IBM

Washington
Systems
Center

Technical
Bulletin

A Perspective on Networking

By J. H. Rutledge

DAPS Code 0911

GG22-9003-00

May 1977

Washington Systems Center

Gaithersburg, Maryland

Technical Bulletin

A Perspective on Networking

By James H. Rutledge

This Technical Bulletin is being made available to IBM and customer personnel. It provides some insight into the issues and complexities of networking. It also discusses how these interact with various implementation configurations.

Please refer any comments or questions on this document to Jim Rutledge, Washington Systems Center. A form is provided in the back for comments, criticisms, new data, and suggestions.

DAPS Code 0911

GG22-9003-00

May 1977

A PERSPECTIVE ON NETWORKING

Introduction

Networking is a widely discussed topic in today's world of information processing. Data processing groups or individuals are often heavily involved in these discussions. Yet many of the concepts, issues, concerns, and interactions of networking are quite alien to those with a traditional data processing orientation. Furthermore, data processors will frequently carry misleading or inaccurate perspectives into their networking discussions. This paper is intended to articulate a data processor's perspective on networking and to explore some of the innate concepts and issues of the networking environment. In accomplishing these goals, many more questions will be posed than will be answered. These questions tend to be those that can only be answered by one associated directly with a particular network. It is hoped that by posing these questions and mentioning parameters of possible answers, some insight can be provided.

A few comments about the assumptions of the paper are appropriate. The reader is presumed to have an elementary acquaintance with teleprocessing concepts and terminology. Before proceeding, I recommend that the reader peruse the glossary to understand the terms used in this paper. Some ideas that are germane are also described. Although the topic is highly technical by its nature, I will endeavor to use non-technical language as much as possible.

As we begin this discussion, it is imperative that the reader understand that we are exploring a general purpose communications network functioning as a utility for its users. The comments made herein are directed to four parties associated with any particular network: the network owner, the network designer, network consultants, and network users. This discussion will be structured around the answers to four questions. They are:

1. What is multiple-CPU networking? Here I will state a simple definition of a network. This is a very important concept because it says so little about the factors and considerations involved.

2. Why do multiple-CPU networking? Here we will discuss the two most common factors providing the initial motivation for a prospective network owner.

3. Are there any unique considerations? Here we are addressing networking compared to other data processing endeavors. There are some significant differences.

4. Finally, what are the major categories of implementation configuration? We will look at many typical networking requirements and consider these in light of actual implementations. Three major categories will be presented and discussed.

What is Multi-CPU Networking?

The problem to be addressed by networking began with the first connection of terminals to computer-based application programs. Quickly, the opportunities for program to program and terminal to terminal communications began to be exploited. From these circumstances, the suspicion arose that telecommunications could be rationalized to a single base case. This leads directly to the any-to-any concept that many prospective network owners envision. A definition and picture of a network are shown on Chart 1.3.0. Both the definition and the picture are important. The definition because it is simple and easily understood. The significance of 'multi-CPU' is that the picture includes more than one box capable of executing instructions, i.e. more than one computer. It does not imply anything about size or type of computer, the presence or lack of an operating system, or anything else about the real computer configuration. The picture is important because it gives us an object that can be characterized, designed, and discussed at length. The picture does represent a general purpose, utility network. The user community may be a corporation, a division of a corporation, or some other group to be served by the network. The network usually has the property of serving user groups not under the control of the network owner. This property has significant implications about the visibility of and the responsibility assumed by the network owner. The difficulties start when individuals look at this network picture and begin to make assumptions and envision requirements about the network services to be provided. These difficulties have their beginnings in many areas. A few of the more common pitfalls are: a failure to appreciate the day-to-day problems of a network, incomplete thoughts about the magnitude of the responsibilities to be assumed by the network owner, unrealistic judgements about what can actually be implemented with the time, people, and dollar resources available.

NETWORKING IS:

Provision of the capability for <u>any</u>

End Use Mechanism

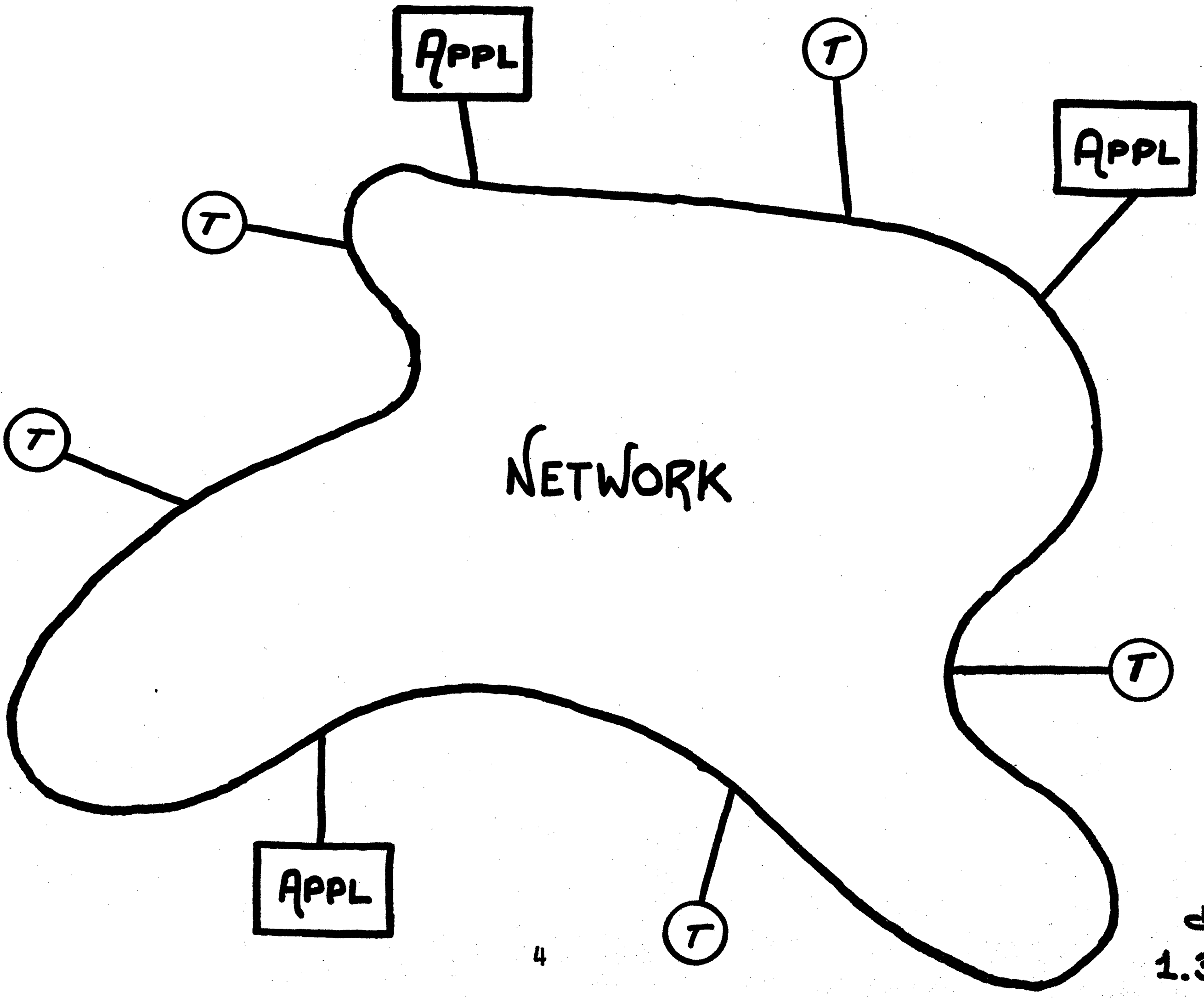
(basically terminals and application programs)

<u>to</u> communicate with <u>any</u> other

End Use Mechanism

in a

<u>Multi-CPU Environment</u>



R
dHR
1.3.0

Why Do Multi-CPU Networking?

There are two characteristics of teleprocessing that initially motivate someone to believe that he might like to be a network owner. The first I call the communications requirement. It is based on the cost of communications services from a carrier. The prospective network owner will express a desire to share communications resources in order to reduce the unit cost of data transmission. This desire is immediately complicated by the existence of two primary carrier technologies: circuit switching and message or packet switching. See Appendix A for a very brief discussion of these two technologies.

The second motivational characteristic I call the data processing requirement. A prospective network owner sees this as a desire to access or improve access to geographically dispersed data and resources in order to increase the value and utility of data processing. This desire is immediately complicated by the question of exactly what function should a network owner provide his user community. This question can only be answered by the network owner. His major sources of input will be his network user community and his network designer(s). The vendor role in this process will be shaped greatly by the product set that the vendor has to offer. This question of function and its interrelation with implementation constraints will subsequently be discussed at length.

It is very important to understand that the desires stated above will seldom result in a cost-justification for a utility network. They will simply provide some feel for the magnitude of the resources affected by such a network. The justification will lie in the fact that the existence of the utility network will aid the user community in achieving their goals. In a corporate environment, this could mean that the existence of the network may result in significant, positive changes to the ways a corporation does business.

Unique Considerations

Having stated a definition of a utility network and discussed the basic motivations, it is now time to discuss networking as an environment compared to more traditional data processing environments. There are several significant characteristics that I believe make networking endeavors unique. They are:

1. There are significant resources to be utilized that are outside the province of the data processing vendor and outside the understanding of the data processors who may be acting as prospective network owners, designers, or users. All of the above parties will tend to view the goal, a utility network, from their data processing perspective. They do not realize that a significant, if not larger, proportion of the bottom line bill often goes for capabilities and functions other than data processing, e.g. carrier-provided communications capability.

2. There are many new transmission developments that are outside the province of the data processing vendor, e.g. value-added carriers, packet switching technologies, new interfaces to be supported, satellites as a transmission vehicle with unique characteristics, etc.

3. Front end processor concepts have great appeal both in configuration and operation. Considerations of these will follow.

4. The diversity of requirements placed on a utility network will most likely be extreme. For example, there may be requirements for long messages that must be delivered within a day with guaranteed delivery, short messages that must be delivered with minimum delay but can be recreated if lost, and middle-sized messages that must be delivered with minimum delay and guaranteed accountability. Initial thinking of prospective network owners is usually that all these requirements will be met by one utility network implemented with common resources.

5. There will very likely be a non-homogeneous mix of equipment. This includes the equipment used to implement the network as well as the equipment used to implement end-use-mechanisms.

6. A desire to accommodate existing environments may be restrictive or constraining to optimum network implementation. Consider the following scenario. A prospective network owner has a stated objective of no impact on existing user community environments. However, his user community has applications that are stable for the foreseeable future. These applications are running on purchased, old hardware using a 270X interface to communications lines. There is no readily apparent reason or justification for changing the implementation of the existing applications. This means that in order to conform to the objective, the new utility network must be implemented on boxes that can appear as 270X on one side and network on the other side. The cost and desirability of doing this must be carefully examined. Existing implementations may not lend themselves to a new environment (a utility network) and new technologies.

7. The cost of a complete, operational system includes items that data processors often forget. Here, the little understood but very significant factor is the visibility of the utility network. The network cannot be considered operational until the network owner understands and addresses parameters associated with user education, problem tracking and resolution with minimal operational impact, non-disruptive network changes, network maintenance, development of new capabilities, etc. Clean, effective management of these areas is essential to a pleased, supportive user community and to successful operations.

8. Before a network owner establishes or commits to a set of network requirements, he must understand the information flow that is required. Existing communications lines may not adequately describe this requirement. Unfortunately, the data may not exist to define the real information flow requirements. Therefore, many assumptions will have to be made regarding network objectives. As

the project progresses, many of these will turn out to be misunderstood or just plain wrong. There is an implementation cost associated with this process that must be understood at the outset. Closely related to information flow requirements is the determination of which end-user connections require immediate services of a host and which connections can tolerate delayed services.

9. The political and organizational concerns will far outweigh the technical issues besetting the network owner. The reason for this emanates from the mission of a utility network. The network owner is trying to provide a service for a user community that probably has its own way of accomplishing these services. The user community may be asked to give up some capability as well as control to the network owner. This is characteristically viewed as empire erosion by the affected user group and thus resisted.

10. The network owner must have the authority to resolve differences and break ties between his separate user groups. The very mission of a utility network ensures that there will be conflicts of interest between user groups and that these conflicts will be highly visible.

Requirements Impact On Implementation

When the prospective network owner feels comfortable with all the concepts and issues mentioned above, it is time to think about what his network is really going to be. The question to be answered is: when I look at the network as an end-user, what do I want to see? There are two mandatory requirements:

1. The network must be able to interface with end-use-mechanisms and to control these interfaces, and

2. The network must have some form of path control which may be very simple or quite elaborate.

However, these two requirements tend to be assumed. Alone, they will not generate much excitement in any prospective user community. Network owners and users frequently expect that their network should have many more facilities, for example:

network services for
    recovery and integrity,
    data collection and distribution,
    word processing,
    message switching,
    formatting and mapping;
network control;
secondary storage for end users;
function and volume growth flexibility.

The prospective network owner will make up a requirements list including items such as those mentioned above. This list is frequently passed to network designers or consultants without further thought. However, many of the desired functions dramatically affect the implementation effort. Only the network owner can assess the worth of implementing them.

Given such a wish list (requirements), a network designer or consultant should very quickly try to understand the real networking environment as the users want to see it. One way to evaluate this might be to pose the following questions to the owner and his user community. Is your networking environment one where:

> The only desired connection is between a terminal and an application program to transfer transactional data?
>
> There is no network services requirement for data recovery, message switching, secondary storage, or data collection and distribution?
>
> There is a standard interface with application hosts external to the network?
>
> There is the opportunity to standardize on a single transmission protocol or terminal type?

In most commercial environments that I have seen, the first answers of a prospective network owner or user community to the above questions are a resounding no on all points. Let us consider this. If all of the points of the question were answered yes instead of no, the resulting list of requirements begins to look very modest in light of the anticipated network functions.

To illustrate this, there is an existing, operational network that has such an apparently modest list of capabilities. It supports nine applications and approximately seven thousand terminals nationwide for such functions as:

accounts receivable,
order entry,
cash flow,
aids for system design,
technical training,
problem fix distribution,
class enrollments, and
message switching.

The network is an internal IBM network known as the Corporate Consolidated Data Network (CCDN). It is used by the FE, DP, and GS divisions of IBM. The initial business case existed because every branch office had three non-compatible terminals for separate applications. This meant that there were three separate communications networks going to the same sites. The initial user community consisted of DP and FE divisions who had their own way of doing things and saw no reason to change or give up function to a corporate group. The prospective network owner, an IBM headquarters group, was given the authority to mandate to the user community that CCDN would be used for the overall good of the corporation.

From this example, you can identify the phenomena of resisting erosion of control, political and organization problems of user groups with conflicting interests, and the financial and corporate motivations. All of these have been mentioned earlier.

Having described CCDN, let us now compare its function with the modest requirements list that arose from answering yes to all the questions previously asked about an imagined networking environment. We will take each point separately.

CCDN only supports connections between terminals and application programs for a transactional data transfer. This means no terminal-to-terminal or application-to-application connections are supported by CCDN.

Network services are limited when compared to the list mentioned earlier. In terms of data recovery, any user data that is in a CCDN stored-program box is lost when that box fails. All data recovery is the responsibility of the end-use-mechanisms, specifically the application programs. Message switching is not built into the network. It is implemented as an application, an end-use-mechanism to the network. The message switching application happens to be owned by the same group that owns the network. The noteworthy point is that network service applications may best be implemented as end-use-mechanisms owned by the network owner. Regarding secondary storage, at no time does CCDN make any secondary storage available to any end-use-mechanism. Nor does CCDN ever put any user data on any secondary storage device. CCDN has no facilities to do large data collection or distribution operations.

CCDN did elect to implement a standard interface with application hosts external to the network. The key word is 'external'. When the decision is made to have hosts external to the network, the network owner has committed himself to provide and maintain the interface between hosts and the network. One reason to make this decision is to isolate network physical implementation from end-user-mechanism physical implementation and possibly avoid conflict of interest circumstances. Having end-user-mechanisms and network functions implemented together can be more flexible and cost-effective but may pose a new series of management interactions.

CCDN did have the opportunity to standardize on a single terminal type for implementation of non-host end-use-mechanisms. Internal network transmission protocols were also standardized.

There are at least two important lessons to be learned from examining CCDN. The first is that a modest set of requirements may be very adequate for a successful utility network. Some parameters affecting the urgency of a given set of requirements are: how much responsibility does the network owner want to assume, how much money is available, how much time is available to bring up the network, how much authority does the network owner have with respect to the user community, etc. The second lesson is that in order to successfully implement the network, considerable changes to current processes may be necessary. Existing solutions to business problems may not be well-suited to new technologies and implementations. Again we see the necessity for a network owner to be politically and organizationally effective.

CCDN has proven to be a successful venture both financially and operationally. Significant monies are saved annually. Overall operational flexibility and reliability are improved over the original networks.

Having discussed CCDN, where are we in this paper? We have posed the question of what is my real networking environment. We have addressed some specific issues and discussed these in terms of prospective network owner initial expectations and in terms of an existing network. Finally, we drew some conclusions from examining a real network, CCDN. Given that the initial expectations of many networks are far greater than the those listed above, what should a networking environment be? The requirements might include functions and services such as:

- interfacing with incompatible terminals and mainframes; the network owner and designer should understand that no matter how this capability is provided, it will cost CPU cycles; we are not yet discussing how these cycles might be packaged:
- an architecture or open-ended design to support new network services; how much are you willing to invest up front to ensure a loosely defined flexibility at some future time?
- the ability to carry bulk data.

As a prospective network owner builds his list of requirements, some that appear simple may dramatically affect his implementation. Consider an example. The prospective network owner states that his network must be able to transfer bulk data. He envisions that an end-use-mechanism can dump messages up to five hundred thousand characters long on the network and request delivery.

Suppose we are designing a network for bulk data only. This objective alone levies several new requirements.

- The network must now have enough storage available to contain these large messages.

- Current storage technology dictates that this will be secondary storage because main storage is not inexpensive enough.

- Accepting very large messages presents the network owner with an integrity issue concerning his users' data. If a large message is the result of a long computer run, the network owner cannot afford to lose this data and ask the user to recreate it. Such a procedure is guaranteed to make users feel that they are victimized by the network instead of served.

The more common expectation is that, not only will the network handle bulk data, but over the same facilities it will handle transactional traffic. This adds requirements to those above.

- There must be some segmenting technique to break up end user work units into economically manageable units. 'Economically manageable' from whose perspective? Everyone is familiar with the concept of avoiding very large messages on a line that also serves terminal operators. The long message would dominate the bandwidth resource. In fact, long messages can dominate any resource used to implement the network, i.e. bandwidth, CPU cycles, or main storage.

- There must be some ability to reassemble segmented messages at the destination. This will involve a commitment of CPU cycles and storage.

- A need for some priority scheme may be envisioned so the network is able to favor some types of traffic with more resources to expedite delivery.

All of these conditions are imposed simply because the network owner wants to support some form of bulk data transfer. These conditions impact network operations at every step. There is more to design, more to implement, more to test, more to educate users about, more to maintain, more that is impacted by configuration changes, more that is affected with additions of new function or changes in technology. All of these represent significant costs. The network owner must now review his real requirements and circumstances. For example, elaborate network recovery capabilities may mean extensive user education on these facilities. How many terminal operators will know what a recovery is? Or care? To avoid the user data integrity concerns, perhaps the network owner should change his vision of bulk data transfer from that expressed above. A logical change would be to place the responsibility on the end-user-mechanisms to break up large messages into economically manageable blocks and to verify successful delivery of each one. Suppose the real networking requirement is 70% bulk data transfer and 30% transactional traffic, then the network owner might consider doing the bulk data job by itself. Or maybe he should consider two separate networks.

The questions and possible answers are endless. It is very important to make decisions keeping in mind that the network will be designed and implemented but once, while it will be used thousands of times per day. This should suggest that the real user community be adequately represented when decisions are made regarding what the network will or will not do. Data processors may have too narrow a view. The successful outcome of the utility network will depend on the positive interaction of user community expectations, network capabilities to satisfy these expectations, and the resources available to implement these capabilities.

How much work should a network work if a network could work net? It should be clear by now that the only person who can answer this is the network owner. Users, designers, and consultants can only provide guidance. In addressing this question, the network owner should be very careful not to get tangled up in implementation before he has identified the problem(s) he is trying to solve. There are two facets of this entanglement: arbitrary implementation constraints and unanticipated external factors.

Arbitrary implementation constraints are best illustrated by some examples. The network owner should be very careful about saying to his designers and consultants things like, "My network has to be built around a front end processor configuration." Before this statement is made, there must be some understanding of the advantages and limitations of FEP configurations, how the FEP configuration will interface with the host end-use-mechanisms, and how the FEP configuration will address short and long term network objectives. "My network must have three nodes." Why three nodes? I have seen situations where this was an objective yet the traffic did not exist to justify three nodes even if the nodes had zero cost. "All nodes must be duplexed with the backup machine hot and able to take over primary function with no disruption to end-users." This is something that can be done but the associated cost will be very high. This capability involves specialized hardware and software that can watch something while not doing anything - not a standard, off-the-shelf capability. As the desired network functions increase, this duplexed capability becomes more and more difficult to implement. If you have direct access queues, how do you resolve the pointers between the two boxes? What will be the takeover criteria? How will the secondary machine pick up in-progress error recovery? There are answers to all of these questions, but the associated cost will be high. Probably too high for a commercial utility network.

The second manifestation of implementation problems is unanticipated external factors. Suppose the network owner spends or asks someone to spend considerable resource on an extensive network topology design. The result shows that an optimum configuration should have seven nodes and they should cost $2,467.89 each. Then a businessman looks at this and says, "We will have three nodes because we have machine rooms in three cities. There is no such thing as a $2500 node when you consider physical facilities, programming, maintenance, etc. You say each node will be unattended. I don't see how that can be when the list of functions for a node looks like a computer advertisement. Besides we want decentralized network control to spread the knowledge around. This provides a growth path for operations personnel and protects the network from being crippled by a single outage." Reality in the form of non-technical constraints may shape what is done. In this case, a more rational approach to the network topological design might have been to understand the business and philosophy constraints first, then anticipate some probable results from the design effort. If the anticipated results did not fall within the constraints, then the design effort or the constraints should have been reconsidered. Here, a three man-month design effort that was essentially useless might have been replaced by a one man-month effort with some useable output.

Unless the network will never have to be cost justified, the network owner must ensure that someone associated with network design is considering the network in terms of available resource, user community relations, and many other business or real life considerations. Network design is much more than a technical job.

Up to now, the discussion has centered around the circumstances of the prospective network owner and the various influences affecting him. Eventually the network owner will have to say he understands all these ancillary factors and address himself to the task of implementing the utility network.

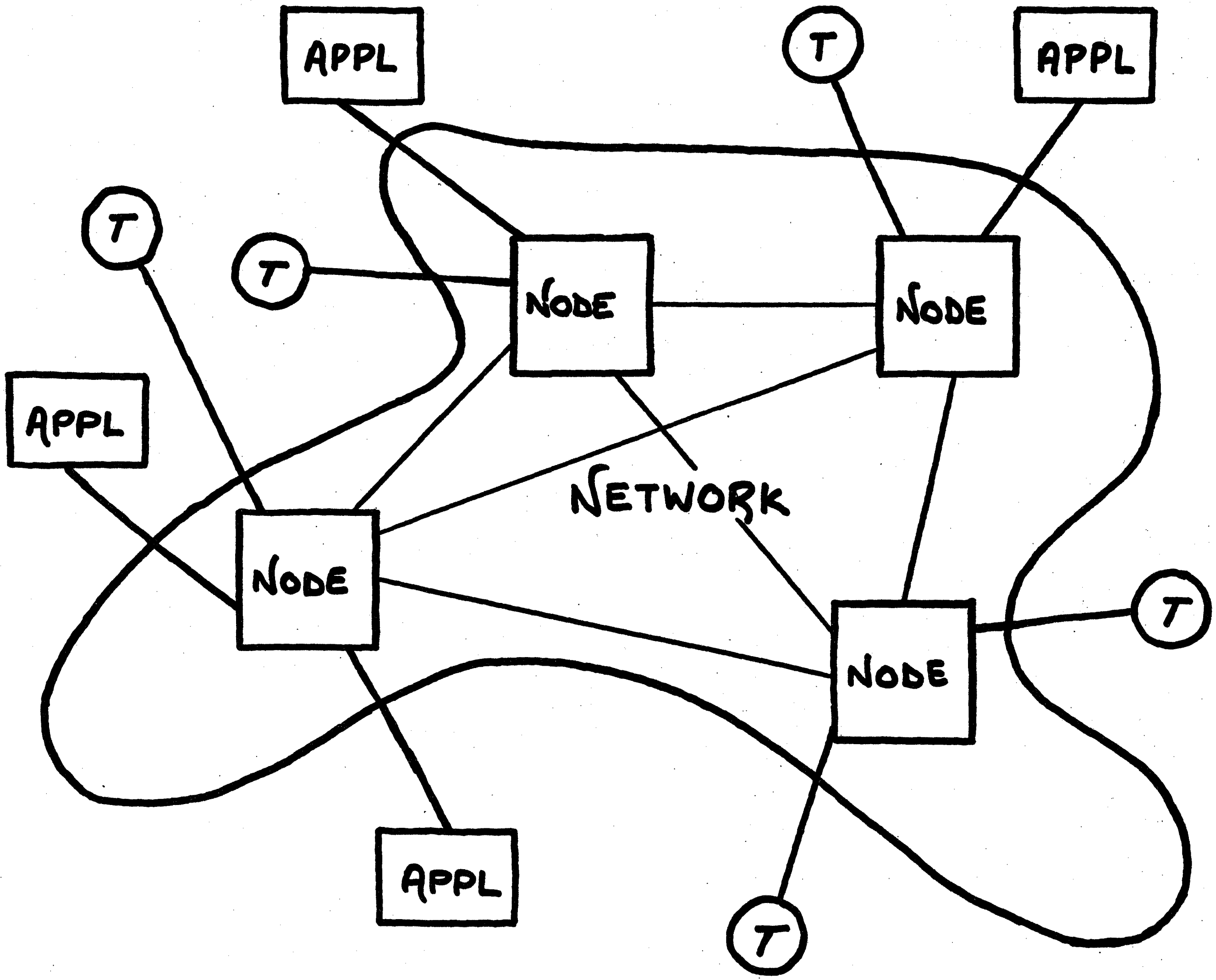
Implementation

The first aspect of implementation is to understand what resources are available to make the network exist and perform. There are only two: bandwidth and hardware. Bandwidth is the transmission capability provided to move data about. An oft used analogy is to compare bandwidth to the size and quality of the pipe one might provide to move fluids about. Most commonly bandwidth is provided to the network owner by carriers but the network owner can set up the communications capability with his own equipment. Included in this term, bandwidth, is all the associated equipment such as modems, TDM's, etc. Hardware has many varieties. One is fixed function hardware such as a 270X transmission control unit. We will not discuss fixed function hardware here because I contend that such hardware is a trivial case of network implementation. Another variety is programmable hardware. The term hardware, as used in this paper, will mean programmable hardware and the associated software to give it character and capability. The general physical characteristics of programmable hardware can be expressed in terms of available CPU cycles and main storage.

The network shown previously in Chart 1.3.0 is implemented with main storage, CPU cycles, and bandwidth. These are the physical resources that can be utilized by a network owner.

Once we accept this concept of available resources, the network as shown in Chart 1.3.0 changes to that shown in Chart 1.13.0. All of a sudden we have nodes which are interconnected by bandwidth and to which end-use-mechanisms are connected by bandwidth. This picture seems logical and intuitive. But the network designer needs more than logic and intuition. What is a node really? A dictionary defines a node as "a point of concentration; a central point." Some networking enthusiasts will define a node as an addressing point within the network. Neither of these is much help in understanding exactly what a node will be. What should a node do? One answer might say a node must provide a network interface to the outside world and perform internal network

THE HARDWARE IMPLEMENTATION, WHATEVER ITS CAPABILITIES, WILL BE CALLED A **NODE!**



from <u>WEBSTER'S</u> <u>NEW</u> <u>TWENTIETH</u> <u>CENTURY</u> <u>DICTIONARY</u>

NODE - a point of concentration; a central point.

R
JHR
1.13.0

functions and services. Minimum function for a node might be multiplexing, path control, and interfacing to end-use-mechanisms. Notice that the last two items are the same two items listed as minimum network function earlier in this discussion. Multiplexing has been added because we are now concerned with an entity, a node, that must do many things concurrently. Have these words helped clarify nodes? Probably not. The reason is that 'internal network functions and services' is extremely vague.

Stepping back a moment, you should understand that nodes are ill-defined in the networking community. There are many available ideas and concepts, most of which have some validity. However, many of these ideas and concepts are very loosely formed with little applied rigor or discipline. This ensures rampant confusion whenever two or more individuals try to discuss networking.

Returning to the theme, consider the following list of internal functions and services that might be expected of a node.

Queuing
Contention resolution
Addressing including broadcast
Polling
Concentration with speed and code conversion
Traffic prioritization
Integrity support
Network management
Journaling or logging
Path management
Performance monitoring
Accounting
Testing
Circuit switching
Data storage

The above list is certainly not complete but does include many things that are commonly considered node functions. Some deserve individual discussion. This should allow demonstration of how deceptively simple objectives expand into large, significant implementation considerations.

For example, consider traffic prioritization. This requirement may be judged necessary because the network owner, designer, or consultant envisions his network as supporting bulk and transactional data transfer. Clearly, the thinking goes, there must be some mechanism by which one traffic type can be favored, some scheme by which overall resource allocation on an end-to-end basis can be accomplished. This means that some identifiable traffic categories exist and some algorithm can be applied by which the network can allocate resources to the flow of that traffic. What resources is the network going to allocate to the traffic flow? The resources it has available, i.e. bandwidth, CPU cycles, and main storage. There are numerous schemes and techniques for traffic prioritization, many of them quite elegant. Few, if any, are well understood in a general environment. What does this mean to the network owner? It probably means that if he decides to implement an elegant scheme to better utilize network resources and provide desirable service levels, he should have R & D money in his budget and R & D time in the implementation plan. Furthermore he must plan to track his user community reactions while the network prioritzation scheme is being tested with live traffic profiles. Prospective network owners in commercial environments are seldom prepared to do this.

Another point deserving individual attention is 'integrity support'. When a network owner specifies that his network must have integrity support, some questions are raised immediately. For example, do you mean system or data integrity? System integrity is the ability of the network to react in some orderly manner to outages and variations within its own resources. Data integrity often means a guarantee of no data loss to the user community. That is a tremendous responsibility. These two integrity categories must be understood and addressed separately. As usual, the impact of a given requirement begins at system design time and remains throughout the life of the system.

Consider alternate pathing as one aspect of integrity support. There are many facets of this that are not widely understood. Are you thinking about dynamic or manual alternate pathing? How do you protect against duplicate transmissions which may result from alternate pathing? Have you considered the increase in network traffic that may result from dynamic alternate pathing? A pitfall of alternate pathing is that when the picture is drawn, it includes one terminal connected to a particular host over path A which will be shifted to path B whenever path A is interrupted. How long does A have to be interrupted for the switch to happen? In real life, instead of one terminal, it will be five hundred. This changes the magnitude of the problem considerably. If path B does not have the available resources (main storage, CPU cycles, and bandwidth) to sustain this new traffic load, then alternate pathing cannot work. How can the network owner ensure resource availability? He can over-commit, which means significant resources idle most of the time. Hopefully the alternate pathing circumstances are infrequent. Or there must be some priority scheme to allow the network to quiesce current traffic on path B in favor of the traffic from path A. The above example was alternate pathing for network connection availability. Network load balancing and resource optimization provide another motivation for alternate pathing. Many of the exposures associated with implementing priority schemes for overall resource allocation would also apply to alternate pathing techniques no matter what the motivation. And that takes us right back to the earlier priority discussion.

Acknowledgments are still another aspect of integrity support. The network owner may state that the network must provide acknowledgements to the origin upon request. The only acknowledgement the network will ever be able to give to the origin end-use-mechanism is that the message was delivered to the indicated address, which was a valid address, and that something at that address received the message. There can be no guarantees about what happened after the message disappeared into the destination end-user address. If the environment is such that one end-user must know something about the data after the network delivered it to the other end-user, e.g. a financial transaction, then a protocol must be worked out by the user group for its use of network provided connections of end-use-mechanisms. Herein lies one of the arguments for making data recovery and integrity the responsibility of the end-use-mechanisms instead of the network.

Network management or control deserves some mention. These terms include capabilities such as problem determination from a network control center, fast reaction to availability changes, security or authorization responsibility, fast response to configuration changes in either the network or end-user-mechanisms, and many others. These all represent programmed capabilities that are non-trivial tasks. The amount of programmed or automatic capability often depends directly on the network owner's view of his operations staff. An unsophisticated operations staff will require extensive programmed operation support which will entail program design, test, maintenance, etc. And it is critical to network operation. Perhaps less automated capability and more sophisticated personnel is a better choice. It is not unreasonable to expect network control implementation to involve three to five man-years of effort.

A final item for individual attention is statistics gathering. Many prospective network owners specify that their network will gather, keep, and process statistics. Once again, it is important to ascertain the purpose for which these statistics are to be gathered and processed. Capacity planning? Billing the user groups? Problem anticipation? Monitoring of errors? Data for tuning? As these questions are asked, the answer in most cases is yes. If the network designer installs numerous accounting exits and testing begins on the network, two things may become obvious. First, there is so much data gathered that it will take a twenty two hour run per day on a large, scientific computer to reduce this data to anything meaningful. Second, there is so much resource dedicated to gathering data that the network cannot perform. This statistics scenario is exaggerated but should serve to illustrate the following points. Gathering statistics has an associated price. Frequently, statistics gathering will cost network resource (main storage, CPU cycles, or bandwidth) at a time when they are least affordable. The network owner must give some careful thought to which statistics or accounting data he really needs and what he is willing to pay for it.

In our discussion, we have listed many functions that are typically expected of a node. We have singled some of them out for individual attention. All of the ones mentioned so far are telecommunications considerations. Equally important are network services kinds of tasks that the network owner may desire to implement in his network, e.g. message switching, word processing etc. These types of functions may well be included in the list of node functions to be implemented in the programmable hardware.

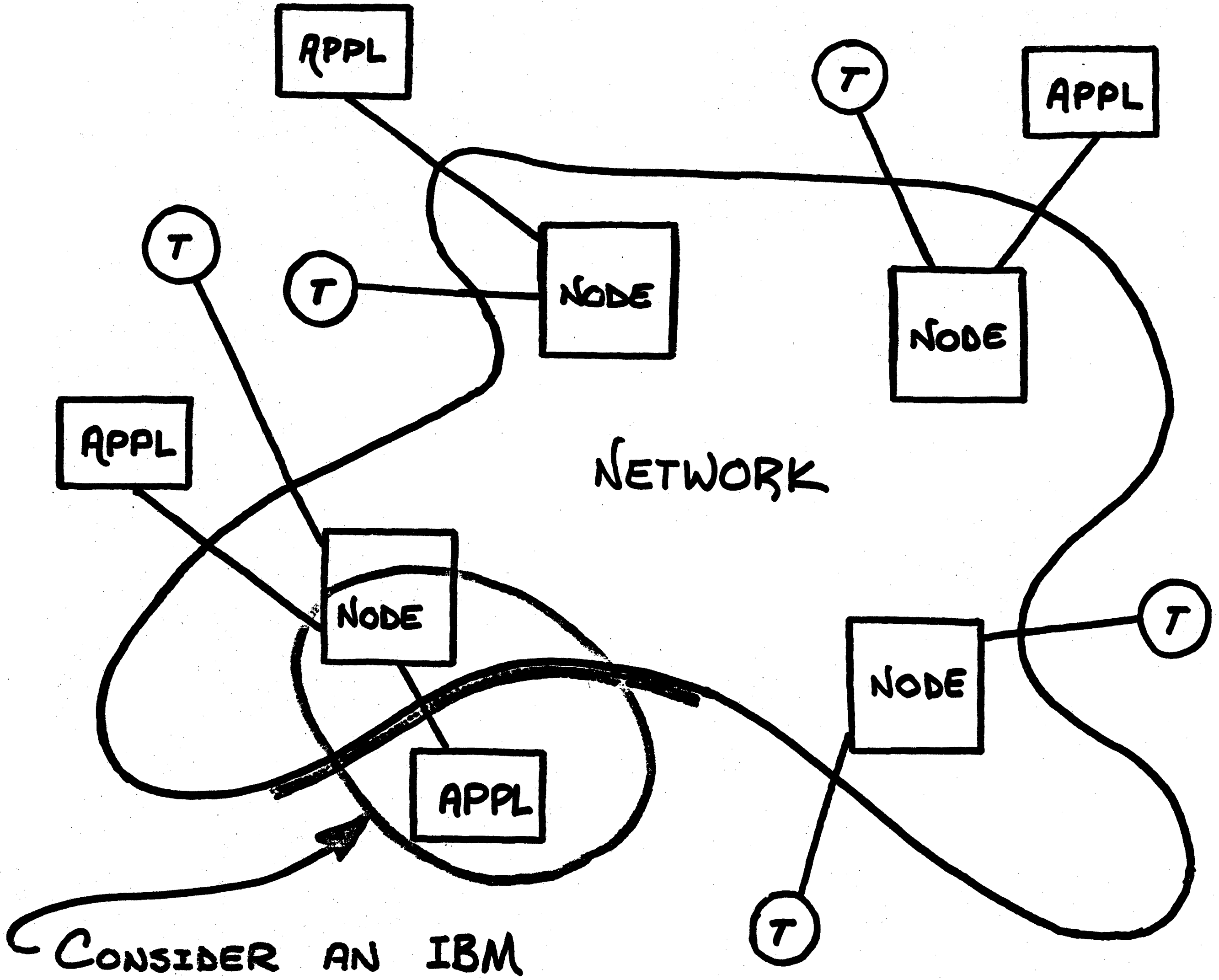
This discussion of nodes should leave the reader with the idea that nodes may come in different sizes with different capabilities. Consider a class A, B, and C node where the C node is minimum function as mentioned earlier and the A node is whatever full function list the network owner designates.

The B node is somewhere in between. For a general utility network, the network owner must understand how to cope with two types of growth; volume growth, which may mean growing a small C node to a larger C node, and function growth which may mean growing a B node to an A node. Volume growth occurs because there is increased demand for the existing set of network services. Function growth occurs because of new technologies, new user group requirements, new network services, etc. Further, there is no requirement that all types of nodes be implemented on the same or compatible types of hardware. Such a requirement would be a decision of the network owner.

Implementation Configurations

Now that we are experts on node functions, it is time to consider some ways that nodes may be implemented physically. Chart 1.18.0 is a necessary view to begin. We will take this logical picture and apply it to physical implementations as we are used to them. The following discussion has a machine-room orientation toward mainframes, associated end-use-mechanism, and communications controllers. Although not included in this paper, similar understandings are necessary for the environment outside the machine room. Before proceeding with the discussion, we need to explore some fallacies that arise from the logical picture. These fallacies are called picture problems. Each picture problem will be described then discussed briefly before moving on. The problems can be visualized with Chart 1.18.0.

NOW THAT WE UNDERSTAND NODES,



CONSIDER AN IBM

IMPLEMENTATION

OF THIS PORTION OF THE NETWORK

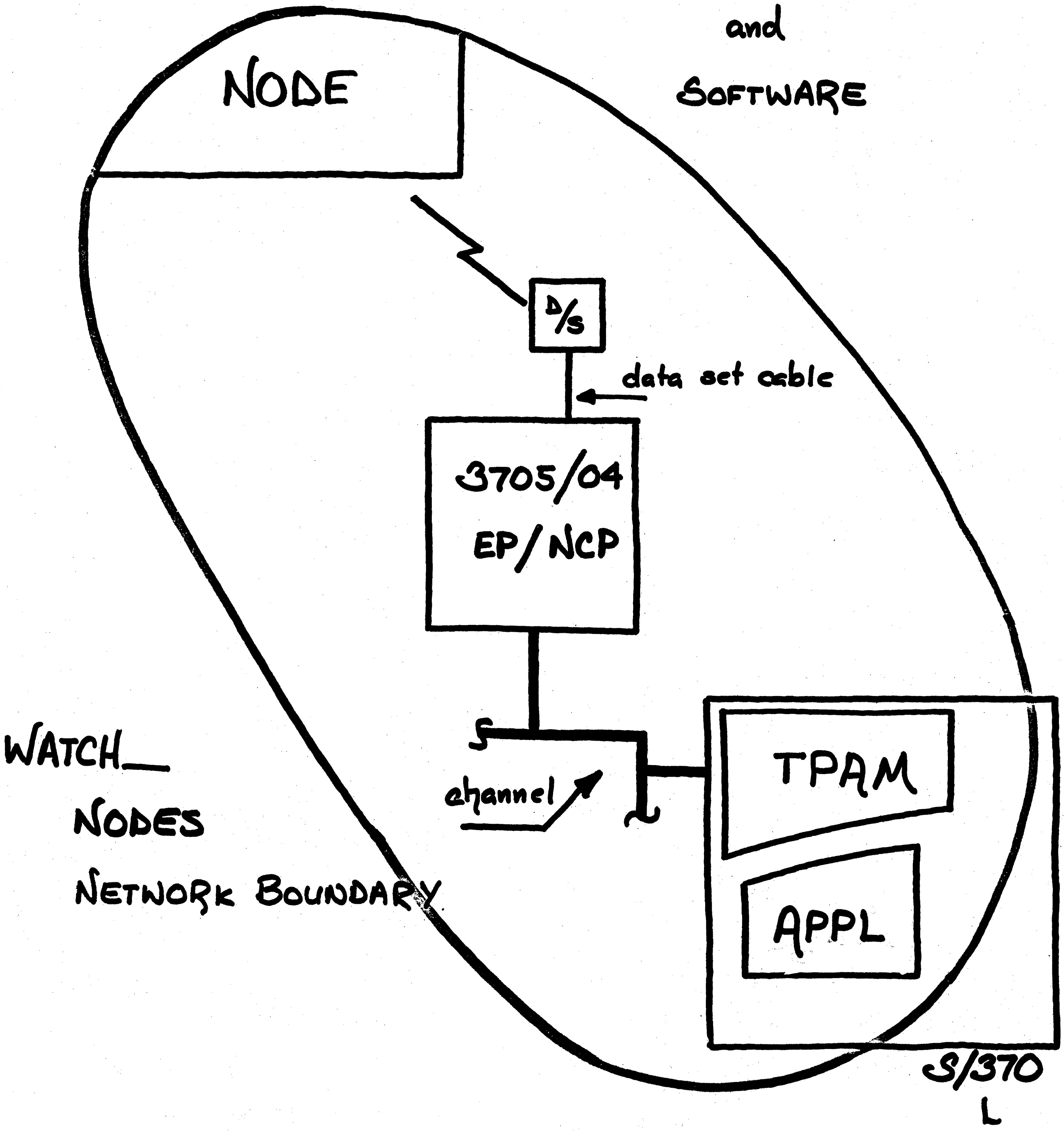
L
JHR
1.18.0

The first picture problem concerns the end-use-mechanisms labelled APPL, the application programs. There is a tendency to translate APPL directly into CPU. The rationale is that everyone knows that an application program (APPL) cannot exist without a computer. The application is an end-use-mechanism outside the network, therefore the host computer (CPU) must also be external to the network. This reasoning is not sound. There are no criteria yet specified that would dictate placing host computers totally external to the network. There are reasons and circumstances why the network owner might elect to specify such a configuration; but there are associated costs as well as benefits. These should be clearly understood by the network owner before he establishes such a constraint.

The second picture problem is very similar. We have shown nodes as boxes totally contained within the network. When the node materializes on the machine room floor, what does 'totally contained within the network' mean? Usually it is interpreted as meaning that the box will be owned and controlled completely by the network owner or his representatives. However, there are no criteria yet specified that would dictate that physical nodes be owned and controlled completely by the network owner. As before, there are reasons and circumstances why the network owner might stipulate such a configuration. And, as before, he must understand the impact of such a stipulation.

The third picture problem is perhaps the most important. It has to do with the line connecting the host end-use-mechanism to the node (network). The fact that the connection is drawn as a simple line makes people think that this connection is physically something with a linear form, e.g. a communication line, a data set cable, or a channel. Considering device attachment as we know it, channel-attached control units and access methods (see Chart 1.22.0), assumptions about the physical characteristics of the connection immediately define the location of the network boundary. For example, assume that the line in Chart 1.18.0 connecting the end-use-mechanism to the network (node) will be implemented as a channel. This means that the network boundary from Chart 1.18.0 appears between the S/370 and the 370X on Chart 1.22.0.

PHYSICAL CONFIGURATION of HARDWARE and SOFTWARE



L
JHR
1.22.0

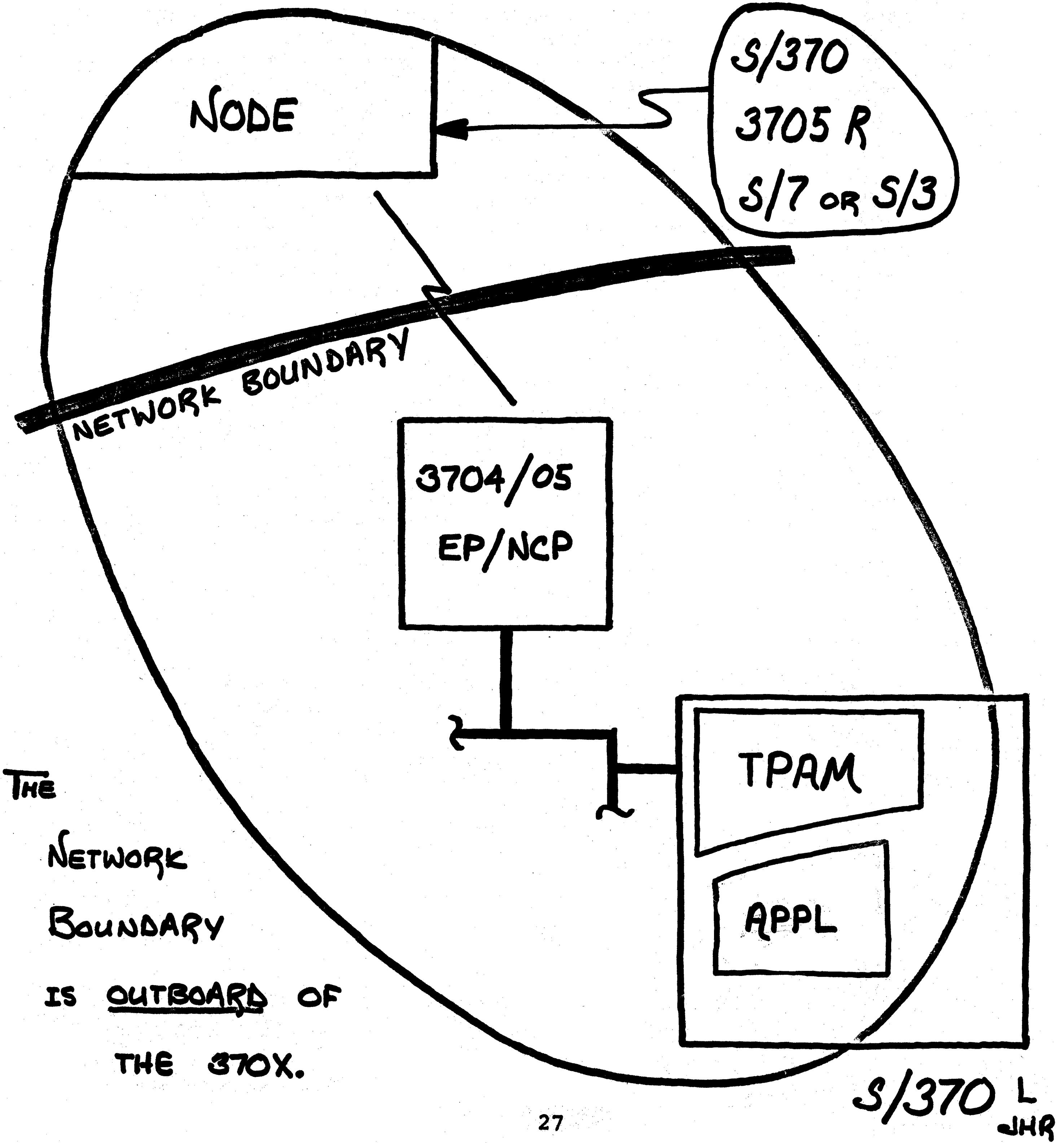
These three picture problems represent assumptions that one might make about physical implementation while studying a logical picture such as Chart 1.18.0. Having exposed the picture problems, we will now consider three major implementation categories. These categories are based on the location of the network boundary relative to a real data processing configuration. These will be discussed individually below. For each boundary location, it is the responsibility of the network owner to determine if the functions available to him are consistent with the network objectives he has established.

Chart 1.23.0 shows the first configuration. The network owner envisions the network boundary outside the communications controller (shown as a 370X). The connection line between the end-use-mechanism and the network is supposed to be a data set cable or a communication line. With some poetic license, I call this the packet-switcher configuration. This choice of network boundary location dictates several conditions. First, with the RS-232 connection to communications facilities, the network owner has probably chosen one of the slower and more unreliable implementations of the network connection. Second, there is much resource outside the network (main storage and CPU cycles) that is necessary to accomplish the desired telecommunications job. The efficacy of this boundary configuration may not be readily discernible to persons without an intimate knowledge of all the parameters.

Regarding the RS-232 type connection to communications facilities, I recognize that more connection flexibility will be possible as new interfaces such as X.25 and X.21 are better defined and come into more widespread use.

An implicit assumption in Chart 1.23.0 is that the box representing the communications controller is a data processing box. This is the traditional view. In the future, it is reasonable to expect carrier communications offerings that challenge this assumption. These offerings will involve the network boundary moving toward the host and reduction of the data processing sphere by replacing the communications controller system with carrier-provided facilities. Change to the interface specifications defining this new boundary implementation are also to be expected.

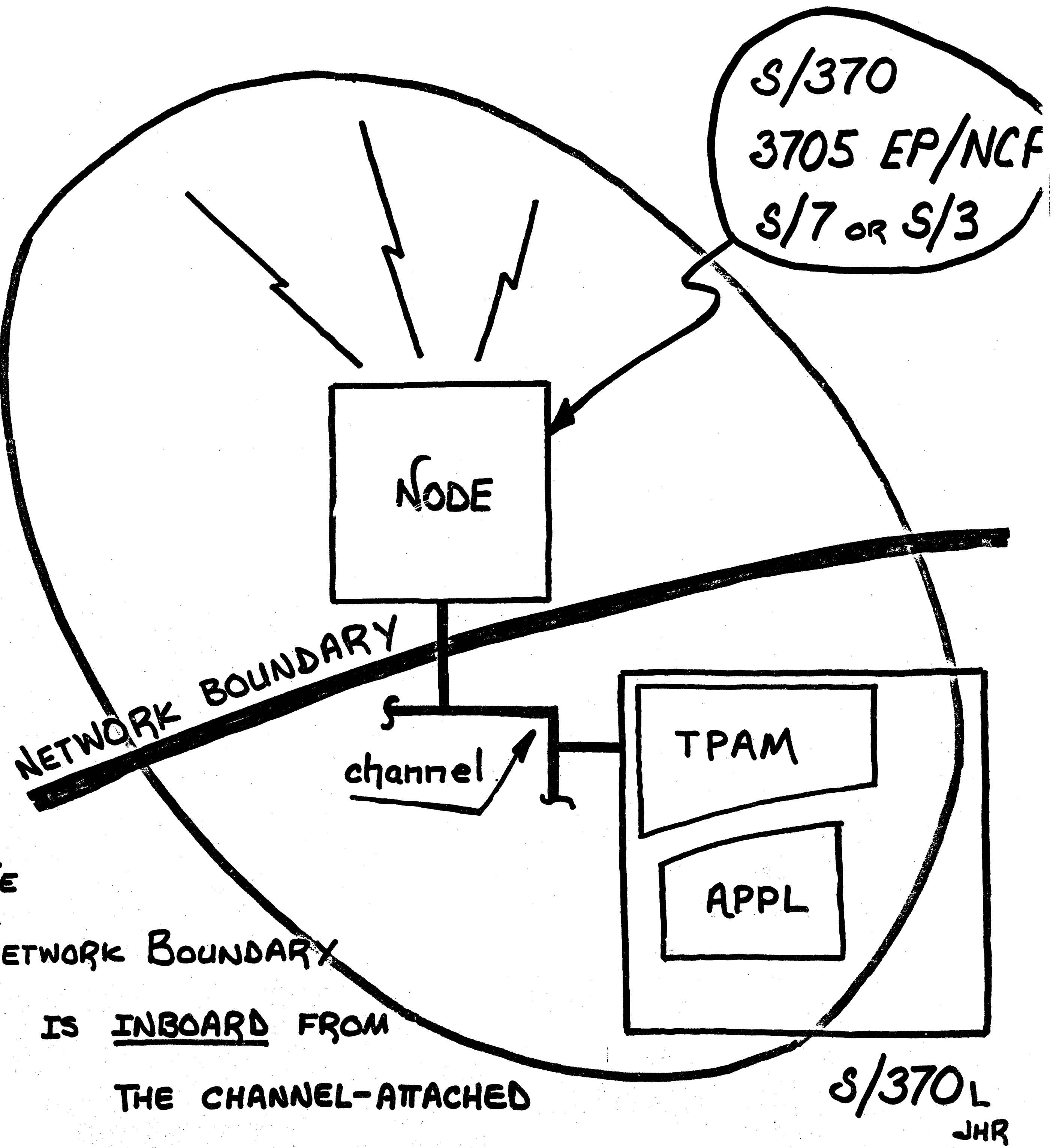
THE CONNECTION LINE REPRESENTS A DATA SET CABLE OR A COMMUNICATION LINE



THE NETWORK BOUNDARY IS OUTBOARD OF THE 370X.

JHR
1.23.0

THE CONNECTION LINE REPRESENTS A CHANNEL



THE
NETWORK BOUNDARY
IS <u>INBOARD</u> FROM
THE CHANNEL-ATTACHED
CONTROL UNIT.

JHR
1.24.0

Chart 1.24.0 shows the second configuration. The network owner envisions the network boundary inside the communications controller but outside the host. The connection line between the host-implemented end-user-mechanism and the network is supposed to be a channel. I call this the front end processor configuration. The connection to the network is a channel, much faster and more reliable than most widely used communications facilities. The box labelled 'node' represents CPU cycle and main storage resource available to implement node or network function. This function is initially specified by the list of node functions prepared by the network owner/designer. The box used to implement the node (Chart 1.24.0) must have capabilities and limitations consistent with ultimate network objectives. This may be much more than CPU cycle time, instruction set, and addressing capability. These limitations include configuration flexibility, compatible enhancements, hardware and software support, maintenance, to name a few.

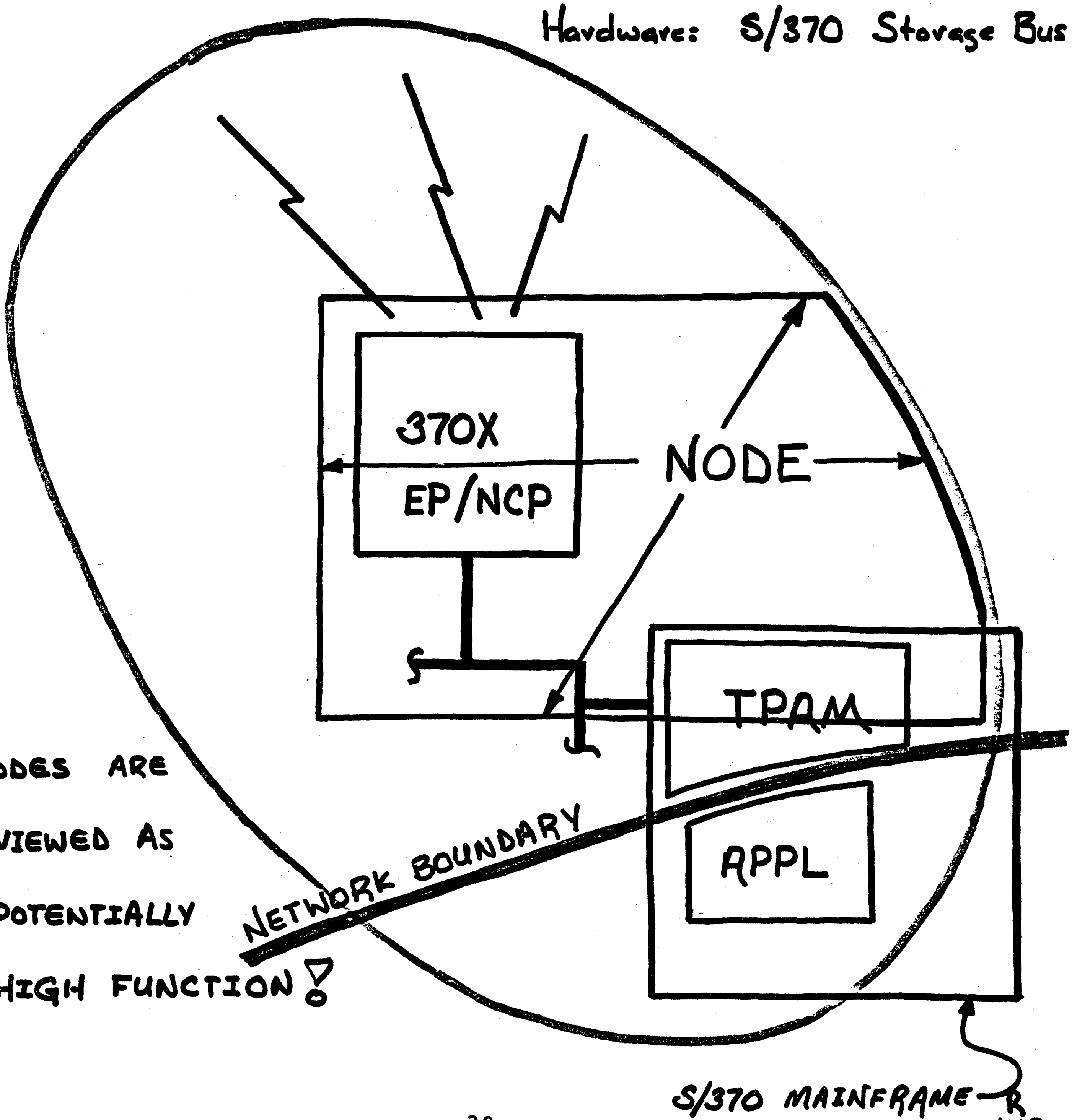
Chart 1.25.0 shows the third configuration. The network owner envisions the network boundary inside the host computer. There are no more line-like physical things for the connection line. The network connection is implemented as a computer storage bus which may be the ultimate in speed and reliability. Nodes are implemented as some combination of the communications controller and the mainframe. The mainframe executes instructions on behalf of the network (node) as well as on behalf of the end-user-mechanism. This configuration is where the IBM product line has evolved. Without exploring all the reasons for this evolution, one does stand out. Nodes, whatever they may be, are viewed as potentially high function devices. Remember the list of node functions offered earlier. Incomplete as it was, it still represents a considerable resource requirement. With such a list of node requirements, we can see that node implementations must have a readily available growth path to more CPU resource.

THE CONNECTION LINE REPRESENTS A ....

THE NETWORK BOUNDARY IS WITHIN THE CPU.

Software: API of TPAM

Hardware: S/370 Storage Bus



NODES ARE VIEWED AS POTENTIALLY HIGH FUNCTION!

JHR
1.25.0

Proponents of one configuration or the other are often driven by the way they package CPU resource. A prospective network owner may feel that he understands the configurations described above. Yet he may still assert that he doesn't like the configuration shown in Chart 1.25.0. Allegations of need for more function out in front are often heard. No matter how well one understands the desire to distribute function, he is still faced with the decision as to how to do it. At this stage, there are some similarities between a vendor and a prospective network owner. The vendor may be trying to implement and evolve a consistent product line while the network owner is looking at implementing a general utility network in a manner that allows minimally disruptive growth and enhancement. Implementation configurations, supportability, attachment and configuration flexibility, enhancement difficulty, and many others are all parameters of decisions that must be made by vendors and network owners. Subsequent remarks should be considered in light of this similarity.

The order in which the next points are addressed implies nothing about their relative significance. You should also realize that we are discussing the mix of function between the host computer and the box designated 370X in Chart 1.25.0. Whether this second box is called a 370X, a communications controller, a front end processor, a node, or an inverse spandrel is not important. The characteristics of it and its ability to provide a consistent network foundation are important. I will continue to call it a communications controller.

In recent years, much attention has been given to the large amount of resource required by a mainframe to control a teleprocessing environment with a 270X interface to communications facilities. This resource was used up because of excessive disk accesses for each error condition and unproductive polling operations. NCP directly addressed these problems. It offloads polling, much error recovery, operations such as code translation, and allows a more efficient data flow between the communications controller and the mainframe. However, to implement NCP involves extensive changes to the mainframe interface. Prior to NCP, the mainframe interface dealt with a device known as a communications line. With NCP, the mainframe interface deals with a device known as a communications controller. The two devices have dramatically different characteristics. Much of the available mainframe code to support the communications line interface (BTAM, RTAM) was simply not suitable for interfacing with a communications controller. Thus changing devices necessitates changing mainframe code, even to move out a relatively small function. The result is that a large change in system configuration is necessary to achieve an apparently small change in visible configuration or operation. This makes the large system change difficult to justify by itself.

Prospective network owners allege that they want still more function moved to the communications controller. This presents a very complex problem. Exactly what function should be moved or implemented differently? Often suggested is network control. Programmed network control function, whatever it may involve, has the characteristic of much code and tables but very low utilization. Now the designer has to answer an interesting question. Does it make sense to put programmed network control in a box, the communications controller, where real storage may be at a premium. The availability of real storage at this point in the system can have significant impact on network performance characteristics. There is no fixed answer to this question but it is a consideration. Suppose the communications control unit has been implemented on a box with a limited addressing capability, e.g. 65K, 128K. When you start to consider hundreds of terminals, dozens of lines, programmed network control, these amounts of storage will not go far. You can end up in a storage bind long before any other resource gets critical.

Futhermore, if a desire to keep hardware costs low has led you to a box with limited addressing capability, this box may also have a limited instruction set. Consider the 3705 with 51 instructions and the S/370 with 183. The 3705 does not have the data sniffing instructions, e.g. decimal arithmetic, translates, storage-to-storage, etc. If the network designer desires to have data sensitive operations in the communications controller, then an unsophisticated instruction set may prove to be a real bottleneck. For example, suppose you desire a decimal arithmetic and move operation that will require execution of a twenty instruction loop with a limited instruction set. If this has to be done very often, then its execution could conceivably consume the CPU cycle resource of the communications controller. Another case would be the requirement to examine each data character for some property. This examination will require CPU cycles. It may be possible for this use of CPU cycles to offset the benefit of a cycle-stealing scanner with block level interrupts.

The above examples should demonstrate that a network owner or designer must give some thought to desired function before he commits himself to a hardware configuration. Data processors often tend to view the CPU cycle resource as a bottomless well. It is a mistaken view. A mismatch of desired function and box capabilities provides a disappointing and expensive result.

Now, the reader may be thinking "This is all garbage, the cost of hardware continues to go down." This I will not dispute. However, it is unit cost of hardware that is going down, i.e. the price per instruction execution. Hardware without software is useless. There is little evidence that software costs or system design costs are small or are decreasing. The network owner must understand that his network is a machine. The components of the machine are bandwidth, stored-program boxes, programs, terminals, and people, among others. There is a cost associated with each of these components. One of the most significant components is the people. They provide the magic, the imagination, the insight. The network owner must be careful not to choose a hardware configuration that has a lower box cost but requires a higher people cost to support it. The bottom-line bill for the network could be much higher or the price could be lost opportunities to serve the user community due to capability limitations.

Making the communications controller a potentially powerful system increases the temptation to take advantage of this power. The network owner must satisfy himself that a configuration involving a potentially powerful communications controller is to his best advantage.

In general, the issue is one of reliability and availability. As a box becomes functionally more complicated, will the reliability of that box decrease? In general, the answer is yes. Will the reliability of that box decrease to an unacceptable level? There is no general answer to that question. The circumstances must be assessed individually by the network owner. The network owner must understand how he will have a high function box and still meet his reliability/availability criteria and his network objectives for service. Service includes responsiveness to new requirements.

An oft-stated requirement is to have a communications controller with a direct access device. There are many valid motivations behind this requirement. From the network designer's perspective, there are many questions to be asked. Exactly how will the disk be used? This is both a short term requirement and a long term requirement. It affects disk size, access speed, transfer rate, and hardware attachment capabilities. More importantly perhaps, what types of organizations will you want for this disk? ISAM? BDAM? VSAM? TCAM-type reuseable and non-reuseable queues? Remember that this question must be answered for the life of the system at the outset. If you want a general capability for various organizations, then you are talking about access methods. The presence of access methods increases supervisory complexity and requires code. If you are willing to provide code to support file organizations as needed, then there is a software development/maintenance cost which is a people cost. A further cost may arise from reduced flexibility.

What are some of the requirments for disk on the communications controller? One common scenario involves improved network availability. The network owner has a six hour window in which to collect data from a large terminal population. His network designer says he needs a disk on the controller that can hold two hours worth of data. The purpose is to allow the controller to keep the network up for two hours in the event of host outages. If the host is to be unavailable for more than two hours, then the controller will be switched to another host to do the data entry job. This appears to be a very reasonable requirement. What happens when the host is available again after an outage? The controller must have the main storage and CPU cycle resource and the supervisory capability available to run the communications network, the disk, and the channel concurrently. This capability is idle except when there has been an outage and recovery is in process. This is considerable function for an inexpensive box with low function supervisory software. Suppose the disk drive on the controller fails. Can the disk be serviced without affecting the rest of the controller? How many other spindles are available, unaffected, and can be used? Suppose the failure occurs at the end of the two hour window and results in the loss of the data that has been gathered. Are you now better or worse off than if the data had been on a host-associated disk or was still out at the data entry point? This scenario presumes a user group with a straight data entry operation that could be implemented with a very low function disk and communications controller. What about another user group to be serviced by the network? They like this network availability provided by the disk. But their data entry is currently supported by TCAM with extensive on-line edit and scrolling capabilities. To this second user group, data entry is useless without these on-line capabilities. To implement these capabilities in the communications controller makes it begin to look like a general purpose computer.

Suppose the design objectives state that the host-controller interface is to be 270X. Remember that this interface is such that supporting host code (the access method) expects a device with the characteristics of a communications line. Furthermore, the access method will contain a definition and topological description of the line network. Yet the network owner/designer may specify a high function communications controller to be the foundation for a high function network implementation. The 270X interface objective and the high function controller implementation may well be dangerously conflicting. This is not to say that the communications controller cannot implement high function network services (queuing, network control, message switching, etc.) and still appear to the host with a communications line interface. However, the stipulation of 270X interface insures that there is host code containing some topological definition and code to control lines while the high function communications controller understands and controls the real network of communication lines. This means there are two topological definitions which must be kept synchronized. This means the controller is executing instructions to control lines. The host is executing instructions to control a pseudo-network of lines. And the controller is executing instructions to translate between the real network of lines and the pseudo network defined in the host access method. Finally, many high function network services involve implementations with traditional data processing requirements. This may mean they are more appropriately implemented in an environment with the supervisory and generation flexibility of a mainframe rather than repeating equivalent capabilities in a communications controller. The circumstances described in this paragraph suggest that there may be extensive, wasteful use of computing power. Given the added complexity and the redundant use of CPU cycles, the network owner must look carefully at the benefits and justification for a configuration involving a high network function implemented in a communications controller with a 270X host interface.

The above discussion illustrates the long term effect and the interrelationship of many typical requirements. Many of these parameters apply to all kinds of additional capabilities desired for the communications controller.

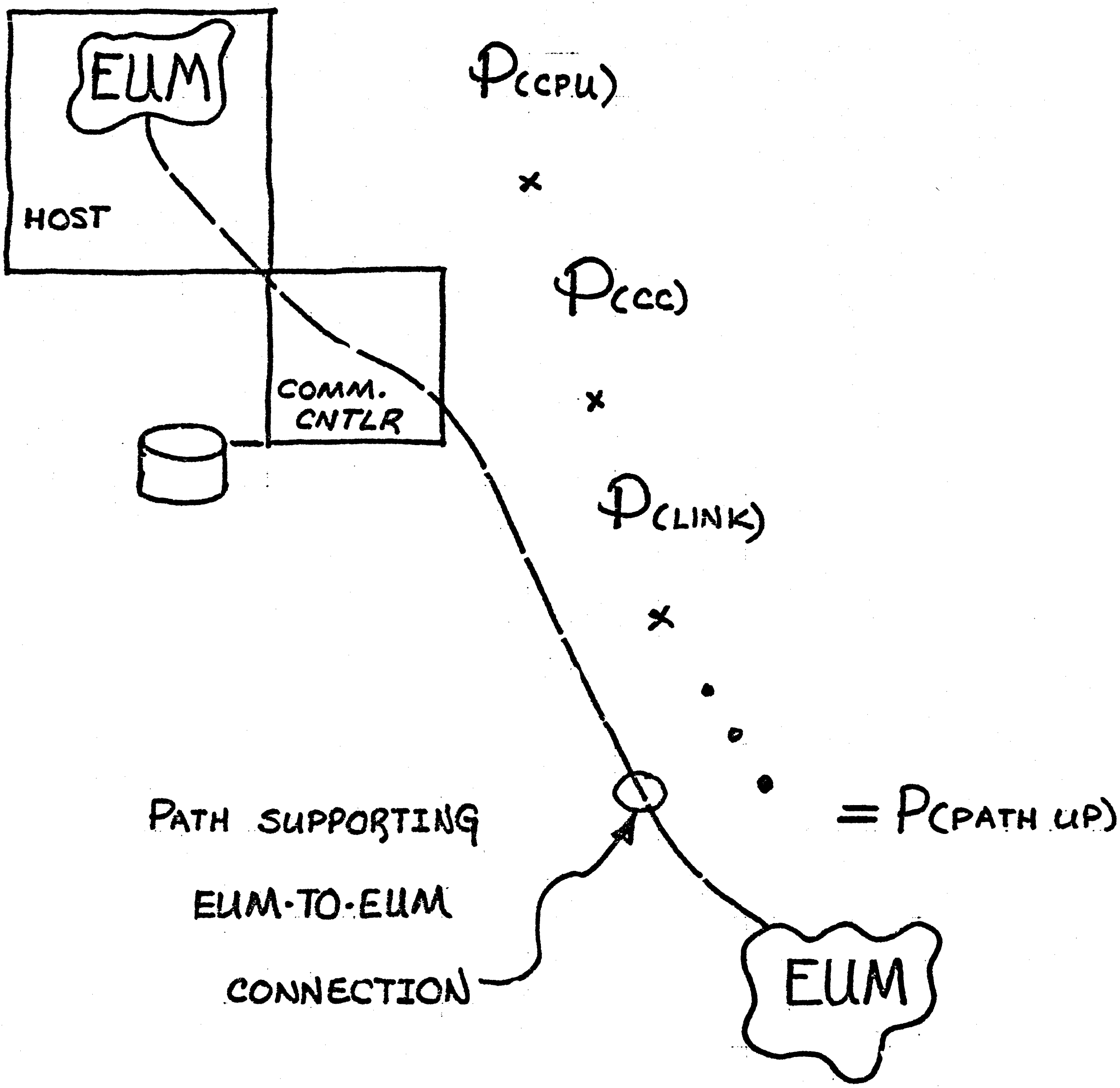
Consider the 3705 and its history. Experience has shown the 3705 to be an extremely reliable box. Some of the reasons for this are: limited function, no elaborate supervisor services, dispatching driven directly by hardware, no mechanical I/O. When the network owner/designer envisions a communications controller with extensive message handling capabilities, he expects high function, elaborate supervisor services to support software services, software controlled dispatching with its associated path length overhead, and mechanical I/O.

Now examine Chart 1.30.0 noticing particularly the end-user to end-user path and the associated components. The probability of this path being operational is the product of all the individual component probabilities;

$$P(\text{path up}) = P(\text{cpu}) \times P(\text{cc}) \times P(\text{link}) \times \ldots \times P(\text{term})$$

If the communications controller grows from a box with 3705 NCP capabilities to one with TCAM capabilities, in general P(cc) will decrease, i.e. the controller will be less reliable. The network owner/designer must understand how this function migration will occur without making P(path up) decrease to an unacceptable figure. This is not to say that it cannot be done. I am suggesting that data processing experience to date indicates that it is more difficult than it may initially appear.

Let us consider an interpretation of current data processing experience. Much attention has been given to the need for a data processing vendors to standardize their teleprocessing product offerings. Indeed, IBM has tried to evolve the product line toward one access method, one facility for gaining access to communications capability. This has been difficult to achieve. In general, the environment is too complicated, there are too many different circumstances, philosophies, and implementations for a single, high function implementation to be viable. Committing to one product, one implementation begins to appear as an unrealistic goal. A single implementation commitment is too inflexible to allow reaction to new technologies over a long product life.



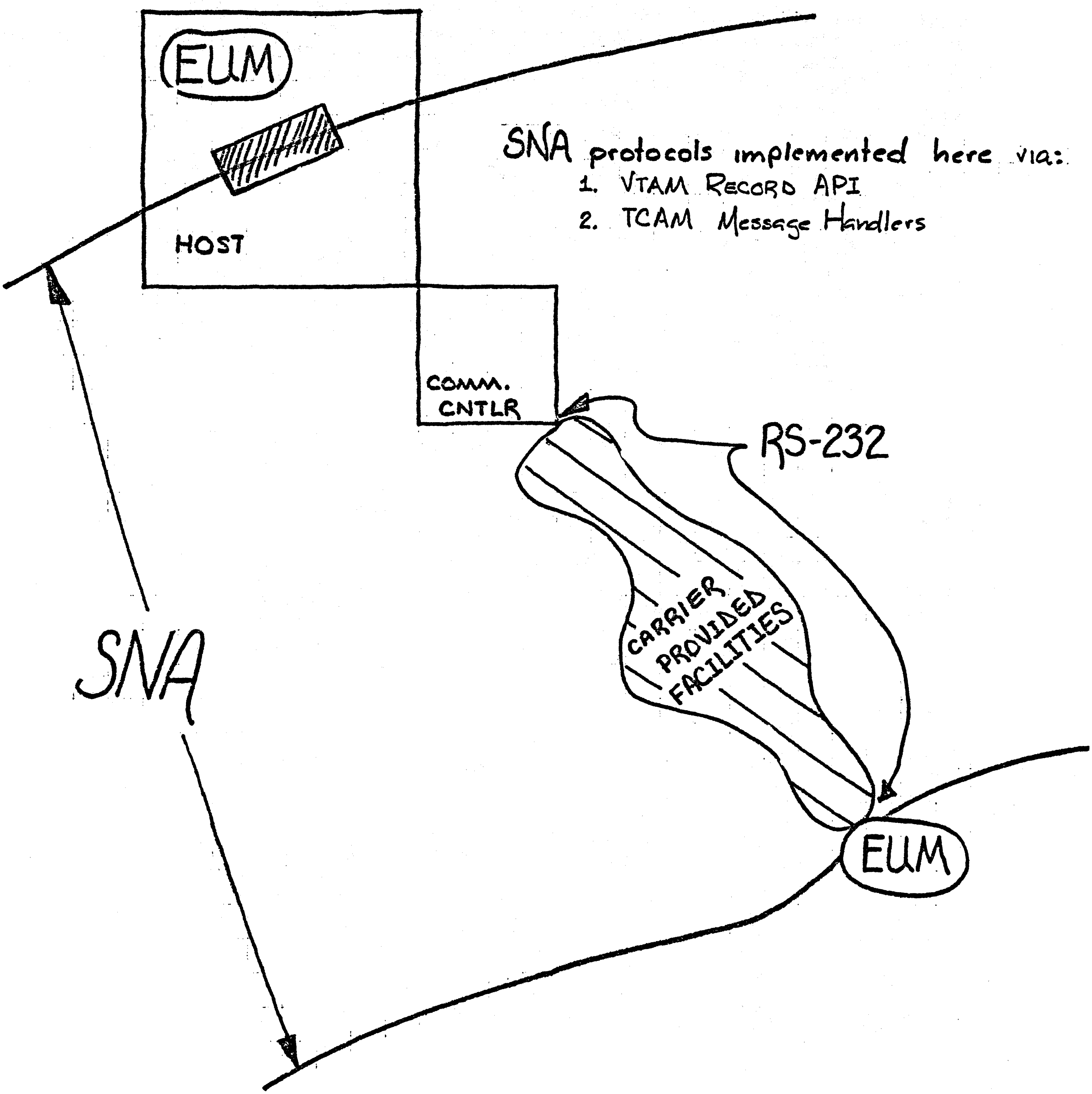
* $P_{(\text{PATH UP})}$ represents the probability that the path will be up or available.

This experience could lead to the conclusion that a single product is not needed. Instead, the need is for a uniform set of rules or protocols under which product development can be done. The purpose is to allow implementation flexibility while conforming to prescribed interfaces and protocols. Such a set of rules is usually known as an architecture. A vendor or a network owner might wish to establish an architecture to assist in providing consistency to the execution of their mission.

IBM's Systems Network Architecture is an example of such an architecture. See Chart 1.32.0. With SNA, IBM has said these are the rules under which we will develop products for moving data between two end-use-mechanisms. The consistency is provided by the SNA protocols for moving data. But the real consideration of an architecture is the implementation flexibility provided.

Consider some SNA-based examples of implementation flexibility. On Chart 1.32.0, within the bounds of SNA, one part of the process that a vendor or network owner may elect not to provide or be constrained from providing is the communications capability typically provided by a carrier. There are standard interfaces for accessing these communications capabilities, e.g. EIA RS-232 or CCITT X.25. Within the current implementation of SNA, the RS-232 interface is the primary one for accessing bandwidth. X.25 and X.21 represent interfaces to new, carrier-provided communications capabilities. As they become standardized and utilization of them increases, a vendor or network owner might elect to support these new capabilities. The important perception is that under an architecture, such as SNA, a new communications opportunity could be supported with minimual to no change to existing implementations of architecture protocols.

The current implementations of SNA are mainframe based, i.e. they will not run without a S/370 computer to house the access method. Another important perception is that new distributions of function, new implementations, can be accomplished as soon as rational configurations can be understood. If the governing architecture is effective, then these new function distributions and new implementations should be minimally disruptive to users of the service implemented under the architecture.


EUM
HOST
SNA protocols implemented here via:
1. VTAM Record API
2. TCAM Message Handlers
COMM. CNTLR
RS-232
CARRIER PROVIDED FACILITIES
SNA
EUM

1.32.0
JHR

The significance of an architecture to allow maximum long range flexibility has been demonstrated. That is not to imply that there is not an associated cost with achieving architectural consistency for long range flexibility.

Conclusion

As stated at the outset, the purpose of this discussion has been to provide improved insight into the field of networking as an aspect of information processing with particular emphasis on areas of importance to network owners and designers. There have been three underlying goals in this paper for accomplishing this objective. First, the reader should have expanded his network focus to include the user community and its relation to the network owner and the services he provides. Second, implementation alternatives have been categorized into three possible configurations. Third, some interactions between desired network objectives and implementation alternatives has been discussed. It is hoped that the content of this paper will aid interested readers in their comprehension of networking for their respective environments.

Appendix A

Circuit switching is the most familiar and widely understood. Carriers using circuit switching technologies provide the capability for a user to purchase a communications channel (bandwidth) for any desired period of time. Circuit switching has the property that physical communications resources are allocated to the purchaser for the specified time period. From the perspective of the owner of the communications resources, the carrier, this is very inefficient because the actual utilization of resources by the purchaser(s) will be very low. The carrier would like to be able to optimize the use of communications resources by allocating them to the purchaser needing them at any particular instant. This leads directly to the second technology.

Message or packet switching techniques are employed to achieve greater utilization of a given communications resource, i.e. serve more purchasers with the same resource. This is accomplished by using stored-program boxes in addition to lines (bandwidth). The stored-program boxes allow at least two additional capabilities: maintaining an awareness of logical connections between users without allocating channel resources and an ability to react to variations in channel availability. Lower unit costs for transmission and improved service are the objectives. These represent added value for communications users, hence the name, value-added carriers or networks.

GLOSSARY

communications controller-	a stored-program computer which provides an intelligence function to implement a specified communications function. It may or may not be a stand-alone box; it may or may not have peripheral i/o. The term 'communications controller' is intended to encompass terms such as front end processor (FEP) and transmission control unit.
data processor-	an individual or group whose primary exposure to information processing has been through large, single host environments. Networking, as an information processing endeavor, is a new concept. It demands a clearer distinction between communications and processing.
host-	a host is a stored-program computer on which end-use-mechanisms will be implemented. The CPU cycle resource of the host need not be dedicated strictly to end-use-mechanism implementations; the host may execute instructions on behalf of the network. For this paper, hosts will not interface directly with a communications facility; the communications controller is required. This restriction is for conceptual clarity, it has no technical foundation.
mainframe-	a stored-program computer that exists behind the communications controller. The communications controller is between the mainframe and the

	communications facilities (bandwidth). A mainframe may or may not be a host; it depends on whether an end-use-mechanism is implemented on the mainframe. Usually mainframes will be hosts.
network-	a general, utility communications service with a single-system appearance to users.
network consultant-	individuals or parties that provide information and advice on techniques, configurations, objectives, implementations, etc.
network designer-	the individuals or parties who design the network for the network owner. 'Design' includes hardware and software configuration, decisions about network services, and selection of communications resources to be used.
network owner-	the individual or party who wants to provide and be responsible for the network services and their availability. This person or party is typically a customer of a data processing vendor.
network user-	an individual accessing network services through some end-use-mechanism.
node-	a logical concept to specify where network services as specified by the network owner will be provided. Nodes may be implemented with some combination of hosts, mainframes, and communications controllers. Nodes may also be

	implemented in other configurations such as cluster contollers or other computers outside the central site configuration of hosts, mainframes, and communications controllers. These are mentioned only briefly in this paper. As a logical concept, nodes have no addressing significance but this can change once node implementation is defined.
user community-	the total collection of parties that the network owner desires to service with the network. The user community is comprised of user groups.
user group-	individual collections of network users. For example, a network user community might be an entire corporation. Each division of that corporation may represent a user group.

READER'S COMMENTS

TITLE: A PERSPECTIVE ON NETWORKING
WASHINGTON SYSTEMS CENTER
TECHNICAL BULLETIN GG22-9003-00

PLEASE STATE YOUR OCCUPATION______________________________________

COMMENTS:

PLEASE MAIL TO: JAMES H. RUTLEDGE
IBM CORPORATION
WASHINGTON SYSTEMS CENTER
18100 FREDERICK PIKE
GAITHERSBURG, MARYLAND 20760