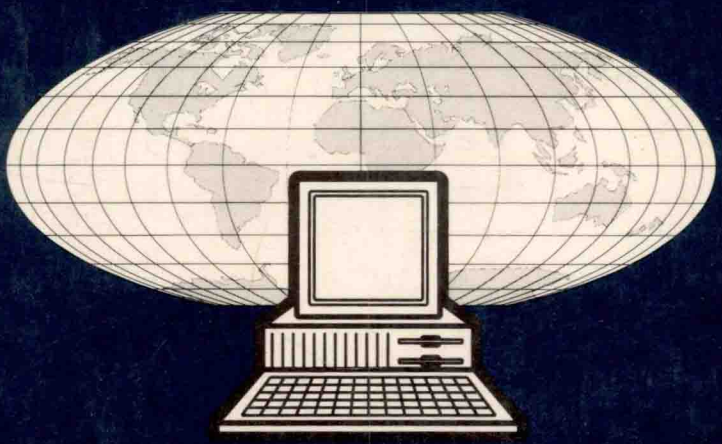
ISSUES AND PROBLEMS IN COMPUTER NETWORKING



BILL HANCOCK

Issues and Problems in Computer Networking

Bill Hancock

amacom
American Management Association

This book is available at a special discount when ordered in bulk quantities. For information, contact Special Sales Department, AMACOM, a division of American Management Association, 135 West 50th Street, New York, NY 10020.

Library of Congress Cataloging-in-Publication Data

Hancock, Bill, 1957–
[Network concepts and architectures]
Issues and problems in computer networking / Bill Hancock.— 1st AMACOM pbk. ed.
p. cm.
Reprint. Originally published: Network concepts and architectures. Wellesley, Mass. : QED Information Sciences, c1989.
ISBN 0-8144-7738-0
1. Computer networks. 2. Computer network architectures.
I. Title.
[TK5105.5.H36 1990]
004.6—dc20 *89-48314*
CIP

First AMACOM paperback edition, 1990.
Originally published as Network Concepts and Architectures *by QED Information Sciences, Inc., 1989.*

Printing number
10 9 8 7 6 5 4 3 2 1

Preface

The idea for this book resulted from one of my very frequent trips to the bookstore. Upon inspection of the various titles on networking and data communications, I failed to find any straightforward books on computer networking or descriptions of the various types of network architectures. While it is true that there were a few titles on the subject of networking, many were on how to use the various timesharing services or theoretical aspects of data communications. I could not find any books on general network concepts. Worse, I found very little intelligible information on the various types of network architectures and where they are applicable in the area of computer networking. Not that there are not good books on networking—there are. Many, however, are too general or too technical. There are few in the middle.

During further conversations with various friends in the industry, I was told that they were having a tough time explaining to their management about how computer networks and computer network architectures work and the difficulties involved with such projects. Many managers and nontechnical personnel do not fully appreciate the issues and problems involved in computer networking including design, preparations, installation and many other issues having to do with computer networks. Not that they don't want to know—the problem is where to go for information.

This book is not the end-all solution for understanding network concepts and architectures. Its purpose is to enlighten the reader on concepts, buzzwords, and topics involved with various aspects of computer networking.

I hope that you enjoy this book. If you have some comments on how this book could be improved, please feel free to contact me at P.O. Box 13557, Arlington, Texas 76094-0557.

Bill Hancock

Acknowledgments

In any undertaking, there are always many individuals who help keep things on track.

First, I need to thank my reviewers: Bill Brindley of the U.S. Navy, Mike Heagney of Marathon Oil International, Sandy Traylor of Target Systems, Jim Ebright of Software Results Corporation, and Ken O'Mohundro of Able Computer. Their comments and insight of the various chapters reviewed was most valuable and enlightening.

I also need to thank Brindley McGowan of Hughes Aircraft for technical support and friendship during the production of this book.

As always, my friend and colleagues at ERI Training, Marty Davis, Bob Branchek, Bob Russo, Fred Sanders, Jill Davis, and Debbie Amiel were their usual highly helpful and professional selves, critical to the production and marketing of this book.

Finally, my thanks to those of you who have purchased this book. I hope that you will find it enlightening, entertaining, and useful in your network endeavors.

Contents

Chapter One

What Is Communication?

INTRODUCTION

"Communication is, son, what you will get across your posterior if you do not clean up your room!"

How about this one:

"I mean, like, you know, like, there was just no communication between us, man. We were on this really weird trip and just couldn't seem to reach the same line pattern; y'know what I mean?"

Of course, there's always this one:

"Thank you for your letter of 9 October. In response, we would like to ascertain the relationship, through differential and qualitative thought, of our mutual and underlying interests regarding the manner in which your office plans to concatenate and otherwise perform an overall and lasting modification of corporate guidelines to reduce the amount of unnecessary and verbose correspondence between office hierarchies."

Finally, we've all heard this one:

"The check is in the mail."

What we have just seen is proof that humans do not communicate properly. In the course of everyday life, we all are faced with instances where communication "breaks down." We are then forced to consider alternative means of communication such as different patterns of speech, reiteration

of verbal communication, or even to communicate our desires and displeasures through an alternative communications media, such as physical actions (sign language, body motions, physical violence). This book is not about how people talk to people; more it is how people talk to machines (computers) and how machines talk to other machines. It is important, however, to understand that people build machines, sometimes in our own image and likeness. As a result, many different types of computing systems and data communication procedures have developed, often leading to a breakdown in communications.

The Human-Machine Experience

There was a time when communicating with computing machines was difficult. Anyone desiring access to the all-powerful computer had to first fill out countless request forms (computer cards) and present them to the computer's representative, the batch stream operator. The operator would then tell the luckless user that they would be ready soon. How soon? "As soon as they are ready!"

Oh.

After waiting a reasonable amount of time, the user would return to the all-powerful computer and ask the operator for the data requested. One of two responses seemed to be the most common in those days: "Sorry, it's not ready yet," or, "What job?"

Realizing that batch computing was slow and produced results that were often not timely, the term "interactive systems" soon became the cry of system purchasers and users worldwide. With an interactive system, users would have use of a terminal to communicate directly to the all-powerful computer and have it execute programs interactively with the user. This led to many users being able to perform work more quickly and efficiently, as well as giving them control over the all-powerful computer. But, with all good things, problems must arise.

More terminals began to appear on systems, meaning that more data was being required more often by more users for management decision making and for everyday production. Soon it became apparent that the computers purchased would not be able to handle the amount of terminals that were being added to them. This led to a general user revolt, in some companies, because there were "never enough terminals" to do work on and the computer system could not support more.

Users were then lured by the prospect of the personal computer (PC). The idea of computing on one's desk was very attractive as was the ability to not have to beg for computer time from the all-powerful computer or surly operators. With PCs, computing could be done by the user, for the user, and without, necessarily, the help of the computer types. With the introduction of user-usable (I avoid the term user-friendly, as most programs are not) programs, spreadsheets, and databases, the ability to effectively use the PC as a serious business tool became viable. Unfortunately, the need did not alleviate the load on the all-powerful computer and a new need surfaced at the same time: getting information between PCs and the all-powerful computer in an easy, efficient manner.

On the other side of data processing were the engineering and scientific communities, who desired the use of computing systems for precise calculations and for other functions such as numerical control, process control, computer-aided design (CAD) and computer-aided manufacturing (CAM). Where computing within the commercial environment tolerated some delays in generation and collection of data, in the engineering and scientific environments minor deviations in measurements or time can be critical to operations. As a result, engineering and scientific programs tend to be more compute-intensive (i.e., making exhaustive use of central processing unit resources) rather than I/O-intensive (making exhaustive use of input-output related peripheral devices, such as disk drives).

Most commercial data processing shops utilize the batch-oriented or interactive ("timesharing" — every user shares an equal amount of time)

computing systems, which were designed for I/O operations and also for multiple terminal requests from users. These two types of computing systems are, unfortunately, inadequate for engineering and scientific operations. This led to the development of what is called "real-time" systems. Real-time systems respond to nebulous items called "events." An event could be the completion of an I/O request, and interruption of the system by an analog collection device, power failure, etc. Essentially, it is important to know that an event is, simply, anything that has the potential of changing the status of the system. Through the use of event-oriented systems, engineers and scientists could now design systems to manage operations that required real-time (immediate) response and action: manufacturing, assembly, chemical reaction control, nuclear reaction and fission control, process control, and many other items. To develop such systems, however, the scientific and engineering personnel needed much more sophisticated development environments than the classic commercially-oriented developer. Problems such as display graphics, flow mechanism, exacting display graphics for computer-aided design, simulation of electrical circuitry and many other high-power consumption needs forced the development of engineering workstations.

Engineering workstations are used for a variety of tasks, but usually in the areas of computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided engineering (CAE). With the influx of CA-oriented workstations has also come the problem of high-speed communications and sharing of data and programming. CA-oriented stations tend to use very powerful PCs, minicomputers, or superminicomputers to provide the compute power necessary to handle the graphics, math-intensive computations, database access, and communications needs of the modern CA-oriented workstation. Along the lines of the CA-oriented workstation is a new line of workstations called computer-aided programming (CAP) stations that are used by programmers and systems personnel to produce sophisticated computer applications through non-traditional means. CAP stations can provide interactive Artificially Intelligent (AI) environments for AI programming, graphic display meta-environments for graphics programming, programming tools and programmer-friendly symbolic

debugging systems. As with the commercial environment, however, the need for communications between engineering workstations and the all-powerful computer (as well as amongst each other) are increasing and the technology is not keeping up with the demand.

As it can be seen, the human-machine interface can be very complex indeed. The way that a personnel specialist communicates with a computer system and the way that an engineer communicates with a computer system can be radically different due to the nature of the job at hand and also the type of computer that is used to perform the job.

The Machine-Machine Interface

If you think that getting humans to communicate with each other is tough, try to get two dissimilar computing systems to "talk" to each other. It is very similar to an American attempting to communicate with a Chinese without either having knowledge of the other's language, customs, and idioms. Needless to say, the attempt at verbal communication is doomed to be worthless, and gestures between the two can sometimes do more harm than good. For example, consider the gesture that most Americans use to indicate the number two — the index and middle finger extended, the rest of the fingers folded, usually with the back of the hand towards the person the gesture is directed at. For most of us, this indicates two (2) of something: hamburgers, french fries, soft drinks, hurricanes, H-bombs, demolition derbys, ad infinitum. Now, just for excitement, travel to Scotland, walk into the local pub, and order two beers using the same gesture. Do not be surprised if the bartender attempts bodily harm on you! The same gesture that indicates two of something in the United States means, basically, "up yer kilt, laddie" in Scotland. The point is that frequently, we humans can communicate with sign language and get results (desirable or undesirable), even though we may not understand the language of the other person we are attempting to communicate with.

Computing systems do not have the luxury of being able to gesture to other computer systems when the language barrier exists. Computers, you

see, are very dumb. After all, all they do is add. Ah, but you say, that's not true. Of course it is. Subtraction is the complement of addition, multiplication and exponentiation are just a bunch of adds, and division is subtractions. Even though the basic unit of work (addition) is the same in virtually every computing system, the results are different and the methods of communicating results to peripheral devices and humans tends to be unique to the computing system that created the results. Since computers cannot gesture very well, it is highly unlikely that one computer, say an IBM system, that expects to communicate in a certain manner will be able to communicate with a dissimilar computer (e.g., an Apple II).

PUTTING IT ALL TOGETHER

What does this mean? Essentially, for computers to communicate with each other, they must have something in common, like a "language" or, more specifically, a protocol. Just like there is protocol at a very formal dinner engagement that we humans might attend, computers expect a very formal protocol so that they can communicate with each other on an efficient basis. It is the job of the engineers and programmers to develop protocols so that computers can "talk" to other computers, terminals, peripherals, and humans. In case you have not met one before, those are the people who sit in their offices and cubicles, speak in unintelligible three and four letter words that no one but engineers and programmers understand, think that users are dumb, consume great deals of caffeine-laced coffee, work odd hours, and are generally considered to be somewhat strange by most users. Most of them will not byte (pun intended) if they are carefully fed massive quantities of junk food and technical publications. They are the creators of the all-elusive protocol, each with their own method of implementation and their own ideas of "what a protocol should be." This is why there is such a veritable plethora of protocols in existence and also why so many of them are incompatible with each other. Still, there are a few engineers and programmers that understand more than one type of protocol. This special type of person is the one who develops the "black box" or protocol emulator that is used for computer-computer communications. The emulator is usually a device or software

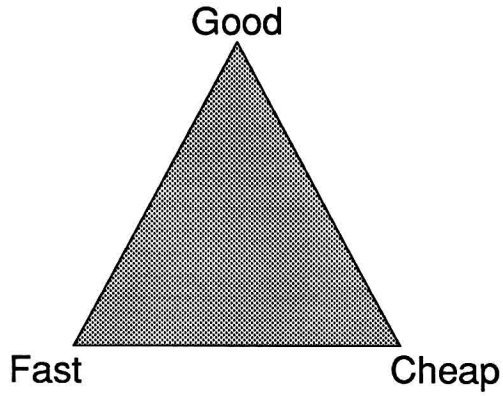
program(s) that is used to "translate" one protocol into another. The idea is similar to the use of an interpreter that speaks English and Chinese. Now the Chinese can communicate with the American and the American with the Chinese. With the use of the interpreter, the two individuals now have something in common — the interpreter. In computer-computer communications, the thing in common would be the protocol emulator or the device that was developed to "interpret" communications between the two machines.

Conclusions and Warnings

It is obvious that communications between computing systems are similar in many respects to interpersonal communications between humans. There is a certain, set procedure in which communications are established, how long they continue, and (usually) a means to terminate communications. While humans communicate in languages, we frequently resort to gestures to emphasize our points or to clarify our communication to other humans. Since computers do not have this luxury, they have to be doubly specific when communicating with each other. Also, just as we know when we do not hear something quite right, we ask for it to be repeated. Computers have to do the same. We can therefore conclude that machine-to-machine communications are similar, in many respects, to human communications, the exception being that they are less tolerant than humans and cannot be physically expressive.

It is the wise implementer that understands the needs of the users and system before implementation of any computer-computer communications scheme. It also should be noted that communications are not a "quick and dirty" process. Careful thought and consideration has to be given to all aspects of pending implementations. Computers cannot do that; humans have to. This also means that implementation of any communications system will not be without expense: be prepared to reach deep into the proverbial wallet when considering communications.

Finally, consider the following diagram:



The diagram illustrated is called "Truman's Triangle," after its inventor Truman Reynolds. Think of "Truman's Triangle" as the guide to implementation of anything that is data processing related. Its use is simple: pick any two corners, such as "CHEAP" and "GOOD." You will notice that the remaining corner has the word "FAST" in it. What this means is that if you want something good and cheap, it will not be fast. In the example of the corners "FAST" and "GOOD," the result will not be "CHEAP."

Plan your communications implementation carefully. Do not be shocked when costs are uncovered — data communication is NOT cheap. Do not rush into data communications. If communications are planned and implemented correctly, they will save you time and money. If they are implemented incorrectly, do the following:

1. Be prepared to spend more to correct problems.
2. Learn to like communications downtime.
3. Stock up on your favorite ulcer medicine and aspirin.
4. Remove all sharp objects from your office.

Chapter Two

The Basics

INTRODUCTION

Understanding data communications is much like trying to learn a new language and culture at the same time. The language part involves the learning curve the newcomer to data communications experiences when trying to decipher all the bits and pieces of jargon that entwine the communications world — DDCMP, BiSYNC, ATLP's, EIA-232D, OSI, ISO, X.25, etc. Culture is something that is developed over a period of time; it involves learning to UNDERSTAND WHY equipment does what it does, protocols work the way they do, users cause the trouble they do, managers don't understand about the $1000.00 per month MODEM, and the like.

There are people who spend their lives in search of the perfect protocol — the one that allows perfect communication between everything and everyone. Others spend their lives trying to create more abbreviations for communications in hopes of reducing the overall length of communications terms. But, by and large, most of us just want to know enough to get our communications working, enhance them when they need it, keep them from breaking down, and find a way to fix our communications problems when they arise. To do this, we must understand some of the jargon of the communications world and also the basics of how to communicate.

DISTRIBUTED PROCESSING AND NETWORKING

One of the most renown problems in the communications industry is the confusion between distributed processing and networking. They ARE

NOT the same! Distributed processing refers to a managerial technique that is used to distribute a data processing workload amongst several locations or internal corporate divisions. An example would be the use of a mainframe system to perform most of the corporate processing and word processors, strategically placed throughout the organization, being used for local word processing and preliminary data collection. It is important to note that there is no direct communication between the mainframe and the word processing systems; still, distributed processing has been implemented. Computing machinery necessary for corporate operations has been distributed to the appropriate levels and users for maximum effectiveness of available resources.

Networking is the proverbial horse of a slightly different color. Networking is somewhat like distributed processing — the systems communicating may or may not be co-located, and they are probably distributed to the appropriate levels of user necessity. The difference here is that the systems actually "talk," or communicate, with each other. The communication usually involves electronic data transfer of some type over a private cable, leased cable, or public phone line. While the implementation of distributed processing concepts may be cheaper, they do not have the power of a network, or the problems. Networked systems have protocol compatibility problems, line failures, growth problems, and myriad other problems much too lengthy to be discussed at this time. For the purposes of this book, we will not discuss the usefulness of classical distributed processing; rather, we will discuss networks and networking. An understanding of networks and networking will make distributed processing techniques easily apparent.

Network Topology

Network topology is the basic outlay or design of a computer network. Think of topology as the architectural drawing of the network components, which is much like the architectural drawing of a home or building (some are simple, some are very complex). This design can be varied in accordance with the company's needs, but certain base elements to configuring the topology of a network still apply.