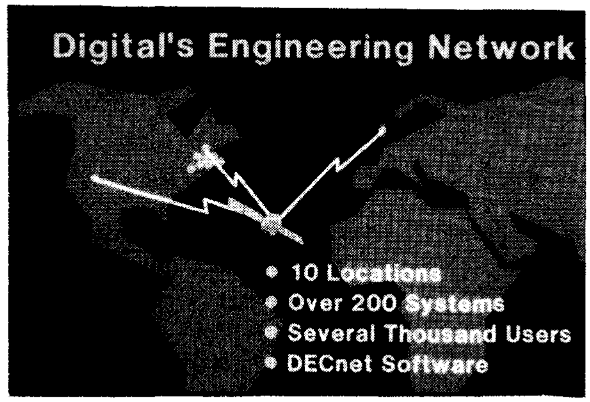
In this way the network can evolve on need rather than being limited by some planner's limited view of the future or some salesman's ability to get the wrong equipment into a site.

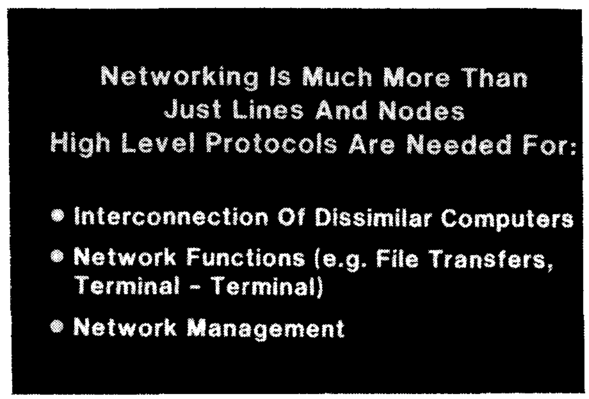
Detailed planning is one of the hardest jobs in evolving and changing organizations, whèther it's adding a new department or product line, or whether people are just moving their desks every day. In many organizations planning is done Russian style: a highly centralized top-down affair that includes the range from a new building to a box of pencils. For the dynamic growth and change that can be expected for computing, centralized planning often creates more problems than it solves.

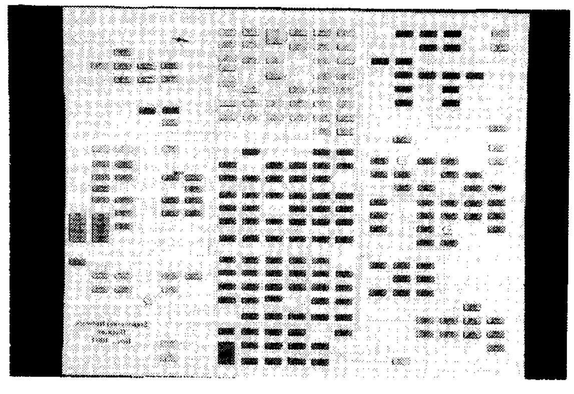
Ethernet technology solves the problem of the dynamic change, allowing tradeoffs in the number and kind of connections, the number of terminals, the number of computers on a day to day basis. The intermediary planners and doer organizations aren't needed: everyone is free to get more work done.

The result: higher productivity by eliminating a function and the interface to that function. Workers can just do the work without begging and negotiating to do work.


Need
Open Ended, High Speed,
Complete Interconnection Of:
All Disbursed Computers


Digital's Engineering Network
• 10 Locations
• Over 200 Systems
• Several Thousand Users
• DECnet Software


Networking Is Much More Than
Just Lines And Nodes
High Level Protocols Are Needed For:
• Interconnection Of Dissimilar Computers
• Network Functions (e.g. File Transfers, Terminal - Terminal)
• Network Management



SLIDE 35

The last story.

Interconnecting numerous species of computers is somewhat different than connecting terminals or personal computers to shared computers.

SLIDE 36

Again, I would like to turn to a homely example. Our Engineering Network at Digital includes over 200 computer systems serving several thousand terminal users. It looks like a bunch of interconnected links and nodes.

SLIDE 37

But a network is more than just lines and nodes despite the fact that I've been trying to show how simple one can be.

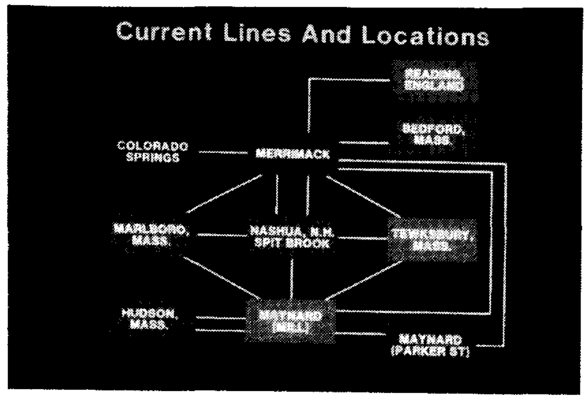
Higher level protocols are needed to support the interconnection of dissimiliar computers, to implement complex network functions such as file and data transfer of all types and terminal-to-terminal communications, and to provide network management.

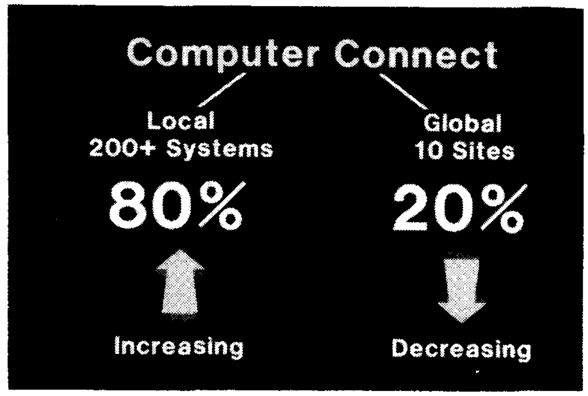
The protocols are complex. But they are a prerequisite for building a network that includes different computer systems. That's why it is critical that local area network communications are completely compatible with high-level networking protocols.

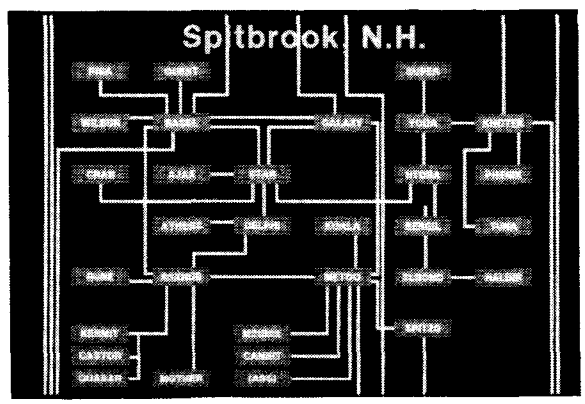
For the Ethernet Standard, we chose the Open Systems Architecture of the International Standards Organization. In addition, our own DECnet architecture is compatible with this standard.

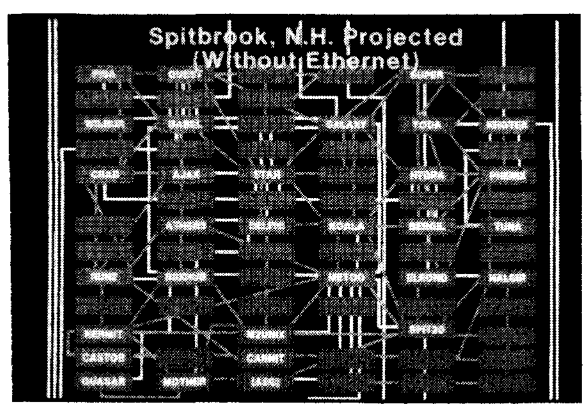
SLIDE 38

Digital's Engineering Network has over 200 computers in 10 different locations.


Current Lines And Locations
READING, ENGLAND
BEDFORD, MASS.
COLORADO SPRINGS
MERRIMACK
MARLBORO, MASS.
NASHUA, N.H. SPIT BROOK
TEWKSBURY, MASS.
HUDSON, MASS.
MAYNARD (MILL)
MAYNARD (PARKER ST)


Computer Connect
Local
200+ Systems
80%
Increasing
Global
10 Sites
20%
Decreasing


Spitbrook, N.H.


Spitbrook, N.H. Projected
(Without Ethernet)

SLIDE 39

There are sites in Massachusetts, New Hampshire, Colorado, New Mexico and England connected by special 52,000 bit per second lines and satellite links.

SLIDE 40

The number of sites is increasing more slowly, while the number of computers at each site is increasing very rapidly, and their rate of increase will accelerate as personal computers replace the simple terminals.

At least 80 percent of all network traffic is local traffic and that percentage will increase.

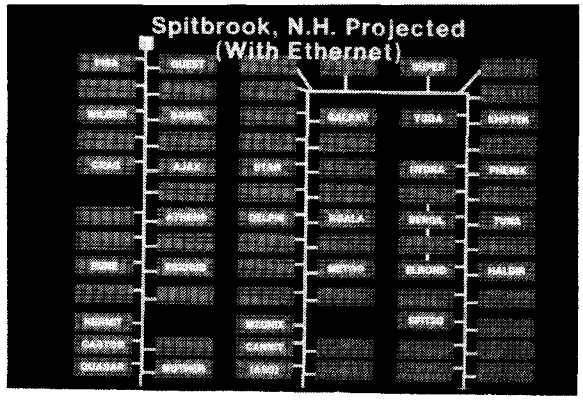
SLIDE 41

Nine links tie the 30 Computers at the Spitbrook site to other network sites.

Notice that what we are trying to achieve is full interconnectivity on a democratic, non-hierarchical basis. If we did this by running wires, 435 wires would be required to interconnect the 30 computers.

SLIDE 42

With 90 computers, 4005 links would be required for total interconnectivity. Also, over 8000 terminating controllers would be required. As you can see, interconnecting these computers on a point-to-point basis results in a topology that's so complex (not to mention so expensive) that it's bound to be ineffective and undesireable.



Ethernet Solves The Computer Interconnect Problem:

- Simplify Installation, While Net Operates
- Provide Very High Speed Communication (x 1000 Improvement)
- Reduce The Load On Computing Nodes Without An Active Switch
- Permit Orderly Network Growth With Only 1 Connection Per Node And NO Extra Links

SLIDE 43

Now see what happens when we install Ethernet. Only one wire and only one terminating control unit is needed per machine. And anyone can make the connection to the cable at any time. Everything is interconnected in a very simple and orderly way. We now have an understandable and workable structure that will provide a number of benefits.

SLIDE 44

Ethernet not only solves the connection problem, but also provides four additional benefits.

One. Systems can be connected and disconnected while the network is in operation.

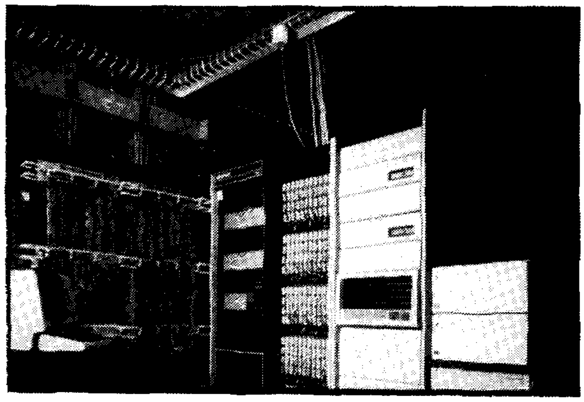
Two. Communications are a thousand times faster than via direct wire or phone line. Radically new use and applications will follow.

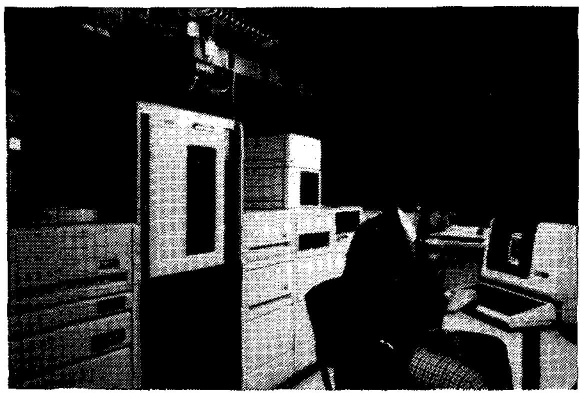
Three. Although costs are reduced we're also getting more computing for each dollar by reducing the switch load on computing nodes. If you look closely at our current network, it turns out that many nodes are primarily switching computers. Ethernet will eliminate the need to use computers as switches. In this way the computers that are doing an overhead function switching messages for their friends can go back to real computing and have fun too. Everyone's productivity is raised.

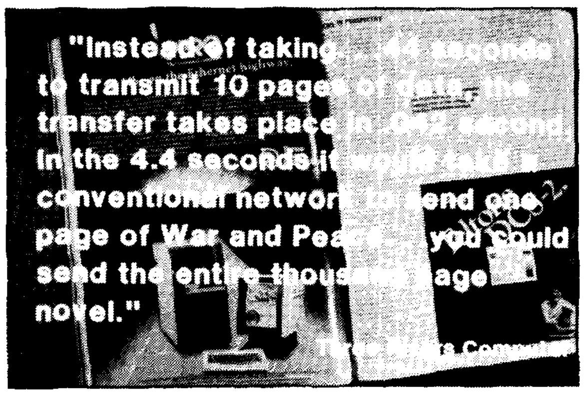
Four. The last point is the most important one. We can't have orderly open-ended growth without having a structure. With Ethernet there is only one connection per node. In traditional network structures there are many connections and equipment must be provided to switch messages. Ethernet provides a fully distributed switch without the pain and limitations of intensive and erroneous planning.

SLIDE 45

It's also been shown that Ethernet works with a variety of computers. In May of last year Digital, Xerox, and Intel had an Ethernet running at the National Computer Conference. Since each of these companies followed the same standard we were able to transfer print files and send messages back and forth between the Digital, Xerox, and Intel booths.






"Instead of taking
to transmit 10 pages
transfer takes place
In the 4.4 seconds
conventional network
page of War and Pea
send the entire
novel."

SLIDE 46

Ethernet is installed in our Central Engineering Department where we're in transition from the data switch to switching concentrators on Ethernet. Here, we also see three generations of switches: the telephone, the data switch and Ethernet.

SLIDE 47

A VAX computer connected to Ethernet.

SLIDE 48

A terminal concentrator manufacturered by one of our competitors is plugged into our Ethernet.

SLIDE 49

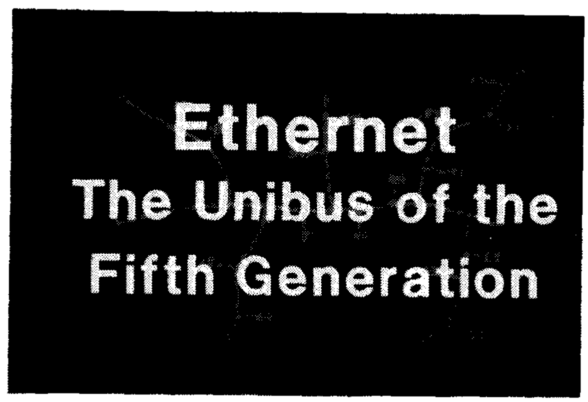
Finally I'd like to show you an ad produced by another manufacturer. Note how they feature Ethernet, and listen to what they have to say. Let me read.

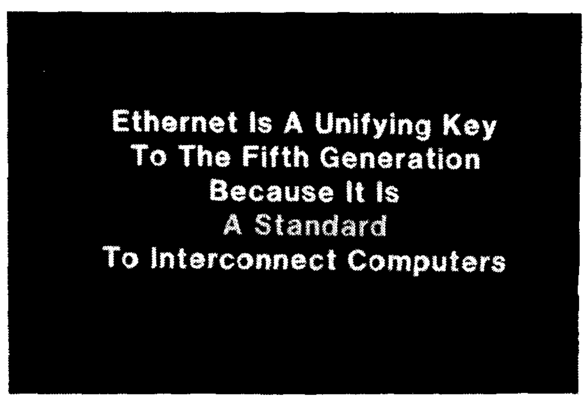
"Ethernet...gives you instantaneous access to all resources on the network, such as files, printers, other I/O devices -- even other mainframes -- plus all the speed of a dedicated single-user computer.

"In real terms, what this means is this. Instead of taking as long as 44 seconds to transmit ten pages of data, the transfer takes place in .Ø42 second. In the 4.4 seconds it would take a conventional network to send one page of **War** and **Peace**, with ... Ethernet, you could send the entire 1ØØØ page novel."

All the people in Xerox's Advertising Department couldn't say it any better. Neither could ours or Intel's. With this performance, with the ease with which you can connect systems to Ethernet, and with the number of different manufacturers lining up behind the Ethernet standard, you're going to see a growing interest in local-area networks.


Ethernet Is An Important Part Of
Fifth Generation Computing
• It Provides For Many Needs
• We Use It And Are Committed To It
• Ethernet Conforms To The International Standards And Digital Network Architectures
• We Will Be Introducing Products Within The Next Few Months
• Moreover, For The Future. . .


Ethernet
The Unibus of the
Fifth Generation


Ethernet Is A Unifying Key
To The Fifth Generation
Because It Is
A Standard
To Interconnect Computers

SLIDE 5Ø

Ethernet provides the needed structure for the Fifth Generation of computers.

It provides for many current needs. The actual use is likely to be quite different.

We use Ethernet and are committed to Ethernet.

Ethernet conforms to the Open Systems Architecture of the International standards Organization, and we believe that because of it's simplicity Ethernet will become the Local Area Network standard.

Digital will certainly be introducing products within the next few months.

Moreover for the future...

SLIDE 51

Since we believe Ethernet is the UNIBUS of the fifth generation,

SLIDE 52

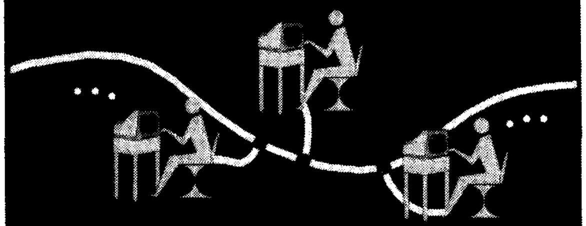
We, therefore, believe Ethernet is the unifying key to the 5th computer generation because it is the right standard to interconnect computers and for Open Local Area Networks.

PERFORMANCE OF A SIMULATED ETHERNET ENVIRONMENT

BILL HAWE, PRINCIPAL ENGINEER
SYSTEMS PERFORMANCE ANALYSIS GROUP
DIGITAL EQUIPMENT CORPORATION

NOTE: Presented at ETHERNET PRESS SEMINAR
New York City, February 10, 1982
Amsterdam,The Netherlands, March 2,1982

Performance of A Simulated Ethernet Environment



Goals

- Establish Traffic Patterns Based On Current Environment
- Predict Growth Capability Based On Ethernet Installation

Results

- Ethernet Can Support Large Numbers Of Users
- Ethernet Delays Are Small
- Few Collisions Are Experienced

SLIDE 1

In this study we investigate the performance of a simulated Ethernet environment. We wish to predict the capacity of the channel in terms of the number of active users that it can support simultaneously. This provides an understanding of the loading one could expect in a particular environment. It also establishes the capacity in the system for future growth.

SLIDE 2

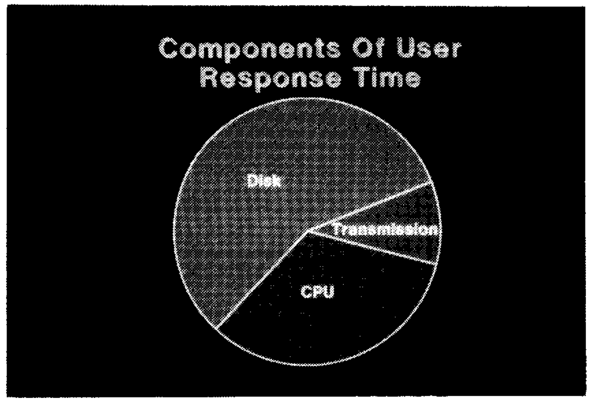
The goals were to establish the traffic patterns in the existing system and to estimate the excess capacity that would allow growth. The traffic patterns were established through measurements performed on operational systems that were interconnected with conventional point-to-point connections. We wish to see how heavily loaded an Ethernet would be if installed as an interconnect mechanism for the hosts, terminals, etc. We were also interested in understanding the additional loading that would take place because of new devices and their use (print and file servers, etc.) along with increased load due to growth in the user population.

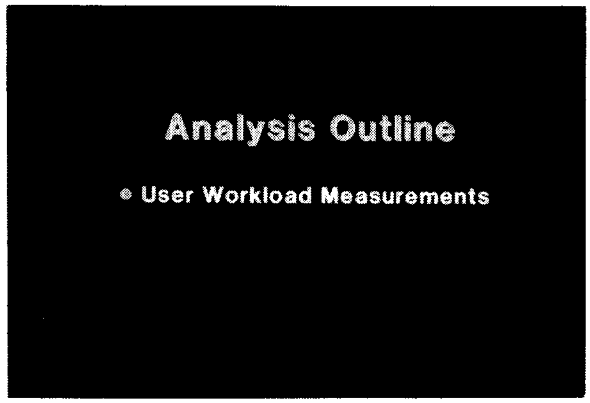
The behavior of users during various periods (such as a busy period) were monitored. The resultant data was then analyzed to produce a profile of the "typical" operations a user performs. From this, a workload which specifies the operations performed (and their frequency) was developed. This includes items such as the rates and sizes of commands, data, etc. that are exchanged between the user and the system.

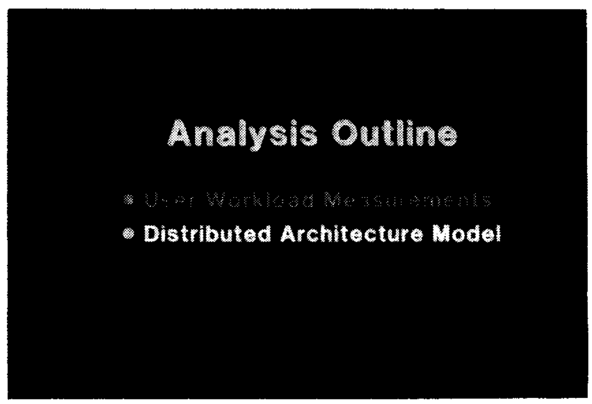
To predict the growth capability present in the system, we simulated the Ethernet using a distributed architecture model and the user workload as the source of traffic. The number of users was then increased until the idle time on the Ethernet channel went to zero.

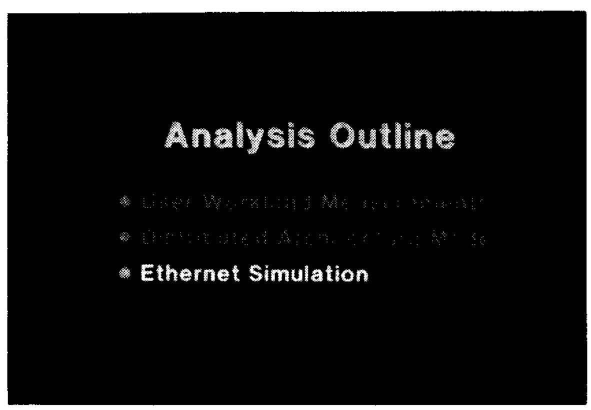
SLIDE 3

The results indicate that the Ethernet has sufficient bandwidth to support a large number of users of the type characterized in this environment. The delays in the Ethernet level of the architecture are small compared to other delays such as disk seeks, application program execution, etc. We also see that there are few collisions, even under heavy load.


Components Of User
Response Time
Disk
Transmission
CPU


Analysis Outline
• User Workload Measurements


Analysis Outline
• Distributed Architecture Model

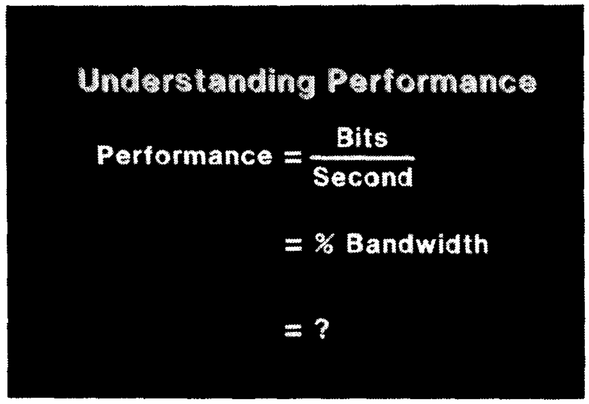

Analysis Outline
• Ethernet Simulation

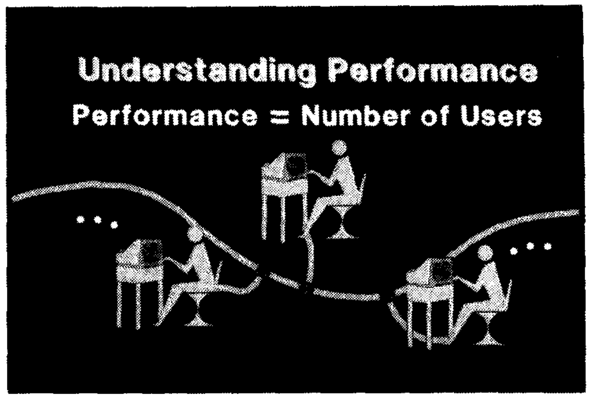
SLIDE 4

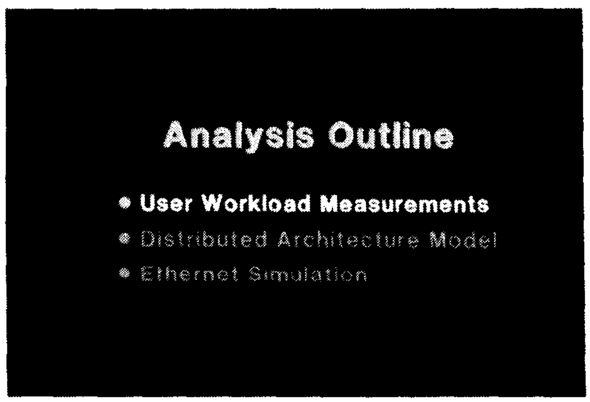
In systems such as backbone networks, the delays in transfering information from node to node are usually dominated by the transmission and propagation delays. Processing time per message at the nodes is small compared to these factors. With the advent of local area networks we see a different relationship. Local area networks are generally built using interconnection mechanisms that have speeds of around 1 to 1Ø Mbps. They are generally confined to a limited geographic area such as few buildings. This means that now the transmission and propagation delays are much smaller in relation to the disk and CPU delays. For this reason, it becomes important to consider all levels in the system when evaluating the performance.

SLIDES 5,6,7

There are three parts to the study. First, measurements were performed to characterize the behavior of users in a program development environment. From this, a user profile (or workload) was developed. Second, the user workload is used as input to a model of the distributed architecture that is used in the Ethernet network. This results in a traffic load placed on the Ethernet. Finally, this load serves as input to a detailed Ethernet simulation. The number of users using the system in the simulation is then increased to observe the effects of increased load. It is assumed that enough hosts, terminals, etc. will be added to the system to support those additional users.


Understanding Performance
Performance = Bits / Second
= % Bandwidth
= ?


Understanding Performance
Performance = Number of Users


Analysis Outline
• User Workload Measurements
• Distributed Architecture Model
• Ethernet Simulation


Program Development Environment
• Edit/Compile/Link/Run
• Mail & Talk
• Copy, Delete, Etc.
• Help, System Status, Etc.

SLIDE 8

Here we are interested in the capacity of the system. There are many ways that one can investigate this aspect of the performance. Often the capacity of a channel is expressed in bits per second or percentage of the bandwidth used on the channel. Metrics such as this are difficult to interpret when one is interested in estimating how many users the system can support.

SLIDE 9

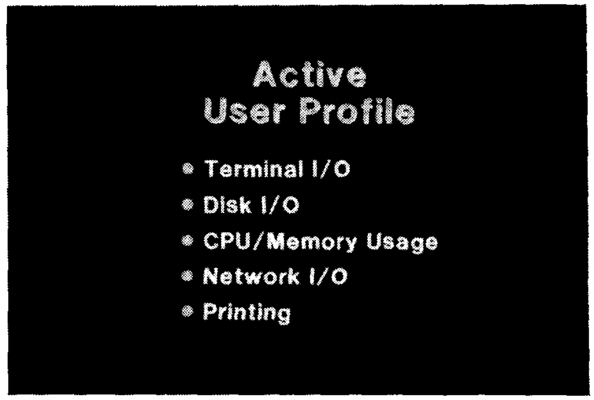
Therefore, in order to understand the capacity of the system we focus on the number of users that it can support. This is especially important when one is interested in determining whether or not there is sufficient capacity in the channel to support the existing user population as well as reserve capacity for future expansion both in the number of users and the types of traffic they generate.

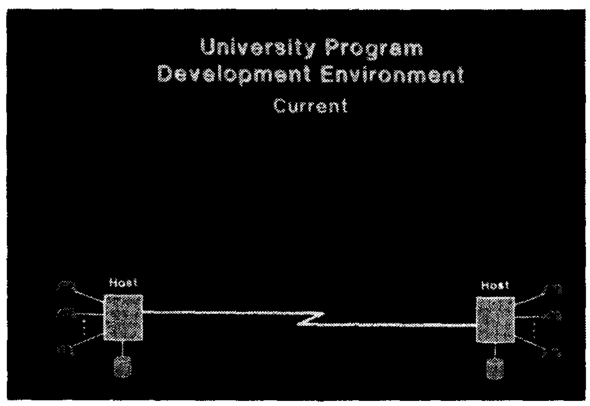
SLIDE 10

There are two ways in which the environment affects the number of users that the system supports. First, it dictates the higher level protocols to be used to transfer information between hosts, terminals, etc. This in turn affects the amount of traffic generated by each user. Second, it specifies the packet size distribution and arrival rate distribution. These play a significant role in determining the performance of the Ethernet.

SLIDE 11

Here we investigate the program development environment. Measurements were performed at several locations which were considered to be representative of this environment. As an example of such an environment we consider a large University. Users in this environment perform the obvious activities associated with the development of programs. This includes editing files, as well as compiling, linking, running and debugging the programs. They also communicate with other users by sending mail and using interactive message facilities such as "Talk". They copy, delete, print and perform other file manipulation operations. In addition to these functions, they also obtain information from the system. This includes help messages, queries about system status, etc.


Active
User Profile
• Terminal I/O
• Disk I/O
• CPU/Memory Usage
• Network I/O
• Printing


University Program
Development Environment
Current
Host
Host

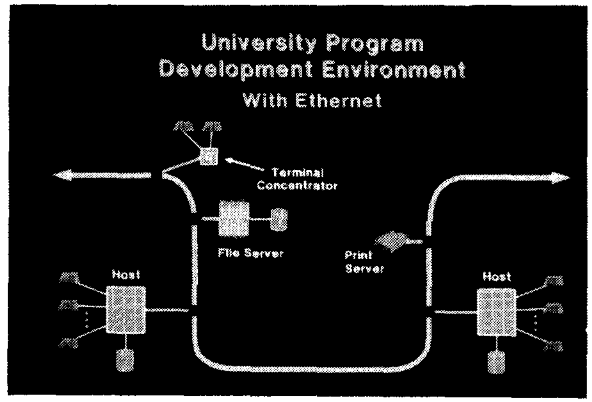
SLIDE 12

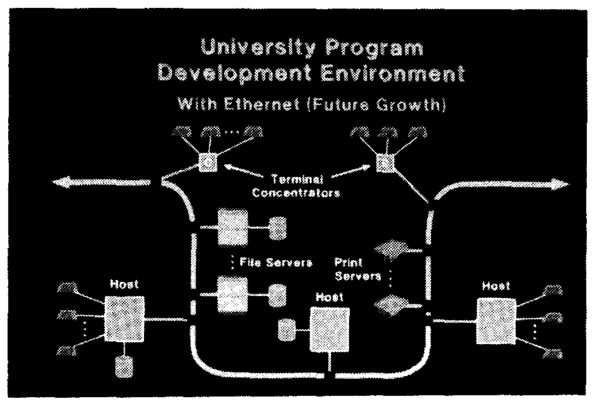
To characterize the activities of a typical user various parts of the system must be monitored. Data was collected at several installations representing this environment. The data was collected at various times during the day so that busy periods could be investigated.

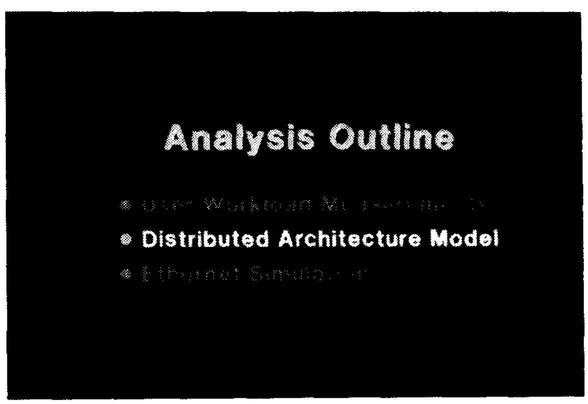
The amount and frequency of information transfer between the terminal and the host was monitored. In addition, the disk I/O that occurs as a result of operations performed by the user was also measured. This includes disk I/O that is for temporary work files such as those generated by programs such as linkers and compilers. Note that when we examine the impact of sending disk I/O over the Ethernet to a file server we do not include this type of traffic. This is because it is more efficient to generate and manipulate those temporary files at the location that the linker or compiler is running. However, the source and destination files can certainly be located on a file server. We also monitored other forms of traffic resulting from user operations. These included CPU usage, printing, network I/O, etc.

SLIDE 13

As we mentioned, the current environment uses conventional methods for interconnecting hosts, terminals and other devices. Terminals are connected directly to the hosts. The hosts are interconnected using point-to-point connections. The network is not always fully connected. However, the routing capabilities of the hosts assure that the network is logically fully connected.


University Program
Development Environment
With Ethernet
Terminal
Concentrator
File Server
Print
Server
Host
Host


University Program
Development Environment
With Ethernet (Future Growth)
Terminal
Concentrators
File Servers
Print
Servers
Host
Host
Host


Analysis Outline
• Distributed Architecture Model

SLIDE 14

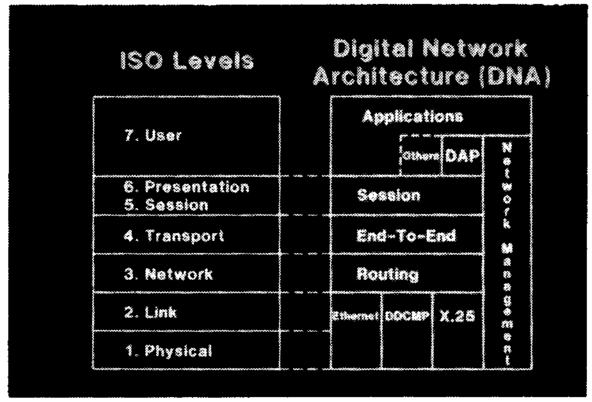
We consider the impact of an Ethernet installation in this environment. The Ethernet will carry traffic between the hosts for remote file and data access, remote logins, printing, etc. It will also carry traffic to and from new devices such as file servers and print servers. Existing terminals which are connected directly to the hosts can access remote hosts, servers, etc. by going through their hosts. Other terminals can also be connected to the Ethernet either directly (with the appropriate interface) or through terminal concentrators. With this approach they are not dependent on any one host's availability for access to the network. Personal computer workstations can also be connected directly to the Ethernet. Their traffic will be somewhat different than the terminal traffic because of the increased intelligence in the workstation. It will appear more like the host to host and host to server traffic. Gateways, routers, and other devices which allow communication outside of the local area network may also be connected directly to the Ethernet. Often hosts implement these functions in addition to their normal duties. The traffic which flows through those devices can be of any of the types already described.

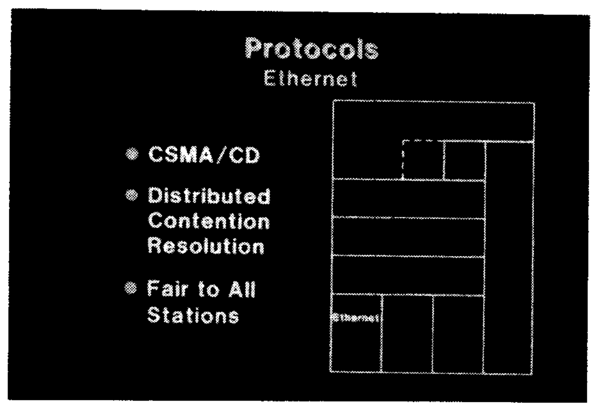
SLIDE 15

As time passes the network will expand in several ways. More devices will be added as the user population increases. This includes terminals, concentrators, hosts, servers, etc. Hosts without local terminals could be added and called computing servers. The other way in which the network will expand is in the traffic patterns. The availability of devices such as file and print servers will stimulate the growth in the traffic associated with those devices. For example, as more files are moved to file servers, so that sharing is easier, the devices will be used more often.

SLIDE 16

We have discussed the users and their environment. Now we discuss the distributed architecture. A distributed architecture is necessary to provide an effective local area network. There must be facilities, for reliable, controlled communications between users and processes inside and outside the local area network. This means that we need mechanisms for a user on the local area network to access information not only on the local network but also at some location that is not local. This would be accomplished by going through a gateway or router. Therefore, Ethernet is only a part of the total network architecture. It represents the lowest layers and is thus the foundation on which the local network is built.


ISO Levels
Digital Network Architecture (DNA)
7. User
6. Presentation
5. Session
4. Transport
3. Network
2. Link
1. Physical
Applications
Others
DAP
Session
End-To-End
Routing
Ethernet
DDCMP
X.25
Network Management


Protocols
Ethernet
CSMA/CD
Distributed Contention Resolution
Fair to All Stations
Ethernet

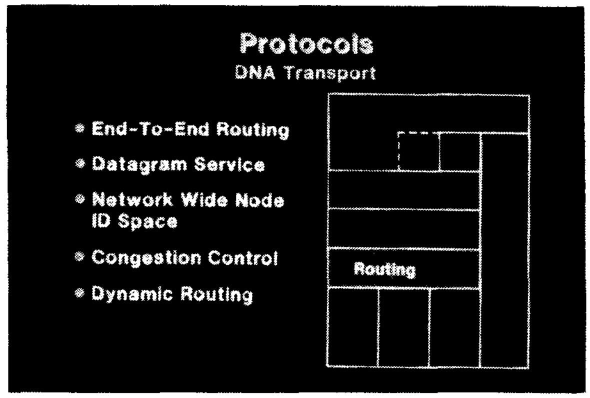
Slide 17

The Digital Network Architecture (DNA) is an example of a complete network architecture. Here we see the relationship between DNA and the ISO layered architecture. The Ethernet comprises essentially the lower two levels for the local area network. Parts of the system which interface to public data networks could use the X.25 services. Other point-to-point links could use the DDCMP facilities. Above the data link is a network wide routing service. This delivers packets to the appropriate destination - either locally or remotely. Above that is an end-to-end service which provides for reliable communications between two processes. The Session layer controls the end-to-end service. Above that we have the applications and special purpose protocols. The network management facility has access to most of the protocol levels. It is used to monitor as well as control and configure them.

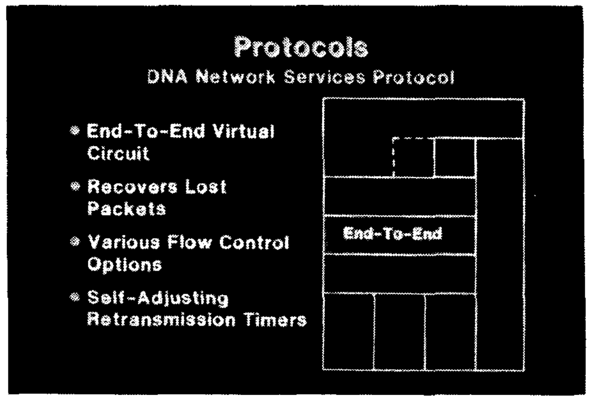
It is very important that all these layers in the architecture be considered when examining the user perceived performance of the local network. This is because each layer will add some additional load to the components of the system. Most will add some amount of additional traffic to the Ethernet. They will also use resources such as CPU cycles and memory space.

SLIDE 18

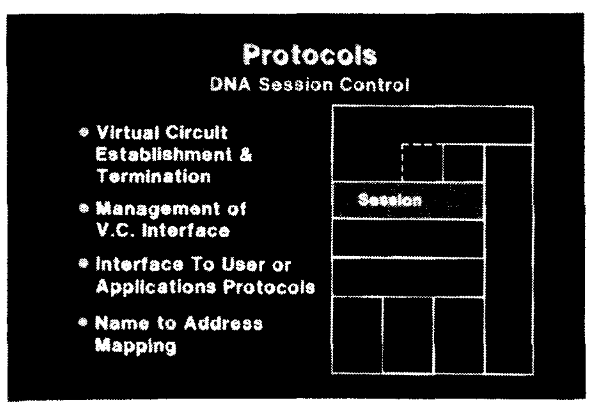
As we said, in DNA the Ethernet implements the physical and data link layers of the network architecture for the local area network. It offers a datagram service with delivery of packets on a "best effort" basis. In that sense it is different than other data link protocols such as DDCMP. The channel is, in general, relatively error free so this protocol is a good match. The Ethernet uses the CSMA/CD protocol to share the 10 Mbps. channel. It uses a distributed algorithm called binary exponential backoff to resolve contention for the channel. This algorithm is executed independently by each station and is fair to all. The specifications allow a maximum of 1024 stations or "taps" on the Ethernet cable. However, as we shall see, there can be more users than taps. This is true of terminal concentrators where several user terminals may share a single tap. Hosts may also have a single tap as well as several users or processes that are generating Ethernet traffic.



Protocols
DNA Transport
• End-To-End Routing
• Datagram Service
• Network Wide Node ID Space
• Congestion Control
• Dynamic Routing
Routing



Protocols
DNA Network Services Protocol
• End-To-End Virtual Circuit
• Recovers Lost Packets
• Various Flow Control Options
• Self-Adjusting Retransmission Timers
End-To-End



Protocols
DNA Session Control
• Virtual Circuit Establishment & Termination
• Management of V.C. Interface
• Interface To User or Applications Protocols
• Name to Address Mapping
Session

SLIDE 19

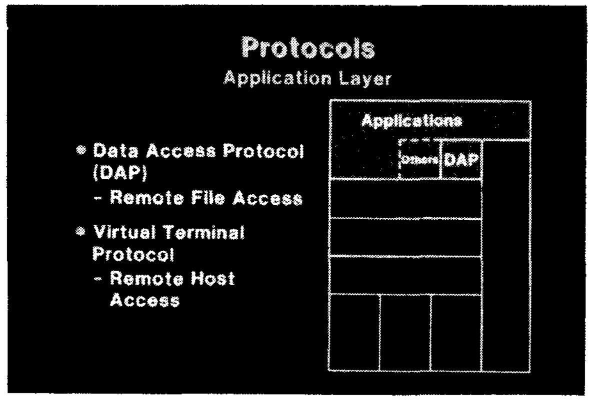
The DNA Transport protocol implements the network wide routing layer of the network. (This includes the local network as well as components that are not connected locally to the Ethernet.) This layer corresponds to essentially the ISO Network layer. It provides end-to-end routing of datagrams and routes packets to a destination even if the node is not on the Ethernet. To do this, it supports a network wide node address space. A node's address can be the same as its Ethernet address if it is on the Ethernet. However, all nodes are not necessarily connected to an Ethernet. Therefore, we need this address space. This layer also prevents congestion within the network and provides dynamic routing to bypass sections of the network that may have failed for one reason or another.

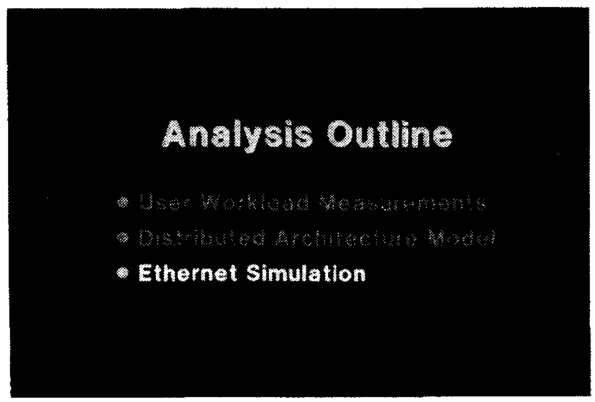
SLIDE 2Ø

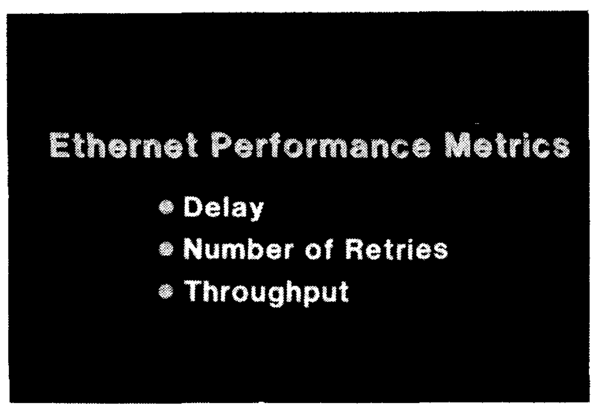
In order to provide effective, error free, and reliable process-to-process communication an end-to-end service is required. This is implemented by DNA's Network Services Protocol (NSP). NSP uses a virtual circuit to provide these features. This assures that packets are delivered to the user in the order they were sent. NSP makes sure that none are lost in the network. This is done by retransmitting lost packets. The timers used to decide when to retransmit a packet are self adjusting. This means that they adjust to the delays in the channel. This has the advantage of limiting the amount of unnecessary retransmissions thus reducing the load on the channel. The protocol also provides various flow control options. This allows the characteristics of the circuit to be tailored to the application. For instance, some applications may require tight control on the rates at which information is exchanged. These data rates impact the amount of resources (buffers, etc.) that must be devoted to the circuit. Flow control is especially important when the receiver is slower than the sender. An example is host to terminal output where the host can usually output data at a rate much faster than the terminal (or user) are capable (or willing) of accepting.

SLIDE 21

The DNA Session Control layer is used to control the virtual circuit service that NSP implements. It allows users to set up and terminate circuits. It validates incoming connect requests and activates the appropriate processes for those that are valid. It manages the interface between the user applications and the circuit. It also provides name to address mapping. For example, if the user requests that a circuit be establised to a node having a particular name, this layer determines the address of that node so that the connect request packet can be sent to the proper destination.


Protocols
Application Layer
Data Access Protocol (DAP)
- Remote File Access
Virtual Terminal Protocol
- Remote Host Access
Applications
Others
DAP


Analysis Outline
User Workload Measurements
Distributed Architecture Model
Ethernet Simulation


Ethernet Performance Metrics
Delay
Number of Retries
Throughput

SLIDE 22

Above the Session Control layer are the applications protocols. The DNA Data Access Protocol (DAP) is one such protocol. It provides remote file access services. This means that the user can use this facility to access files as if they were stored locally on his system. The operation of the network is completely transparent. Another example of an application protocol is a virtual terminal protcol. This allows the user to connect to remote hosts through the network. The user then appears to be connected locally to that remote system.

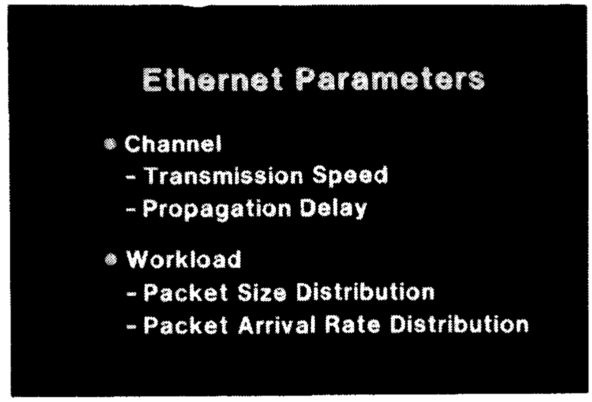
The network management part of the architecture is used to monitor and control the various protocol layers. It can be used by the network manager to monitor the traffic in the network and thus is useful for capacity planning. It is also used to tune the network for better performance.

SLIDE 23

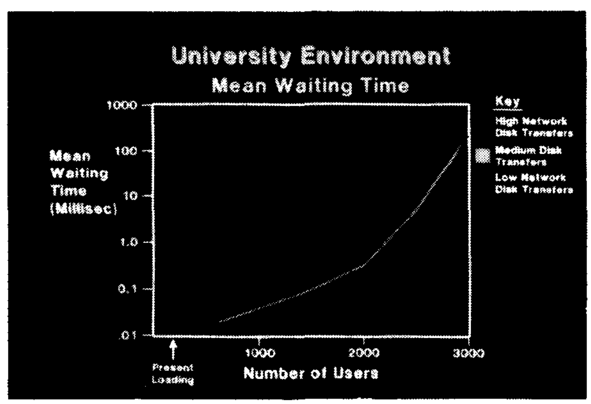
We use the user workload as an input to the distributed architecture model. The output of this is a load on the Ethernet. This consists of the user information being transfered between points on the network as well as various control and data packets associated with the protocol layers in the distributed architecture. The Ethernet simulation simulates the transmission of these packets.

SLIDE 24

To understand the behavior of the channel there are several metrics one can examine. The delay experienced in transfering a packet between stations is of obvious interest. The number of retries necessary to accomplish that transfer is also important. Retries occur whenever there is a collision between two or more packets. The specifications indicate that after 15 retries (ie: 16 attempts) the packet will be aborted. At that point, the higher layer protocols must retransmit that packet. In this case it is NSP that will do the retransmission. The number of retries then gives us an indication of how the channel is behaving.



Ethernet Parameters
Channel
- Transmission Speed
- Propagation Delay
Workload
- Packet Size Distribution
- Packet Arrival Rate Distribution



University Environment
Mean Waiting Time
Mean Waiting Time (Millisec)
1000
100
10
1.0
0.1
.01
Key
High Network Disk Transfers
Medium Disk Transfers
Low Network Disk Transfers
Present Loading
1000
2000
3000
Number of Users

SLIDE 25

The values of the performance metrics such as delay, retries, etc. are determined by variables that come from two general sources. The first are those associated with the Ethernet itself. These are the transmission speed and the propagation delay. Here the transmission speed is 10 Mbps. The `propagation delay depends on the size of the network. There is a maximum size that the network can have and therefore the worst case propagation delay is bounded. A transmitter must continue to transmit a packet long enough so that it can propagate to the farthest parts of the network. This way all stations can detect that a packet is being transmitted. However, another station may have started to transmit a packet before the signal from the first one reached it. In that case there is a collision. The collision must propagate back to the sender while it is still transmitting. This way it will know that its packet has been corrupted. The sender must therefore transmit a packet long enough so that it can propagate to the end of the network and any collision can propagate back. This time is called "the slot time" and it is about the round trip propagation delay for the largest network. (The slot time is 51.2 microseconds in the Ethernet specification.)

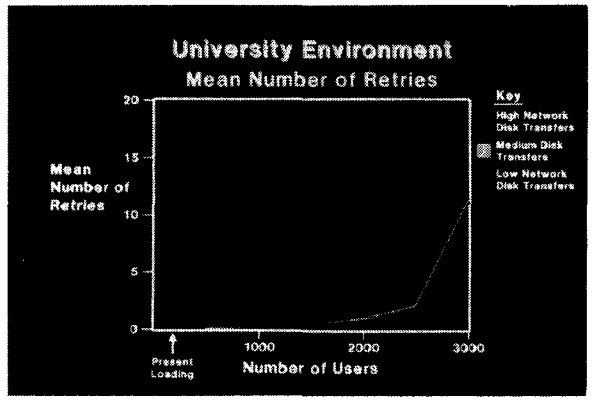
The other factor which determines the performance is the workload. This is the combination of the user workload and the traffic from the distributed architecture. The packet sizes and the rates at which they arrive for transmission over the Ethernet combine to present an given "offered load" to the Ethernet.

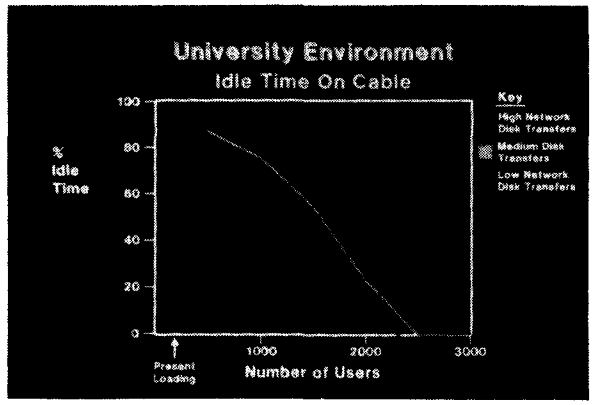
SLIDE 26

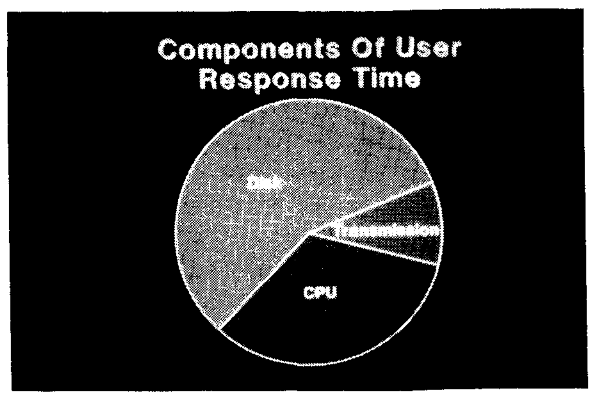
Here we see the mean waiting time on the Ethernet as a function of the number of users. The waiting time is the time from when a packet first becomes ready for transmission until its starts a successful transmission. It includes any time used in deference or collisions. We show three curves based on three levels of remote file traffic. Notice that for up to around 2000 users with this workload, the average waiting time is small when compared to typical delays at disks or in executing application programs or in processing protocol messages.

It is important to remember that the "users" in these curves are active users. This means they are logged in and actively working. Generally, the number of users that are actually using a system at any given time is only a fraction of the total user population. This is true not only for this program development environment but for other environments as well. For example, capacity planning of telephone systems uses knowledge of the relationship between the number of active users and the total user population.

Also note that the system can support more than 1024 users. As mentioned previously, the Ethernet specifications indicate that a maximum of 1024 taps may be connected to the cable. However, we have noted that taps can be shared by several users.


University Environment
Mean Number of Retries
Mean Number of Retries
20
15
10
5
0
Present Loading
1000
2000
3000
Number of Users
Key
High Network Disk Transfers
Medium Disk Transfers
Low Network Disk Transfers


University Environment
Idle Time On Cable
% Idle Time
100
80
60
40
20
0
Present Loading
1000
2000
3000
Number of Users
Key
High Network Disk Transfers
Medium Disk Transfers
Low Network Disk Transfers


Components Of User Response Time
Disk
Transmission
CPU

SLIDE 27

The number of retries a packet experiences is another indicator of the channel performance. A retry occurs whenever a packet has been involved in a collision. Here we see the mean number of retries plotted versus the number of active users for the three levels of remote file traffic. Note that for large numbers of users the average number of retries is still close to zero.

SLIDE 28

At some point when the number of active users is increased to a large enough number, the idle time in the channel will go to zero. This happens when the resources are all used in successfully transmitting packets and in overhead (such as collisions). Here this is plotted for the three levels of remote file traffic. Generally, one chooses an operating point at a point that allows fluctuation in applied load as well additional growth. We see that the Ethernet has ample room for growth at this particular installation based on its operating point. In other studies, such as the measurements of the PARC Ethernets, it has also been observed that the loading on the Ethernet in this and other environments is low.

SLIDE 29

It is important that one keep the Ethernet performance data in the proper perspective. Consider a simple example of a file transfer from a file server over the Ethernet to a host or workstation. The "transmission component" includes the actual transmission time of all the packets in addition to the waiting time for each packet. There will be data and control packets from the various distributed architecture layers. The "CPU component" includes the processing time for each packet as well as any application overhead such as that due to the file system and application protocol. This also includes queueing for the CPU that will occur because there are multiple processes sharing that resource. The slower the CPU, the larger this component will be. The "disk component" includes the disk seek delays in addition to the rotational latency and transfer times for the data. It also includes queueing for the disk that occurs because it is shared. Comparing the CPU and disk components to the transmission component, it is not uncommon to observe that the ratio can easily be 4 to 1 or even 20 to 1 or higher - even when the Ethernet is heavily loaded which makes the waiting time longer.

Other scenarios such as terminal I/O have similar relationships. There the disk component may or may not be as large. This depends on how much disk traffic the user generates. Linking and compiling programs, for example, can generate large amounts of disk traffic. The application program overhead in the CPU component can also be large.

Conclusions

- Ethernet Can Support Up To Several Thousand Active Program Development Users
- Ethernet Delays Are Small Compared To Disk And Other Delays
- Few Collisions Are Experienced, Even Under Heavy Load

SLIDE 30

To summarize, we have seen that the Ethernet is capable of supporting a large number of users of the type characterized in this environment. We have also seen that the delays associated with the Ethernet are typically small when compared against delays other parts of the system. We also note that few collisions are experienced. Therefore, the Ethernet seems well suited for this environment. It has ample capacity and performs well.

Ethernet Is A Unifying Key
To The Fifth Generation

Ethernet Is A Unifying Key
To The Fifth Generation
Because It Is
A Standard
To Interconnect Computers

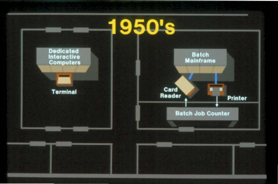
	Unibus	Ethernet
Purpose:	Interconnect A Computer	Interconnect Computers To Form A Network
Distance:	15 Meters	2,500 Meters
Components Connected On The Bus:	Processor Primary Memory Disks Printers Terminal Interface Communication Interface Real Time Interface	Computer Workstations File Servers Print Servers Terminal Servers Gateways Real Time Equipment

A Local Area Network Is A

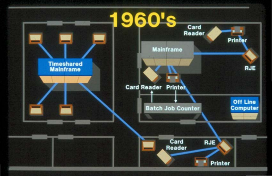
Set of Information Processing Nodes, Distributed In A Single Area And Fully Interconnected Via High Speed Links

Interconnect Evolution

Computing Has Evolved To Require
Complete, High Speed Interconnectivity
Among All Information Processing Nodes
To Form A Network

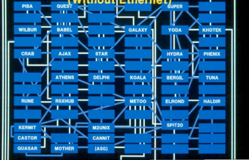


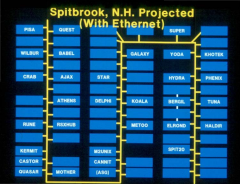
1950's
Dedicated Interactive Computers
Terminal
Batch Mainframe
Card Reader
Printer
Batch Job Counter


1960's
Timeshared Mainframe
Mainframe
Card Reader
Printer
RJE
Card Reader
Printer
Batch Job Counter
Off Line Computer
Card Reader
RJE
Printer

Timesharing on Minicomputer
for Personal Computing

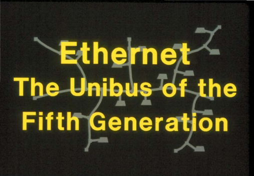
Spitbrook, N.H. Projected (Without Ethernet)



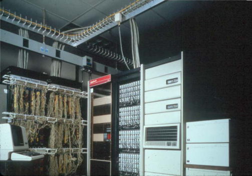

Spitbrook, N.H. Projected
(With Ethernet)
PISA
QUEST
SUPER
WILBUR
BABEL
GALAXY
YODA
KHOTEK
CRAB
AJAX
STAR
HYDRA
PHENIX
ATHENS
DELPHI
KOALA
BERGIL
TUNA
RUNE
RSXHUB
METOO
ELROND
HALDIR
KERMIT
M2UNIX
SPIT20
CASTOR
CANNIT
QUASAR
MOTHER
(ASG)

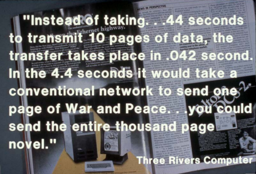
Ethernet Solves The Computer Interconnect Problem:

- Simplify Installation, While Net Operates
- Provide Very High Speed Communication (x 1000 Improvement)
- Reduce The Load On Computing Nodes Without An Active Switch
- Permit Orderly Network Growth With Only 1 Connection Per Node And NO Extra Links


Ethernet
The Unibus of the
Fifth Generation

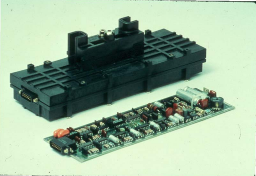

digital
We change the way the world thinks.
See the change Ethernet makes




"Instead of taking. . .44 seconds to transmit 10 pages of data, the transfer takes place in .042 second. In the 4.4 seconds it would take a conventional network to send one page of War and Peace. . .you could send the entire thousand page novel."
Three Rivers Computer

Ethernet Is An Important Part Of Fifth Generation Computing

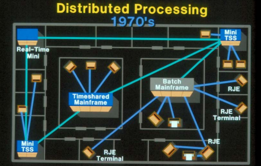
- It Provides For Many Needs
- We Use It And Are Committed To It
- Ethernet Conforms To The International Standards And Digital Network Architectures
- We Will Be Introducing Products Within The Next Few Months
- Moreover, For The Future. . .

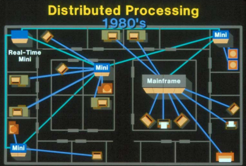


Ethernet In Daily Use At

digital


See the change
Ethernet makes

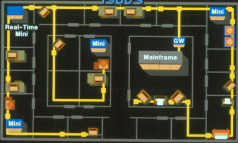

Distributed Processing
1970's
Real-Time Mini
Mini TSS
Timeshared Mainframe
Batch Mainframe
RJE
RJE Terminal
Mini TSS
RJE Terminal
RJE


Distributed Processing
1980's
Real-Time
Mini
Mini
Mini
Mini
Mainframe

Local Area Networks Address These Needs:

- High Speed Interconnection Among Dispersed Computers (i.e. High Performance Networks)
- Simplified Interconnection Of Terminals And Personal Computers To Host Processors
- Interconnection Of Evolving Computer Controlled Equipment With Other Information Processors

Distributed Processing
1980's



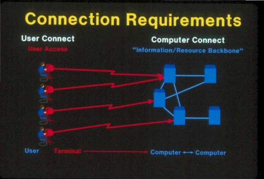
Needs

Open Ended And Complete Interconnection Of:

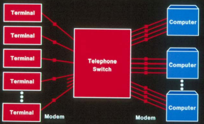
- Terminals And
- Personal Computers

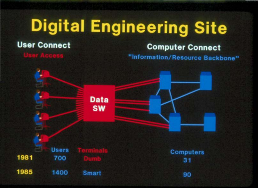
To:

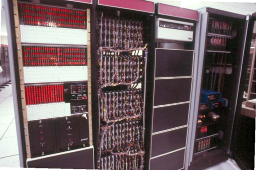
- Host Processing Facilities And
- Special Processing Facilities


Connection Requirements
User Connect
User Access
Computer Connect
"Information/Resource Backbone"
User
Terminal
Computer
Computer

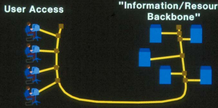
Terminal To Computer Connection




Digital Engineering Site
User Connect
User Access
Computer Connect
"Information/Resource Backbone"
Data SW
Users
Terminals
Computers
1981
700
Dumb
31
1985
1400
Smart
90



Future Connection With Ethernet



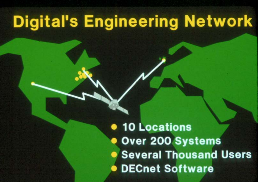
Ethernet Solves The User To Host Or Computer Connection Problem:

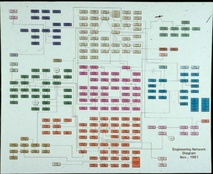
- Simplify Installation, While Net Operates
- Provide High Speed Communication For Interactive Computing
- Permit Open-Ended Network Growth By Direct Cable Access, Without Equipment
- Permit Tradeoff In The Number And Kind Of Connections

Need

Open Ended, High Speed,

Complete Interconnection Of:
All Disbursed Computers

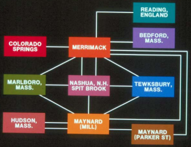

Digital's Engineering Network
10 Locations
Over 200 Systems
Several Thousand Users
DECnet Software

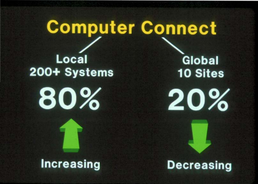

Engineering Network
Diagram
Nov., 1981

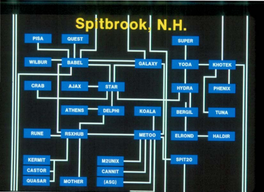
Networking Is Much More Than Just Lines And Nodes
High Level Protocols Are Needed For:

- Interconnection Of Dissimilar Computers
- Network Functions (e.g. File Transfers, Terminal - Terminal)
- Network Management

Current Lines And Locations




Computer Connect
Local
200+ Systems
80%
Increasing
Global
10 Sites
20%
Decreasing



Spitbrook, N.H.
PISA
QUEST
SUPER
WILBUR
BABEL
GALAXY
YODA
KHOTEK
CRAB
AJAX
STAR
HYDRA
PHENIX
ATHENS
DELPHI
KOALA
BERGIL
TUNA
RUNE
RSXHUB
METOO
ELROND
HALDIR
KERMIT
M2UNIX
SPIT20
CASTOR
CANNIT
QUASAR
MOTHER
(ASG)

THE CENTRAL FACILITY

- Large, Shared Data Base
- Archiving for Personal or Organizational Computing
- Program Facility for a Few, Distributed Users
- Quality Printing (Typesetting) and Special Facilities
- Very High Performance Processing
- General Facility for Casual Users

GROUP LEVEL FACILITES

- Shared, Project Data Base
- Specialized Facilities (eg. Microprocessor Debug)
- Programs Run in Common for Group
- Intra-Group Communications
- Communications wtih Central and Personal Computers
- High Performance Processing
- Personal Computing for Many of the Group

PERSONAL COMPUTERS PROVIDE:

- PERSONAL DATA BASES AND SECURITY
- FAST RESPONSE TO RELATIVELY COMPLEX REQUESTS (E.G. EDITING)
- PROGRAM ENVIRONMENT FOR ENTERING AND GETTING INTO PRODUCTION

A Timeshared Computer Consists Of:

- Processor And Primary Memory
- File Memory (Programs And Data)
- Communication Links For Intercommunication And To Terminals For Human Input And Output

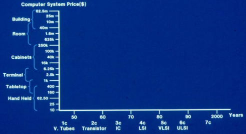
Used Interactively By Several Persons In A Shared Fashion And Belongs To A Group.

A Personal Computer Consists Of:

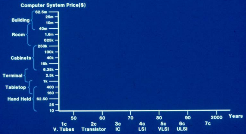
- Processor And Primary Memory
- File Memory (Programs And Data)
- Communication Links (Optional) For Intercommunication And Facilities Sharing
- Transducers For Human Input And Output

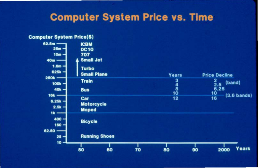
Used Interactively By One Person At A Time And May Either Belong To One Person Or A Group

Computer System Price vs. Time



Computer System Price vs. Time




Computer System Price vs. Time
Computer System Price($)
62.5m
25m
10m
40m
1.6m
625k
250k
100k
40k
16k
6.25k
2.5k
1k
400
160
62.50
25
10
ICBM
DC10
707
Small Jet
Turbo
Small Plane
Train
Bus
Car
Motorcycle
Moped
Bicycle
Running Shoes
Years
3
4
8
10
12
Price Decline
2
2.5
6.25
10
16
(band)
(3.6 bands)
50
60
70
80
90
2000
Years

The Transitions

- Technology Transition
- Transition to Distributed Computing Based on Local Area Networks (NI)
- Transition to Personal Computers from Minis & Mainframes
- Transition from Conventional Rack & Stack 16-Bit Computers
- Transition to Software for End Use Versus Programmer Tools
- Transition in Hardware Design Skills

ETHERNET PRESS SEMINAR
WORLD TRADE CENTER—NEW YORK CITY
FEBRUARY 10, 1982

Schedule of Events

8:15– 9:00	Coffee & Danish
9:00– 9:40	Introduction: **The Impact of Very Large Scale Integration (VLSI) Technology on Communications** *Dr. Robert Noyce*, Vice Chairman, Intel Corporation
):40– 10:20	**The Evolution of Distributed Processing and Local Area Networks** C. *Gordon Bell*, Vice President, Engineering, Digital Equipment Corporation
10:20– 10:30	Intermission
10:30– 11:15	**Productivity in the Office Environment** *Mr. David E. Liddle*, Vice President and General Manager, Office Systems, Office Products Division, Xerox Corporation
11:15– 12:00	Panel Question and Answer Session
12:00– 1:15	Luncheon

ETHERNET EDITORIAL SEMINAR
WORLD TRADE CENTER—NEW YORK CITY—FEBRUARY 10, 1982

Schedule of Events

1:30–1:50 Introduction: **The Impact of Very Large Scale Integration (VLSI) Technology on Communications**
Dr. Robert Noyce, Vice Chairman, Intel Corporation

1:50–2:10 **The Evolution of Distributed Processing and Local Area Networks**
C. Gordon Bell, Vice President, Engineering, Digital Equipment Corporation

2:10–2:30 **Productivity in the Office Environment**
Dr. David E. Liddle, Vice President and General Manager, Office Systems, Office Products Division, Xerox Corporation

2:30–2:45 Intermission

2:45–3:05 **The Changing Economics for Computer and Terminal Interconnection**
Phillip Arst, Intel Corporation

3:05–3:25 **Ethernet Performance Based on a Simulated User Environment**
William Hawe, Principal Engineer, Systems Performance Analysis, Digital Equipment Corporation

3:25–3:45 **Dispelling Ethernet Myths**
Bob Prentis, Manager of Network Standards

3:45–4:15 **Panel Question and Answer Session**
Moderator: *Ronald T. Yara*, Strategic Marketing Manager, Intel Corporation

4:15–5:00 Hospitality Reception

ROBERT N. NOYCE
Intel Corporation
Santa Clara, California

Robert N. Noyce is Vice Chairman of the board of directors of Intel Corporation, Santa Clara, California. A co-founder of Intel Corporation in 1968, Dr. Noyce was President until 1975 and chairman of the board from 1975 to 1979.

Dr. Noyce is co-inventor of the integrated circuit with Jack Kilby. They have jointly received the Ballantine medal of the Franklin Institute, and the Cledo Brunetti Award of the IEEE for this work. With Gordon Moore he has received the AFIPS Harry Goode award for leadership in computer science. Dr. Noyce was awarded the National Medal of Science and the I.E.E. Faraday Medal in 1979, and the IEEE Medal of Honor in 1978. He is a member of the National Academy of Science, the National Academy of Engineering, the American Academy of Arts and Sciences, and is a Fellow of the IEEE.

Dr. Noyce was born in Iowa in 1927. He received a B.A. degree and membership in Phi Beta Kappa at Grinnell College (Iowa) in 1949, and a Ph.D. in physical electronics at the Massachusetts Institute of Technology in 1953. He did research at Philco Corporation until 1956 when he joined Shockley Semiconductor Laboratory, Palo Alto, California, shortly after its founding, to work on transistor technology. (The lab was founded by William Shockley, co-inventor of the transistor at Bell Telephone Laboratories.)

In 1957, Dr. Noyce co-founded Fairchild Semiconductor Corporation, Mountain View, California. He was research director until early 1959, when he became vice president and general manager. By 1968, the sales for Fairchild Semiconductor had risen to over $100 million.

As research director of Fairchild Semiconductor, Dr. Noyce was responsible for initial development of the firm's silicon mesa and planar transistor product lines. Also, his inventions in the integrated circuit field enabled Fairchild to produce the first commercial integrated circuit.

In July, 1968, Dr. Noyce co-founded Intel Corporation with Gordon E. Moore, who had also been a co-founder of Fairchild Semiconductor and a member of the Shockley laboratory staff. (Dr. Moore succeeded Dr. Noyce as President and then Chairman of Intel.)

Their goal was to make LSI technology a practical reality. At the time, LSI was still in its early stages of development and used primarily to produce custom circuits. Intel developed the Schottky barrier bipolar and the silicon gate metal-oxide-semiconductor technologies which allowed several thousand transistors to be integrated on a single chip of silicon with a relatively high production yield. Intel used the silicon gate MOS technology to produce the first high density memory components and the first microprocessor. It now produces most of its LSI products with advanced versions of this technology.

Dr. Noyce holds 16 patents for semiconductor devices, methods and structures.

Intel has grown to approximately 16,000 employees. In 1979 revenues totaled $663 million and net income $77.8 million. Intel manufactures and markets large scale integration (LSI) and VLSI semiconductor devices, such as microprocessors and memory components, and systems built with LSI devices.

Impact of VLSI on Communications

We are on the eve of major developments in worldwide data communications on all fronts—within the factory and office, between buildings and cities, and between countries. Indeed, we now hear of the "Second Industrial Revolution," the "Paperless Society," the "Information Age," and the "Knowledge Revolution" from popular writers or news reports, from conferences of industry and labor leaders, and studies of governments and learned societies. All of these global forecasts point to an increased need to communicate and to expand those communications facilities to reach a much broader community of users.

The semiconductor industry is, in large part, responsible for the enormous demand for increased data communications. In its first decade, the microprocessor has been designed into more than 100,000 products. The development of standard VLSI building blocks has allowed manufacturers to introduce microcomputer-based products at an unprecedented rate. The fact that these new systems are becoming increasingly interdependent will result in data communications networks (to interconnect those systems) becoming as pervasive as the microprocessor is today. It is important, therefore, that the same orientation toward global optimization, which resulted in the development of standard microcomputer building blocks, be continued by the semiconductor industry, equipment manufacturers, and end users in the defining and implementing of advanced data communications capabilities.

The impact that VLSI will have on communications must be viewed in the context of the impact the semiconductor industry has had on computing. Through standard building blocks, manufacturers were able to drive costs down, while increasing capabilities, to change the economics of computing from "one for many" to "one for one." Whereas the large mainframe and expensive system resources imposed a "one for many" environment, and the lower-cost minicomputer a "one for few" relationship, VLSI (microprocessors memories and software) is now makng possible the era of the personal work station, a "one for one" relationship. The impact of VLSI on communication, then, can be seen as an opportunity to provide cost-effective interconnection of those personal work stations and the centralized capabilities supported by minicomputers and mainframes, providing uniform access to information, resources, and services.

Global optimization in communications will lead to interconnection that achieves:

A. Location-independent access to
   —information
   —resources
   —services
B. Media-independent access
   —telephone wire
   —coaxial cable (TV cable)
   —fiber optic cable
   —others
C. Interoperability—different equipment from different manufacturers communicating with each other.

The benefits of such a solution are:

A. Timely access to, and distribution of, information, independent of where the user is or the transmission medium used.
B. Cost-effective sharing of both distributed and centralized resources and services.
C. Optimized end user solutions that can include equipment from multiple manufacturers.

How will such a solution come about? As with other established markets, it will be an evolving process. Because of the growing demand for data communications, however, the development of capabilities and architectures by various manufacturers will be rapid, running well ahead of the actual installation of such equipment. It is important, therefore, to start now in the definition of standardized interfaces that will lead to both location and media-independent access and interoperability. This will also allow standard VLSI communications building blocks to be developed, resulting in an impact on communications analogous to the impact that microprocessors had on computing—that of driving the economics of communications from "few to one" to "any to any."

The three companies; Digital Equipment Corporation, Intel Corporation, and Xerox Corporation, have made that first step in the area of standardized, high-speed communications within the building: Ethernet. Starting with the basic Ethernet technology that had been under development and testing at Xerox since 1975, the three companies entered into a cooperative agreement. That agreement involved the development of a high-speed, Local Area Network, and publishing the specification to encourage general, widespread implementation. The goal of the three companies was to achieve interoperability within a building through a standard, high-speed, Local Area Network.

Intel's contribution to the cooperative effort has been focused in two areas. We currently have an extensive family of LSI communications peripherals that support existing protocols, including HDLC/SDLC and Bisynch. Using that base of knowledge, we are utilizing our VLSI expertise to develop a high-speed Local Area Network Controller which will support the Ethernet specification, which we expect to sample before the end of the year. We have also implemented Ethernet in several systems products, the first of which is a distributed microcomputer development system that begins customer shipment this quarter.

The efforts of the three companies, I believe, reflect the industry "sense" of the problem to be solved and the need for cooperation. In many respects, we are sitting in the same position the railroad industry was in when they saw the opportunity to provide freight and passenger service throughout the country—the time was at hand to agree on the width of the railroad tracks. Ethernet provides that "standard width" for *integrated* solutions within a building. Similar efforts are required to allow the strengths of VLSI solutions to be properly focused to provide "cost effective one for one" computing *and* "any to any" communications, thereby achieving the "Wired (World) Community."

THE IMPACT OF VLSI ON COMMUNICATIONS

WE ARE ON THE EVE OF MAJOR DEVELOPMENTS IN WORLDWIDE DATA COMMUNICATIONS

—Within the Factory and Office

—Between Buildings and Cities

—Between Countries

—"Paperless Society"

—"Second Industrial Revolution"

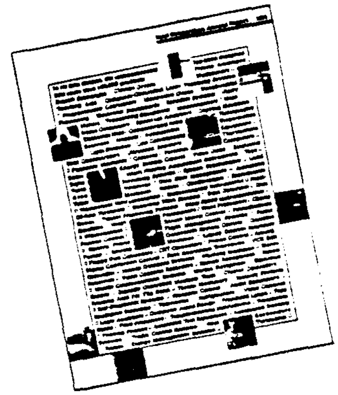
—"Information Age"

—"The Knowledge Revolution"

UNPRECEDENTED DEMAND FOR COMMUNICATIONS

intel®

IN ITS FIRST DECADE, THE MICROPROCESSOR HAS BEEN DESIGNED INTO MORE THAN 100,000 PRODUCTS



intel®

DATA COMMUNICATIONS WILL BECOME AS PERVASIVE AS MICROPROCESSORS ARE TODAY

- Orientation Toward Global Optimization
  - —Semiconductor Industry
  - —Manufacturers
  - —End Users

intel®

THE SUCCESS OF THE SEMICONDUCTOR INDUSTRY

—A Healthy Mix of Competition and Cooperation

intel®

"ALL COMPETITORS WHO BOTH COEXIST AND ARE PROFITABLE OVER TIME ARE SIGNIFICANTLY DIFFERENT."

—BRUCE HENDERSON

- THE SEMICONDUCTOR INDUSTRY PLAYERS ARE DIFFERENT IN:

—Objectives

—Strategic Approach

—Management Style

—Products and/or Markets

intel®

AGAINST THIS BACKDROP OF INTENSE COMPETITION, A HISTORY OF COOPERATION

—Conscious and Deliberate

AN ORIENTATION TOWARD GLOBAL OPTIMIZATION

intel®

CONSCIOUS AND DELIBERATE COOPERATION

- Cross Licensing and Second Sourcing
- Joint Ventures
- Standards for Packages, Functions, etc.

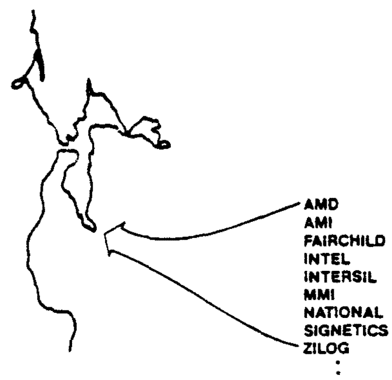
AGAINST THIS BACKDROP OF INTENSE COMPETITION, A HISTORY OF COOPERATION

—Conscious and Deliberate

—Unconscious

AN ORIENTATION TOWARD GLOBAL OPTIMIZATION

SILICON VALLEY GEOGRAPHIC CLUSTERING



AGAINST THIS BACKDROP OF INTENSE COMPETITION, A HISTORY OF COOPERATION

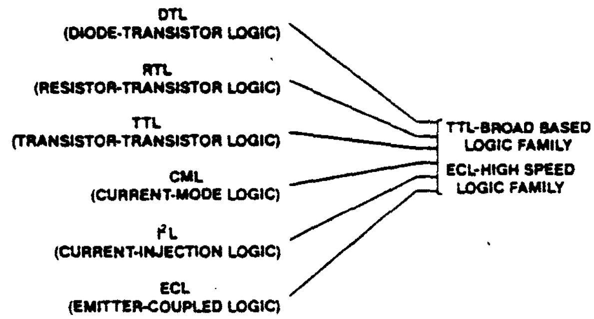
—Conscious and Deliberate

—Unconscious

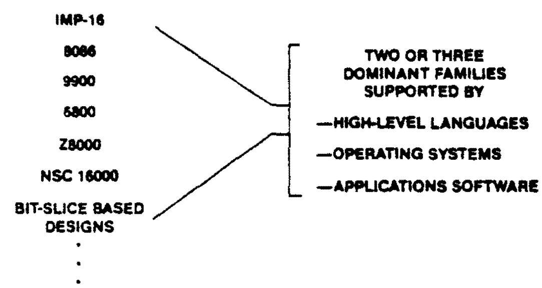
—Market Consolidation—Defacto Standards

AN ORIENTATION TOWARD GLOBAL OPTIMIZATION

MARKET/TECHNOLOGY CONSOLIDATION —BIPOLAR TECHNOLOGIES



MARKET/TECHNOLOGY CONSOLIDATION —16-BIT MICROPROCESSORS



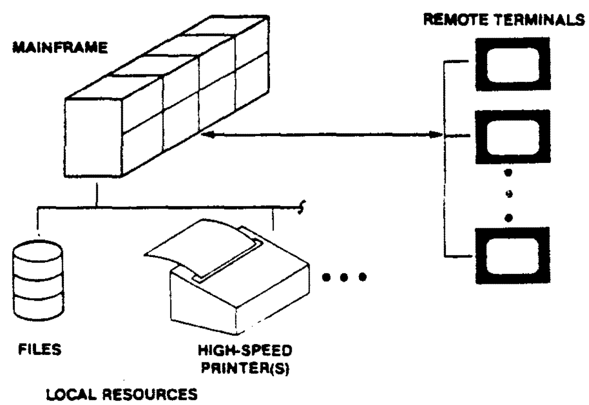
BENEFITS OF GLOBAL OPTIMIZATION

- Semiconductor Industry
  - —Better Use of Available Resources
  - —More Competitive, on a Broader Front
- Our Customers
  - —Standard Building Blocks with Wider Application
  - —Develop Better Solutions, Sooner and at Lower Cost

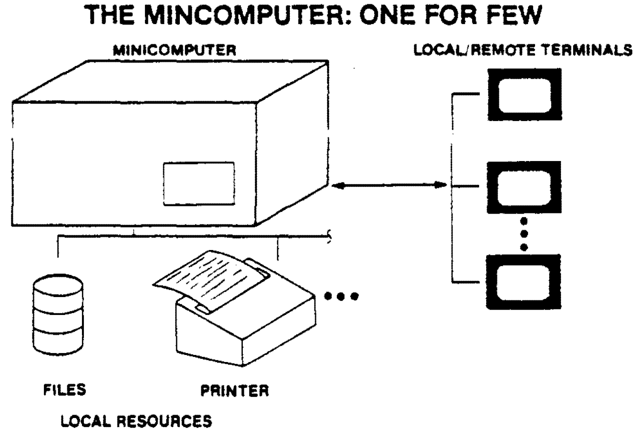
IMPACT OF THE SEMICONDUCTOR INDUSTRY ON COMPUTING:

Changing the Economics of Computing from "One for Many" to "One for One"

THE MAINFRAME: ONE FOR MANY



THE MINCOMPUTER: ONE FOR FEW



intel®

THE PERSONAL WORKSTATION: ONE FOR ONE

- Local Resources and Local Computer
  - —Files
  - —Keyboard
  - —Printer
  - —Display

intel®

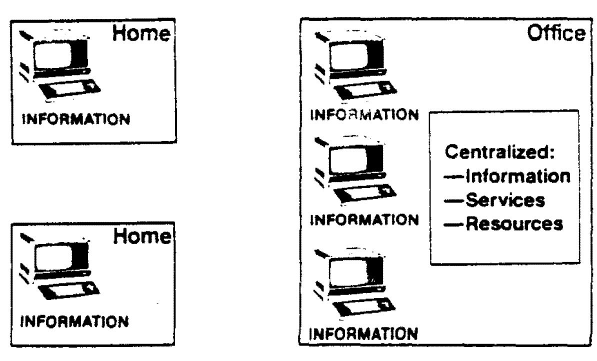
THE OPPORTUNITY FOR COMMUNICATIONS IS TO PROVIDE COST-EFFECTIVE INTERCONNECTION

"ONE FOR ONE" (Distributed) AND "ONE FOR MANY" (Centralized)

- Information
- Resources
- Services

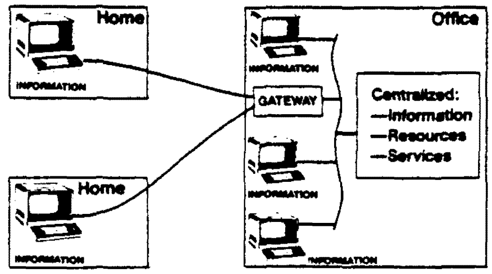
intel®

A PREREQUISITE TO MANAGING INFORMATION IS THE ABILITY TO ACCESS IT



LOCATION AND MEDIA INDEPENDENT ACCESS TO:

- Information
- Resources
- Services



INTEROPERABILITY—ANY TO ANY COMMUNICATIONS USING EQUIPMENT FROM DIFFERENT MANUFACTURERS

- Providing Access to
  - —Information
  - —Resources
  - —Services

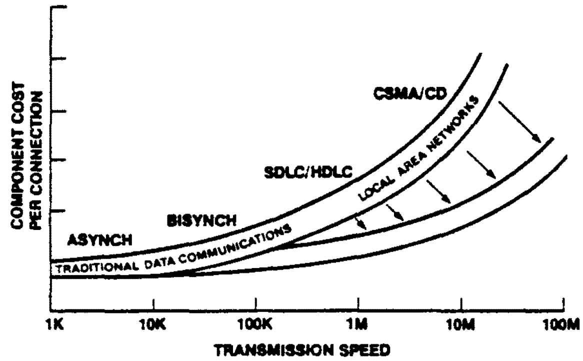
THE KEY—STANDARDIZED INTERFACES

—Orientation Toward Global Optimization

—Develop Standards that Lead to Location and Media Independence, e.g., Transmission Standards for Color T.V.

—Standard VLSI Communications Controllers: Driving the Economics of Communications from "Few to One" to "Any to Any"

VLSI IMPACT ON COMMUNICATIONS



ETHERNET: A FIRST STEP TOWARD STANDARDIZED HIGH-SPEED COMMUNICATIONS WITHIN A BUILDING

DIGITAL/INTEL/XEROX COOPERATION

- Develop a High-Speed Local Area Network
  —Thorough Correctness Proof
- Publish the Specification for General Widespread Implementation

GOAL: Interoperability Within a Local Area Network

INTEL CONTRIBUTION

—Current LSI Communications Peripherals Support Standard Protocols—HDLC/SDLC, BISYNCH, etc.

—Developing a VLSI LAN Controller Which Supports the ETHERNET Specification

—Develop System Level Products Utilizing ETHERNET

SUMMARY

—The Impact of VLSI on Computing has been Significant

—There will be a Similar Impact of VLSI on Communications

—An Orientation Toward Global Optimization will be Central to Achieving the "Wired Community"

Philip L. Arst
Intel Corporation
Santa Clara, California

Philip L. Arst is manager of the Data Communications Product Line at Intel Corporation, Santa Clara, California. He is responsible for setting corporate direction for the company's data communications activities, as well as, engineering and marketing responsibilities for a line of data communications systems products. Mr. Arst is also responsible for managing Intel's Ethernet program and has led this program since its inception. This responsibility includes work on the Ethernet specification, interfacing to standards bodies and bringing a family of products to market.

Mr. Arst has an extensive background in the data communications field, including work on a series of projects which resulted in one of the earliest commercial implementations of a local area network (LAN).

As the data communications product manager in the Data Systems Division of Xerox Corporation, El Segundo, California, Mr. Arst developed a front-end communications processor for their mainframe computer product line. And, as the data communications product planner at the company's Integrated Office Systems Division, El Segundo, California, he was responsible for the formulation of strategies and products incorporating Ethernet and global data communications capabilities for integrated office systems.

While associated with the Collins Radio Company, Newport Beach, California, Mr. Arst was active in the system and software designs of early message switching systems. And, this is where he was involved in one of the earliest commercial implementations of an LAN, the Collins C-System TDM Loop.

Mr. Arst has received a BSEE degree from the U.S. Naval Academy and an MBA from the University of Chicago.

The Changing Economics for Computer and Terminal Interconnection

The combination of VLSI components and the inherent systems cost reductions provided by the Ethernet architecture will materially lower the costs of data interconnection and switching within localized geographic areas such as buildings, factories or laboratories. This approach offers such significant advantages over today's telecommunications based techniques, that we foresee Local Area Networks being installed in all business establishments and the LAN interface component becoming a standard part within personal computers and workstations destined for use in the business establishment.

The VLSI design process is typically a three year program. Current estimates are that VLSI devices for Ethernet, the simplest of today's crop of LAN protocols, will be approximately 50% more complex than Intel's 8086 16-bit microprocessor. Bringing this device to market therefore represents a formidable design and product challenge for the semiconductor manufacturer. The stability and simplicity of the Ethernet protocol makes this practical.

Current estimates are that this controller component will implement the full Ethernet protocol (i.e., the entire Bluebook) with the exception of the physical link (transceivers and cabling). In this manner, not only will 80–100 ic's and a full circuit board of today's implementation be replaced, but also the user will be freed from any programming at the Ethernet data-link level as the component manufacturer will have done it all for him/her. We therefore foresee the electronics cost of the Ethernet data-link dropping to a $30–$40 level by 1985 if LAN demand provides for the production volumes we believe they will.

However, the electronics interface is only the tip of the cost iceberg. While it is incorrect to compare today's telecommunications based solutions to local networks (because the local network provides needed high bandwidth data services for computer to computer communications which are beyond the capability of the digital PABX) it is still useful to examine the relative cost components of each.

In today's telecommunications based systems, the Electronics Interface is cheap, but modems, dedicated ports on PABX equipment (at $500–$1000 per port) and front end or message switching computers are also required to transmit, route and distribute data between distributed and centralized data processors and user workstations. Rewiring and reconfiguration are also an important portion of today's cost equation as they are often required to accommodate change and growth.

The Ethernet bus architecture eliminates the cost of the switching function provided by the centralized PABX unit or the front end/message switching processor by building a distributed switching capability into the controller electronics of each workstation. This is accomplished by interconnecting all processors and workstations on a single shared channel. In this manner, each receives the traffic of all other stations on the net and selects only traffic which bears its address.

An additional unique capability of the Ethernet Architecture is its transceiver design which permits easy reconfiguration. This permits it to avoid expensive rewiring and switching equipment reconfiguration when needs change or equipment is relocated. However, the transceiver design of Ethernet has its drawbacks as these devices are currently expensive (approximately $300 in small lots). Fortunately, the transceiver is also susceptible to considerable cost reduction. The first step will be the integration of its electronics into a single or a few chips. But since the major cost of the transceiver is in its mechanical parts (i.e., housing, connectors, separate circuit board and power supply), a systems approach can be taken to lower this class of costs.

These systems approaches are typically based upon sharing a single transceiver between many stations. Products of this category consist of:

- Transceiver multiplexers which permit the sharing of a single transceiver by 4, 8 or more stations (or its elimination entirely in small systems).
- RS-232c interfaces for multiple "dumb" terminals which share a single transceiver and set of electronics (such as the Ungerman-Bass Network Interface Unit).
- Packaging the transceiver electronics (i.e., chip) within the workstations and bringing a flexible version of the Ethernet cable to a tap on the cabinet. By clustering these "cables to the cabinet" terminals and then interfacing them to the main Ethernet cable via a simple repeater, significant cost reduction can again be achieved. Through utilization of these techniques, we foresee an Ethernet interface consisting of a VLSI controller and a separate transceiver selling in volume OEM quantities in the $120–$150 per node range in 1985 and in the cable to the cabinet configuration of $30–$50 per node.

A further cost reduction of the Ethernet VLSI component will be obtained by applying its basic CSMA/CD (Carrier Sense Multiple Access with Collision Detection) technology to other applications. For example, we foresee CSMA/CD LANs being built within cabinets of electronics, such as a personal workstation. The CSMA/CD LAN would interconnect the station processor, its floppy disk, printer and other devices. Intel products will support these non-Ethernet applications, thereby further building product volumes and lowering Ethernet costs.

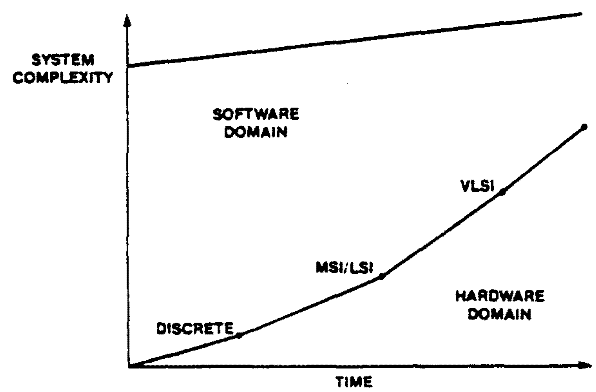
These cost levels, plus the higher functionality provided by the Ethernet architecture, will, in our opinion, make the Ethernet controller the computer terminal interface of the 1980s.

"THE CHANGING ECONOMICS FOR COMPUTER AND TERMINAL INTERCONNECTION"

Philip L. Arst
Intel Corporation

THE DATA COMMUNICATIONS CHALLENGE

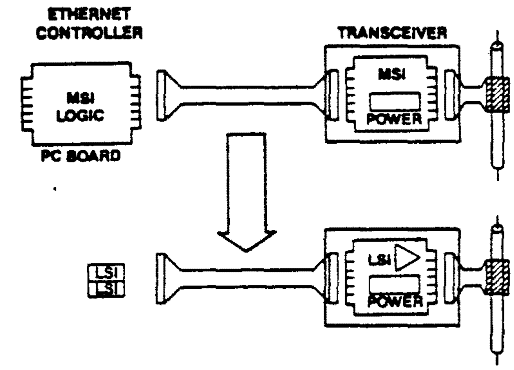
A Systems Solution



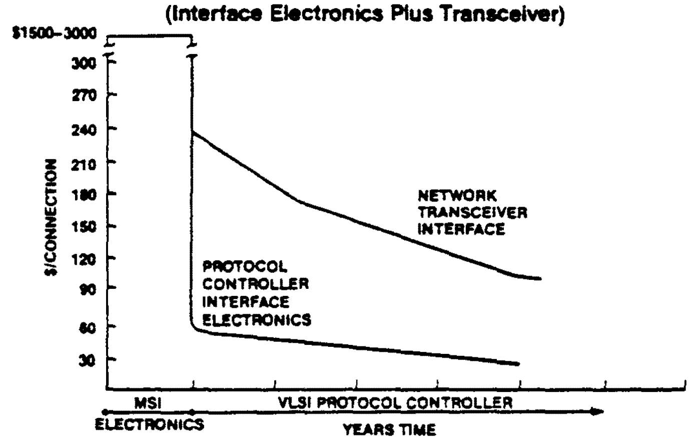
DESIGN TRENDS

Date	Device	Part Type	Device Complexity	Design Cycle-time (Years)
1976	UART	8251	2,800	1.5
1979	USART	8274	24,000	2
1979	16-Bit Microcomputer	8086	29,000	3
1981	Ethernet Local Network	—	43,000	3

VLSI COST REDUCTION



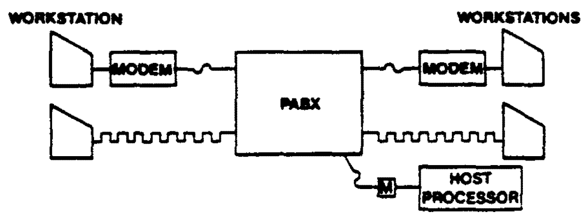
ETHERNET INTERFACE COST TRENDS
(Interface Electronics Plus Transceiver)



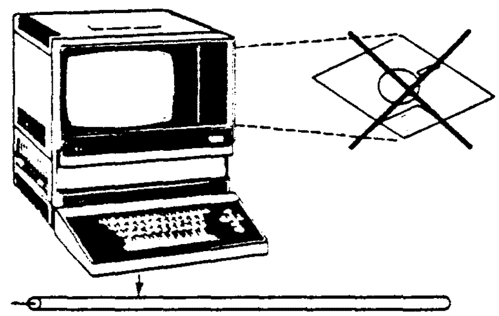
TELECOMMUNICATIONS-BASED WIRING AND SWITCHING (Con't)

- Today's Approach
  - —Full Voice Service (Blocking)
  - —$500 to $1000 Per Port
  - —High Initial Cost
  - —Modems Required for Data
- Voice Data Approach
  - —Full Voice/Medium Speed Data (Non-blocking)
  - —Generally Requires Rewiring
  - —High Initial Cost
  - —Cost Per Port?

TELECOMMUNICATIONS-BASED WIRING AND SWITCHING



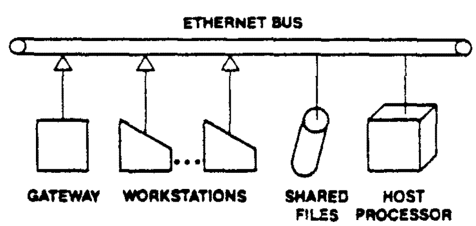
TOMORROW'S PERSONAL WORKSTATION



All Electronic Workstation
With a High-speed
Ethernet Interface

- Shared Resources
  - —Files
  - —Data Output
  - —Data Input

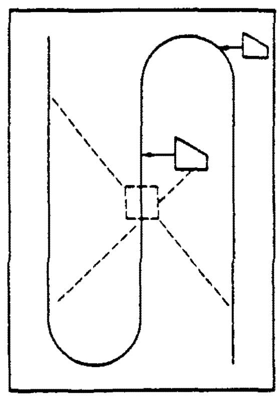
LOCAL AREA NETWORK-BASED WIRING AND SWITCHING



ETHERNET APPROACH

—Switching on Bus
—Modem Function in Transceiver
—High Bandwidth
—Short Wiring Runs
—Relocatable Bus Connectors
—Coexistence With Existing Voice Systems

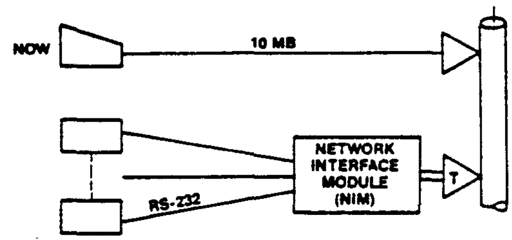
ETHERNET CONFIGURABILITY



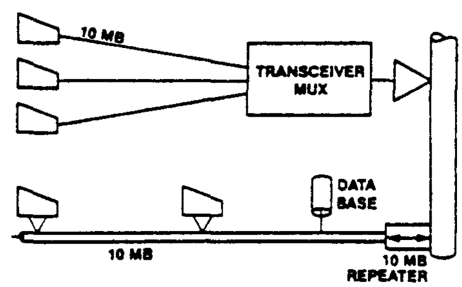
ETHERNET INTERFACE COST ASSUMPTIONS

—Estimated OEM Average Selling Price
—Interface Built Into Terminal
—Shares Microprocessor and Memory

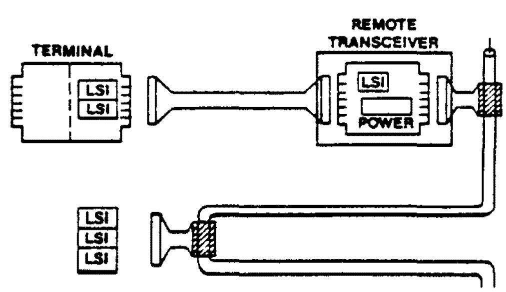
SYSTEMS CONFIGURATION ALTERNATIVES



SYSTEMS CONFIGURATION ALTERNATIVES (Con't)

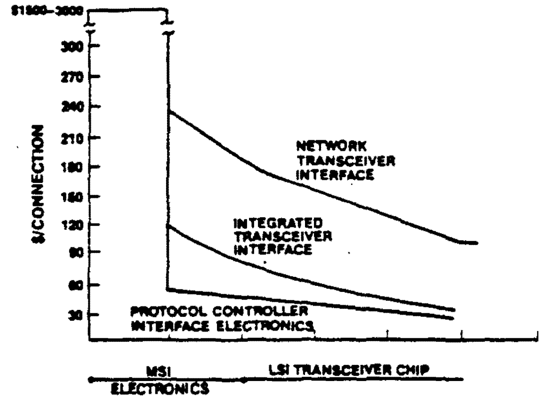


CONFIGURATION ALTERNATIVES YIELDS COST REDUCTION



ETHERNET INTERFACE COST TRENDS

(Interface Electronics Plus Transceiver)



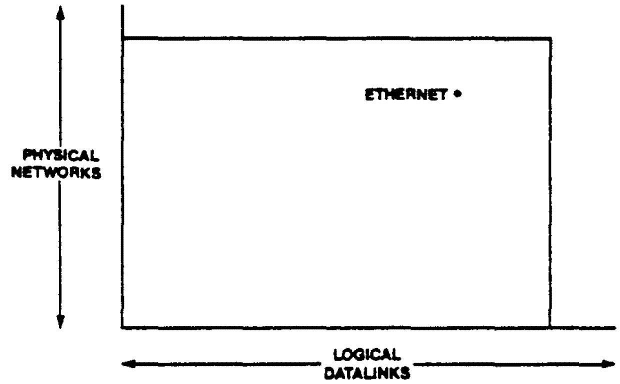
ETHERNET—THE COSTS TUMBLE

- ☐ Simple Field-proven Protocol—Suitable for VLSI
- ☐ High Bandwidth and Configurability Enable a New Generation of Applications
- ☐ Ethernet Cost Projections
  - —Improves Upon Today's Telecommunications-based Costs
  - —VLSI to Drop Costs Further
  - —Systems Configuration Tradeoff's Used to Go to Rock Bottom

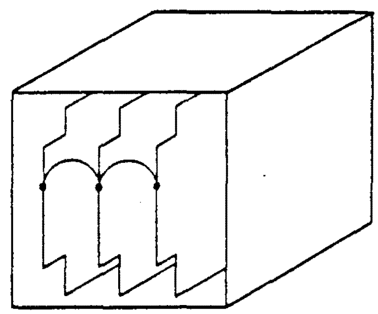
CHANGED

"THE ~~CHANGING~~ ECONOMICS FOR COMPUTER AND TERMINAL INTERCONNECTION"

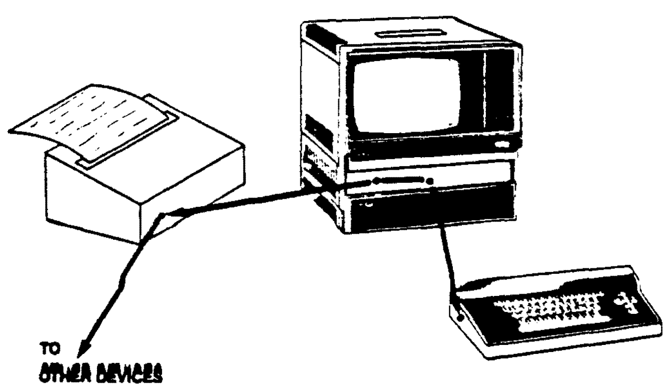
CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION



SERIAL BACKPLANE WITH CSMA/CD



INTRACABINET INTERCONNECT WITH CSMA/CD



CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION

- Just Like Verbal Conversations
  - —Shared Channel
  - —Listen Before Transmit
  - —Transmit Only When No Carrier Present
  - —If Collision, Stop and Wait a Random Time
  - —Retransmit If No Carrier Present

INTEL'S VLSI EFFORT IN LANs

INTEREQUIPMENT COMMUNICATIONS

- <$50 per ETHERNET Connection
  - —Interoperability
  - —High Performance

INTRACABINET COMMUNICATIONS

- <$25 per ETHERNET Subset
  - —Same VLSI
  - —Performance/Cost Tradeoffs

DIGITAL AND ETHERNET

Gordon Bell
William R. Hawe

BIOGRAPHICAL INFORMATION

C. GORDON BELL

Vice President, Engineering

C. GORDON BELL, 47, is Vice President, Engineering for Digital Equipment Corporation.

In this position, Bell has responsibility for the company's research, design and development activities in computer hardware, software, and systems and is a member of the Operations Committee, Digital's 13-member senior management team.

Bell joined Digital in 1960 as Manager of Computer Design, a position he held for six years. He took a leave of absence from Digital in 1966 to join the faculty of Carnegie Mellon University in Pittsburgh. He rejoined the company in 1972 as Vice President of Engineering.

Prior to joining Digital, Bell held several engineering positions including that of research engineer at the MIT Speech Communications and Electronic Systems Laboratories.

Bell earned his B.S. and M.S. degrees in Electrical Engineering at Massachusetts Institute of Technology in 1956 and 1957 respectively.

He is a widely published author on computer architecture, and computer design. His books include the McGraw-Hill book, "Computer Structure," co-authored with Allen Newell; and the

Digital Press books, "Designing Computers and Digital Systems, Using PDP-16 Register Transfer Modules," with John Grason and Allen Newell; and "Computer Engineering: A DEC View of Hardware Systems Design," co-authored with J. Craig Mudge and John McNamara.

Holder of several patents in the computer and logical design areas, Bell has also served the U.S. Government as a member of COSINE Committees of the National Academy of Science for computer engineering education, and the National Science Foundation, Office of Computer Activities and Computer Science and Engineering Researching Study (COSERS). Bell has also served on the Council for International Exchange of Scholars.

Among his professional affiliations, Bell includes the National Academy of Engineering, Fellow of the Institute of Electrical and Electronic Engineers, member of the National Research Council's Computer Science and Technology Board, fellow of the American Association for the Advancement of Science, the Association for Computing Machinery, and is listed in "American Men of Science" and "Who's Who."

He is a resident of Lincoln, Massachusetts.

#####

January 1982

WHY DIGITAL IS COMMITTED TO ETHERNET FOR THE FIFTH GENERATION

A summary of remarks made by Gordon Bell, Vice President—Engineering, Digital Equipment Corporation, at the Xerox/Intel/Digital Seminar.

New York, February 10, 1982

WHY DIGITAL IS COMMITTED TO ETHERNET FOR THE FIFTH GENERATION

ABSTRACT

What the Digital Unibus did for minicomputers, Ethernet will do for the Fifth Generation.

Nearly all recent computers are organized around a single, high speed bus (Unibus-type structure) which provides communications among its processors, memory, disks, and interfaces to the external environment. This simple structure has been one factor in the rapid evolution and proliferation of computers. Unfortunately, a bus for interconnecting computer components within a cabinet, is not suitable for interconnecting a network of computers within a building.

Ethernet is a high speed, 10 megabits per second, standard bus providing the first two levels of the ISO Open Systems Architecture. It permits the dynamic connection of computers at a site to form a local-area network (LAN) in an open-ended fashion without the need of centralized equipment or planning and control. In the Fifth Generation, the network becomes the system and Ethernet is a key prerequisite of the generation.

Ethernet will be used initially, in an evolutionary fashion, to interconnect networks of today's computers to each other and to terminals and personal computers. Since Ethernet is a factor of 1000 higher speed standard than today's network links, and easily used to form networks, we expect a rapid transition to a tightly integrated network, where the network is the system. In this generation, separate function computers (eg. personal workstation, file server, print server, real time, timeshared) will be tightly integrated, interchanging many types of messages, such as, files, computed graphics, pictures, and voice. This kind of network will permit a radically different use of computers, and only then can we be certain that this is the Fifth Computer Generation.

Because Ethernet is so important to the Fifth Generation, Digital is committed to it as a standard. We use these networks and will be providing products in the near future.

Ethernet Is The Unibus
Of The Fifth Generation

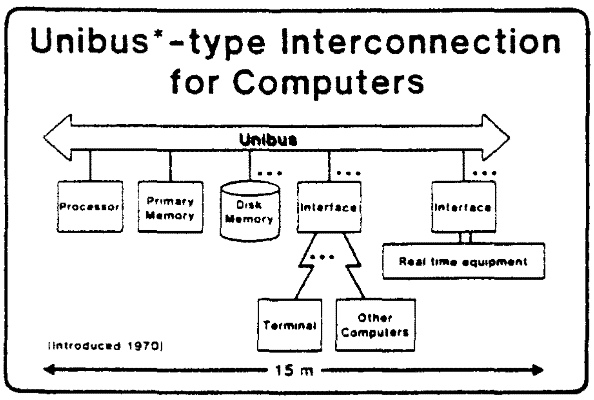
"You have to look at Ethernet as a standard..."

Ethernet is one of the keys to the development of the Fifth Generation because it provides a standard for the interconnection of all sizes and types of computers in a passive, local-area network.

Up until now, interconnection has been a very difficult task simply because there has been no standard.

A standard is a blueprint that shows you how to build the components that will go into a system or onto a network.

"... system components are connected by a single high-speed bus in an open-ended fashion."

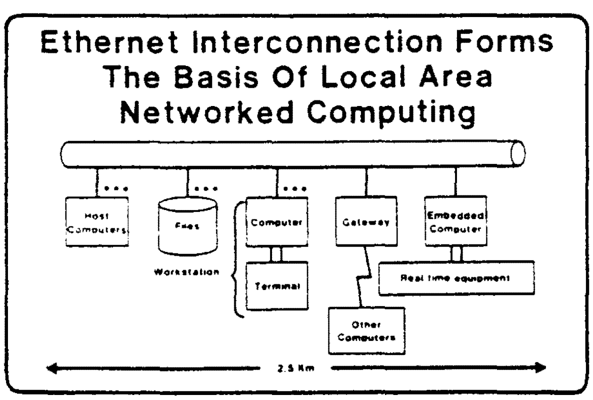


If you look at current computer architecture you will find standards. One such standard is the Digital Unibus that defines the architecture used in the largest selling series of minicomputers ever built—the PDP-11 series. The Unibus standard made it possible for our users and a number of different manufacturers to build memory boards, communications interfaces, and other components that can be plugged directly in a PDP-11 system in an open-ended fashion.

If you look at the Unibus-type architecture, or any competitive implementations of the Unibus idea such as Intel's Multibus or Motorola's Versabus, you will find that all system components—processors, system memory, data storage, and data communications interfaces—are connected by a single, high-speed data path or bus.

This bus enables the computer to move data within the system at very high speeds. Unfortunately there has been no standard bus to move data between systems at the similar speeds. Ethernet communications won't replace Unibus or any competitive busses but Ethernet will solve the local-area networking problem.

"Think of Ethernet as an extended bus..."



Ethernet is an extended bus. Up until now busses have provided high-speed computer communications within a very limited area—a single cabinet or room. Ethernet provides an extended bus that will link information processing nodes throughout a building, campus, or industrial complex.

The system components don't change. You have the same components in an Ethernet as you have in a single system. The only difference is that you now have more components and they're dispersed over a wider area. Where a Unibus system has a single processor, an Ethernet can have many.

Where a Unibus system has local data storage, an Ethernet will support databases distributed throughout the network.

Where a Unibus system interfaces to other computers, an Ethernet interfaces to other networks through gateways.

"The network becomes the system…"

In other words, with Ethernet, the network becomes the system. And when this happens, we will have a whole new computer generation—The Fifth Generation.

We—that is Digital Equipment Corporation—want to be a leader in the development of this new generation just as we were the leader in interactive computing and the development of the minicomputer generation that made distributed data processing possible.

Let me take a minute to define what I mean by a computer generation.

A new generation of computers comes about when there is a convergence of Technology and Need that forms a new Structure that is then followed by general Use.

With Ethernet and VLSI—Very Large Scale Integrated Circuits—we have the technology. That technology is needed to build and network an ever-growing number of computers, terminals, intelligent workstations, and personal computers that are being bought to solve many of the productivity problems facing business today.

There is also a new structure, the local-area network. Just as minicomputers and distributed processing changed the way computers were used in the 70s, local-area networks and personal workstations will change the way computers are used in the 80s.

The final requirement for a new computer generation is customer acceptance. Will the new technology and the new structure come into general use? In this particular case, I am convinced it will. Just as I'm convinced that Ethernet is the technology that will make this happen.

Ethernet provides the simplicity, speed, and universality needed in local-area networking.

Unlike other local-area networks, Ethernet is open-ended. It allows the user to build a local-area network from the bottom up without making a large capital investment or developing an inflexible long-range plan.

As I mentioned earlier, Ethernet is a passive communications medium. An Ethernet is really nothing more than a coaxial cable and standard protocols that define the way data is transmitted. For example, the Ethernet protocol defines packet size. It defines the way packets are addressed. It's really very simple. And its been tested for 10 years and it works.

Ethernet can carry a great deal of information at very high speeds. But you don't have to take my word for it. I'd like to read you part of an advertisement written by another computer manufacturer who adopted the Ethernet standard.

"Instead of taking . . . 44 seconds to transmit 10 pages of data, the transfer takes place in .042 second. In the 4.4 seconds it would take a conventional network to send one page of War and Peace . . . you could send the entire thousand page novel."

It is not difficult to see the benefits. You can transmit entire files from a computer to a personal workstation almost instantaneously. You can transmit photographs, data sheets, engineering drawings, or even voice messages.

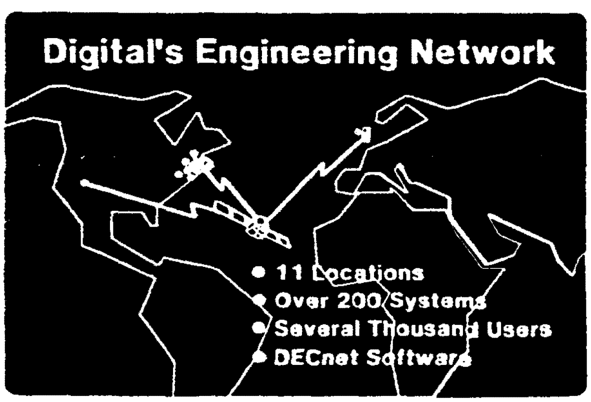
The key is universality. Any manufacturer who follows the Ethernet standard can build equipment to go onto the network.

But it is important that we realize that a network is more than just lines and nodes. Higher level protocols are needed to support the interconnection of dissimilar computers; to implement complex network functions such as file transfers and terminal-to-terminal communications; and to provide network management capabilities.

"Higher-level protocols are needed . . ."

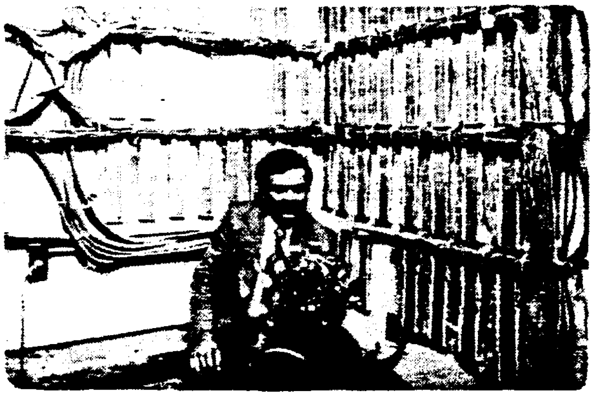
These protocols are complex. But they are a prerequisite for building a network such as the one that serves Central Engineering at Digital. One of the reasons we are committed to Ethernet is that it fits into the framework defined by Digital Network Architecture. We don't have to change the higher level protocols that are being used to support tens of thousands of DECnet nodes around the world. We can make Ethernet part of DECnet. We have a fit. And we have the range of capabilities required to implement complex computer networks.

Let's look at an example.



Digital's Engineering Network is made up of over 200 different systems serving about eight thousand terminal users. But interestingly enough, 80% of the traffic on this network is local traffic—only 20% of the traffic is between locations.

Local-area networking addresses the local problem. It provides high speed interconnection among computers within the same building or complex, and it simplifies the interconnection of terminals and processors to host computers.



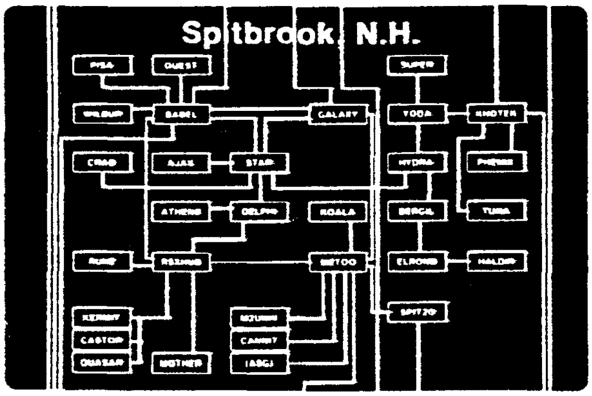
"Ethernet can eliminate this complexity while providing the flexibility needed for future growth."

Let's look at the computer-to-computer and terminal-to-computer interconnection problem in a little detail.

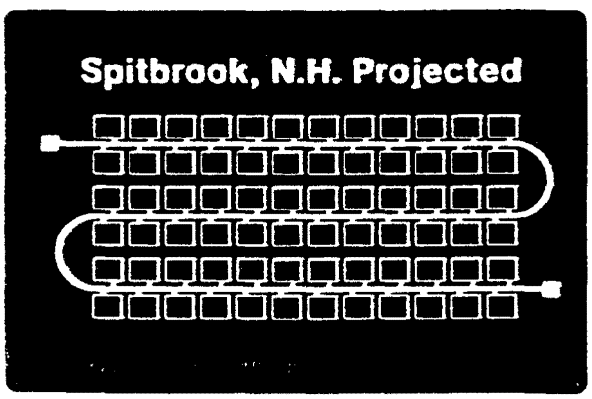
The problem is a wiring problem. It is one thing to connect A to B; quite another thing to connect A to B through Z. Before you know it you have a very complex maze of wires and switches. This is the wireroom in our Spitbrook, New Hampshire facility. As you can see, interconnecting a large number of devices is—at best—a very messy and, I might add, very expensive, business. You have fixed wires running all over the place. It's difficult to add systems or make changes.

Ethernet can eliminate this mess and provide needed flexibility. Ethernet will let us replace all this wiring with a single coaxial cable that will run throughout the building. When we want to add a terminal we'll just tap into the cable. It won't be necessary to run wires back to a central location. And we'll be able to add terminals to the network without interrupting network operations.

But we—like most other large organizations—are starting to provide individual users with intelligent workstations or personal computers rather than simple terminals. A simple terminal is usually a low speed device that can operate over telephone-type wiring. After all I can only read and write just so fast. I can type 50 words a minute. I can read about 200 words a minute. 9,600 bit per second transmission is more than fast enough for me as long as I only have a simple terminal. But when I have an intelligent terminal that can deal with information a lot faster than I can. I need to be able to communicate at computer speeds. Ethernet provides the speed needed to support intelligent user devices. The speed needed to transfer entire files or complex graphic images in a fraction of a second. I need Ethernet communications.



“This is how our computers are connected today.”



“Ethernet eliminates message switching.”

At the same time Ethernet solves the problem of interconnecting computer systems.

This is how the computers at Digital's Spitbrook, New Hampshire facility are connected today. As you can see messages have to be routed through the network. This creates computer overhead. Many systems spend much of their time switching and forwarding messages. And as more and more systems are added to the network this overhead just keeps growing and growing.

Fortunately, Ethernet can eliminate the overhead problem because it eliminates message switching and forwarding. This is how Spitbrook will look when we install an Ethernet.

As you see each system is connected directly to the Ethernet. There is no message switching. No routing. No forwarding. No computer overhead.

Instead of a maze of wires you have a high-speed, high-capacity extended bus that serves the entire complex. As you see Ethernet is changing the very definition of a system. With Ethernet, the network becomes the system.

We have a new technology. A pressing user need. And, a new structure. Three of the four prerequisites for a new computer generation. The fourth requirement is use. There are currently about 100 Ethernets in operation. There are going to be thousands. We've already talked to our customers. We know what they want and we know that many of them are going to install Ethernets. That's why I believe that we're looking at a new computer generation.

"Within the next few months we will be introducing our first Ethernet products."

We're going to build that generation. That's why we joined with Xerox and Intel to develop the Ethernet Specification. That specification conforms to both The Open Systems Architecture proposed by the International Standards Organization and Digital Network Architecture used in thousands of networks around the world. Right now we are implementing Ethernet as a part of Digital Network Architecture and within the next few months we will be announcing our Ethernet program and introducing our first Ethernet products.

I believe that Ethernet is one of the keys to the development of the Fifth Generation just as the Digital Unibus was one of the keys to the development of the minicomputer generation.

PERFORMANCE OF A SIMULATED ETHERNET ENVIRONMENT

A summary of remarks made by William R. Hawe, Principal Engineer in the Systems Performance Analysis Group, Digital Equipment Corporation, at the Xerox / Intel / Digital Seminar.

New York, February 10, 1982

BIOGRAPHICAL INFORMATION

WILLIAM R. HAWE

William R. Hawe is a Principal Engineer in the Systems Performance Analysis Group at Digital Equipment Corporation. He is involved in the modeling and performance analysis of networks and distributed systems. His interests are in local area network architectures and the performance of higher level protocols. He is a member of the Local Area Networks Performance Working Group of the IEEE Project 802 Local Area Networks Standards Committee. Prior to joining Digital he was a member of the faculty at Southeastern Massachusetts University where he performed research in X.25 packet switching for personal computers.

He received a BS and a MS in Electrical Engineering from Southeastern Massachusetts University. There he was president of the local chapter of IEEE and vice-president of Eta Kappa Nu. He received SMU's IEEE Outstanding Electrical Engineer award while he was research director of the Murail project. This research involved the development of distributed collision avoidance and dynamic routing systems for automated train systems.

PERFORMANCE OF A SIMULATED ETHERNET ENVIRONMENT

William R. Hawe, DIGITAL EQUIPMENT CORPORATION

SLIDE 1

In this study we investigate the performance of a simulated Ethernet environment. The goal is to predict the capacity of the channel in terms of the number of active users that it can support simultaneously. This provides an understanding of the loading one could expect in a particular environment. It also establishes the capacity in the system for future growth.

SLIDE 2

The goals were to establish the traffic patterns in the existing system and to estimate the excess capacity that would allow growth. The traffic patterns were established through measurements performed on operational systems that were interconnected with conventional point-to-point connections. We wish to see how heavily loaded an Ethernet would be if installed as an interconnect mechanism for the hosts, terminals, etc. We were also interested in understanding the additional loading that would take place because of new devices and their use (print and file servers, etc.) along with increased load due to growth in the user population.

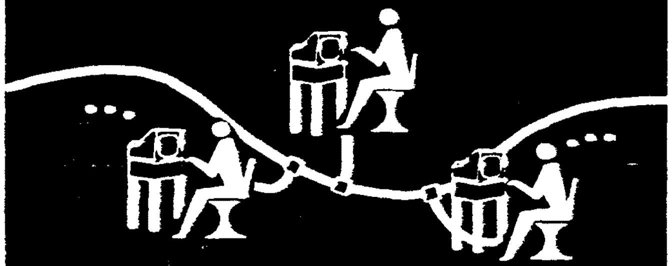
The behavior of users during various periods (such as a busy period) were monitored. The resultant data was then analyzed to produce a profile of the "typical" operations a user performs. From this, a workload which specifies the operations performed (and their frequency) was developed. This includes items such as the rates and sizes of commands, data, etc. that are exchanged between the user and the system.

To predict the growth capability present in the system, we simulated the Ethernet using a distributed architecture model and the user workload as the source of traffic. The number of users was then increased until the idle time on the Ethernet channel went to zero.

SLIDE 3

The results indicate that the Ethernet has sufficient bandwidth to support a large number of users of the type characterized in this environment. The delays in the Ethernet level of the architecture are small compared to other delays such as disk seeks, application program execution, etc. We also see that there are few collisions, even under heavy load.

Performance of A Simulated Ethernet Environment



-1-

Goals

- Establish Traffic Patterns Based On Current Environment
- Predict Growth Capability Based On Ethernet Installation

-2-

Results

- Ethernet Can Support Large Numbers Of Users
- Ethernet Delays Are Small
- Few Collisions Are Experienced

-3-

NOTES

SLIDE 4

In systems such as backbone networks, the delays in transfering information from node to node are usually dominated by the transmission and propagation delays. Processing time per message at the nodes is small compared to these factors. With the advent of local area networks we see a different relationship. Local area networks are generally built using interconnection mechanisms that have speeds of around 1 to 10 Mbps. They are generally confined to a limited geographic area such as few buildings. This means that now the transmission and propagation delays are much smaller in relation to the disk and CPU delays. For this reason, it becomes important to consider all levels in the system when evaluating the performance.

SLIDES 5,6,7

There are three parts to the study. First, measurements were performed to characterize the behavior of users in a program development environment. From this, a user profile (or workload) was developed. Second, the user workload is used as input to a model of the distributed architecture that is used in the Ethernet network. This results in a traffic load placed on the Ethernet. Finally, this load serves as input to a detailed Ethernet simulation. The number of users using the system in the simulation is then increased to observe the effects of increased load. It is assumed that enough hosts, terminals, etc. will be added to the system to support those additional users.

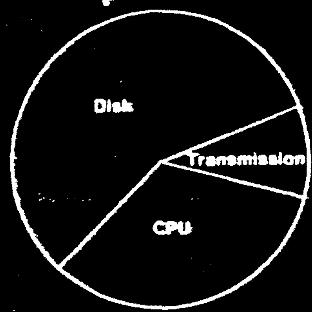
SLIDE 8

Here we are interested in the capacity of the system. There are many ways that one can investigate this aspect of the performance. Often the capacity of a channel is expressed in bits per second or percentage of the bandwidth used on the channel. Metrics such as this are difficult to interpret when one is interested in estimating how many users the system can support.

SLIDE 9

Therefore, in order to understand the capacity of the system we focus on the number of users that it can support. This is especially important when one is interested in determining whether or not there is sufficient capacity in the channel to support the existing user population as well as reserve capacity for future expansion both in the number of users and the types of traffic they generate.

Components Of User Response Time



-4-

Analysis Outline

- User Workload Measurements

-5-

Analysis Outline

- User Workload Measurements
- Distributed Architecture Model

-6-

NOTES

NOTES

Analysis Outline

- User Workload Measurements
- Distributed Architecture Model
- Ethernet Simulation

-7-

Understanding Performance

$$\text{Performance} = \frac{\text{Bits}}{\text{Second}}$$

= % Bandwidth

= ?

-8-

Understanding Performance

Performance = Number of Users

-9-

SLIDE 10

There are two ways in which the environment affects the number of users that the system supports. First, it dictates the higher level protocols to be used to transfer information between hosts, terminals, etc. This in turn affects the amount of traffic generated by each user. Second, it specifies the packet size distribution and arrival rate distribution. These play a significant role in determining the performance of the Ethernet.

SLIDE 11

Here we investigate the program development environment. Measurements were performed at several locations which were considered to be representative of this environment. As an example of such an environment we consider a large University. Users in this environment perform the obvious activities associated with the development of programs. This includes editing files, as well as compiling, linking, running and debugging the programs. They also communicate with other users by sending mail and using interactive message facilities such as "Talk". They copy, delete, print and perform other file manipulation operations. In addition to these functions, they also obtain information from the system. This includes help messages, queries about system status, etc.

SLIDE 12

To characterize the activities of a typical user various parts of the system must be monitored. Data was collected at several installations representing this environment. The data was collected at various times during the day so that busy periods could be investigated.

The amount and frequency of information transfer between the terminal and the host was monitored. In addition, the disk I/O that occurs as a result of operations performed by the user was also measured. This includes disk I/O that is for temporary work files such as those generated by programs such as linkers and compilers. Note that when we examine the impact of sending disk I/O over the Ethernet to a file server we do not include this type of traffic. This is because it is more efficient to generate and manipulate those temporary files at the location that the linker or compiler is running. However, the source and destination files can certainly be located on a file server. We also monitored other forms of traffic resulting from user operations. These included CPU usage, printing, network I/O, etc.

Analysis Outline

- User Workload Measurements
- Distributed Architecture Model
- Ethernet Simulation

-10-

NOTES

Program Development Environment

- Edit/Link/Compile/Run
- Mail & Talk
- Copy, Delete, Etc.
- Help, System Status, Etc.

-11-

Active User Profile

- Terminal I/O
- Disk I/O
- CPU/Memory Usage
- Network I/O
- Printing

-12-

SLIDE 13

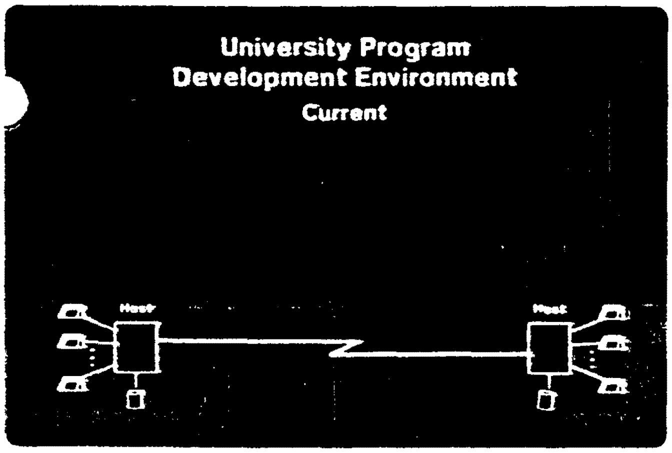
As we mentioned, the current environment uses conventional methods for interconnecting hosts, terminals and other devices. Terminals are connected directly to the hosts. The hosts are interconnected using point-to-point connections. The network is not always fully connected. However, the routing capabilities of the hosts assure that the network is logically fully connected.

SLIDE 14

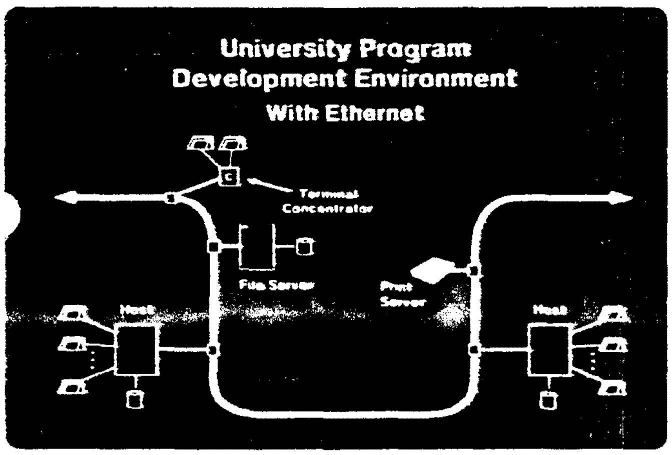
We consider the impact of an Ethernet installation in this environment. The Ethernet will carry traffic between the hosts for remote file and data access, remote logins, printing, etc. It will also carry traffic to and from new devices such as file servers and print servers. Existing terminals which are connected directly to the hosts can access remote hosts, servers, etc. by going through their hosts. Other terminals can also be connected to the Ethernet either directly (with the appropriate interface) or through terminal concentrators. With this approach they are not dependent on any one host's availability for access to the network. Personal computer workstations can also be connected directly to the Ethernet. Their traffic will be somewhat different than the terminal traffic because of the increased intelligence in the workstation. It will appear more like the host to host and host to server traffic. Gateways, routers, and other devices which allow communication outside of the local area network may also be connected directly to the Ethernet. Often hosts implement these functions in addition to their normal duties. The traffic which flows through those devices can be of any of the types already described.

SLIDE 15

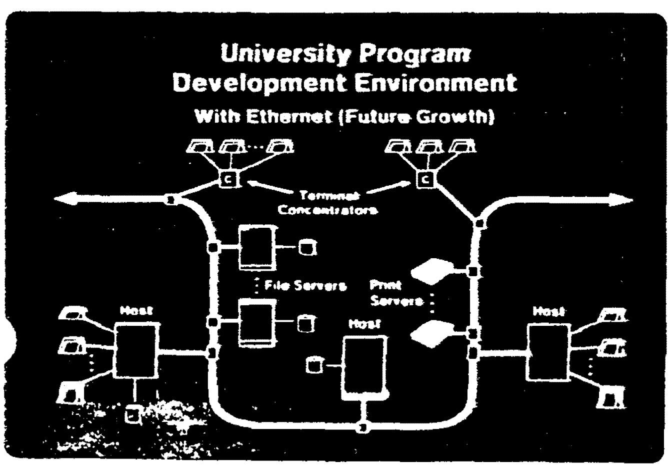
As time passes the network will expand in several ways. More devices will be added as the user population increases. This includes terminals, concentrators, hosts, servers, etc. Hosts without local terminals could be added and called computing servers. The other way in which the network will expand is in the traffic patterns. The availability of devices such as file and print servers will stimulate the growth in the traffic associated with those devices. For example, as more files are moved to file servers, so that sharing is easier, the devices will be used more often.



- 13 -



- 14 -



- 15 -

NOTES

SLIDE 16

We have discussed the users and their environment. Now we discuss the distributed architecture. A distributed architecture is necessary to provide an effective local area network. There must be facilities for reliable, controlled communications between users and processes inside and outside the local area network. This means that we need mechanisms for a user on the local area network to access information not only on the local network but also at some location that is not local. This would be accomplished by going through a gateway or router. Therefore, Ethernet is only a part of the total network architecture. It represents the lowest layers and is thus the foundation on which the local network is built.

SLIDE 17

The Digital Network Architecture (DNA) is an example of a complete network architecture. Here we see the relationship between DNA and the ISO layered architecture. The Ethernet comprises essentially the lower two levels for the local area network. Parts of the system which interface to public data networks could use the X.25 services. Other point-to-point links could use the DDCMP facilities. Above the data link is a network wide routing service. This delivers packets to the appropriate destination - either locally or remotely. Above that is an end-to-end service which provides for reliable communications between two processes. The Session layer controls the end-to-end service. Above that we have the applications and special purpose protocols. The network management facility has access to most of the protocol levels. It is used to monitor as well as control and configure them.

It is very important that all these layers in the architecture be considered when examining the user perceived performance of the local network. This is because each layer will add some additional load to the components of the system. Most will add some amount of additional traffic to the Ethernet. They will also use resources such as CPU cycles and memory space.

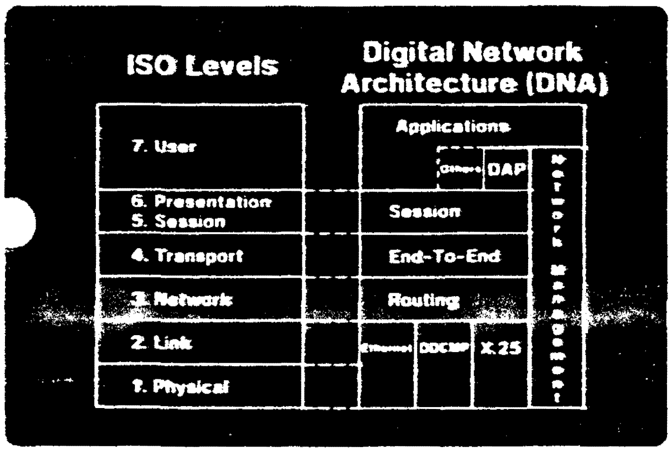
SLIDE 18

As we said, in DNA the Ethernet implements the physical and data link layers of the network architecture for the local area network. It offers a datagram service with delivery of packets on a "best effort" basis. In that sense it is different than other data link protocols such as DDCMP. The channel is, in general, relatively

Analysis Outline

- User Workload Measurements
- Distributed Architecture Model
- Ethernet Simulation

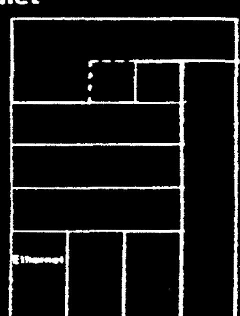
-16-



-17-

Protocols
Ethernet

- CSMA/CD
- Distributed Contention Resolution
- Fair to All Stations



-18-

NOTES

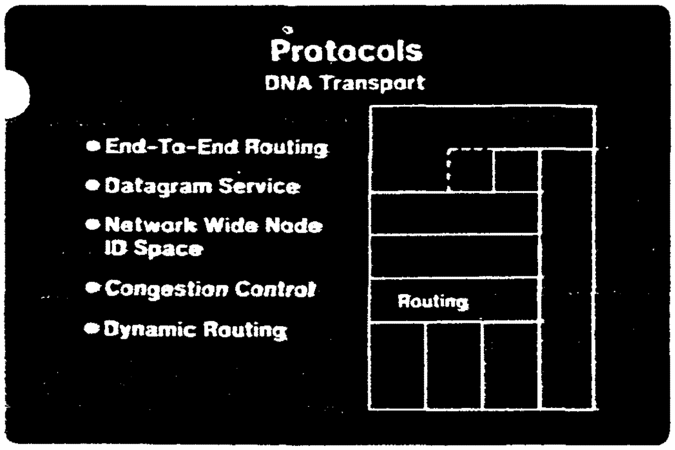
error free so this protocol is a good match. The Ethernet uses the CSMA/CD protocol to share the 10 Mbps channel. It uses a distributed algorithm called binary exponential backoff to resolve contention for the channel. This algorithm is executed independently by each station and is fair to all. The specifications allow a maximum of 1024 stations or "taps" on the Ethernet cable. However, as we shall see, there can be more users than taps. This is true of terminal concentrators where several user terminals may share a single tap. Hosts may also have a single tap as well as several users or processes that are generating Ethernet traffic.

SLIDE 19

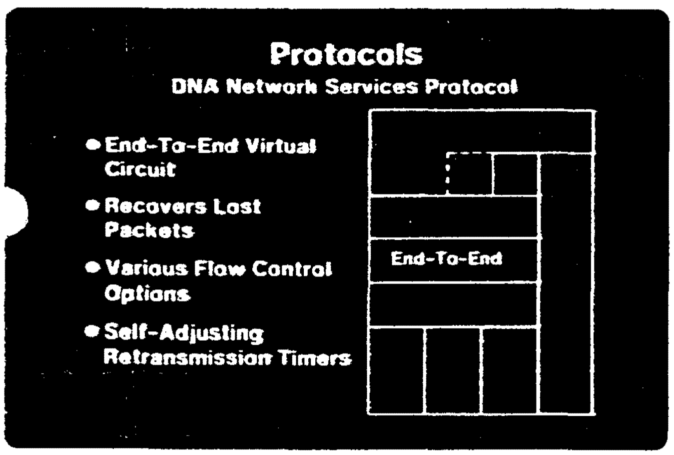
The DNA Transport protocol implements the network wide routing layer of the network. (This includes the local network as well as components that are not connected locally to the Ethernet.) This layer corresponds to essentially the ISO Network layer. It provides end-to-end routing of datagrams and routes packets to a destination even if the node is not on the Ethernet. To do this, it supports a network wide node address space. A node's address can be the same as its Ethernet address if it is on the Ethernet. However, all nodes are not necessarily connected to an Ethernet. Therefore, we need this address space. This layer also prevents congestion within the network and provides dynamic routing to bypass sections of the network that may have failed for one reason or another.

SLIDE 20

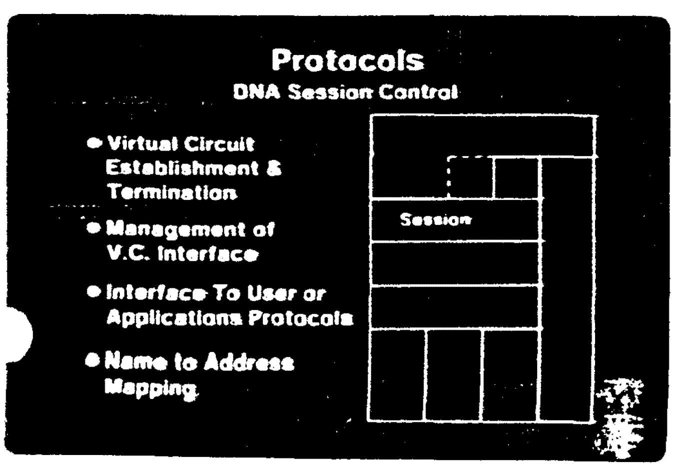
In order to provide effective, error free, and reliable process-to-process communication an end-to-end service is required. This is implemented by DNA's Network Services Protocol (NSP). NSP uses a virtual circuit to provide these features. This assures that packets are delivered to the user in the order they were sent. NSP makes sure that none are lost in the network. This is done by retransmitting lost packets. The timers used to decide when to retransmit a packet are self adjusting. This means that they adjust to the delays in the channel. This has the advantage of limiting the amount of unnecessary retransmissions thus reducing the load on the channel. The protocol also provides various flow control options. This allows the characteristics of the circuit to be tailored to the application. For instance, some applications may require tight control on the rates at which information is exchanged. These data rates impact the amount of resources (buffers, etc.) that must be devoted to the circuit. Flow control is



-19-



-20-



-21-

NOTES

especially important when the receiver is slower than the sender. An example is host to terminal output where the host can usually output data at a rate much faster than the terminal (or user) are capable (or willing) of accepting.

SLIDE 21

The DNA Session Control layer is used to control the virtual circuit service that NSP implements. It allows users to set up and terminate circuits. It validates incoming connect requests and activates the appropriate processes for those that are valid. It manages the interface between the user applications and the circuit. It also provides name to address mapping. For example, if the user requests that a circuit be establised to a node having a particular name, this layer determines the address of that node so that the connect request packet can be sent to the proper destination.

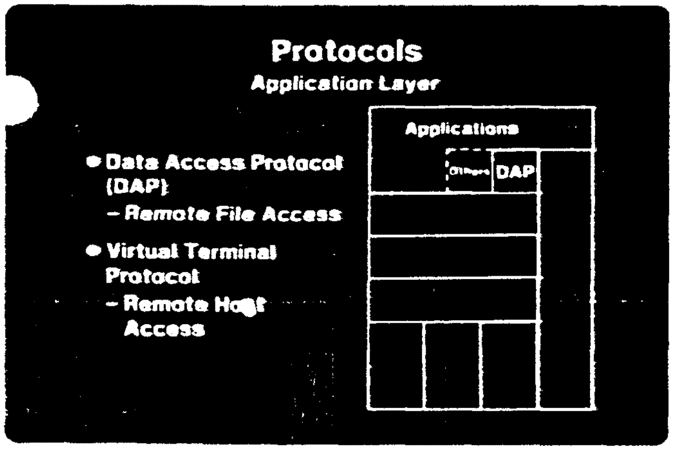
SLIDE 22

Above the Session Control layer are the applications protocols. The DNA Data Access Protocol (DAP) is one such protocol. It provides remote file access services. This means that the user can use this facility to access files as if they were stored locally on his system. The operation of the network is completely transparent. Another example of an application protocol is a virtual terminal protcol. This allows the user to connect to remote hosts through the network. The user then appears to be connected locally to that remote system.

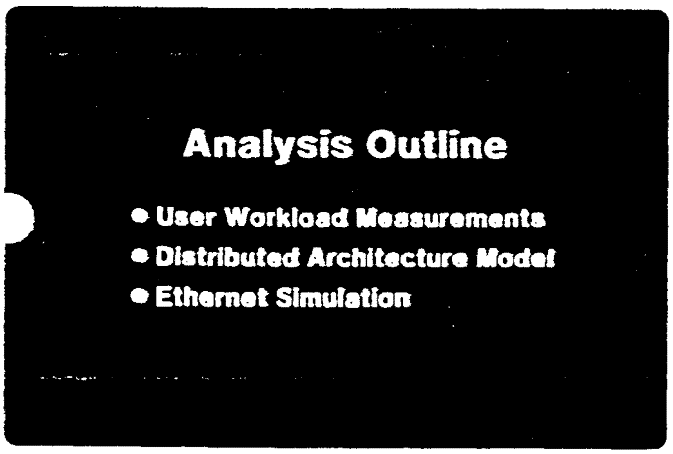
The network management part of the architecture is used to monitor and control the various protocol layers. It can be used by the network manager to monitor the traffic in the network and thus is useful for capacity planning. It is also used to tune the network for better performance.

SLIDE 23

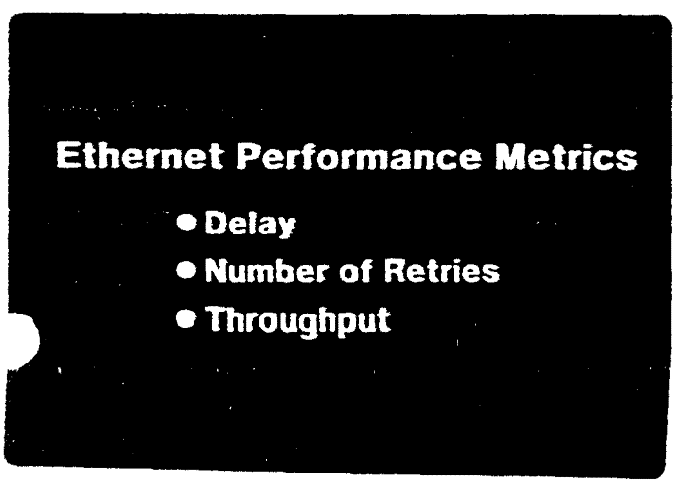
We use the user workload as an input to the distributed architecture model. The output of this is a load on the Ethernet. This consists of the user information being transfered between points on the network as well as various control and data packets associated with the protocol layers in the distributed architecture. The Ethernet simulation simulates the transmission of these packets.



-22-



-23-



-24-

NOTES

SLIDE 24

To understand the behavior of the channel there are several metrics one can examine. The delay experienced in transfering a packet between stations is of obvious interest. The number of retries necessary to accomplish that transfer is also important. Retries occur whenever there is a collision between two or more packets. The specifications indicate that after 15 retries (ie: 16 attempts) the packet will be aborted. At that point, the higher layer protocols must retransmit that packet. In this case it is NSP that will do the retransmission. The number of retries then gives us an indication of how the channel is behaving.

SLIDE 25

The values of the performance metrics such as delay, retries, etc. are determined by variables that come from two general sources. The first are those associated with the Ethernet itself. These are the transmission speed and the propagation delay. Here the transmission speed is 10 Mbps. The propagation delay depends on the size of the network. There is a maximum size that the network can have and therefore the worst case propagation delay is bounded. A transmitter must continue to transmit a packet long enough so that it can propagate to the farthest parts of the network. This way all stations can detect that a packet is being transmitted. However, another station may have started to transmit a packet before the signal from the first one reached it. In that case there is a collision. The collision must propagate back to the sender while it is still transmitting. This way it will know that its packet has been corrupted. The sender must therefore transmit a packet long enough so that it can propagate to the end of the network and any collision can propagate back. This time is called "the slot time" and it is about the round trip propagation delay for the largest network. (The slot time is 51.2 microseconds in the Ethernet specification.)

The other factor which determines the performance is the workload. This is the combination of the user workload and the traffic from the distributed architecture. The packet sizes and the rates at which they arrive for transmission over the Ethernet combine to present an given "offered load" to the Ethernet.

SLIDE 26

Here we see the mean waiting time on the Ethernet as a function of the number of users. The waiting time is the time from when a packet first becomes ready for transmission until its starts a successful transmission. It includes any time used in deference or collisions. We show three curves based on three levels of remote file traffic. Notice that for up to around 2000 users with this workload, the average waiting time is small when compared to typical delays at disks or in executing application programs or in processing protocol messages.

It is important to remember that the "users" in these curves are active users. This means they are logged in and actively working. Generally, the number of users that are actually using a system at any given time is only a fraction of the total user population. This is true not only for this program development environment but for other environments as well. For example, capacity planning of telephone systems uses knowledge of the relationship between the number of active users and the total user population.

Also note that the system can support more than 1024 users. As mentioned previously, the Ethernet specifications indicate that a maximum of 1024 taps may be connected to the cable. However, we have noted that taps can be shared by several users.

SLIDE 27

The number of retries a packet experiences is another indicator of the channel performance. A retry occurs whenever a packet has been involved in a collision. Here we see the mean number of retries plotted versus the number of active users for the three levels of remote file traffic. Note that for large numbers of users the average number of retries is still close to zero.

SLIDE 28

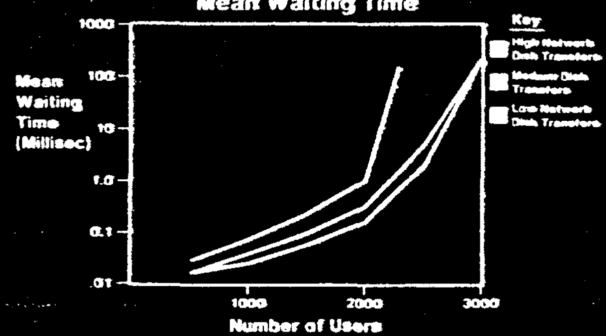
At some point when the number of active users is increased to a large enough number, the idle time in the channel will go to zero. This happens when the resources are all used in successfully transmitting packets and in overhead (such as collisions). Here this is plotted for the three levels of remote file traffic. Generally, one chooses an operating point at a point that allows fluctuation in applied load as well additional growth. We see that the Ethernet has ample room for growth at this particular installation based on its operating point. In other studies, such as the measurements of the PARC Ethernets, it has also been observed that the loading on the Ethernet in this and other environments is low.

Ethernet Parameters

- Channel
  - Transmission Speed
  - Propagation Delay
- Workload
  - Packet Size Distribution
  - Packet Arrival Rate Distribution

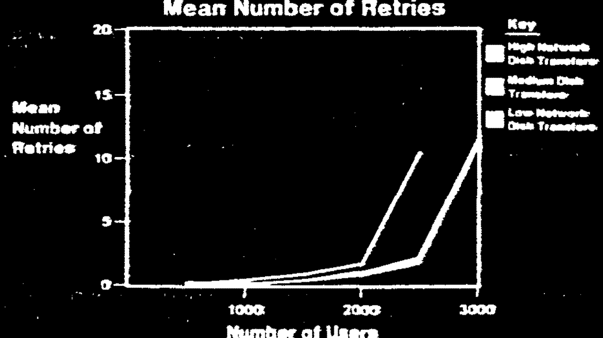
-25-

University Environment
Mean Waiting Time



-26-

University Environment
Mean Number of Retries



-27-

NOTES

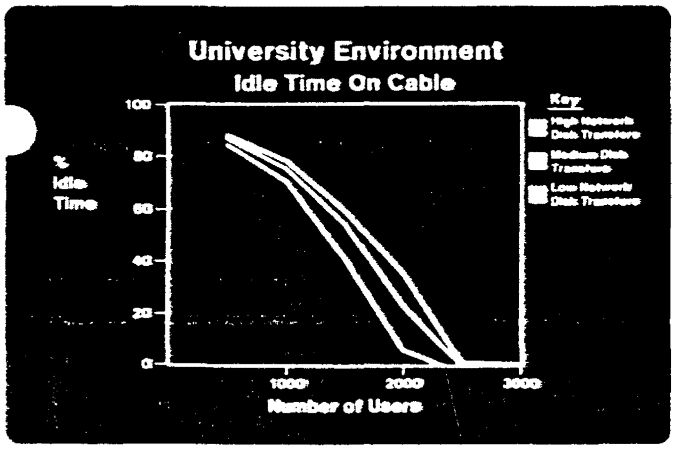
SLIDE 29

It is important that one keep the Ethernet performance data in the proper perspective. Consider a simple example of a file transfer from a file server over the Ethernet to a host or workstation. The "transmission component" includes the actual transmission time of all the packets in addition to the waiting time for each packet. There will be data and control packets from the various distributed architecture layers. The "CPU component" includes the processing time for each packet as well as any application overhead such as that due to the file system and application protocol. This also includes queueing for the CPU that will occur because there are multiple processes sharing that resource. The slower the CPU, the larger this component will be. The "disk component" includes the disk seek delays in addition to the rotational latency and transfer times for the data. It also includes queueing for the disk that occurs because it is shared. Comparing the CPU and disk components to the transmission component, it is not uncommon to observe that the ratio can easily be 4 to 1 or even 20 to 1 or higher - even when the Ethernet is heavily loaded which makes the waiting time longer.

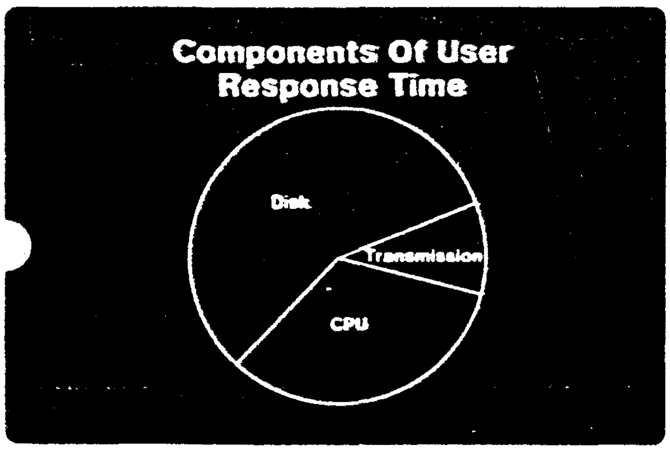
Other scenarios such as terminal I/O have similar relationships. There the disk component may or may not be as large. This depends on how much disk traffic the user generates. Linking and compiling programs, for example, can generate large amounts of disk traffic. The application program overhead in the CPU component can also be large.

SLIDE 30

To summarize, we have seen that the Ethernet is capable of supporting a large number of users of the type characterized in this environment. We have also seen that the delays associated with the Ethernet are typically small when compared against delays other parts of the system. We also note that few collisions are experienced. Therefore, the Ethernet seems well suited for this environment. It has ample capacity and performs well.



-28-



-29-

Conclusions

- Ethernet Can Support Up To Several Thousand Active Program Development Users
- Ethernet Delays Are Small Compared To Disk And Other Delays
- Few Collisions Are Experienced, Even Under Heavy Load

-30-

NOTES

PREDICTED CAPACITY OF ETHERNET IN A UNIVERSITY ENVIRONMENT

Madhav Marathe and Bill Hawe

Digital Equipment Corporation
Systems Performance Analysis
1925 Andover Street
Tewksbury, MA Ø1876

ABSTRACT

Local area networks are becoming increasingly popular as mechanisms for interconnecting a broad variety of devices within a moderate geographical area. The Ethernet* is one of the major access methods currently being used for this purpose. Terminals, hosts, personal computer workstations, gateways, and various types of servers have all found their way onto the Ether. The number of devices that one may attach to the channel is limited by several factors. Finite bandwidth, limitations of the contention resolution algorithm, physical constraints, etc. all impose certain limits. The number of users that may use those stations or "taps" for communication is also limited by these and other factors such as the layered protocol architecture, the physical system architecture, the user workload, etc. Here we examine the limits imposed on the number of users due to the finite bandwidth of the channel. This study is performed for users in a time-sharing environment. Measurements were performed to estimate the characteristics of that environment at a large University currently using conventional direct connections between hosts and terminals. We wish to estimate the limitations on the number of users when the system uses an Ethernet for the interconnection of hosts, terminals, etc. The characteristics of the user environment were coupled with a distributed architecture model and used as input to an Ethernet simulation. The results of the simulation give an upper bound on the number of users which can be supported in this environment. This of course assumes that there are a sufficient number of hosts, etc. so that those resources are not a bottleneck.

Keywords & Phrases

Ethernet, Ethernet performance, Ethernet simulation, higher level protocols, layered architecture, user level workloads, time-sharing, interactive program development.

* Ethernet is a trademark of the Xerox Corporation.

OVERVIEW

Local Area Networks

Local Area Network interconnection schemes such as the Ethernet provide the the framework in which one can construct systems which provide sharing of resources in an effective manner. Two aspects of the Ethernet which help achieve this goal are its speed and the fully-connected nature of its configurations.

To date, no one has come up with a standard definition of local area networks. However, most Local Area Networks do exhibit some general characteristics. Generally, they span areas of up to a few square kilometers. They are often contained completely in one or a small number of buildings. They usually have data rates in the range of 1 to 1Ø megabits/second. One group or organization almost always has complete control over the operation of the network. Since users are generally from one organization, there is a strong desire to access shared devices such as print servers, file servers, gateways, hosts, databases, etc. As a result, full physical connectivity is desirable. Because of the technology employed and the restricted size of the network, one observes lower bit error rates compared to conventional long-haul networks.

Because of the Local Area Network's speed it usually gets used for not only the traditional network communication but also for handling I/O traffic for shared disks, printers, etc. The personal computer workstations of the future will introduce a new class of traffic on the network. However, in the near future, the traffic on the local area network will consist of host/terminal traffic, host to host file transfers, mail, etc., specialized device traffic (print servers, etc.) and gateway traffic. We make use of this fact in modelling the workload on these networks. More information on Local Area Network technology and architectures can be found in [COTT80] or [FREE80].

(This paper will be presented at the SOUTHCON/82 conference.)

PREDICTED CAPACITY OF ETHERNET IN A UNIVERSITY ENVIRONMENT

Ethernet

In this paper we are concerned with a Local Area Network built using an Ethernet [DIGI80], [METC76]. Ethernet uses a broadcast mechanism (coaxial cable) and a distributed access procedure to allow for sharing of the channel. The procedure is called Carrier Sense, Multiple Access with Collision Detection (CSMA/CD). Nodes on the Ethernet can sense on-going transmissions and defer theirs until the channel is idle. They also have the ability to monitor the channel while transmitting to determine if any other stations are also attempting to transmit. Once an idle channel is sensed a station may transmit. Because of the propagation delay on the wire, two or more stations may sense an idle channel and attempt to transmit simultaneously. This results in a collision. In order that all stations (including the one transmitting the packet) can "hear" the collision it is required that all packets be greater than a certain minimum size. That size is determined by a parameter called the "slot time". The slot time is slightly greater than the round trip propagation delay. Any station involved in a collision must stop sending the packet and reschedule the transmission. The algorithm used to determine when the next attempt should be made is called the truncated binary exponential backoff algorithm. Basically, every time a station is involved in a collision it backs off (ie: waits) a random amount of time whose mean is doubled every time it experiences a collision. The backoff time is reset after a successful transmission. This algorithm has the advantage of being fair to all nodes on the Ethernet since it is executed by all. Ethernet performance is fairly robust. It degrades slowly and recovers well from momentary overloads [MARA80], [SHOC80].

The day to day operational performance of a 3 Mbps Ethernet is reported in [SHOC80]. It is interesting to note that the utilization of the channel was quite low. Less than 0.03% of the packets transmitted were involved in collisions while 99% acquired the channel with no latency.

One of the main reasons for Ethernet's popularity is because it uses a passive broadcast medium. This results in very reliable operation. Ethernet interfaces can be built using VLSI technology and thus made fairly inexpensive. Multi-vendor environments can be implemented by adhering to interface specifications at any of several levels. For instance, one may chose to provide compatibility at the wire tap, the transceiver cable, the port, higher level protocols, etc. Because of the heterogeneous enviornments in which Ethernets are used one can expect to see a great variety of traffic distributions. In this paper we study the traffic generated in a University environment and predict the performance of the Ethernet when used to satisfy the needs of that enviornment.

Methodology

This study deals with the behavior of Ethernet in the interactive time-sharing and program development environments. There are many installations which fall in this category. Our analysis is based on the measurements at one such installation - a large University with a number of large hosts presently connected to each other by conventional direct connections. We asked the question: "What will the traffic on the Ethernet at this University look like if an Ethernet was installed today?". We hypothesized that for the near future, the university will still have the dumb terminals (asynchronous, character mode) that are being used today and that these will be connected through terminal concentrators to the Ethernet. Others will still have direct connections to hosts since it is not likely that existing hardware will be thrown away. However, the users of those terminals still will generate Ethernet traffic in transfering files, sending mail, etc. The hosts will continue to have local secondary storage which will be used for user files and temporary workfiles. We assumed some level of file transfers and mail messages between hosts. Since we could not extrapolate the current traffic of this type into the superior sharing environment of the Ethernet, we assumed three somewhat arbitrary levels for traffic of this type.

Our principal objective is to predict the maximum number of users supported when the limiting resource is the Ethernet. In other words, we wish to estimate the number of users that can be supported on the Ethernet when all other resources such as terminals, processors and secondary storage are available in sufficient quantities so as not to be bottlenecks. There are two ways in which the environment affects the number of users supported. First, it dictates the higher level protocols to be used while executing the commands given by the user. This in turn affects the amount of traffic generated by a user. Second, it specifies the packet size distribution which has a significant role in determining the performance of Ethernet.

In estimating the Ethernet traffic we assumed that typical layered network protocols would be used. We coupled the user level workload with this model of the distributed arhitecture to estimate the average number of packets per active user per second. The packet level Ethernet simulation is then executed while increasing the number of users until the idle time goes to zero. Since the existence of Ethernet will cause more sharing and thus more host to host file and mail traffic, this workload alone is not sufficient to predict the total Ethernet load. We therefore study the network behavior with three levels (low, medium and high) of host to host file and mail traffic.

Note that in estimating the number of users a system will support one must also examine the user perceived response time and determine if it meets the requirements for the applications, environment, etc. Other bottlenecks such as disk delays, host processing of protocol messages, application program contention for memory and CPUs may play a larger role than the Ethernet in determining the user perceived delay. Those other possible bottlenecks may limit the number of users able to be supported to a smaller number than predicted here.

Performance Metrics

As mentioned above, here we concentrate on the performance at the Ethernet level. The delay through the Ethernet and the throughput as functions of offered load are two important performance metrics. The delay is often small compared to the delays in the higher levels. The main parameter controlling Ethernet performance is the ratio of the one way propagation delay (ie: half the slot time) to the average packet transmission time. This is called "alpha". The performance improves as this ratio is made smaller [MARA80], [SHOC80]. This is because packets are exposed to collisions only during the first slot time of their transmission. Once a packet has been on the wire for that length of time it should not experience a collision. Under heavy load the throughtput will be better if alpha is smaller [SHOC80].

The number of collisions a packet experiences in attempts to transmit is another interesting metric. Each collision causes the backoff range to be doubled. One would hope that, on the average, a packet does not experience many collisions. Measurements [SHOC80] and simulations [MARA80] have shown that there are few collisions in typical systems.

One could devise other metrics relating to the higher level protocols such as number of packets transmitted for each user message, etc. However, here we examine worst case scenarios and do not pursue that topic. It should be noted that the higher layers often dictate the performance of the network and therefore they should be carefully studied [MQUI80]. They will produce extra packets for each user packet transmitted. These control packets contend with the data packets for the limited resources of the shared channel (Ethernet). They also contend with other applications for resources (CPU cycles and memory) at the transmitter and receiver. Here we only address the issues relating to the shared channel.

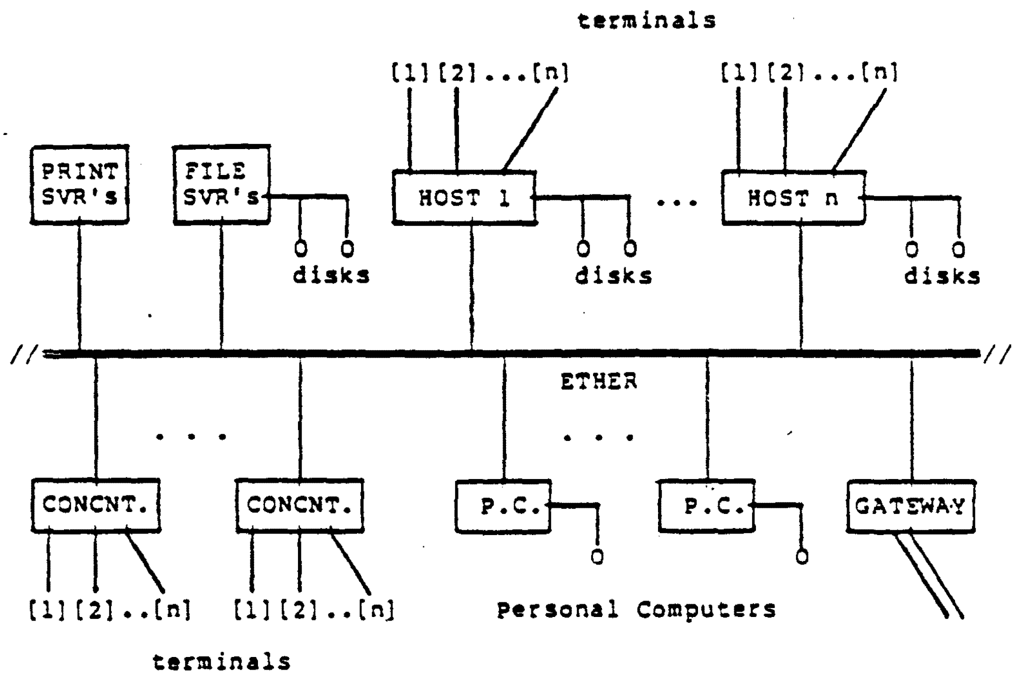


Figure 1. Local Area Network Components.

PROGRAM DEVELOPMENT ENVIRONMENT

System Components

Figure 1 depicts a typical collection of components found in a Local Area Network. Users can be connected to the Ethernet through terminal concentrators, hosts, or through personal computer workstations. Disk requests made on the behalf of a user can be directed towards a local disk (on a personal computer or a host), or they may be directed towards a file server. Swapping and paging traffic is assumed not go over the Ethernet since the hosts have local disks for "system related" operations. In this study we assume that the disk requests generated by the users are, for the most part, satisfied at the host with which they are communicating. However, remote file access and transfer (for mail, etc.) does use the Ethernet. Initially, the Local Area Network will not contain all the devices depicted in Figure 1. However, as time passes file servers, etc. will be added to the system.

User Profile

The workload contains descriptions of the activities of the users. User perform operations such as file edits, links, compiles, executes, etc. They also perform typical "house keeping" operations such as directory listings, file copies and deletes, etc. They send and receive mail and communicate with other users using interactive message facilities. The characteristics of the users were measured during heavy usage periods for several days at the University. I/O as well as program image related data was collected. Table 1 summarizes some of the major points of interest in the user I/O characteristics. The table contains the mean value of several interesting statisitics. It is important to note that many of these statistics had bimodal, trimodal, etc. distributions This means that more than the mean : required to fully understand the data.

PREDICTED CAPACITY OF ETHERNET IN A UNIVERSITY ENVIRONMENT

PARAMETER	VALUE
1) Avg. Session Duration	1307 seconds
2) Avg. Input Size (Term -> Host)	10.7 bytes
3) Avg. Input Rate (Term -> Host)	0.16 inputs/sec
4) Avg. Output Size (Host -> Term)	25.5 bytes
5) Avg. Output Rate (Host -> Term)	0.34 outputs/sec
6) Avg. Printed Character Rate	2.91 chars/sec
7) Avg. Remote File Access Rate (Assumed Light Usage, See Text)	0.00567 accesses/sec
8) Avg. File Access Size (Directed Locally or Remotely)	3584 bytes/access

Table 1. "Per-User" Workload Summary

In deriving the total network traffic generated by each user, the data and control packets generated at each protocol layer as a result of a user transaction were totaled and used to drive the Ethernet simulation. The amount of disk traffic present on the Ethernet will change with time as more intelligent servers and workstations are added to the system and as usage patterns change due to those new capabilities. We therefore have varied the load due to disk traffic in the simulation. Various amounts of the user disk traffic were sent over the network. This traffic is normally channeled to/from the host's local disk and the host. Access rates of 0.00567, 0.0085 and 0.017 accesses/second/user were used. This corresponds to 3.3%, 5%, and 10% of the traffic a given user generates at the local disk on the host.

Figure 2 contains a histogram of the Ethernet packet sizes generated by the user interactions coupled with the protocol model. The packet size includes user data (if any) the preamble, CRC and all other protocol fields. The protocol model was based on the examples contained in architectural specifications. (See [DAP80], [DECN80], [DIGI80], [NSP80], and [SESS80] for details of the architecture. See [WECK80] for an overview and description of its features and capabilities.) The model used assumes worst case examples. For instance, no acknowledgements are piggybacked. We also assume that each data packet transmitted requires its own acknowledgement and therefore there are no acknowledgements of multiple data packets. All of these assumptions are clearly worst case. They all increase the load on the Ethernet as well as the transmitter and receiver CPUs and memories.

RESULTS

Figure 2 contains a histogram of the Ethernet packet sizes generated by the user interactions coupled with the protocol model. The packet size includes user data (if any), the preamble, CRC and all other protocol fields from all protocol levels. The main contributor to the relatively large number of small packets (64 to 100 bytes) is the higher level protocol control packets. As mentioned previously, we have assumed the worst case for all protocol exchanges. This means that there are no piggybacked acknowledgements, etc. This imposes the heaviest load due to protocol control traffic. Since these are generally small packets, this distribution poses a demanding load on the Ethernet and should produce conservative results for this user workload.

Figure 3 shows the Ethernet offered load versus the number of users for this workload. The Ethernet specifications indicate that a maximum of 1024 taps may be connected to an Ethernet. The simulation conforms to that rule. Note that several users can share a tap. This is the case with terminal concentrators and hosts that have local terminals generating Ethernet traffic. In the figures presented here, the "number of users" corresponds to actual users - not to physical transceiver taps (of which there is a maximum of 1024).

Figure 4 shows the mean waiting time versus the number of users. Figure 5 shows the 90th percentile of the waiting time. The waiting time is defined as the time from when the packet becomes ready for transmission until it begins successful transmission. It includes all time spent defering, colliding and backing-off. As mentioned previously, three levels of remote file traffic were simulated. The "low level" corresponds to an access rate of 0.00567 accesses/user/second. The other two are for one and a half and three times the load due to that component. Note that with this time-sharing workload, the number of users supported is quite large.

Figure 6 shows the idle time on the Ethernet going to zero at the overload points. Again note that this occurs for an unusually large number of users. Figure 7 shows the number of attempts required to successfully aquire the channel as a function of the number of users. The number of attempts includes all collisions as well as the one successful attempt which aquires the channel. Note that even at an overload point with 2000 users, a given packet experiences an average of only one collision per successful transmission. Figure 8 shows the 90th percentile of the number of attempts.

CONCLUSIONS

The results of the simulation indicate that the Ethernet has sufficient bandwidth to serve large numbers of users of the type characterized by the time-sharing workload. In practice, one generally does not operate the system with the steady state load near the system limits. The finite rate at which the hosts, disks, users, etc. can generate and process information will prevent the steady state loading from achieving this level.

The waiting time experienced in attempting to gain access to the channel was shown to be within reasonable bounds. The number of collisions experienced by a packet attempting to acquire the channel was also shown to be quite low - even in the heavily loaded regions.

In summary, we can say that the Ethernet seems to be well qualified to carry the type of traffic experienced in the time-sharing environment. It has the capacity to support large numbers of users in this environment.

Discussion

Here we have shown that the Ethernet is capable of handling the traffic generated in this time-sharing environment. To build an effective network, the operation of the higher level protocols must be examined. The delays encountered due to processing and queueing can result in poor user perceived performance if care is not taken in their implementation. One should also examine other environments to see how similar or different they might be and how this affects performance. For example, the office environment is very important.

ACKNOWLEDGEMENTS

We would like to thank our colleagues in the Systems Performance Analysis Group, especially Rollins Turner, for obtaining the workload measurements as well as their help in analyzing the large amounts of data. We also wish to thank them, and others in Distributed Systems, for insights regarding the modelling of Ethernets in this environment. Finally, the people in Systems Performance Analysis and Distributed Systems Product Management who reviewed this paper deserve special thanks for their many useful comments and suggestions.

FIGURES

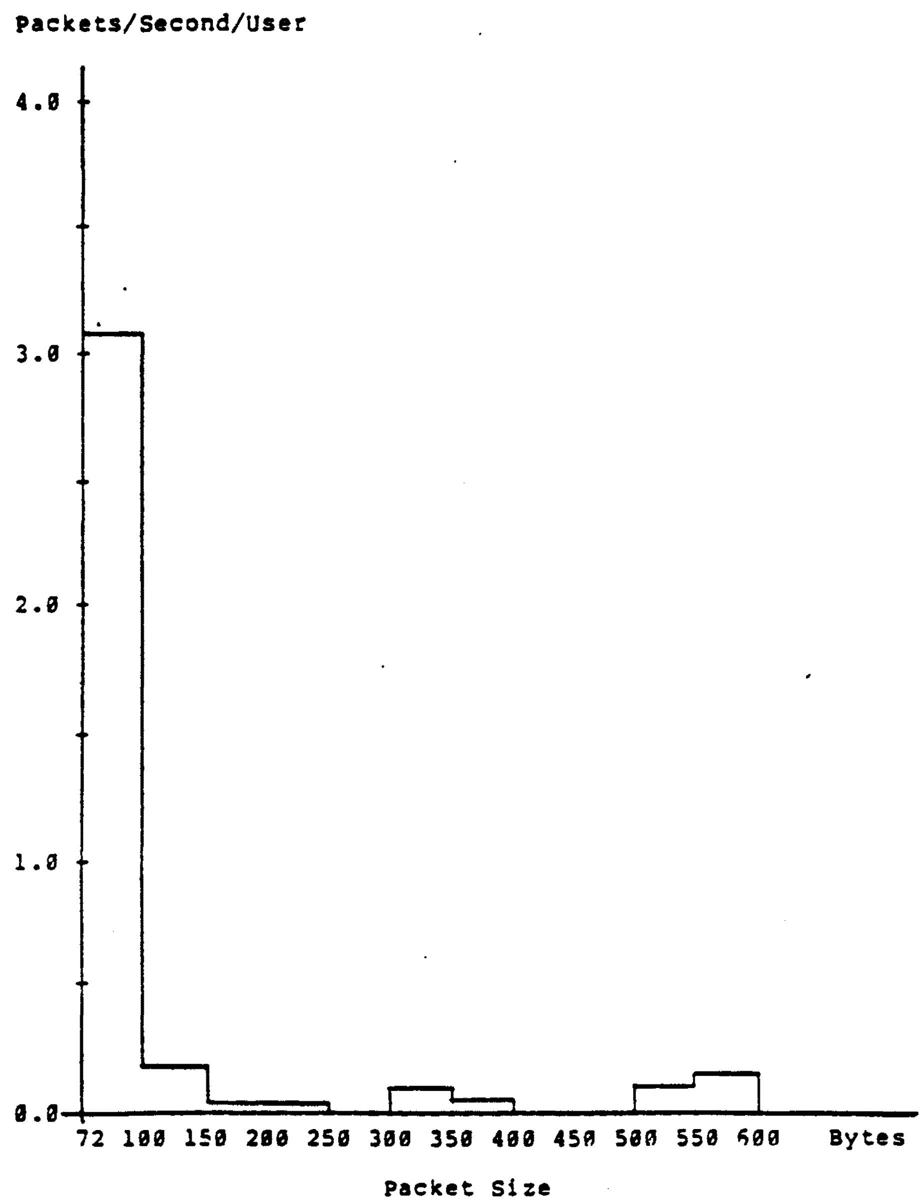


Figure 2. "Per-User" Ethernet Packet Size Frequencies
(Low Remote Disk Traffic)

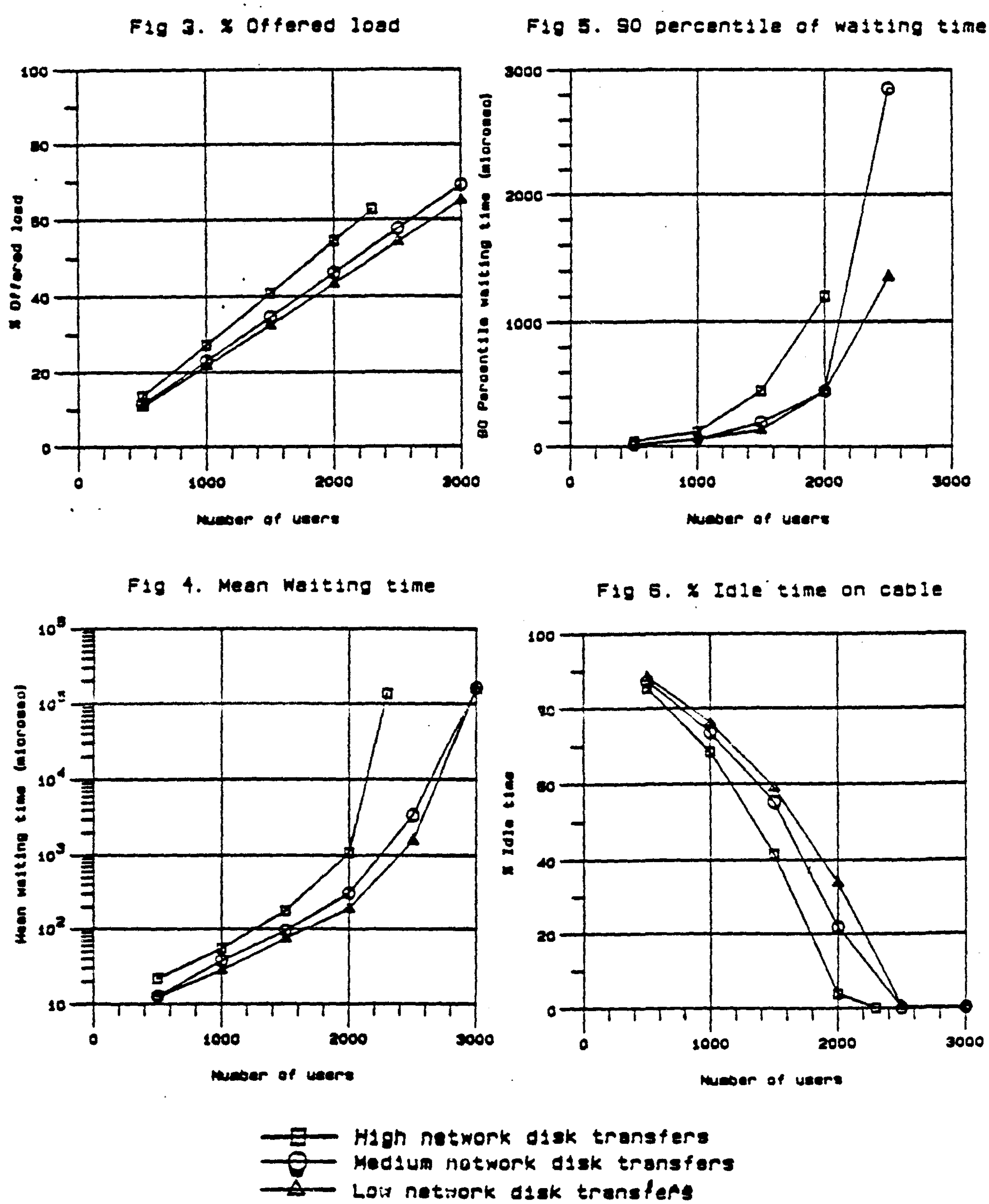


Fig 3. % Offered load
% Offered load
Number of users
Fig 5. 90 percentile of waiting time
90 Percentile waiting time (microsecs)
Number of users
Fig 4. Mean Waiting time
Mean waiting time (microsecs)
Number of users
Fig 6. % Idle time on cable
% Idle time
Number of users
High network disk transfers
Medium network disk transfers
Low network disk transfers

Fig 7. Mean number of attempts

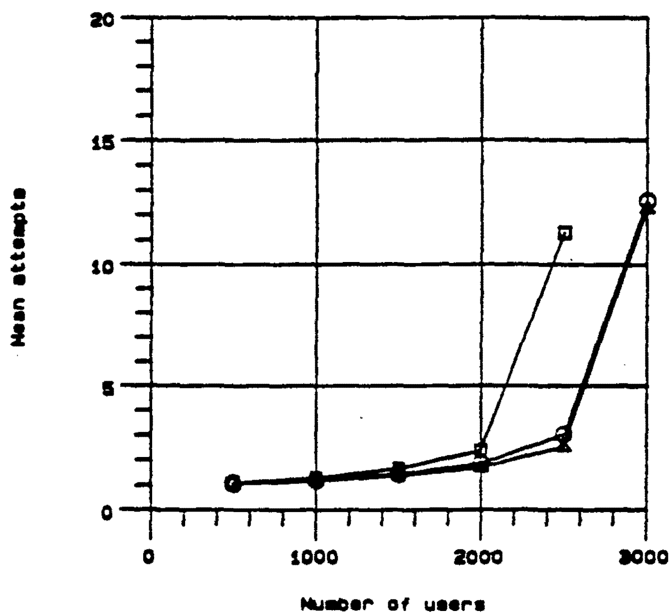
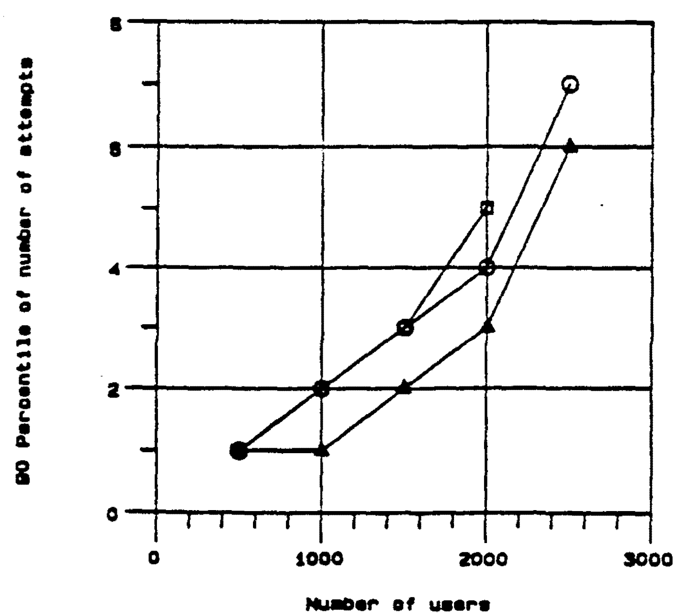
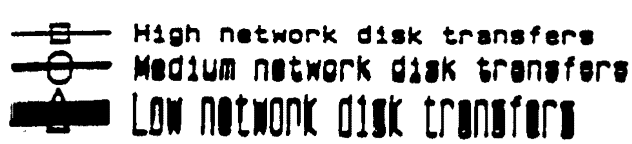


Fig 8. 90 Percentile of number of attempts





REFERENCES

[COTT80] TECHNOLOGIES FOR LOCAL AREA COMPUTER NETWORKS, I. Cotton, Computer Networks, Vol.4, No.5, Oct/Nov 1980, pgs. 197-208.

[DAP80] DECNET DATA ACCESS PROTOCOL FUNCTIONAL SPECIFICATION, Version 5.6.0, Digital Equipment Corporation, October 1980.

[DECN80] DECNET TRANSPORT FUNCTIONAL SPECIFICATION, Version 1.3, Digital Equipment Corporation, March 1980.

[DIGI80] THE ETHERNET - A LOCAL NETWORK, Version 1.0, Digital, Intel, Xerox, September 1980.

[FREE80] UPDATED BIBLIOGRAPHY ON LOCAL COMPUTER NETWORKS, H. Freeman, K. Thurber, ACM Comp. Comm. Review, Vol.10, No.3, July 1980, pgs. 10-18.

[MARA80] DESGIN ANALYSIS OF A LOCAL AREA NETWORK, M. Marathe, Comp. Network Symp., Wash. D.C., Dec. 1980, pgs. 67-81.

[METC76] ETHERNET: DISTRIBUTED PACKET SWITCHING FOR LOCAL COMPUTER NETWORKS, R. Metcalfe, D. Boggs, Comm. ACM, Vol.19, No.7, July 1976, pgs. 395-404.

[MQUI80] LOCAL NETWORK TECHNOLOGY AND THE LESSONS OF HISTORY, J. McQuillan, Computer Networks, Vol.4, No.5, Oct/Nov, 1980, pgs. 235-238.

[NSP80] DECNET NETWORK SERVICES PROTOCOL FUNCTIONAL SPECIFICATION, Version 3.2.0, Digital Equipment Corporation, October 1980.

[SESS80] DECNET SESSION CONTROL FUNCTIONAL SPECIFICATION, Version 1.0.0, Digital Equipment Corporation, November 1980.

[SHOC80] MEASURED PERFORMANCE OF AN ETHERNET LOCAL NETWORK, J. Shoch and J. Hupp, Comm. ACM, Vol.23, No.12, Dec. 1980, pgs. 711-721.

[WECK80] DNA: THE DIGITAL NETWORK ARCHITECTURE, S. Wecker, IEEE Trans. Comm., COM-28, No. 4, April, 1980, pgs. 510-526.

P‾ƆGRAPHICAL INFORMATION

MADHAV MARATHE

Madav Marathe is a Consulting Engineer in the Systems Performance Analysis Group at Digital Equipment Corporation. He is involved in performance analysis of local area networks and data base systems. His present research interests are in distributed data base architectures, data base machines and file server architectures for local networks. He is a member of ACM and the Local Area Networks Performance Working Group of the IEEE Project 802 Local Area Networks Standards Committee.

He received a Ph.D. from Carnegie-Mellon University for research in the performance of the hardware architecture and operating systems kernel levels of a computer system. While there (and at Digital), he also did research in memory and data contention in multiprocessor systems.

DAVID E. LIDDLE

Dr. Liddle received his undergraduate degree in engineering from the University of Michigan in 1966. He later received a M.S. and Ph.D. from the University of Toledo, the latter being awarded in 1972. From 1970 to 1972, Liddle was Project Manager at Owens-Illinois, Inc. in Toledo, Ohio.

In 1972, Dr. Liddle joined Xerox Corporation as a research scientist in the Palo Alto Research Center, where he worked on the "POLOS" office system project, early design issues for the Alto, and various file servers.

In 1975, Liddle became Manager, System Architecture in the Systems Development Department where he was responsible for the development, definition, and specification of the overall OIS architecture for Xerox. In January of 1976, he wrote and published the "OIS Architectural Principles". Until 1979, Liddle was Manager of the Systems Development Department. This organization developed the Mesa programming environment and the Ethernet communications network. It also produced the Pilot Operating System, and was responsible for a new family of software products for office applications.

In 1979, Liddle was appointed Vice President of the Office Products Division of Xerox. In this position, he had continued responsibility for the development of Ethernet, Integrated Network Services, and Advanced Information Processing Systems for OPD. He became a member of the Technology Review Group, a corporate committee which review strategic technical issues on a corporate-wide basis.

Currently, Dr. Liddle is Vice President and General Manager, Office Systems Business Unit for the Office Products Division. He has general management responsibility for office automation systems and products, including network services and professional workstations.

ETHERNET EDITORIAL SEMINAR

PRODUCTIVITY IN THE OFFICE ENVIRONMENT

In discussing 'productivity in the office', it's important to first clarify just what is meant by those terms. In this context, the concept of the extended office is being used: any structured association of people working with information. Essentially, improving productivity equates to producing more work, of higher quality, at less cost.

Business Week reports office costs are rising at the rate of 12-15% a year and will probably double over the next six years. Those costs are rising faster than any other cost factor - even faster than the costs associated with generating business revenue. Direct costs of office operations in 1980 were over $920 billion and are likely to rise to $1.5 trillion by the end of the decade. What's important to look at is that this productivity factor in the extended office is really people productivity.

Over 50% of American workers now work with information on a full-time basis. Nevertheless, nationwide, over ten times as much is invested in technology for factory workers as for office workers. When companies first attempt to "automate", their attention traditionally has been on the secretary. However, secretarial functions account for only 23% of office costs and only around 12% of salaries. At the opposite end of the spectrum, managers and executives have also benefitted somewhat from technology. The mainframe computer has typically provided data processing reports of various forms for use by the manager.

The person in the middle of the office hierarchy, the professional, has not had the benefit of technology even though professionals make up 80% of payroll costs - and their numbers are expected to grow 30% during this decade. Can their tasks benefit from technology and thereby, make them more productive? A number of studies say "yes".

To increase professional productivity, the tools they use must be improved, and the barriers to productivity must be eliminated or minimized in the four basic areas of information processing: creating knowledge, reproducing it, getting it in and out of files, and distributing it to others. Putting these two thoughts together says that the "better tools" must be able to work together; Ethernet provides the interoperability for that solution.

Every network user has the option of selecting the piece of equipment that best meets his or her own individual needs, whether that be the need for a recording typewriter for short letters and memos, or a personal computer to run accounts payable or inventory, or access to a mainframe computer. The user must not be limited to equipment from just one vendor.

It's important that the network and the products on it can grow in an evolutionary manner. A company should not be penalized by starting small. The evolution into automated office systems integrated on a network should not require a massive, all-encompassing galactic plan.

Ethernet has over 7000 person-years of testing and user experience. All of this experience supports the important premise of ease of growth and interoperability. The specifications for Ethernet were published jointly in September, 1980, by Intel, Digital Equipment Corporation, and Xerox. Since that time, over 275 requests for license applications have been made; over 70 applicants have paid their license fee; 22 have publicly announced their intentions to build Ethernet compatible products. This speaks for itself; no other network technology has attracted such a broad allegiance. The fact is, it works. Over 50 installations of Ethernet networks within the past four months prove it.

Ethernet has provided a truly integrated approach to automating office tasks. Systems connected to the Ethernet operate simultaneously, and can be both standalone office machines and part of the network system, sharing resources or files or printing devices. The open architecture of Ethernet allows multi-vendor connectibility. The specifications have been published to allow other vendors that ability. The Ethernet customer is not forced to purchase all their equipment and services from one vendor. The higher level protocols that Xerox recently published take this connectibility a step further and allow any vendor to be truly compatible with other products on the Ethernet.

Ethernet's interconnectibility and interoperability is transparent to the user. The barriers to productivity can all be hurdled. Input, output, filing, retrieval, distribution - all can be accomplished from any system on the Ethernet. A secretary can print on the laser printer from an electronic typewriter. A manager can call up records files from a mainframe and manipulate them on a personal computer. The professional can access massive stored reports and extract information to prepare a summary report, complete with graphics, on a professional workstation. And everyone can distribute information to every other workstation on the system without the delays of mails and unanswered telephone calls. Network capabilities are driven by user needs, and Ethernet provides these integrated services critical for office productivity.

David E. Liddle
Vice President & General Manager
Office Products Division
Xerox Corporation

ETHERNET EDITORIAL SEMINAR

PRESENTATION HANDOUTS

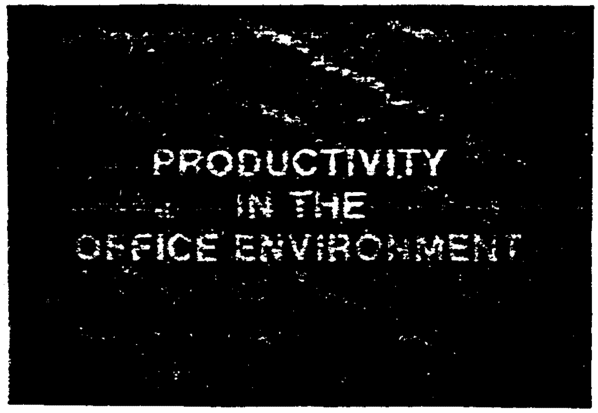
PRODUCTIVITY IN THE OFFICE ENVIRONMENT

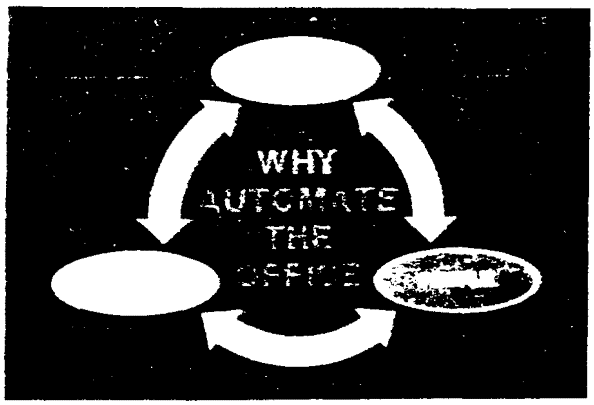
DAVID E. LIDDLE

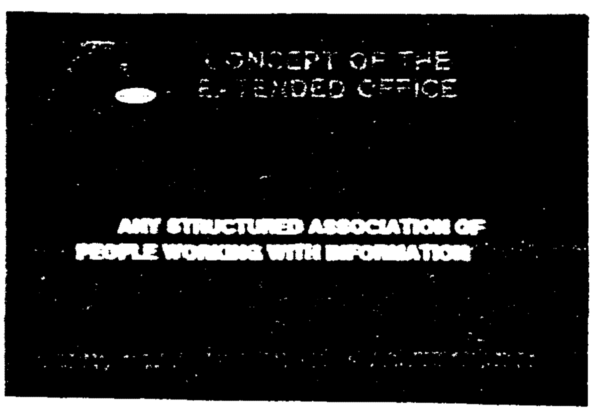
XEROX CORPORATION

OFFICE PRODUCTS DIVISION

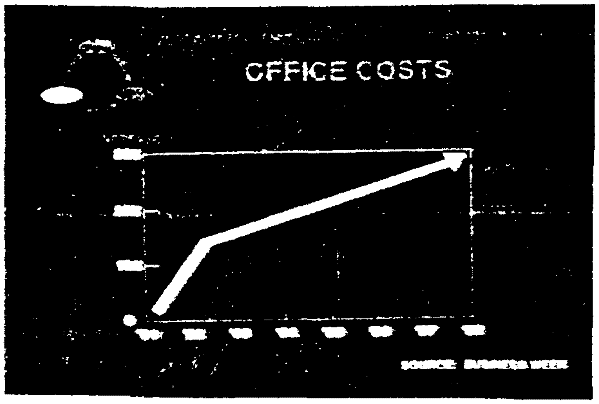
February 10, 1982



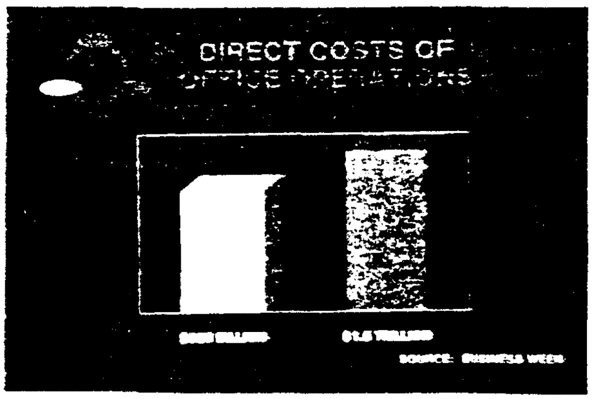




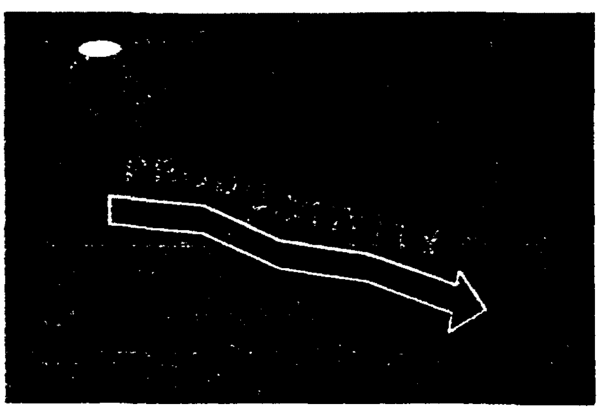
The extended office concept goes beyond the reference to an individual's separate office or a separate office function and is meant to include the entire structured association of information handlers.



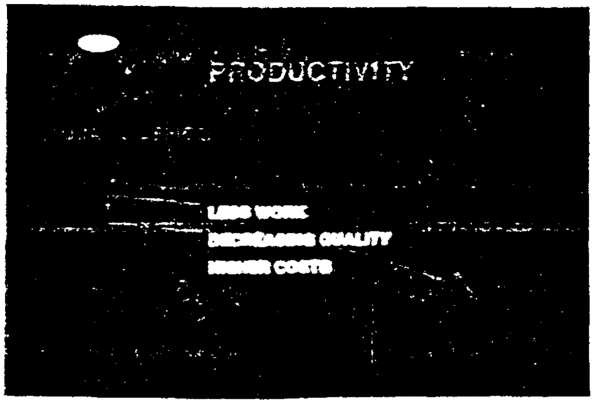
Office costs are rising 12-15%, doubling over the next six years. These costs are rising faster than any other cost factor, even faster than the costs associated with generating business revenue.

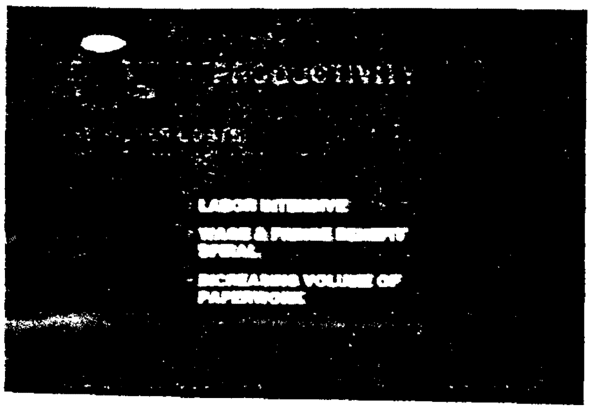


Direct costs of office operation in 1980 were over $920 billion. Overhead expenses are expected to rise to **$1.5 trillion** by 1990.

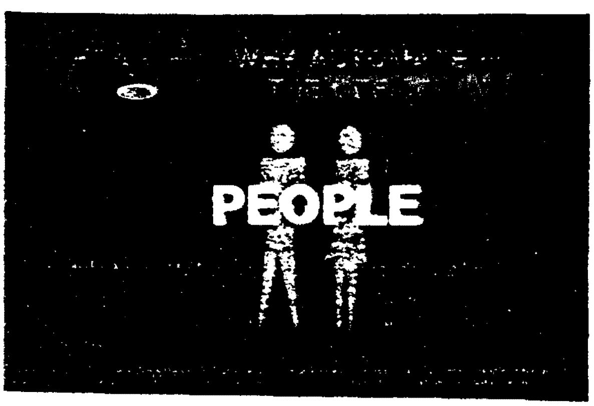


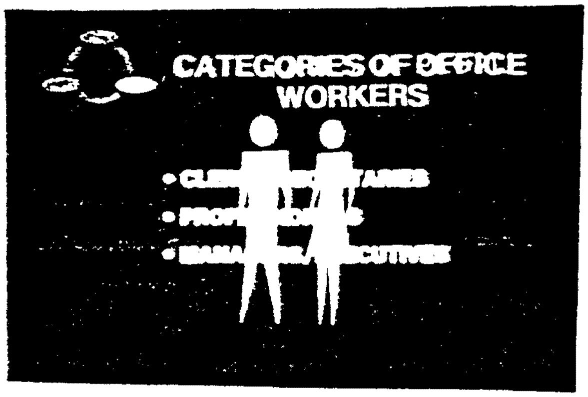
While office costs are rising, however, office productivity is declining.



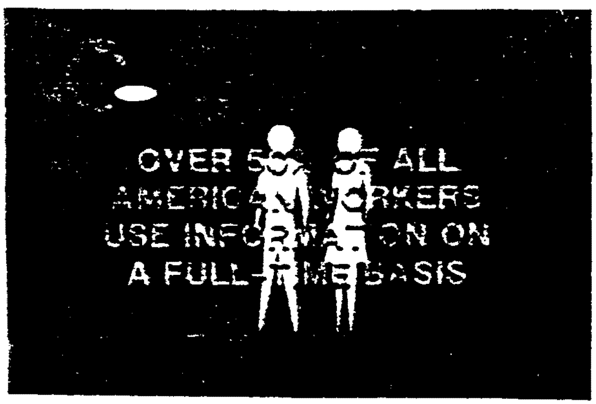


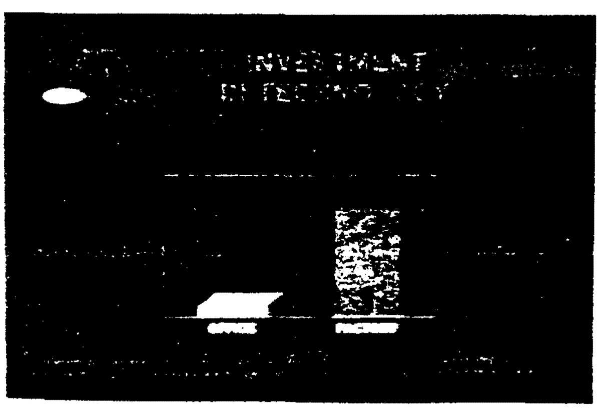
This explosion of costs is making a significant negative impact on bottom-line profits for American business. It's important, however, to understand these problems of productivity in terms of real people.



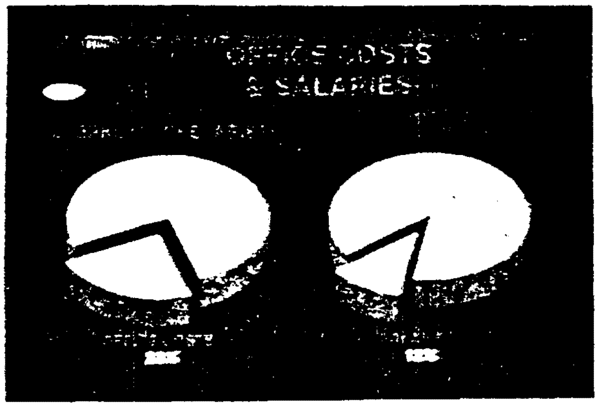


These office people fall into five categories: The **clerk** and **secretaries** that gather data in the form of numbers and information; the **professionals** who create ideas based on information; and the **managers** and **executives** who make decisions based on the ideas and information from their staffs.



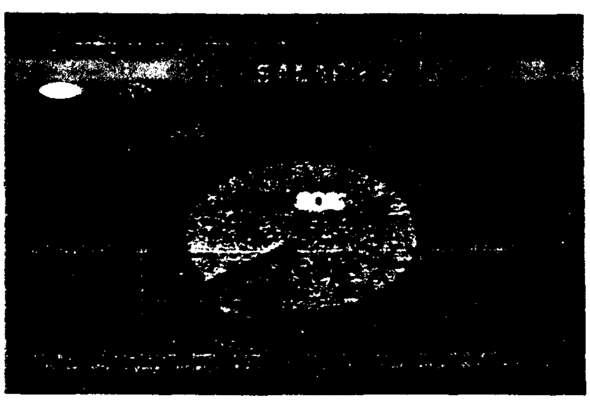


Although the majority of workers are in the office, over ten times as much is invested in technology for the factory worker as for the office worker.

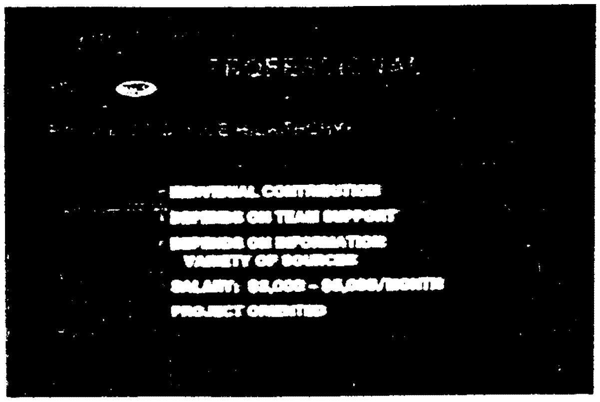


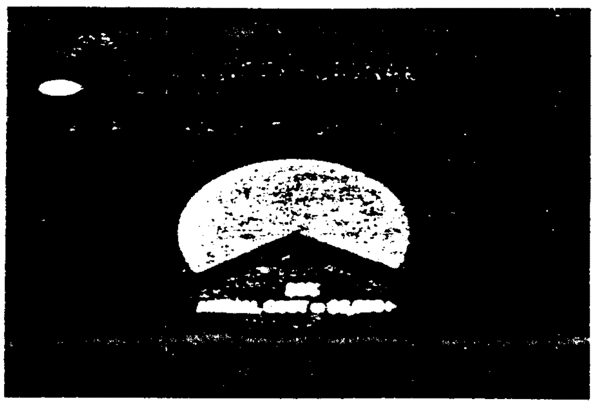
When companies first attempted to automate their offices, their attention traditionally has been on the secretary. However, secretarial functions account for only 23% of office costs and only around 12% of salaries.

At the opposite end of the spectrum, managers and executives have also benefitted from technology. The mainframe computer, with its elaborate processing power, has typically provided data processing reports of various forms for use by the manager/executive.

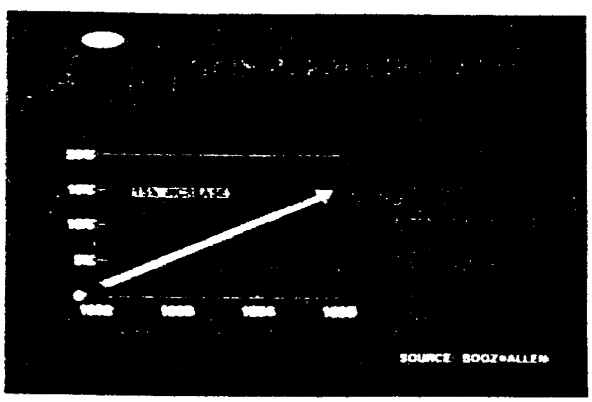


The person in the middle of the office hierarchy, the **professional**, has not had the benefit of technology even though they make up 80% of payroll costs, and their numbers are expected to grow 30% during this decade.

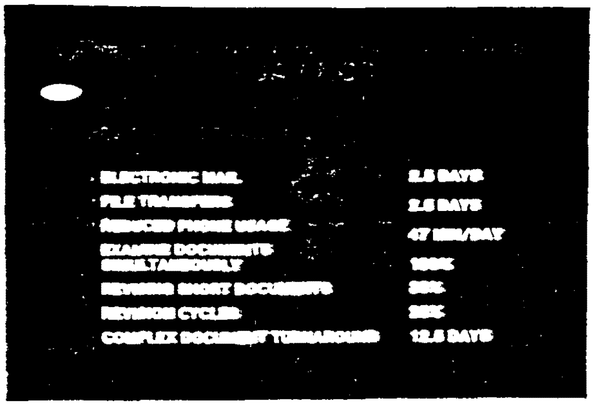




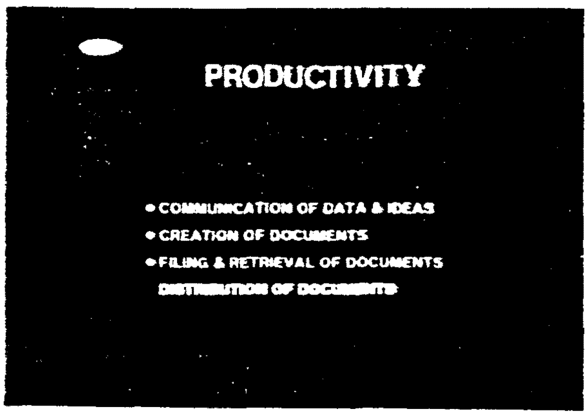
Can the professionals' tasks benefit from technology and thereby, make them more productive? A number of studies say "yes". For example, approximately one third of a professional's time is spent in creating documents at an average cost of $6000/professional.



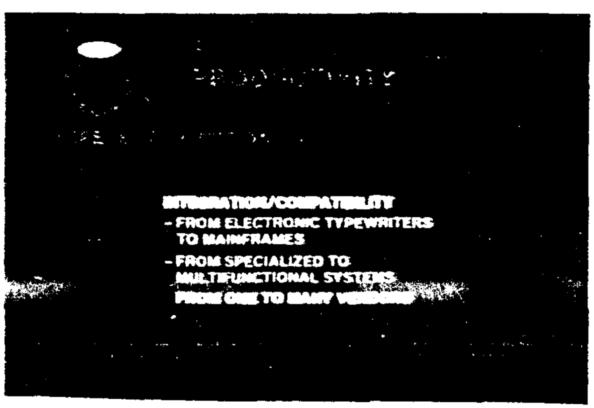
A recent Booz Allen study indicated that by utilizing office automation technology, a 15% gain in professional productivity could be realized by 1985. That's an average annual savings of $5,500/information worker.



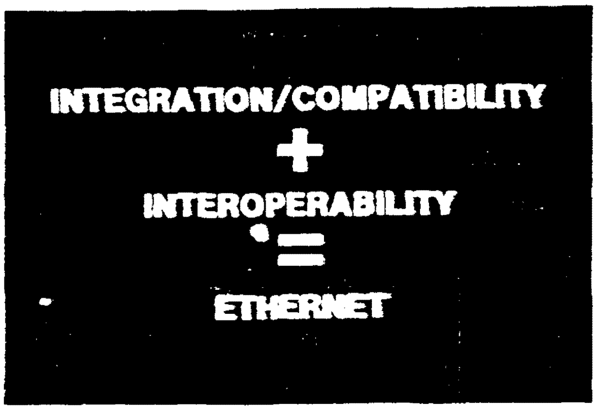
In internal Xerox probe locations utilizing professional workstations and network services, these productivity gains were realized.



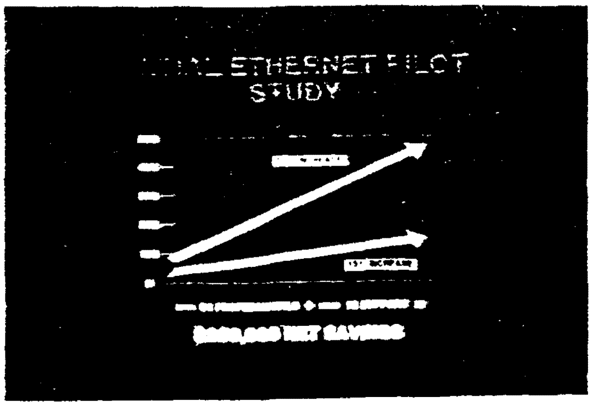
These probes and studies indicate that to increase productivity, the barriers to productivity must be eliminated or minimized in the four basic areas of information processing: creating and communicating ideas and data, creation of documents, filing and retrieval of documents, and the distribution of documents.



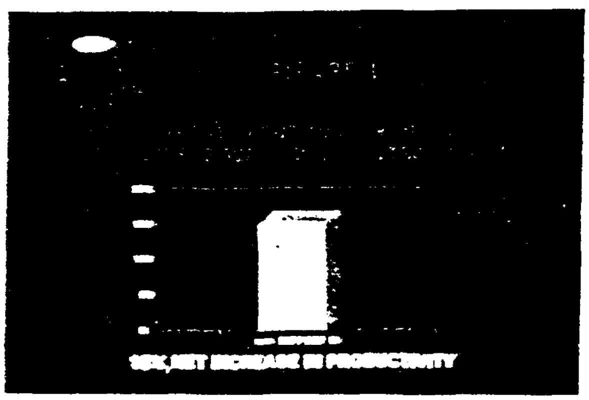
The office tools must be improved, and these tools must be able to work together if all four areas of information handling are to be impacted.



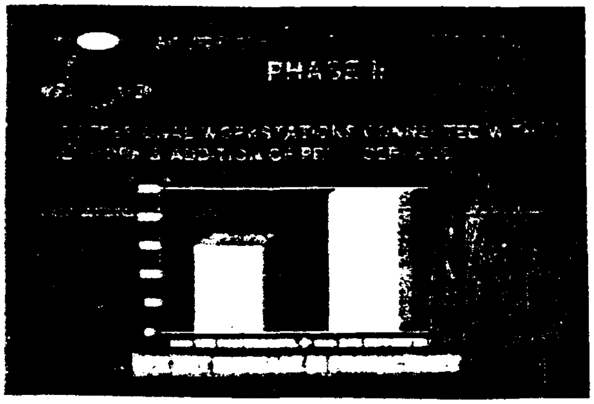
The integration of compatible products and the interoperability of products are available today on Ethernet.



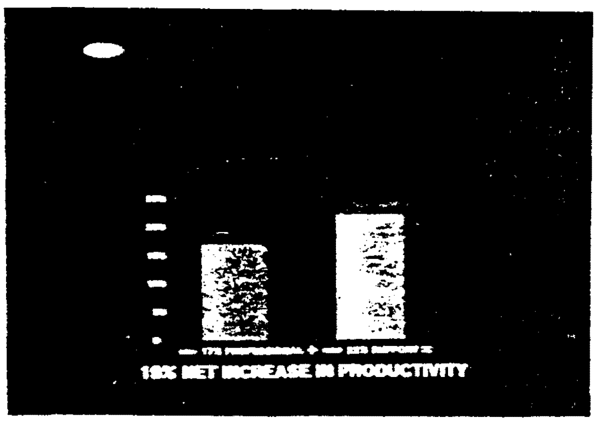
An Ethernet installation in a Fortune 100 manufacturing company showed a $250,000 net savings during the initial year. Because of that immediate realization of productivity increases, we asked Booz Allen to analyze this installation in light of their original 1985 projections.



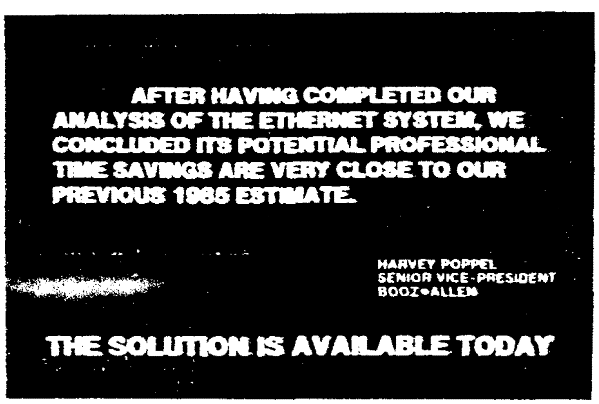
In Phase I of their extrapolation, a 15% increase in productivity could be expected with the support group workstations and file server connected to the network.



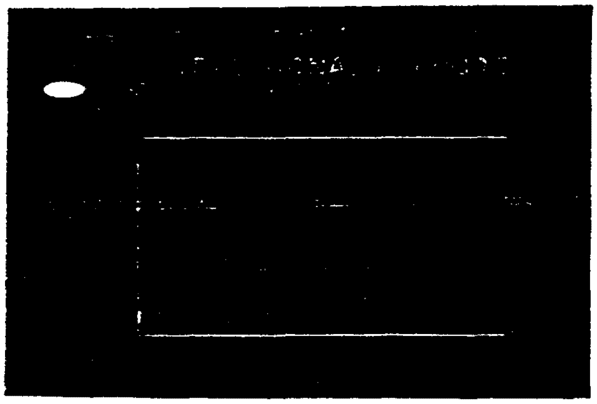
Phase II projects a 17% net increase in productivity with all professional workstations on the net and the addition of print services.



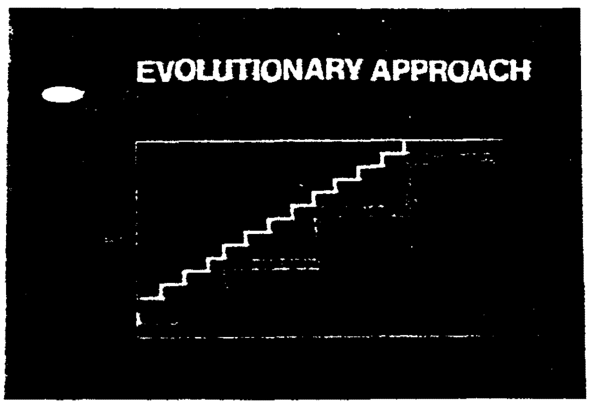
With the further addition to the net of electronic typewriters, personal computers for managers/executives, and communication services, an additional 19% net increase in productivity was realized.



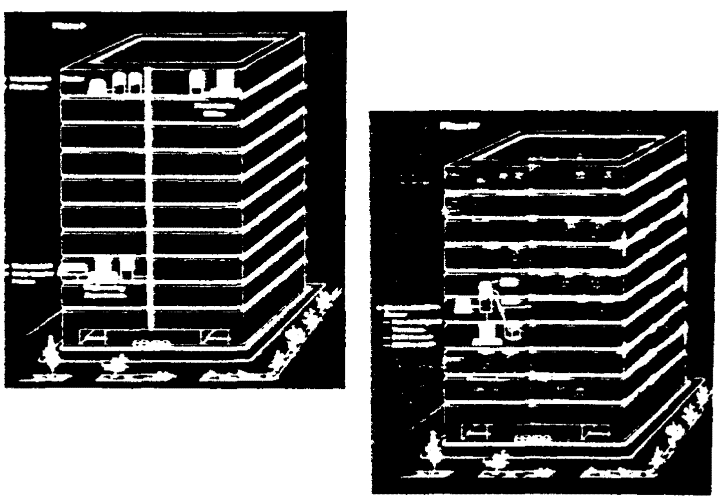
This analysis clearly indicates the technology to increase office productivty exists now and is being utilized by Ethernet customers today.



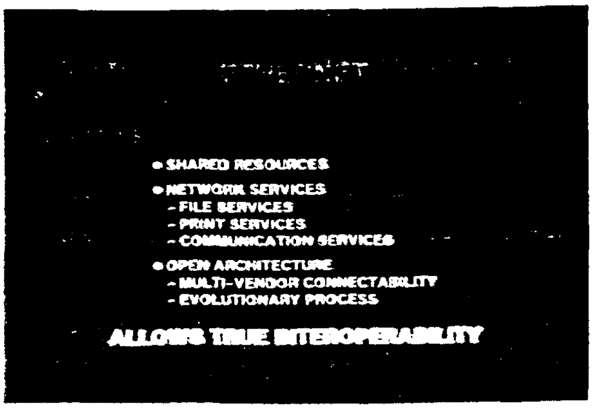
The traditional approach to "buying" technology was that the user had to buy giant pieces at a time; when the user began optimizing all of the capabilities, another big piece of equipment was purchased. The classic example of this is the mainframe computer.



Ethernet, however, allows the user to start small, one work group or a department at a time. It is not necessary to have a comprehensive, long-term automation plan to begin automating an office.

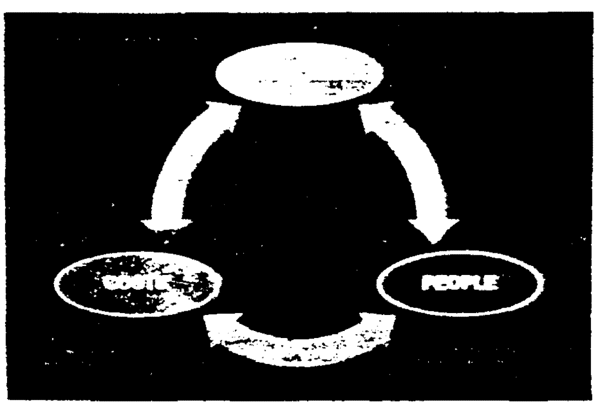


A sample phased Ethernet installation.



Ethernet provides generic local area network capabilities and with the published availability of higher protocol specifications, interoperability is a reality.

By providing integrated services and capabilities and interoperability with "foreign" products, Ethernet allows productivity improvements for **all** user groups.



Ethernet has a proven history of providing the integrated services and capabilities, the true interoperability necessary to increase office productivity for all user groups.

DISPELLING ETHERNET MYTHS

WILLIAM C. LYNCH

MANAGER, TECHNICAL PLANNING

OFFICE SYSTEMS BUSINESS UNIT

OFFICE PRODUCTS DIVISION

XEROX CORPORATION

Often when some new idea, some new concept is introduced you often hear a list of stories that are told about this new concept. The stories are repeated often, embellished and after a time, they become believed by everyone. What I am going to discuss with you today are some of these types of stories and embellishments, the myths we have heard that have collected over the last two years that we have been working on Ethernet. Frankly, some of them are rather remarkable. You have heard or will hear from the Intel speakers where we are in the program, that we are close to having a chip. As you well know, Xerox has been delivering Ethernet products. We are on our way to seeing Ethernet become a recognized standard. So I'd like to clear up as much as I can about the misconceptions that exist about Ethernet. What I want to do is go through several of these items, tell you what the myths are, and then what the truth is.

Before I do this, I would like to remind you of what the Ethernet specifications are. Ethernet exists in the two lower protocol levels of the ISO model, the physical and data link layers. They meet the ISO architecture, and it is an open architecture. We published these specifications jointly with DEC and Intel in September of 1980. The specifications of Ethernet define the electrical and mechanical rules so that when you connect machines together that meet the specifications, they work. We specify in Ethernet a protocol called CSMA/CD. Its purpose is to multiplex data between machines.

This brings me to the first misconception: Ethernet costs too much. It costs too much to attach a terminal or a device to the network. You heard the presentation by Intel which covers that item.

Another myth about Ethernet is that it performs poorly under heavy load. DEC has presented a paper regarding performance, and perhaps some of you have also seen reports of our own Xerox experiments that were reported a year or two ago on the 3Mb prototype. The traffic patterns are about the same between the 3Mb prototype and today's 10Mb Ethernet. You have heard that in all performance evaluations, Ethernet performed suberby.

The next misconception I'd like to dispel is the idea that Ethernet has limited bandwidth. After many years of study and experience with Ethernet, we found that there is more than enough band width to handle applications that we perceive for the next ten years.

Some opponents to Ethernet say it is statistical rather than deterministic. The first thing to understand is the term deterministic. How is it presented to you? It is presented to you in the following way: I can guarantee that when I have a message to deliver to you that I can give you an upper bound on when that message is going to be received by you at your terminal or work station. DEC has gone through a very detailed presentation on the Ethernet performance. The really important issue is waiting time. Does the message get there fast enough for the application that we want? If it gets there fast enough, it doesn't really matter if it is statistical or deterministic. It ges there. It does the job. That is the issue I am bringing out here.

What you are told is that because Ethernet is statistical, it can not do certain things. I can give you an example of token ring and busses, etc., that are also statistical. For example, a token bus. As long as there are, say, ten workstations on my bus, and I'll never get any more than ten, and everybody is sending the same stream of traffic all the time, that is, a terminal user at a constant rate pushing the same button - the return or enter button - what I will get is a stream of data coming out from everybody. And if nothing breaks and nobody else wants to get on the bus, I can guarantee there will be a response time that is fixed. But now you've got to solve the following problem. You come in to your office in the morning and want to check your mail. All the people in your office probably do not come in the same time every morning. You come in and flip the switch and ask, "What is my mail today?" But then, what about messages that are coming back to you? I don't think that a deterministic process will handle this situation, either.

My point is there is nothing deterministic in this business. The reason these channels are shared is to take advantage of the fact that what you are going to do is probabilistic. But you can't predict in advance. Otherwise, you would give a fixed pair of wires for every terminal that is going to use the network. If you really want that - you can do it.

I want to show you this slide to give you an average to look at. These are numbers, response time. Response time is waiting time plus transmission time. You already heard that transmission time for Ethernet is a small value. Waiting time is a probabilistic number: some number of tenths to milli-seconds. If the question really is, "Can I guarantee that my message will get there in a tenth of a second if I use Ethernet?", the answer is: Yes. Data can get from sender to receiver in a tenth or hundreth of a second, almost always. Conservatively, data can get there in one-hundredth of a second, upwards of 90% of the time.

On the other hand, what does micro-second response time mean? It means that you have this packet that if only a few bits wide. What can you get in a few bits - maybe the preamble.

Another myth that we hear is that we have put too much intelligence into the terminal. What is happening now is that VLSI is allowing us to inexpensively locate a lot of intelligence at the work station. Let's take advantage of that. Improve performance and improve capability. More and more of this pattern is showing up. As a matter of fact, the trend is to put evem more of it in the work station. You will find smarter, not dumber work stations in the future.

Next Myth: Ethernet protocol has no error control. Truth: Ethernet uses the Autoden 2, 32 bit FCS for error detection. Next truth: RS232 (X21 bis) has the same error control. This is exactly what the standards have been using for years. You have heard at least four of the presentations talking about layering. What has happened is that with computer networks, we are allowed to start layering some of these functions, with the error detection at the lowest layer and error correction at the middle layers. Xerox provides error correct in the Transport Layer.

We have found that there are certain applications that don't require error control. Time of day, for instance. We send a clock down the network every few milliseconds. I am sure that I don't want to have to retransmit that because I lost it. I know another one is coming later. What we have done is take advantage of the kinds of things that you do on a network. You don't just send files, you also send control information from time to time. Or just plain information. We have a packet we call "Breath of Life" - it sort of floats around to initiate down-line loading. The communication server just sends it out and says here I am, does anybody want me? If it's lost, it's lost; another one is coming.

Myth: The Ethernet protocol (CSMA/CD) does not work with any other media. We have seen since at least 1969 CSMA/CD on every media that you can think of. Xerox implemented CSMA/CD on a fiber optic technology. We have had a Fibernet experiment running since around the 1977. There are a number of other vendors outside of Xerox looking at putting Ethernet on fiber. There are going to be differences in physical architectures, but a CSMA/CD takes advantage of multiple access. Second, broadcast or broadband technology has been using CSMA/CD technology for a long time. Miternet is an example. Of course, Wang's Wangband is CSMA/CD - it's the same protocol. That says something. We must be right. It is a basic way to tie computers on a network and communicate in a multi-access environment. In the Wang or Miternet implementation - in the first two levels of the protocol - what is different is that the physical channels, the physical implementations, are going to be different.

Question: How do you configure this local network called Ethernet? A misconception is that Ethernet has limited topology and topography, that there are only a few ways you can configure Ethernet. That is utter nonsense. We have Ethernets in high rise buildings, and in single floors. The installation we have of the Ethernet, for example in Palo Alto - at my office - is essentially a single cable - a snake between the floors. One Ethernet. We have installations where on each floor, there is a backbone Ethernet going down an elevator shaft and a single Ethernet on every floor. I think I know the source of this particular myth: If you look in the specifications, it says 100 taps per segment and then you look at the next picture - a maximum of 1500 meters cable in a network - linear -difference between two stations. And people start counting because the configuration you see in the specs is 500 meters, 500 meters - where do you get the 1000? It must be limited. Therefore, you can only get 1500 meters between stations. Again, that is nonsense. You can have at least three dimensions of a network topology.

Myth: Ethernet has a limited number of attachments. Truth: Ethernet has 1,024 tap locations. Each of these tap locations can interface several terminals. You can have just about any application to topology that you want. You have seen already from the DEC presentation that 2000 terminals is not a problem. Actually, more than that is not a problem. Referring back to my deterministic slide, remember that response time is in such a small time value range that the applications never see this response time. You sit at the terminal, working out what you are going to do next, which process to serve next. It will always take you longer to do that than it takes Ethernet to send the message on the network. So you are allowed all kinds of applications.

Myth: Application coverage is limited. This has the flavor of, "Ethernet can't do factory applications." That is, of course, not true. Ethernet has been used in factory applications. As a matter of fact, right now there is an Ethernet network in a manufacturing facility in Dallas that is wrapped around a power distribution cable. And every now and then the big switch on the wall goes "girchunck". And the packets keep going and there is absolutely no problem. This takes us back to response time: a tenth of a second - no problem. Ethernet can be used in some applications of manufacturing. Ethernet is planned to be used and is being used outside of the office, dispelling the myth of Ethernet being suitable only for office applications.

These next issues have to do with the acceptability of Ethernet itself, the Ethernet protocol, how has it been received in the public, how has it been received in industry. The first myth here, that Ethernet has limited acceptance by the business and communities, is nonsense. You have already heard that at least 22 vendors that have publicly declared their intentions to be compatible or make components with Ethernet. There are more that are not yet public. There are numerous companies who intend to and who are investigating supporting Ethernet in the field as a product. This slide shows the types of products or components that are currently offered by non-Xerox vendors. You can see that the entire spectrum of the things that you need to do with an Ethernet are avaiable: transceivers, controllers, controller chips, cable, systems, compatible stations. For instance, there are at least five transceiver vendors world-wide. By the way, there is an overlap in this list. Some of the vendors that are making transceivers are also making controllers. There are at least nine controller vendors. We have four chip vendors, three cable vendors. The system vendors consist of people who have decided to supply complete Ethernet compatible systems. They range from software, computer-based systems all the way to the entire network. And the compatible station vendors are people making smart terminals to talk on Ethernet, which range from highly talented terminals or work stations to fairly low functions but direct connections.

All this reinforces Dave Liddle's earlier statement: our goal is interoperability. Open up the marketplace and let other vendors get into this. For example, Xerox does not make transceivers. We buy transceivers. Xerox is into the chip business. We buy chips. So it is important to us that we have received this wide acceptance of Ethernet. The point is that the acceptance in the community has been very high, very wide, and very complete where companies have committed money, time, and people to support Ethernet.

Finally, the last myth: Ethernet is just a development project and will never be implemented. We actually had a question asked of us recently -when will the first Ethernet be installed? The answer is, of course, yes, we have had Ethernet commercially installed for over one year. Two major companies that have discussed their experience with Ethernet to the U.S. press are TransAmerica and Arco. When I made this slide, there were 35 other networks that were up and running. There are another 50-60 networks in different stages of installation and operation. This does not count networks we have inside Xerox.

It is true. There is an Ethernet. It works, and it works reliably. What I have tried to do was go down the list of what I consider the really crucial myths, discuss them, and tell you the facts that dispel these myths. Ethernet itself is certainly no longer a myth - it is a reality.

Thank you.

ROBERT S. PRINTIS

Dr. Printis was born in Washington, D.C. He received a B.S. in Electrical Engineering from Howard University, an M.S. in Electrical Engineering from the University of New Mexico, and a Ph.D. in Electrical Engineering in 1974 from the University of Maryland with major research interest in systems theory. He is a member of Tau Beta Pi and Phi Kappa Phi honor societies.

Printis conducted research in systems theory and applied mathematics, with specific interest in applications of systems theory to large scale systems, at IBM Research Center in Yorktown Heights, New York, between 1973 and 1977. He spent a short time at Bell Laboratories, in Holmdel, New Jersey, in 1977-1979 working in the ACS project on network control and network management protocol development. He has been employed at Xerox Corporation since 1979, working in the area of communications networks. His work assignments included the design and development of the communication protocols for Xerox' long-haul data communications networks.

Dr. Printis is currently Manager, Network Standards in Xerox Corporation's Office Product Division, Systems Development Department in Palo Alto, California and is developing network management standards for local networks.

ETHERNET EDITORIAL SEMINAR

DISPELLING ETHERNET MYTHS

On September 30, 1980, Digitial Equipment Corporation, Xerox Corporation, and Intel Corporation published version 1.0 of the Ethernet Specification. Despite the support of the standard by a wide range of institutions, both commercial and academic, there is still some confusion expressed about Ethernet, its design, and its operation. In today's presentation, several of these misconceptions will be discussed, using the experience gained from the installation and operation of the network since 1975 for the experimental network and since 1980 for the commercial 10Mbit version.

This discussion is separated into six areas of concern: design, configurability, application coverage, acceptability, performance, and costs of Ethernet. Any local area network technology must deal with these concerns. We will discuss how Ethernet addresses these issues. The presentations by DEC and Intel will have addressed the issues of performance and costs. The Xerox presentation will address the remaining four areas.

Robert S. Printis
Manager, Network Standards
Office Products Division
Xerox Corporation

DISPELLING ETHERNET MYTHS

Robert S. Printis
Xerox Corporation
Office Products Business Unit
Palo Alto, California

MYTH CATEGORIES

- DESIGN
- CONFIGURABILITY
- APPLICATION COVERAGE
- ACCEPTABILITY
- PERFORMANCE
- COSTS

DESIGN

- ETHERNET HAS LIMITED BANDWIDTH (capacity)

  -- 10Mbit per second capacity more than adequate for local computer network applications envisioned for next ten years

- ETHERNET PROTOCOL IS "STATISTICAL" (rather than 'deterministic")

  -- Definitions

    -- Waiting Time -- elapsed time from the time that the packet is ready for transmission until the packet successfully begins transmission.

    -- Transmission Time -- propagation time of the packet on the medium.

    -- Response Time = Waiting Time + Transmission Time.

    -- Deterministic System -- Waiting Time known to a fixed upper bound. Therefore, the Response Time is bounded, under normal operation of the channel.

    -- Statistical System -- Waiting Time's upper bound known with probability.

--Response time requirements met by Ethernet

.100 sec	Yes, unless system is broken
.010 sec	Almost always, unless system is broken
.001 sec	Misses this requirement if long packet (Maximum packet size = 1518 bytes)
.0001 sec	Possible for small packets
.00001 sec	Forget it

--For a point of comparison, 9600 baud line

.100 sec	no, if message exceeds 120 bytes

- PARTITION OF ETHERNET FUNCTIONS PLACES TOO MUCH OF THE COMMUNICATIONS RESPONSIBILITY ON THE 'TERMINAL'

  -- Reduction in price-size-performance due to VLSI permits introduction of more communication function in the station.

  -- Permits the design of more efficient communcations.

  -- The direction of the future is to place more, not less, communications in the station.

- ETHERNET PROTOCOL HAS NO ERROR CONTROL FOR DATA TRANSMITTED ON THE CABLE

  -- The situation is the same as that in RS232C data communications.

  -- The Autodin II 32 bit FCS is specified by the Ethernet Specification.

  -- It is common in computer communications networks to place error recovery in the transport layers of the communications protocols.

- ETHERNET PROTOCOL (CSMA/CD) CANNOT WORK ON OTHER MEDIA

  -- Fiber Technology -- Fibernet

  -- Broadband Technology(coaxial cable)

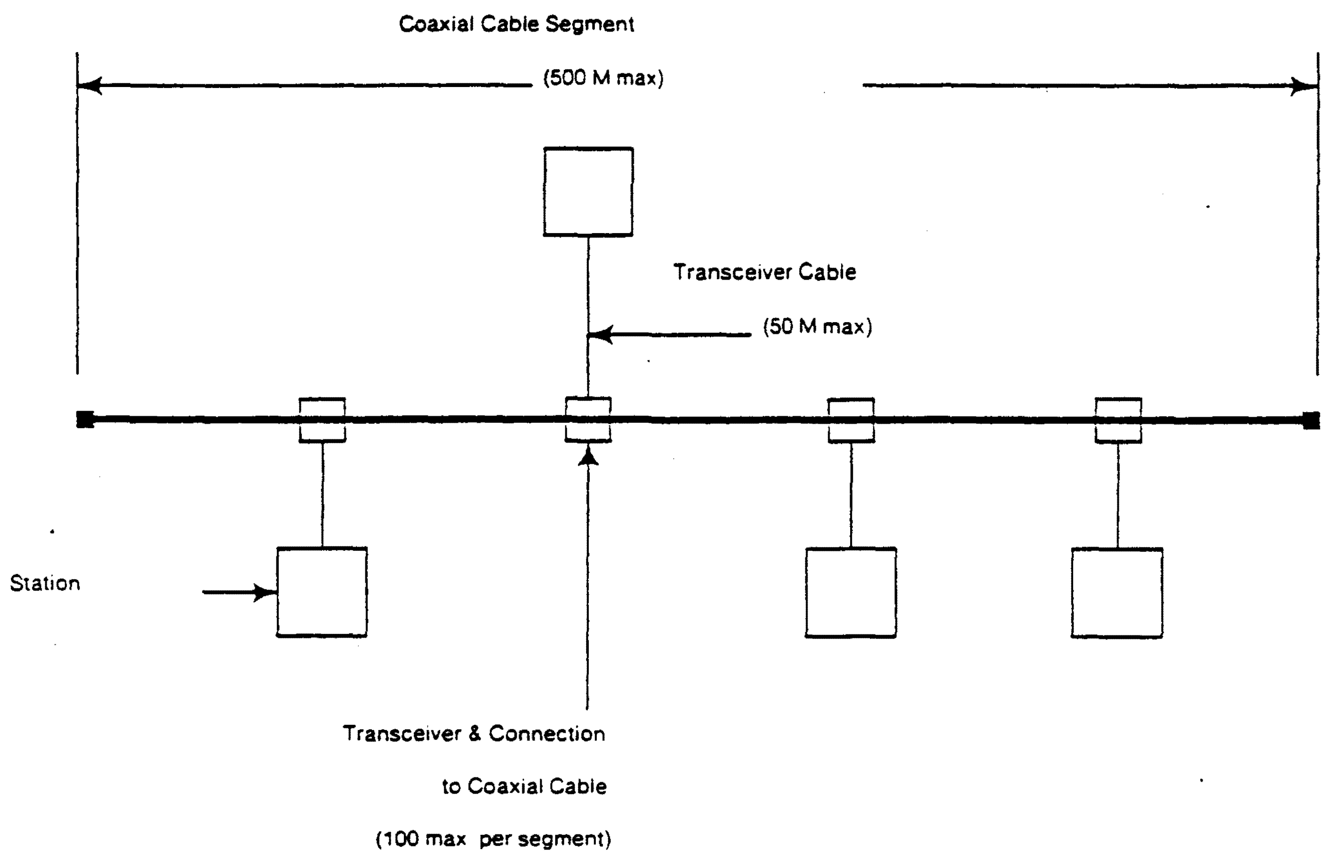
    -- Mitrenet

    -- Wangnet's Wangband

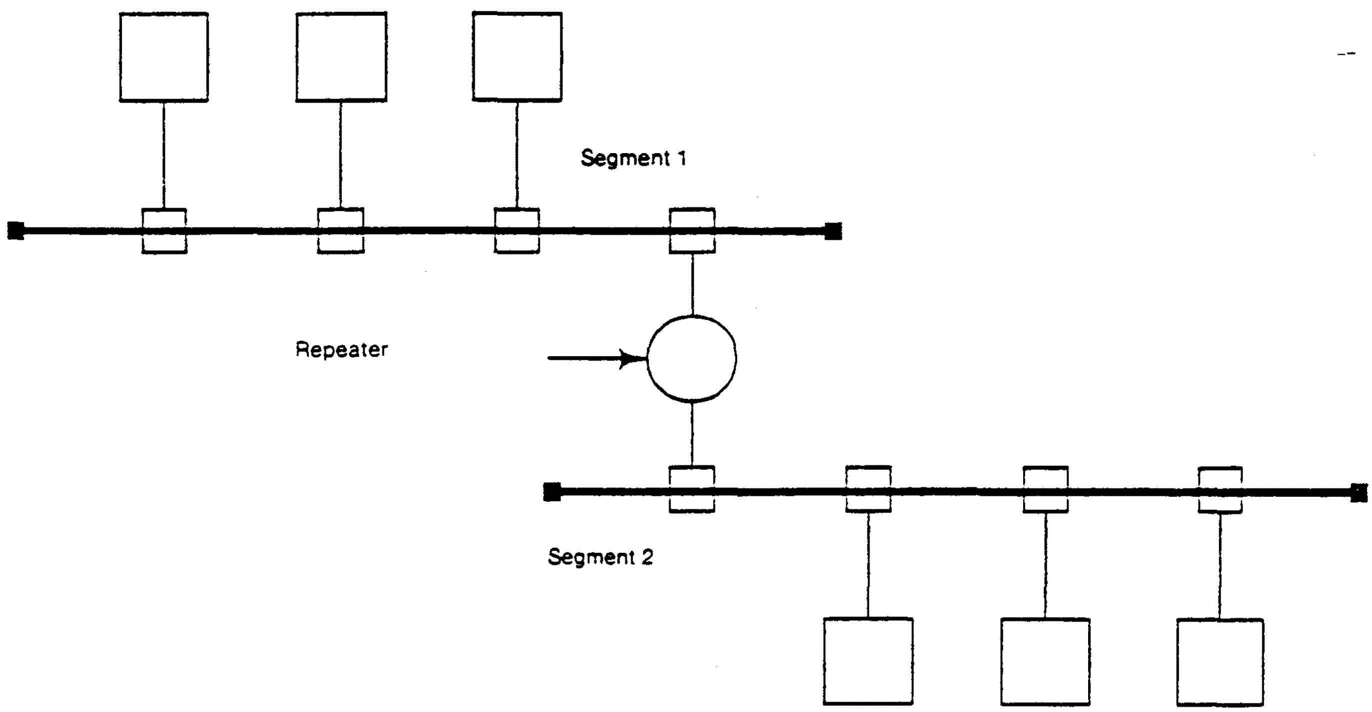
  -- Layering permits this -- must define physical channel interface, but protocol is media independent.

CONFIGURABILITY

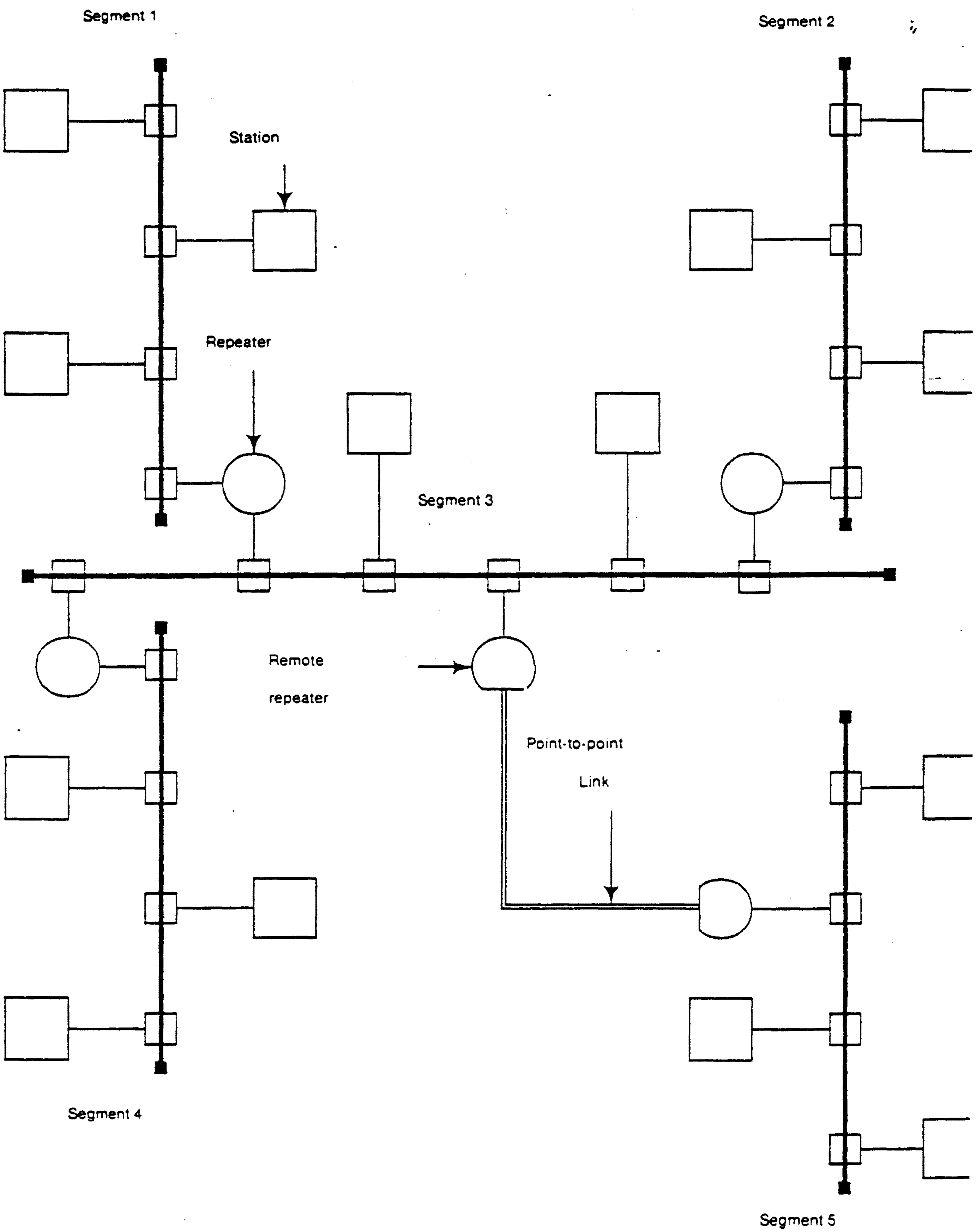
- ETHERNET HAS LIMITED TOPOLOGY AND TOPOGRAPHY



Small Ethernet Installation



A Medium-scale Ethernet Installation



Large-scale Ethernet Installation

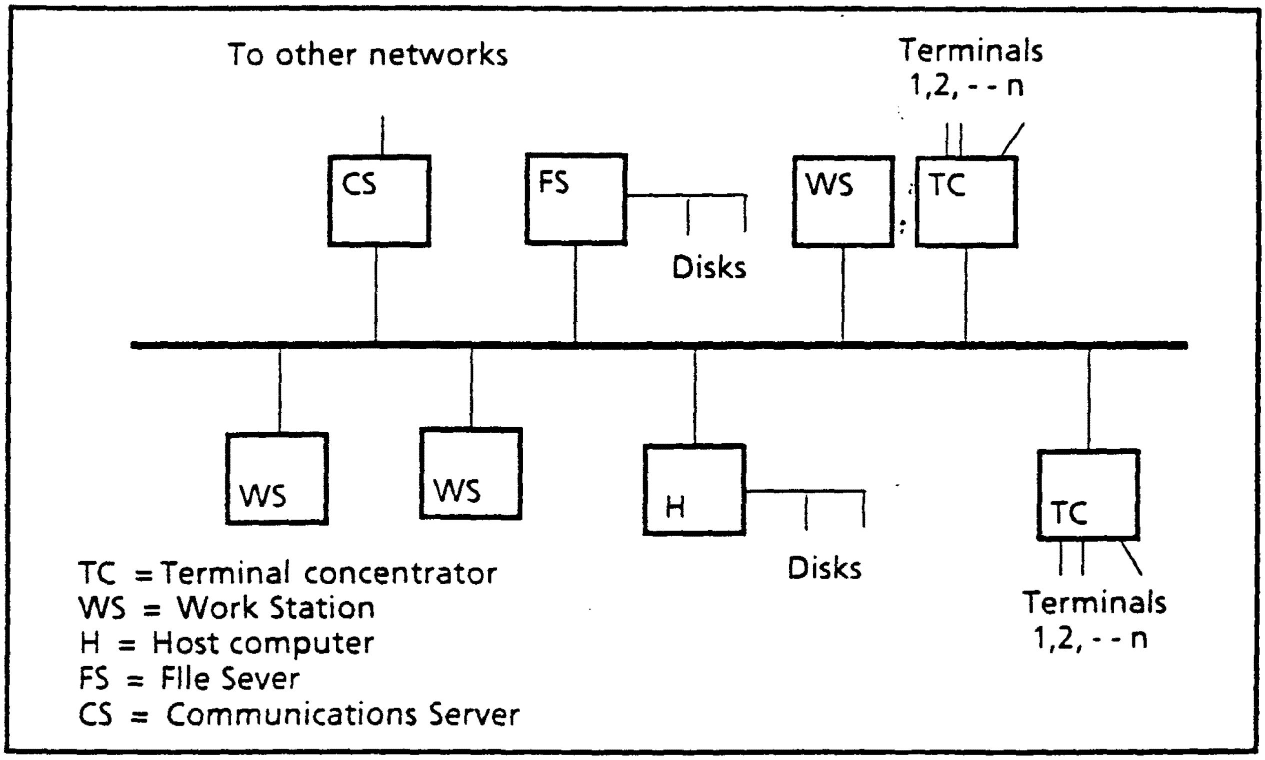
- ETHERNET HAS LIMITED NUMBER OF POSSIBLE ATTACHMENTS

  --1024 attachments possible, i.e., tap locations

  --Several stations can share a tap.

  --For example, with the Xerox 873 Communications Server -- 8192 RS232C Ports

  --Numbers of users depends upon applications



To other networks
Terminals
1,2, - - n
CS
FS
WS
TC
Disks
WS
WS
H
TC
Disks
Terminals
1,2, - - n
TC = Terminal concentrator
WS = Work Station
H = Host computer
FS = File Sever
CS = Communications Server

APPLICATION COVERAGE

- ETHERNET CANNOT BE USED IN REAL-TIME PROCESS CONTROL APPLICATIONS

--If "real-time" means message delivery by 0.1 sec then, may use Ethernet

-- Ethernet can be used in some applications in manufacturing.

-- DEC and Intel briefings give more examples of Ethernet in other enviroments.

- **ETHERNET IS SUITABLE ONLY FOR OFFICE APPLICATIONS**

  DEC intends to use Ethernet as local network for applications which include the office and traditional data processing

  Many of the companies licensed to use Ethernet are not in the office automation business, but are data processsing companies

  No technical reason which restricts use to office applications.

ACCEPTABILITY

- ETHERNET HAS HAD LIMITED ACCEPTANCE BY THE BUSINESS AND TECHINICAL COMMUNITIES

Twenty two vendors announced their intention to provide compatible systems

Transceivers	Five vendors
Controllers	Nine vendors
Controller Chips	Four vendors
Cable	Three vendors
Systems	Ten vendors
Compatible stations	Ten vendors

- **ETHERNET IS JUST A DEVELOPMENT PROJECT AND WILL NEVER BE IMPLEMENTED OR INSTALLED**

  TransAmerica

  Arco

  35 other nets