Revised Viewspec Cards

If you have suggestions, please let me know.

Revised Viewspec Cards

Jeanne Beck hasbrought the little viewspec/Keyset cards up to date and made some changes in the fomat, improvements as I see it. I amseneding the revision thrugh the review process, but with luck everyone will OK it and we can send the file to DDSI thursday night. The draft is in <userguides,viewspeccard,> and some further explanation is in (journal,jrnl22,j24262) 1

Revised Viewspec Cards

(J24266) 23-OCT-74 09:10;1, >, 23-OCT-74 09:38 XXX ;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DIRT( [ ACTION ] ) JOAN( [ ACTION ] Please put this and 24262 in the dirt notebook) ; Sub-Collections: SRI-ARC DIRT; Clerk: DVN;

Thanks a bunch, Guys

Will whoever borrowed my SRI phone book please return it. I would be ever so enormously peeved if I had to reconstruct my personal phone book, which resides on the back cover, and current updates to ARC addressess and phone numbers, which resides inside the front cover. Doncha got no class???

Thanks a bunch, Guys

BRING BACK MY SRI PHONEBOOK! 1

Thanks a bunch, Guys

(J24267) 23-OCT-74 10:08;1, >, 23-OCT-74 10:31 XXX ;;;; Title: Author(s): Jeanne M, Leavitt/JML; Sub-Collections: SRI-ARC; Clerk: JML; Origin: < LEAVITT, SHOOFLY.NLS;1, >, 23-OCT-74 10:03 JML ;;;;####;

Liaison with Pullen and Bourne

Stan,
It's me again - different topic this time. Charlie Bourne called and wanted to know whether Keats Pullen had ever made arrangements for him (Charlie) to use the BRL slot at Office-1 to Collaborate on some of the problems Pullen had outlined during his visit here. Since you are the Architect I assume this would be cleared through you. Can Charlie have access to the system through your slot? Would appreciate knowing what if anything has transpired on this matter. Thanks, Jake 1

Liaison with Pullen and Bourne

(J24268) 23-OCT-74 10:50;1, >, 23-OCT-74 11:28 XXX ;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SMT( [ ACTION ] ) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

For A user otion to Turn off Journal notification

I think the feature of the journal interrupting your work to tell you when it delivers is a pain in the ass; there should be a useruption to defend users against it. Nor do I like delivery itno the classes information and action. It is bad enough trying to force items into those blunt catagories when you send them. 1

For A user otion to Turn off Journal notification

(J24269) 23-OCT-74 11:22;1, >, 23-OCT-74 11:31 XXX ;;;; Title: Author(s): Dirk H, Van Nouhuys/DVN; Distribution: /FDBK( [ ACTION ] ) HGL( [ ACTION ] ) JDH( [ ACTION ] ) ; Sub-Collections: SRI-ARC; Clerk: DVN;

rfc number system

i cant get any rfc numbers out of office-1, the error message "openlock failed: <JOURNAL>RFCNUMBERS,.," seems to be the problem. 1

rfc number system

(J24270) 23-OCT-74 12:01;1, >, 23-OCT-74 16:33 XXX ;;;; Title:
Author(s): Jonathan B. Postel/JBP; Distribution: /JDH( [ ACTION ] ) ;
Sub-Collections: SRI-ARC; Clerk: JBP;

The 1 Jul 74 NSW Proposal is not in its 23352 journal record

Asking Dick to take personal responsibility for requested action,

The 1 Jul 74 NSW Proposal is not in its 23352 Journal record

I wanted to get online access the the NSW propoal: SRI No. ISU 74-132, 1 July 1974, SRI-ARC 23352. Jump Link returned "File Not On Line." The sequence of messages below says that a) there is nothing journalized under 23352, and b) the number anyway appeared to have been reserved for some other purpose. 1

J22-1107 ENGELBART: Retrieve 23352 from archive
Distribution: OPERATOR, engelbart
Rcvd: 22-OCT-74 11:07:05 1a

Thanks, Doug 1a1

J22-2307 KEENEY: FILE
Distribution: ENGELBART
Sent: 22-OCT-74 2307-PDT 1b

[...other business...] HOWEVER, 23352 WAS NOT ARCHIVED, NOR HIDDEN IN ON OF THE JOURNAL FILES. I SENT JEFF A MESSAGE CONCERNING THIS. SO YOU MIGHT CHECK WITH HIM TO SEE IF HE WAS ABLE TO TRACK IT DOWN.
MARCIA 1b1

J23-1001 PETERS: 23352
Distribution: ENGELBART, keeney
Sent: 23-OCT-74 1001-PDT 1c

DOCUMENT #23352 - "IMLAC INTEREST GROUP NOTE INDEX" WAS NOT JOURNALIZED. - JEFF 1c1

This is a significant failing in the "integrity" of our recorded-dialogue system. It is important to locate the correct version of the text and to put it into the Journal under this number. I'd say that it ought to go in under the date that it is actually Jouralized, with a comment that explains the discrepancy between the date of the issued, hard-copy proposal and the Journal-publication date. 2

I'm asking Dick Watson, as the responsible author, to take personal responsibility for guaranteeing that the Journalized version is exactly the same as what was published. 3

A similar incident ocurred on a proposal or report about a year ago, and what was put ino the Journal was not the final version. During the last days before something like this is finalized, a lot of changes can occur, and it is easy to lose track of what files, in whose directory, contain the final version. 3a

The 1 Jul 74 NSW Proposal is not in its 23352 journal record

(J24271) 23-OCT-74 14:26;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /RWW( [ ACTION ] ) JCN( [ INFO-ONLY ] ) CHI( [ INFO-ONLY ] ) JCP( [ INFO-ONLY ] ) KIRK( [ INFO-ONLY ] ) DVN( [ INFO-ONLY ] ) ; Sub-Collections: SRI-ARC; Clerk: DCE;

Test

Ha Ha Ha 1

Test

(J24272) 23-OCT-74 16:59;1, >, 23-OCT-74 17:03 XXX ;;;; Title:
Author(s): Harvey G. Lehtman/HGL; Distribution: /JDH( [ ACTION ] ) ;
Sub-Collections: SRI-ARC; Clerk: HGL;

# MAXIMIZING CAPABILITIES: towards a standard for decision

by Kirk Kelley

NLS Journal number -- 24273
1975
Augmentation Research Center SRI
Menlo Park, California 94025
(415) 326-6200 x3506

## Introduction

There is at times a need to have a standard for decision. This effort develops the philosophical framework leading up to and describing such a standard. A more rigorous mathematical foundation and some mathematical techniques for applying the standard to specific situations are to be part of a later work.

Standards for decision come in various flavors. In applying the science of decision making, the standard most universally recognized as being a poor one is the one most often used because of it's easy measurability: maximization of money. Maximizing people is another standard popular because of deeply etched evolutionary tradition. Standards for decision determine what is morally or ethically right or wrong. Any goal is a standard for decision because alternatives are decided in relation to the goal. Goals are contained in models of the future universe. Your model of the universe is your standard for decision because all of your interactions with the universe are based on and modify that model.

As your model of the universe grows ever more accurate, it tends to become more dissonant with old models of the universe. A large percentage of the human population seem to have a desperate need to identify in a real everyday motivational way with the process of universal life. This need is evidenced by the continued existence in ever growing contradiction to reality of many old models retaining their forms as religions. The growing dissatisfaction with these old models is evidenced by the decreasing membership in established churches and the proliferation of "new" models. Certain phenomenon very recent on a human scale, such as the population explosion and the industrial revolution, seem to account for this dissatisfaction. The fact that there is a proliferation of new models or standards for decision indicates that the need for a new model still exists but none of those available is yet fully satisfactory.

The standard for decision documented here looks as if it may be measurable enough to replace standards such as the Gross National Product and yet it is based on a model of the universe accurate enough to provide a meaningful alternative to old moral and ethical standards.

We begin by showing the universe to be in the process of maximizing alternatives. Basic terms are defined leading up to the definition of a capability as a controlled alternative. The probability equation for survival is developed and applied to the process of the universe showing that in order to maintain or improve its probability of survival in an environment maximizing alternatives, an alternative control process must maximize the alternatives it controls. This is called the law of survival.

One of the distinguishing features of a conscious alternative control process is that it has a model of itself as it's standard for decision. A model of yourself as a conscious alternative control process is described in which every decision you make is in order to maximize your capabilities. Thus, if this is your model of yourself, then your standard for decision is to maximize your capabilities.

The relationship of this standard for decision to various topics is then discussed. Some of the topics are: truth, knowledge, learning, expanding consciousness, freedom, progress, omniscience, futility, techniques, and happiness.

## The Process of Time: MAXIMIZING ALTERNATIVES

As time increases, the number of alternatives in the universe tends toward a maximum.

All known isolated physical systems outside of black holes change irreversibly toward the most probable state. This is the only law of physics by which we can determine the direction of time and was called by Eddington "times arrow" [1].

The probability of a system being in a particular state is determined by the number of alternatives in the state divided by the total number of alternatives in the system. In an isolated system, the most probable of all its states is the one containing the maximum number of alternatives. The universe, being all that exists, is an isolated system.

Since the most probable state is the one containing the maximum number of alternatives and to change toward some maximum is to maximize, we say the universe outside of black holes is maximizing alternatives.

Since the direction of time is defined by the process of maximizing alternatives, we call this the process of time. It is described as the maximization of entropy by the second law of thermodynamics.

[1] Blum, Harold F. "Time's Arrow and Evolution," pp. 14-33, Princeton University Press, Princeton, New Jersey, 1968.

Capability: an alternative Controlled: an alternative where the response to its decision is fed back to the stimulus for its decision. To help visualize, follow this sequence of models.

Change: modeled by an arrow:

```
    ->
```

Stimulus: the beginning of a change,
modeled by a circle to the left of an arrow:

```
  O ->
```

Response: the end of a change,
modeled by a circle to the right of an arrow:

```
    -> O
```

Process: a change from beginning to end,
modeled by an arrow between two circles:

```
  O -> O
```

Entropic Process:

```
       O
  O ->
       O .
```

Alternative: one of at least two responses for the same stimulus, modeled by one of the circles following the arrow.

Bit of entropy: any set of two equally probable mutually impossible alternatives, an undecided bit, modeled by two circles placed one above the other.

```
       O
       O
```

Decision Process:

```
  O
    -> O
  O      .
```

Symbol: one of at least two stimuli for the same response, modeled by one of the circles preceding the arrow.

Bit of Information: the result of a decision between two equally probable mutually impossible alternatives, a binary digit.

```
                    0
```

Standard for decision: that which determines which alternative a decision process chooses.

Feedback Process:
a process where the stimulus is a response to its response. Said another way, the response is the stimulus for its stimulus; modeled by an arrow looping back to its source:

```
.- - - - - - - -.
'--> 0 -> 0 --'
                 .
```

Positive feedback loop: a feedback process which continuously chooses the same alternative without interruption.

```
.- - - - - - - - .
'--> 0 ->        !
            0 --'
     0           .
```

Negative feedback loop: a control process: a feedback process oscillating between alternatives in order to keep some variable within limits determined by its standard for decision.

Modeled by:

```
.- - - - - - - -.
!    0          !
'-->    -> 0 --'
     0           .
```

a bit of entropy & a bit of information combined — an alternative that symbolizes itself.

Capability: an alternative controlled: modeled by one of the circles in front of the arrow.

A controlled alternative is one where the response to its decision is fed back to the stimulus for its decision: an alternative decided in response to a standard for decision in a negative feedback loop.

Examples:

A symbol is one of at least two responses for the same stimulus. You can visualize this by looking at your pen and noticing that the word "pen" on this page and your pen are both stimuli for the same response somewhere in your mind. They are both symbols for each other.

A coin can represent a bit. If we do not allow it to be on it's side, it is impossible for it to be both heads and tails and if we flip it into the air, the probability that it is either heads or tails is equal. I give you a coin and say, "If I correctly call out 'heads' or 'tails' before it lands, I win the coin. If I'm incorrect, you win it." After you flip it but while it is still in the air, it represents a bit of entropy in that it is changing too fast for us to tell if it is heads or tails and so is undecided. After it lands it represents a bit of information because it is decided to be either heads or tails. If you can see it's current status after it lands in your hand, you can win no matter what I call out because you have the capability of making it either heads or tails. You respond to whether it is head's or tails, and it responds when you turn it over. In such a case, it represents an alternative controlled.

Another example of a bit is the light switch. A light can be either on or off but it is impossible for it to be both on and off at the same time. If you are blind, the light switch represents a bit of entropy in that you do not know for sure whether it is on or off. If you can see the light, it becomes a bit controlled because it responds to your stimulus and your stimulus depends on whether it is on or off. That is, you respond to the status or decision of the switch that, in turn, responds to you.

An example of a standard for decision can be found in a "thermostat -> air conditioner -> room air -> thermostat" alternative control process or negative feedback loop. The standard for decision is the setting of the thermostat. The temperature at which the thermostat is set is compared with the temperature of the room and a decision to have the air conditioner on or not is made in order to maintain the room temperature at the setting. Given any particular room temperature, the setting of the thermostat determines which alternative the process will choose thus providing the standard for decision.

## The Law of Survival

Let x represent the total number of possible alternative states of a system and c the total number of alternatives controlled in that system. Some percentage v of x are essential variables which must be kept within vital limits. These are called vital alternatives because they are vital to the survival of c the alternative control processes. Some percentage p of c intersect with v. See Figure 1 below. pc represents vital alternatives which are also controlled alternatives. This is an important distinction. If an uncontrolled vital alternative occurs, all of c is eliminated. If a controlled vital alternative occurs all of c is not eliminated.

Each alternative has a probability 1/x of occuring at any one time. The Probability of survival of c at any one time is represented as

$$S = 1 - ((vx-pc)/x) .$$

That is, one minus the number of uncontrolled vital alternatives, divided by the total number of alternatives.

```
.---------------------------------------.
!   x = total number of alternatives     !
!                  .------------------.  !
!                  ! c = controlled   !  !
!                  !   alternatives   !  !
!  .---------------!--------------.   !  !
!  !    v = vital  !  pc = vital  !   !  !
!  ! alternatives  ! alternatives !   !  !
!  !               !  controlled  !   !  !
!  !               !              !   !  !
!  '---------------!--------------'   !  !
!                  '------------------'  !
'---------------------------------------'
```

(Figure 1)

The second law of thermodynamics is described above as a process maximizing alternatives. This means x tends toward a maximum at some rate k as time increases. If the probability of survival S is also to increase, then (vx-pc)/x = v - pc/x must decrease. Therefore, in order to improve its current probability of survival, c must tend toward a maximum at a rate more than v/p times greater than k, the rate at which the total alternatives are maximizing.

If the probability of survival is to remain constant or increase over time, then the change in c must be greater than or equal to (vk/p)

That is, in order to maintain or improve its probability of survival in an environment maximizing alternatives, an alternative control process must increase the number of alternatives it controls at a rate equal to or greater than the rate of increase of the total number of alternatives in the environment times the percentage of alternatives which are vital to the survival of the alternative control process divided by the percentage of those vital alternatives which are also controlled.

In order to survive, an alternative control process must maximize the alternatives it controls.

The Definition of Life: Life is the process of maximizing capabilities.

Those familiar with Ashby's book "An Introduction to Cybernetics", can view the Law of Survival as encompassing the Second Law of Thermodymics and Ashby's Law of Requisite Variety. Whereas Ashby's law is independent of natural phenomena and holds for any hypothetical system whose variety can be measured, the Second Law of Thermodynamics is verified empirically by observation of the universe. The Law of Survival is also verifiable by empirical observation of life.

Lets take the specific example of the DNA process. The DNA process can be visualized as a feedback loop which cycles once per generation. The environment continually changes obeying the second law of thermodynamics. Each generation, certain organisms fail to continue the cycle. Those that remain, do so because they adopt to the changing environment characterized by the increase in the number of possible states that environment can assume. This is the evolutionary process of natural selection. The net result is a process called the evolution of life which is continually maximizing the alternatives it controls.

Eventually the standards for decision in alternative control processes became sophisticated models of the process. This conscious aspect is discussed next.

Your Standard for Decision: to maximize your capabilities.

A standard for decision determines which alternative a decision process chooses; that towards which the decisions of a process lead: its goal. If every decision a process makes is in order to go toward a certain state, that state is the ultimate goal of the process. An alternative control process is a decision process where the response to the decision is fed back to the stimulus for the decision. A conscious alternative control process has a model of itself as its standard for decision.

Since you are a conscious alternative control process, your standard for decision is your model of yourself. In other words, your standard for decision is what ever you think it is. In particular, it is your model of your future self, or your goal, which is based on your model of your past. Note that you are not just a part of the alternative control process, but you are the entire process including the alternatives it controls.

Your environment is maximizing alternatives as explained above in the description of the process of time. Alternative states of your environment transmitted by the senses to your brain represent your interaction with the environment. Incoming stimuli representing such alternative states create electrical vibrations in short term memory. These electrical vibrations are compared to encoded vibrations (sometimes called long term memory) which contain your self model. This model is of your future self as well as your past and present and thus contains your standard for decision.

If electrical vibrations do not match the encoded vibrations of your model, then there are at least two responses for the same stimulus in the form of dissonant alternative vibrations. Since every dissonant electrical vibration represents an alternative for which there is no matching encoded vibration, there is no model of controlling that alternative. Thus every dissonant electrical vibration represents an alternative you do not control.

Dissonant vibrations are not immediately absorbed and continue to reverberate firing more neurons. Each time a neuron fires, adrenaline is released into the blood stream. The rising blood-adrenaline level eventually motivates a decision or series of decisions until the dissonance has been eliminated.

This process causes the vibration representing the alternative and the decision associated with it to be added to the existing encoded vibrations thus updating your model. Whenever that alternative occurs, it also triggers the vibration representing the decision which eliminated the dissonance so you can now choose to make that decision, based on that model, and have the response to the decision fed back and compared to the model to see if the decision was successful. In other words, you now control the alternative and this capability has been added to those you already have. Every decision you make is motivated by this neural-adrenal mechanism.

For example:

> You are thirsty. This stimulus creates electrical vibrations in your brain which are dissonant with your model of your interaction with the universe: to continue in your comfortable position not being thirsty. The dissonance represents the currently uncontrolled alternative of remaining where you are and not being thirsty. The stimulus does not go away, you are still thirsty. Eventually the dissonance is so great that your blood-adrenaline level rises to a level where your current position is no longer that comfortable. You decide to modify your model to include the vibrations that you are thirsty. In response to this decision you get up, drink some water, and return to your comfortable position. The dissonance is eliminated and you now have added to your capabilities that of sitting in your comfortable position not being thirsty.
>
> Thus the dissonance with your model of your interaction with the environment motivated a decision to update your model and as a result increased your capabilities.

Every decision you make is in order to add to or maximize your capabilities. An ultimate goal is that towards which every decision is intended to lead. Your ultimate goal is therefore to maximize your capabilities.

If the electrical vibrations in your short-term memory as a result of understanding this description of yourself do not create irresolvable dissonance with the encoded vibrations representing your model of your self, then your model of yourself is that you seek as your ultimate goal to maximize your capabilities.

Since as a conscious alternative control process your standard for decision is your model of yourself, your standard for decision is to maximize your capabilities.

## What does it mean to maximize capabilities?

There are many ways of saying it.

"Maximizing capabilities" can be a more precise way of saying many currently vague, popular goals or standards for decisions.

We would like a way of specifying the standard for decision so that we know exactly what we are talking about. One that can be communicated to other people hopefully providing a language basis with which to resolve questions in a rational way that are currently being resolved arbitrarily by brute force. We would like a way of stating it that would be measurable enough to help resolve every day problems and replace the maximization of money as the existing value standard in the equations of systems analysts. In defining "an alternative controlled" we have taken a large step in this direction. It appears that capabilities can be measured in bits as information and entropy are measured. Developing techniques for doing this is the next area of research.

Since truth is a word used to describe models which accurately reflect reality, "seeking the truth" is the same thing as minimizing dissonance between reality and the model of it. As shown above, this is a process of maximizing capabilities.

Knowledge is contained in the models of individuals. The amount of knowledge an individual has can be measured by the number of capabilities s/he has. To seek knowledge is to maximize capabilities. This is also what it means to learn.

Since you are a conscious alternative control process and that includes the alternatives you control, maximizing your capabilities means maximizing yourself. "Expanding your consciousness" is another way this is expressed.

Progress could be defined as that which increases the capabilities of individuals. This would seem to be better than some vague notion of "bigger" or "more" technology.

Do more with less is a maxim made popular by R. Buckminster Fuller. One problem with this maxim as a standard for decision is it doesn't specify what is to be maximized. Maximizing capabilities could be a more practical way of saying it.

Individual freedom is measured in the number of "available" alternatives or capabilities. Thus, individual freedom is won by maximizing capabilities.

You cannot maximize your capabilities by restricting the capabilities of others. Those who try to maximize their own capabilities by limiting the freedom of others do so because they fail to see but a limited number of alternatives for themselves. They are blind to the process of time ensuring that there will never be a shortage. Such people, by limiting the alternatives of others, can keep the ecosystem of which they are apart from maximizing the alternatives it controls. Unstemmed, this results in the death of the system and therefore in their own death. People who limit the alternatives consciously controlled by others are therefore not only committing mass murder, they are committing suicide.

It is not surprising that the best way to maximize the alternatives you control is to maximize the alternatives controlled by the universe in general. The more alternatives you can provide for others to control, the more alternatives they will have to give, and be willing to give, in return.

Life is a process maximizing capabilities. To have as your ultimate goal to maximize your capabilities, is to have as your ultimate goal to live. Life is your standard for decision. This is to identify in a real, every day, motivational way with the process of universal life.

What does it mean to control the maximum alternatives?

It means controlling all of them. Doing everything with nothing. To be all knowing: omniscient. It is also logically impossible.

It is logically impossible to completely model the modeling process. In order to model the process doing the modeling, you need another model. And in order to model that, you need another one and so on in a never ending series of models of models. The best that can be done is to model different parts at different times. But it is impossible to model everything at the same time.

To seek to control the maximum alternatives is a futile goal. It can never be reached. The goal is to increase the accuracy of the model bearing in mind that it can never be completely accurate.

Techniques

1. Maximizing your capabilities is best accomplished by doing what feels good. You are a natural capability maximization process.

Happiness is one of the most frequent words people use to describe their ultimate goal. This may be because it is the emotion felt when dissonance has been successfully eliminated. Dissonance creates feelings of boredom, anxiety, and worry. Maximizing capabilities by eliminating the dissonance causing these feelings creates happiness.

2. Determine the alternatives you control and choose the one that controls the most.

One way of doing this is to view your future alternatives as an ant crawling up the "tree of time" might view the branches of the tree. The ends of the tree branches grow as fast as the ant can crawl. To maximize your capabilities, always pick the branch that contains the most branches.

3. Get together with others who are maximizing capabilities. Discuss techniques. Support or do research into techniques for measuring capabilites. Pass on this information to others. Notify us if you would like to get in touch with other such people interested in this standard for decision.

The purpose of this article has been to put forth a model towards a standard for decision. If this model creates unresolvable dissonance with your model, you can improve the accuracy or communication of the model. Please get in touch.

Since it is impossible for the model to ever be completely accurate, it is expected to evolve and grow more accurate by finding flaws and improving the model. One proposed medium for doing this is via the Whole Universe Catalog which was conceived, developed, and is maintained as an application of this standard for decision. See "The Whole Universe Catalog: a way of looking at things" by Kirk Kelley.

Summary of the SIGART Newsletter Experiment ####;

Summary of the SIGART Newsletter Experiment

For almost two years, from October 1972 until June 1974, the Augmentation Research Center conducted an on-going experiment as the online intermediary between the editing staff of the Newsletter for the ACM's Special Interest Group in Artificial Intelligence (SIGART) and the SIGART Newsletter's readership. 1

This included providing the following services. 2

1. Input of raw offline material into NLS
2. Translating online sections and reports written on other systems
3. Compiling, Editing and Formatting the Newsletter online
4. Printing a final copy formatted for paper
5. Storing both the copy formatted for paper and an optimal online format for "paperless" viewing
6. Providing an online retrieval system for both copies of each issue (People had access to the current issue as well as the "sneak preview" issue in preparation and all of the back-issues done online.)
7. Providing a dialog support mechanism allowing pointers to reader comments to automatically appear by the referenced section of the Newsletter. Online readers could insert and read comments, and comments on comments.
8. Keeping statistics on how often the online Newsletter was accessed. 2a

The SIGART Newsletter is published every-other month and contains approximately 50 pages the size of this page. There are several hundred members of SIGART and a paper copy of each issue is sent to each member. The number of members with access to the Arpanet, and hence, the online copy of the Newsletter, was unknown. Online accesses were in the realm of 20-30 per issue. The commenting capability was not used significantly. This was due in part to it's late introduction but mostly it was felt that among the number of users who read the Newsletter online, the desire to spend the effort necessary to use the commenting facility was not sufficient. At any rate, letters to the editor were sure to be published in the next issue where everyone, offline as well as online, would see them. 3

In the course of the experiment, the "Comment" program was written to augment the NLS dialog support system. Procedures were established for generating, maintaining, contributing to and accessing the newsletters. Much was learned about interfacing between offline and online formatted documents, interfacing between the desires of the editors and the capabilities of the system, and how to train people to use NLS for offline data-collection. 4

The Editors of the Newsletter were Steve Coles and Richard Fikes of the SRI Artificial Intelligence Center. The Online Editor was Kirk Kelley of the SRI Augmentation Research Center. In addition five typists were trained over the course of the experiment to use NLS for inputting raw offline material. 5

The experiment was terminated when the rotating Newsletter editorship left SRI. 6

Summary of the SIGART Newsletter Experiment

(J24274) 12-DEC-74 19:50;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JOAN( [ ACTION ] dpcs notebook) DPCS( [ INFO-ONLY ] ) ;
Sub-Collections: SRI-ARC DPCS; Clerk: KIRK; Origin: < KELLEY,
SIGART-REPORT.NLS;2, >, 12-DEC-74 19:49 KIRK ;;;;

1 THE WHOLE UNIVERSE CATALOG
An On-Line University with Access to the Universe 1

2 A WAY OF LOOKING AT THINGS
The Whole Universe Catalog (WUC [rhymes with luke]) is a way of looking at things. Imagine an electric typewriter in front of a portable TV on a table in your room. The TV is hooked to a two-way cable which connects to a network of computers. To the side of the typewriter is a hand-sized box with three buttons on top called a "mouse". The mouse has wheels in it which make it easy to roll across the table. As you roll the mouse around with your hand, a small spot on the TV moves around among the following words: 2

2a
Emergency
Health Care
Calendar
Mail
People
News
Want Ads
Yellow Pages
Bank Account
Legal Help
University
Entertainment 2a

2b Moving the spot to "University" and pushing the first of the three buttons on the mouse causes the previous view to be replaced by the following major branches of the University knowledge tree: 2b

2b1 University 2b1

2b1a Assumptions
Logic
Mathematics
Physics
Chemistry
Biology
Psychology
Sociology
Whole Systems 2b1a

2c You can continue pointing and pushing the first mouse button to see "outline" views of any branch or sub-branches. 2c

2d When you point to a word and push the second button on the mouse, the definition of that word appears. This is followed by a full view of the entire branch of information defined by that word. Pushing the first two mouse buttons at the same time displays the next "page" in the subject you have thus specified. 2d

2e Pushing the third button on the mouse takes you back to any view you have previously seen. 2e

3 LINKS TO TOOLS

References or "links" to related information in other subjects appear in the paragraphs. Pointing to these and pushing the first button displays the referenced information. The term "information" here is used loosely to refer to tools such as apples and scissors as well as information services such as games, computer tools, and voice-video links to people. 3

3a These tools include commands which allow you to compose, copy, digest, and produce the personal "information space" in which you can do knowledge work such as generating and publishing your own sections for the Whole Universe Catalog in your field of study. Among some of the tools currently available are one to send mail to those who are on-line and another to have pages automatically printed on the electric typewriter for off-line distribution. 3a

3b To use a tool available via the Whole Universe Catalog, point to or type the name of the tool you want. You are then passed on to that tool until you are finished using it. 3b

4 TYPING WORDS

Typing any word terminated by the Carriage Return key on the electric typewriter causes the definition of the word to appear the same as pointing to a word and pushing the second mouse button. Terminating the word with the first mouse button instead of a Carriage Return causes an "outline" of the branch of information defined by that word to appear. A range of search specifications can be typed in addition to single words. 4

4a Using the typewriter is not necessary to get information. You can access all of the information available via the Whole Universe Catalog simply by pointing to what you want to see with the mouse. A baby on a tricycle with three big buttons on the handle bars could use the Whole Universe Catalog accessing system with appropriately colorful picture information projected on a wall. On the other hand, WUC can be used exclusively from the on-line typewriter if you don't have a mouse. 4a

5 THE UNIVERSITY

The University section of the Whole Universe Catalog attempts to coherently provide a concise, accurate model of the universe with access to alternative models; a place to stand for observing and participating in the process of the universe and improving the range and accuracy of the model; a way of looking at things. 5

5a The University is structured for both a classical and an inter-disciplinary education depending on how the student wishes to learn a subject. Each branch of the knowledge tree points via links to supporting information classified in other branches. This allows the student to ask the inexhaustible "why?". For example, a student with a question about the basis of economics could easily end up taking a coherent path through the branches of politics, psychological motivation, biological psychology, biochemistry, atomic physics, calculus, set theory, and logic before satisfying his curiosity. 5a

5b Our current goal is to provide an interface to the universe which consists of information up to the level of first-year college survey courses. This is expected to take two person-years. 5b

6 THE SPACE-TIME CONTINUUM

One section of the Whole Universe Catalog called the "Space-Time Continuum" allows specification of any space-time location in the universe via pointing to maps in the center of the TV screen, or a list of dates on the side of the screen. At any point, a cultural over view of that place and time is available with the second mouse button along with links to sets of specific things such as music, architecture, news and literature. Within this coordinate system, there is a place for any specific thing and any general concept with many different alternative classification hierarchies for finding information easily. As of this writing, it is mostly a bare outline. It would take a person-year to turn the Space-Time Continuum into a meaningful educational tool. 6

7 VIDEO TAPE OF THE WHOLE UNIVERSE CATALOG.

A 30 minute video-tape of an on-line, real-time journey through the prototype Whole Universe Catalog is available. It takes you through the default cable television user's initial branch including glimpses of the places for news off the AP hot wire, entertainment, medical information, legal aid, shopping, bank accounting, want ads, personal, regional, and universal directories of people. A tour of the University follows with a free access education session crossing the inter-disciplinary boundaries of communications, economics, psychology, physiology, biology, chemistry, and nuclear physics. 7

7a There is a session in the Space-Time Continuum. Starting at the origin of the universe you watch it unfold to the present. Zooming down through maps of galaxies, stars, the solar system, to the earth, watch the continents drift as you travel across time to the middle ages. After viewing a textual overview, you move on to the present and view some of the current headlines. Going back to pictorial maps, you zoom in on continents, cities, streets, buildings, into the room where the WUC demonstration is being given, into the protected personal information of the demonstrator including medical records and references to the university that go through organs, into cells, down to molecules and finally to a view of the Hydrogen atom. Towards the end of the tape is a trip to the future plans for the Whole Universe Catalog as contrasted to some predictions by George Orwell. 7a

7b For a copy of the tape, send $30 or a blank 1/2 inch reel of video tape to Kirk Kelley, J2029 SRI, 333 Ravenswood, Menlo Park, CA 94025. 7b

8 GETTING ACCESS

The Whole Universe Catalog currently runs on various PDP-10 computers on the ARPA network. Officially the ARPANET is only to be used by DoD contractors though it is possible to find friendly people who will let you use their terminals, passwords, and accounts. We will try to publish a list of these people in the future. 8

8a The Augmentation Research Center sells the services of NLS (the environment in which WUC lives) to people that fit into the Augmented Knowledge Workshop Community. They are especially looking for non-DoD customers who will buy some portion (preferably all) of a "slot" on the computer. These slots currently cost approximately $40,000/year each. (That could be less than 20 cents a minute if it was sold by the minute.) This price is continually falling. Besides computer time, the price includes training in how to use NLS and a say in how it develops. Contact 8a

8a1 The Assistant Director in charge of Applications
Augmentation Research Center
333 Ravenswood
Menlo Park, CA 94025 (415) 326-9716 8a1

8b HARDWARE YOU NEED

You should have a teletype-like computer terminal and/or a display work station. A display work station consists of a video terminal, a mouse, a modem (interface for the terminal to the computer) and a box called a "Line-Processor" which ties the modem, CRT, and mouse together. There are many manufacturers of teletype-like terminals of which Texas Instruments (TI) in San Antonio, Texas is one of the best. Cheap video terminals that work are the Hazeltine "2000", Datamedia "Elite 2500", and Lear Siegler "ADM". They cost about $1000 each or rent for around $100/mo. The mouse ($300) and Line Processor ($2500) are both available from Cybernex, 922 Industrial Av., Palo Alto, California. Datamedia corporation is at 7300 N. Crescent Blvd., Pennsauken, NJ 08110. 8b

9 WEC, LWEC, WUC and WEE

The name "Whole Universe Catalog" was picked to answer the question "What comes after the Last Whole Earth Catalog?" This was some time before the Whole Earth Epilog appeared. 9

9a In a warm-up version of WEE in Harper's wraparound under the title "EDITOR BREAKS PROMISE" Stewart Brand lists four reasons for resurrecting the Catalog. The last one is "After burning our bridges we reported before the Throne to announce, 'We're here for our next terrific idea.' The Throne said, 'That Was It.'" 9a

9b Time to look around for a better throne? 9b

9c It should be noted that WUC has no official relations with WEC or WEE and was conceived and developed to it's current stage independent of the Point foundation (though Richard Austin, a member of Point, did come out and take a look in 1974). 9c

9d Though their functions overlap, the emphasis is different in WUC partly due to the tremendous differences in media. Since it lives in an on-line environment, and does not have the limitations of paper, WUC is meant to be more encompassing than WEC though computers certainly have their own limitations. The Whole Universe Catalog was conceived (before its name) to be more of a free access holistic education tool and on-line library, rather than an access to tools for an alternative life style for which WEC has developed a reputation. WEC is also concerned with self teaching and even begins with the "Understanding Whole Systems" section. To complete the overlap, WUC certainly has lots of room for providing access to tools for an alternative life style. 9d

9e What is copied directly from WEC to WUC are 1) the integrated, holistic way of viewing and comparing things, 2) providing alternatives to, and a co-evolution with, the existing system, and 3) the spirit behind the business honesty which resulted in the tradition of publishing an overview of the money accounts at the end of every issue. 9e

The Whole Universe Catalog: a way of looking at things

(J24275) 18-JUN-75 20:19;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Sub-Collections: SRI-ARC; Clerk: KIRK;

Froufra

Never count your chickens until you catch your rooster. (You may add that to other eggs I'v laid) 1

JAKE 23-OCT-74 19:20

Froufra

(J24283) 23-OCT-74 19:20;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /JHB( [ ACTION ] immediately if not sooner!) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

New Jornal header; alternatives; call for comments and additional input.

Comments received by FRiday 25Oct74 will be included for next pass which will include the KWAC. Thank you for contributing.

New Jornal header: alternatives; call for comments and additional input.

Various alternatives for a new journal header. 1

One consideration is to allow each person to pick his own form. This would be stored in his ident record. The journal system (already having it loaded) would read the reference format name and use the appropriate "rel" file. If the "rel" file is not known or if none is specified, a default form will be used. 1a

There are two questions, what are the fields of information desired and what are the formats of these fields. These question are partially separate but not wholely. 1b

Some overall criteria to use for determining what fields are useful and where they should go. 1c

Citation should be as short as possible. 1c1

A form which is suitable for sorting. 1c2

Conform to other citations in the literature. 1c3

Take advantages of level and line capabilities. 1c4

Although a new journal system might not developed for some time, pressure for maintaining the same format will be very high and thus the future situation should be heavily considered. 1c5

e.g. multiple hosts 1c5a

Below is a list of various fields, synoyms on the left. Arguments for or against this field to be included are found on the levels below it. 1d

AIDENT author ident 1d1

PRO: Considered by most to be very important and should be on first line. 1d1a

ORG acronym of author's organization 1d2

PRO: Easily obtained from ident record (which is available to journal system). 1d2a

PRO: For a large user community, the organization might be more meaningful than the person's ident or even his full name. 1d2b

PRO: This field is almost always included in most citations in the literature. 1d2c

NAME last name or full name 1d3

PRO: Most citation in the literature include full last name and initials 1d3a

CON: With the ident system one only needs to show record for ident. 1d3b

DATE day, month, and year when mail item was sent. (dd-mm-yy) 1d4

PRO: Used by many as a sort parameter. 1d4a

CON: Should not be on first line since it does not add to the recognition of what this mail item is. For those you like to sort on it, programs can be coded even if it appears on a second or third line. 1d4b

TIME time that the mail item was sent (xx:xx:xx); 24 hour clock or AM/PM 1d5

PRO: Gives another means for uniquely (almost) identifying mail item (especially sndmsg items). 1d5a

ZONE time zone 1d6

PRO: Users of the system are scattered among various time zones. 1d6a

PRO: It is possible that different computers might be in different time zones. 1d6b

CON: The journal system should maintain only one time zone for dating of mail items. 1d6c

DAY day of the week that mail item was sent (MON TUE ....) 1d7

RDATE: date and time when mail item was received. 1d8

CON: Not needed, one can use the signature of the statement. 1d8a

PRO: Signatures are not widely known and are costly for TNLS users. 1d8b

JNUM journal number 1d9

PRO: Useful for sorting and uniqueness of item; only of value in first line, 1d9a

CON: Duplicates information in LINK, 1d9b

CON: LINK might be better on first line, 1d9c

LINK complete journal reference in form of link 1d10

PRO: Should be in first statement of citation in order for jump to link to work when only first line is bugged or referenced, 1d10a

PRO: Even for messages that are delivered with the citation it should be present. This enables one to delete the message and still have the link, 1d10b

PRO: For messages that are delivered with citation ((ncluding SNDMSG mail) this should be a link with only viewspecs that opens up the view to show the whole message, This is predicated on the user having only a clipped view initial, 1d10c

TITLE the title or subject of mail item 1d11

PRO: Nearly everyone agrees this is the most useful field, 1d11a

COMMENT comments 1d12

PRO: to be put at a level below main citation, 1d12a

DIST distribution list of idents 1d13

TO distribution list of idents receiving mail as action 1d14

PRO: Distinction should be made between list of people receiving item and those receiving an information only copy, 1d14a

CC distribution list of idents receiving mail item as information only 1d15

PRO: Distinction should be made between list of people receiving item and those receiving an information only copy, 1d15a

CON: Can use uppercase and lowercase to distinguish action and info copies, 1d15b

PRO: For uppercase only teminals, uppercase/lowercase will not distinguish 1d15c

TYPE the word ACTION and INFO, for action or information 1d16

PRO: For a shorter citation just a short word might suffice 1d16a

PRO: Useful on the first line of citation to quickly determine whether to read mail or not. 1d16b

PRIV privacy type (priv or public or blank) 1d17

PRO: Might be nice to know if item is private or not. 1d17a

ORGIN name of originating host computer 1d18

PRO: might be nice when there are many hosts (Office-1,2,3,4,5...) 1d18a

REF references 1d19

PRO: Often used in many memos and useful for recipent. 1d19a

KEYW keywords 1d20

Leading contenders for the new journal citation format and comments (author ident at beginning). 1e

(JEW) provide several options and have the journal use your prefered format. (This is clearly the way to go.)(KIRK)(CHI)(RLL) 1e1

In addition, have a special directory containing userprogram formats ok'd by the journal programmer. 1e1a

(NDM) JFORM3.CA (matches MESSAGE.SUBSYS format) 1e2

DATE TIME AUTHORIDENT: The title begins here terminated by a CR and 3 spaces
Distribution: ACTION IDENTS ARE UPPER CASE info only idents are lower case
Received at: 12-OCT-74 04:31
(JJOURNAL,12345,1:w) 1e2a

Text of Message is a substructure statement. Note and Comments are also seperate statements in the order listed below. Note: in the statement above this, indentation does NOT represent a change in level. 1e2a1

Note: [ ACTION ] 1e2a2

Comments: Comments would appear last. 1e2a3

A Journal link in the following form would replace the 6th line for documents. 1e2b

Location: (MJOURNAL, 12345, 1:w) 1e2b1

(KEV) modification of jform2 1e3

AUTHOR-IDENT: The title here would be terminated by a carriage return
DAY DATE TIME <Message -- 12345,>
TO: myident( comment to me ) BUGS abc def 1e3a

Comment: date and time would contain the day of the week. 1e3a1

Message: The message occurs after the comment and is a statement in the substructure. For Journal links, <Message-- 12345,> would be replaced by <JJOURNAL, 12345,> 1e3a2

(JHB) places most parameters on first line including beginning of title. 1e4

DATE SENT (ONLY) AUTHOR(S) JNUMBER The title begins here terminated by a CR and 3 spaces
   Received: TIME DATE; Sent: TIME
   TO: Idents of recipients for action followed by a CR and
3 spaces
   CC: Idents of recipients for info only followed by a CR
and 3 spaces
   Link or message is appended to citation here. Note
indentation does not represent a change in level. 1e4a

New statement here is for comment. 1e4a1

New statement here is for notes. 1e4a2

(KIRK) closely resembles standard reference formats 1e5

AUTHOR-IDENT, Title begins here after a comma and has no CR following it. <JJOURNAL, 12345,> SITE DAY DATE TIME 1e5a

Distribution: UPPER CASE IDENTS FOR ACTION lower case idents for info-only 1e5a1

Note: this is where a note would appear. 1e5a2

Comment: this is where comments would appear. 1e5a3

For messages, "<JJOURNAL, 12345,>" is replaced by "Journal Number 12345" and the text of the message is located here. Distribution, Note, Comment, and Message are all separate statements in substructure. 1e5a4

(xxx) Just to see if all fields can fit. 1e6

AIDENT ORG TITLE, LINK, DATE TIME ZONE DAY ORGIN <CR><SP><SP><SP> RDATE TYPE PRIV<CR><SP><SP><SP> TO: identlist<CR><SP><SP><SP> CC: identlist[next level down] COMMENT[next statement] REF[next statement] KEYW[next statement]Message (if delivered with citation) 1e6a

1e6b

Example. 1e6c

```
RLL (SRI-ARC) A Note on the future of journal headers,
(JJOURNAL,12345,1:w), 22-OCT-74 1332 PDT WED at OFFICE-1
   Received at: 22-OCT-74 1356 PDT   for ACTION (PRIVATE)
   TO: ABC DEF GHI
   CC: JKL MNO PQR SRT
```
1e6d

Comments: Just a test for fun. 1e6d1

REFERENCES: (MJOURNAL,34567,1:w) 1e6d2

KEYWORDS: test, journal, header 1e6d3

New Jornal header: alternatives; call for comments and additional input.

(J24284) 23-OCT-74 22:38;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /SRI-ARC( [ ACTION ] ) ; Sub-Collections: SRI-ARC; Clerk: RLL;

bug: playback record/ calculator

In using the playback command with the calculator, I happened to bug an invalid number. The system respondded wth that message. Left my playback hangingg. I tried contrrl O. this cause the system to loop on "?". additional ^O did not good (waited 5 minutes wall time). This happened twice so is repeatable. In future I will be more careful but I thought ^o would free me from playback. 1

bug: playback record/ calculator

(J24285) 23-OCT-74 23:08;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FDBK( [ ACTION ] ) JCN( [ INFO-ONLY ] ) ; Sub-Collections: SRI-ARC; Clerk: RLL;

Faileur to Properly Journalize the NSW Proposal

It's my fault that the NSW proposal is not online under its correct number (23352), It was originally printed with the number preassigned to Mil and a simulated journal header for publicatioin purposes. Sometime in September Dick asked Joan to journalize it corrctly. She tried to do so with my help, but the Journal system was suffering from a bug at the time so that when it failedto act on our request for Mil's preassigned number and instead gave us a new number, it did not give an error message, Some time later we discovered we had failed, I did not get arround to trying again unil, as a matter of fact, yesterday. The journal will normally grant access to a preassigned number either if the number is assigned to an author or if th sender is connected to the assignee's directory, Of course Mil is not an author and I discovered yesterday that her directory no longer exists, I expect this journal item will reach Dave Hopper and he will advise m how to proceed, When I hear from him I will journalize the proposal under the right number forwith, 1

Faileur to Properly Journalize the NSW Proposal

(J24290) 24-OCT-74 08:41;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JDH( [ ACTION ] ) JOAN( [ ACTION ] please put this in he DPCS notebook) DCE( [ INFO-ONLY ] ) RWW( [ INFO-ONLY ] ) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN;

re J24269: Journal notification & info/action branches

I agree with Dirk's notion aout these 2 new features, particularly info and action. These are arbitrary categories at best, and serve to force additional compensatory decisions on the real users. IT's difficult not to be snide, but I don't recall any debate or even discussion about this.
I hope that the items that have been sent by myself and others on the Journal citation format are ok. If not let's have some discussion on them ! 1

re J24269: Journal notification & info/action branches

(J24291) 24-OCT-74 08:56;;;; Title: Author(s): James H. Bair/JHB; Distribution: /FDBK( [ ACTION ] ) HGL( [ ACTION ] ) JDH( [ ACTION ] ) DVN( [ ACTION ] ) RLL( [ INFO-ONLY ] ) ; Sub-Collections: SRI-ARC; Clerk: JHB;

Is Documentation Holding Up NLS-8?

I feel I have lost touch with why NLS-8 has not come up as the running system at Offie-1. If it is waiting for documentation I would like t know about it. The state of documentation is essentially as described in (mjournal,24247,2) except that a draft of the two-page document (mjournal,24247,2h) now exists and a draft of the revised viewspec cards (JRNL22, J24266:gw) and (journal,jrnl22,j24262) is in review. 1

Is Documentation Holding Up NLS-8?

(J24292) 24-OCT-74 09:04;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /RWW( [ ACTION ] ) JCN( [ ACTION ] ) JOAN( [ ACTION ] please put this in te dirt notebook) DIRT( [ INFO-ONLY ] ) ; Sub-Collections: DIRT SRI-ARC; Clerk: DVN;

Is Aything Happening on COMing the DCA Paper?

Is anything happening? Since I don't presently have anything else to go to COM, I'm not planning to send the file you made tonight unless I hear from you otherwise. 1

Is Aything Happening on COMing the DCA Paper?

(J24293) 24-OCT-74 09:28;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /SRL( [ ACTION ] ) JOAN( [ ACTION ] Would you add this to the dpcs notebook please?) JCN( [ INFO-ONLY ] ) NDM( [ INFO-ONLY ] ) EKM( [ INFO-ONLY ] ) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN;

journal headers

< POSTEL, JUNK.NLS;1, >, 24-OCT-74 09:40 JBP ;;;; 1

I like the example shown last in your message with the following changes: use the Authors last name on the first line and add a FROM: line for his ident. 1a

Postel (SRI-ARC) A Note on the future of journal headers, (JJOURNAL,12345,1:w), 22-OCT-74 1332 PDT WED at OFFICE-1
Received at: 22-OCT-74 1356 PDT for ACTION (PRIVATE)
TO: ABC DEF GHI
CC: JKL MNO PQR SRT
FROM: JBP 1b

Comments: Just a test for fun. 1b1

REFERENCES: (MJOURNAL,34567,1:w) 1b2

KEYWORDS: test, journal, header 1b3

journal headers

(J24294) 24-OCT-74 09:58;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /RLL( [ ACTION ] ) ; Sub-Collections: SRI-ARC; Clerk: JBP;

<POSTEL>MTR6722.NLS;1, 26-SEP-74 13:12 JBP ;
Survey of Network Control Programs in the ARPA Network

## 1.0 Introduction

### 1.1 Purpose of this report

This report describes the function of a Network Control Program (NCP) in the Advanced Research Projects Agency Computer Network (ARPANET), and surveys several representative implementations of such programs. This work was accomplished as part of MITRE task 810A.

The network control program is the operating system module that interfaces user programs to the communications network by providing system calls to invoke communications functions specified by the network wide host to host protocol.

The purpose of this report is to document the implementation of the network control program and the optional strategies used in different implementations. This report should be of use to individuals implementing network control programs for other computers, and to individuals designining network control programs for other computer networks.

The survey investigated network control program implementations in eleven systems: TENEX at Bolt Beranek and Newman (BBN); Multics at Massachusetts Institute of Technology (MIT); the IBM systems at University of California, Santa Barbara (UCSB), University of Califoria, Los Angeles (UCLA), The RAND Corporation (RAND), Systems Development Corporation (SDC); the Burroughs system at University of California, San Diego (UCSD); the DEC system at the Center for Computer-based Behavorial Studies (CCBS) at UCLA; the Control Data system at Lawerence Berkeley Laboratory (LBL); the ARPA Network Terminal System (ANTS) at University of Illinois(UI); and the Terminal Interface Processor (TIP) at BBN.

The following table summarizes the organizations (places), operating systems, and computer hardware included in the survey.

| Place | System | Computer |
|---|---|---|
| BBN | TENEX | DEC PDP 10 |
| MIT | Multics | H 6180 |
| UCSB | OS/MVT | IBM 360/75 |
| UCLA | OS/MVT | IBM 360/91 |
| RAND | OS/MVT | IBM 370/158 |
| SDC | VM | IBM 370/145 |
| UCSD | MCP | B 6700 |
| CCBS | DEC | DEC PDP 10 |
| LBL | BKY | CDC 6600 |
| UI | ANTS | DEC PDP 11 |
| BBN | TIP | H 316 |

## 1.2 ARPANET Overview

The ARPANET is an advanced computer communications system connecting together a set of computer centers in the United States and Europe. The following description of the network touches on several aspects: the physical implementation, the scope and size, and the functional goals.

The ARPANET is implemented using packet transmission technology. At each network site there is an Interface Message Processor (IMP) which is a store and forward packet routing computer. Each IMP is connected to between 1 and 5 other IMP's via common carrier circuits. These circuits are normally 50 kilobit per second channels. The IMP's are Honeywell 316 or 516 computers modified and programed by Bolt, Beranek, and Newman (BBN) <Heart>. Also connected to an IMP may be between 0 and 4 hosts. A host is a computing sysyem which currently ranges in size from a DEC PDP11 to a IBM 370/195. A Terminal Interface Processor (TIp) is an IMp and a minihost combined in one processor <Ornstein>.

The ARPANET has grown almost continuously from the time it began with the installation of the first IMP in September 1969. As of this writing the network consists of 48 IMPs of which 21 are

TIPs, and 54 hosts. The network extends from Hawaii in the west to London and Norway in the east. One communication channel has been upgraded to 230.4 kilobits per second, and the communications between California and Hawaii, and between the United States and Europe are via satellite channels. The satellite channel to Europe is 7.2 kilobits per second, while the satellite channel to Hawaii is the normal 50 kilobits per second.

The goal of the ARPA computer network is for each computer to make every local resource available to any computer in the network in such a way that any local program available to local users can be used remotely without degradation. That is, any program should be able to call on the resources of other computers much as it would call a subroutine. The resources which can be shared in this way include software and data, as well as hardware <Roberts>.

The process of sucessful communication requires the use of some rules of behavior in order to permit the communicating entities to properly interpret the conversation. These rules of behavior may include both constraints on the sequencing of the units of conversation, as well as the structure and content of the communication (e.g. the grammar and the meaning). In the ARPANET these rules of communication behavior are called protocols.

Communications in the ARPANET are of two types; those associated with two directly connected entities (e.g. IMP to IMP, IMP-host, system-process) and those between more widely separated entities (e.g. system to system, process to process). This second type of communication is sometimes said to be supported by a virtual communications channel. For example the virtual process to process communication channel is really a process-system, host-IMP, IMP to IMP, IMP-host, system-process channel.

The protocols in the ARPANET build up the capabilities of the network in a series of levels or layers. The lowest of these is the IMP to IMP protocol which provides for reliable communication among the IMPs. This protocol handles transmission error detection and correction, flow control to avoid congestion, and routing.

The next level is the IMP-host protocol which provides for the passage of messages between hosts and IMPs in such a way as to create virtual communication paths between the hosts. With the IMP-host protocol, a host has operating rules which permit it to send messages to specified hosts on the network and to be informed of the dispensation of those messages. In particular, the IMP-host protocol constrains the hosts in their transmissions in order to make make good use of available communications capacity without denying such availability to other hosts.

The next higher level is the host to host protocol, implemented by the network control program. The host to host protocol is the set of rules whereby hosts construct and maintain communications between user processes running on separated computer systems. One process requiring communications with another on some remote computer system makes requests on its local operating system to act on its behalf in establishing and maintaining those communications using the host to host protocol <Crocker>.

If this brief introduction to the ARPANET is not sufficient, the reader is urged to turn to the references <Heart>, <Roberts>, <Crocker>, and <Ornstein>, before reading the body of this report.

## 1.3 Report Organization

The report is organized to first discuss the general characteristics of operating systems and network control programs, and then to discuss the specific implementations surveyed.

First is a discussion of the purpose of a network control program. To place this in some context the relevant characteristics of an operating system are discussed.

A network control program is then discussed in abstract terms, that is without reference to any particular machine or operating system. This description of an idealized NCP is then used as a reference in the descriptions of the surveyed systems.

The findings of the survey are presented next, first in terms of the system structure and NCP implementation strategy, and finally in terms of the physical characteristics of the various implementations.

The final section summarizes the findings of the survey.

At the end of the report there is a glossary of terms and a list of references.

## 2.0 The Network Control Program (NCP) Environment

This section discusses the purpose of the NCP and how it relates to the operating system and other protocols.

### 2.1 Purpose of a Network Control Program

The function of network control program (NCP) is to implement the host to host protocol. That is the NCP is to provide a common interface across the various operating systems to the user level processes, and to provide to the operating systems the means to communicate among themselves the control information necessary to establish, regulate, and terminate the communication between user processes.

The development of the network in the computer science research environment provided an ample collection of well developed interactive timesharing systems for the initial set of systems to become part of the network. These early interactive systems generally possessed a process structure and interprocess communications mechanism. It seemed that the network should naturally extend these interprocess communications mechanisms to allow communication between processes in different systems. There were some difficulties, however, for the various systems did not all have a uniform scheme for interprocess communication or even a common way of naming the destination of a communique, Several of the systems that were later added to network did not allow communication between processes, indeed, a few systems did not support the concept of a process (or so their programmers claimed).

A standard interprocess communication mechanism and a standard naming scheme were needed, Further there was a need for a common language such that the operating systems of the various hosts could talk to each other about the interprocess communications they support,

These needs are filled by the host to host protocol as implemented by the network control programs of the various hosts, The host to host protocol specifies a language of commands with parameters which are exchanged between NCPs to arrange, manage, and terminate process to process communication, The host to host protocol also specifies a common name space called sockets for indicating the source and destination of interprocess communications, The host to host protocol provides an interprocess communication mechanism called connections.

2.2 Relationship of the NCP to Telnet and File Transfer

Two functional capabilities are desirable in a computer network: to be able to use interactive terminals with programs on remote computers "as if you were there", and to transfer between computers large collections of data or files (which may be programs or data), The first capability is provided by the Telnet protocol and the second by the File Transfer Protocol (FTP) <NIC7104>,

2.2.1 Telnet

The implementation of Telnet is to the NCP almost indistinguishable from any pair of communicating processes in the network, At the computer where the human user sits at his terminal there is a program called User-Telnet which talks to the terminal on one side and talks to the NCP on the other side, At the computer where the serving program is located there is a program called Server-Telnet which talks to the NCP on one side and talks to the serving program on the other side as if it (Server-Telnet) were a terminal. This last requirement, that the serving program believes it is talking to a terminal and not the network, is a tricky one. Some systems have constructs which allow a process to act as a terminal to another process, in these systems this is a simple requirement to implement. In other systems however the implementation of this capability has been so difficult that the Server-Telnet function has been implemented in the system with the NCP. The importance of this requirement is that it permits programs constructed for use from interactive terminals with no thought of the network to be used by remote users via the network.

### 2.2.2 File Transfer

The File Transfer Protocol (FTP) is designed to fill the need for a mechanism to transfer files containing programs or data between computers. The FTP utilizes a Telnet connection (pair) to allow the exchange of control information (requests and replies) and a separate data connection for the actual transmission of the file. The NCP is not normally aware of the fact that this data connection is used for the FTP.

## 2.3 Description of an Operating System

An operating system consists of program modules which augument the hardware and provide an environment for processes. Among the operating system modules of interest are a terminal control program (TCP), a file control program (FCP), and of course, a network control program (NCP). The interfaces between these operating system modules and user processes take the form of system calls and returns, and sometimes pseudo interrupts. System calls are implemented in a variety of ways, but often it is a special hardware instruction that invokes a system call (e.g. SVC, UUO, JSYS, MME). In higher level programming languages a system call is often indistinguishable from a subroutine call.

In some cases the form of the system call is quite different for each different module of the system.

A process is a program in execution with its associated address space, a location counter, some general registers, and usually some open files (or devices) . Processes may be created by users, though there are often processes which have been programmed by systems programmers for particular functions, and some of these may be initiated by the system when it begins running. Some processes may have access to greater (or lesser) capabilities than those created by normal users. In general there may be elaborate regulations provided by the system to control passing of the capability or permission to access particular resources between processes.

One important aspect of an operating system that has a great impact on the implementation of network functional capabilities is the provision for interprocess communication. Generally processes are viewed as independent computational units that need interact only with the operating system in a few very constrained ways (i.e. via system calls), however often it would be useful to build a new capability based on a combination of existing programs. One way of extending the usefulness of the process structure is to allow processes to communicate between themselves such that several processes may cooperate to accomplish a computational goal. The form of communication supported by an operating system very much influences the extent to which processes actually cooperate and, therefore, the extent to which use of the ARPANET is a natural extension of the programming enviroment. The network host to host protocol seeks to make available to processes a particular form of interprocess communication called connections.

There are many other features and essential functions of operating systems which will be ignored in this discussion because they are not relevant to the implementation of an NCP. Any multiprogramming or multiple process system must have a scheduler, some form of memory management, and provide for accounting, security, protection, and privacy.

## 3.0 Description of a Network Control Program

This section begins with a general overview of the operation of a network control program and gradually refines the definition. The functional components of an NCP are described first in a general way followed by a description of a typical set of system calls for the user-NCP interface and a description of the operation of the NCP and finally a detailed description of the functions of each NCP component.

### 3.1 NCP Functions:

The NCP must provide several functions to interface between the user process on one side and the IMP on the other side. Among these functions are device handling, formatting, error control, flow control, multiplexing, and synchronization.

The IMP is connected to the host computer much as any input or output device (since the IMP is a full duplex device it might be interfaced as two simplex devices). This implies that there must be in the lowest levels of the system a program module to control the IMP interface on the input output instruction and interrupt level.

The IMP-host protocol requires that a standard format be used for messages exchanged between the host and the IMP. The standard format has a 32 bit leader at the beginning of each message that contains control information indicating such things as the message type, source or destination host and logical link number. In addition to the leader the host to host protocol requires an additional 40 bits of prefix information for each message. This additional prefix indicates the byte size and number of bytes in the text of the message.

As data becomes available to send to the IMP, it is formatted into messages and queued for transmission to the IMP, and as messages are received from the IMP, they are queued for processing.

The messages received from the IMP are of various types. The two most frequently received are REGULAR and RFNM. A REGULAR message is used to transfer data. A RFNM (request for next message) is used to indicate that the previously sent message on this logical link to this host was sucessfully received by the destination IMP. There are several other message types to indicate error conditions.

The REGULAR messages are of two catagories: user data or NCP to NCP control information. These are distinguished by the logical link number in the leader. All NCP to NCP control messages are transmitted on logical link zero. All user data messages are transmitted on a logical link number in the range 2 through 71.

The NCP control messages have several functions: to establish connections between pairs of processes in the network, to regulate the flow of data over these connections, to terminate connections, and to convey some special signals between the NCPs.

The leader of a data message must be examined to determine to which user process buffer the text of the message should be appended.

The user process interacts with the network by issuing system calls to the NCP to establish, use and terminate connections. When the user process issues calls which cause the NCP to send data to a distant process the NCP must include information in the leader that will enable the receiving NCP to determine for which process the data is intended.

The sending and receiving processes and hosts might not operate on the same sized quanta of information and they might operate at differing speeds. In such a situation it is natural to use buffers to smooth the flow of information and to allow each entity to operate using its prefered quantum size. It is the responsibility of the NCPs to manage these buffers and to regulate the flow from sending to receiving host such that the receiving host is able to buffer all the data sent with out undue difficulty.

Many of the NCP control messages received and many of the system calls will require the NCP to send NCP control messages to the foreign host.

The NCP must maintain information about each active connection in a connection table. The elements of each entry in this table include information about the user process using the connection, the foreign host, the buffer location and fullness, and the state of the connection.

If any errors are detected the NCP must act to protect itself from harmful consequences, but it must also act to provide reliable service to all the user processes. In any case the NCP should record the relevant information about the error and the circumstances (e.g. time and day). The NCP must also report abnormal events to the the computer operator and be able to receive instructions from the computer operator.

The NCP should make available to users and the computer operator the status of hosts and connections.

The NCP should gather statistics on the usage of various elements of the protocol and resources allocated to it (e.g. buffers).

3.2 System Calls

The following discussion of the NCP and system calls is at a level comparable to that in the basic specifications of the IMP-host protocol <BBN1822> and the host to host protocol <McKenzie>. Note that the following sections are modeled closely on Network Working Group Request for Comments note number 55 <Newkirk>.

The system calls assumed to be available to user written processes are described.

LISTEN(PORT,AEN,CODE)

The local socket of this process with this AEN is associated with this process PORT. A return value is given in CODE. If there is a pending call the connection may be opened immediately and the process notified, if there is no matching pending call the NCP will notify the process when a matching RFC arrives.

CONNECT(PORT,AEN,FS,CODE)

The local socket of this process with this AEN is associated with this PORT, and the specified foreign socket (FS) is also associated with this local socket, defining a connection. If there is a pending call matching these parameters the connection is opened and the process so notified. A return value is given in CODE. If there is no matching pending call the NCP communicates this request to the foreign host and notifies the local process when a matching request is received and the connection is opened.

SEND(PORT,BUFFER,LENGTH,CODE)

The data starting at BUFFER and extending LEGNTH bits is transmitted on the connection associated with this PORT in accordance with the allocation values. CODE is set with a return value.

RECEIVE(PORT,BUFFER,LENGTH,CODE)

Data received on the connection associated with this PORT is stored into the processes address space starting at BUFFER an extending for LEGNTH bits. A return value is set in CODE.

CLOSE(PORT,CODE)

Activity on the connection associated with this PORT is stopped. A return value is set in CODE.

INTERRUPT(PORT,CODE)

A special interrupt signal refering to the connection associated with this PORT is sent on a logically parallel data path. A return value is set in CODE.

STATUS(PORT,INFO,CODE)

The relevant status information from the connection table entry associated with the PORT is returned in INFO. A return value is set in CODE. This allows a user program to monitor the state of a connection, of special interest are the allocation values and the NCP buffer used and free values.

### 3.3 NCP Operations

Presented here are discriptions of the operations conducted during the three major phases of network usage: opening, communicating, and closing.

#### Opening

In order to establish a connection for data transmission, a pair of RFC's must be exchanged. An RTS must go from the receive side to the send side, and an STR must be issued by the send side to the receive side. In addition, the receive side in its RTS must specify a link number, and the send side in its STR must specify a byte size. These RFC's (RFC is a generic term encompassing RTS and STR) may be issued in either order.

A provision must also be made for queuing pending calls (i.e. RFC's which have not been dealt with by the user program). Thus, when a user is finished with a connection, he may choose to examine the next pending call from another process and decide to either accept or refuse the request for connection. A problem develops because the user may choose to not examine his pending calls; thus they will merely serve to occupy queue space in the NCP. Several alternative solutions to this problem are discussed later.

Utilizing the framework of the typical system calls described above, at least four temporal sequences can be envisioned for obtaining a successfully opened connection:

The user process may issue a LISTEN indicating that it is willing to connect to any process which sends an RFC specifying this local socket. When an RFC of interest arrives the NCP responds with a matching RFC and notifies the user processs of the now open connection. The user can, of course, inspect the parameters of the connection (using the STATUS system call, for example) to determine if it really wants the connection, and if not the user can CLOSE the connection.

If upon processing a user request for a LISTEN, the NCP discovers that a pending call exists for this local socket, the NCP immediately sends the matching RFC and notifies the user of the open connection.

The user may issue a CONNECT, specifing a particular foreign socket that he would like to connect to. An RFC is issued. If the other NCP accepts the request, it answers by returning an RFC. When this acknowledging RFC is received the connection is opened.

When processing the CONNECT, the NCP may discover that a pending call exists from the specified foreign socket to the local socket in question. An acknowledging RFC is issued and the connection is opened.

In all the above cases the user is notified when the connection is opened, but data flow cannot begin until buffer space is allocated and an ALL command is transmitted.

Any of these connection scenarios will be interrupted if either the other NCP sends a CLS command when an RFC is expected or the user issues a CLOSE system call before the connection is opened, as discussed under Closing.

Communicating

Data can only flow when a connection is fully opened (i.e. when two RFC's have been exchanged). It is assumed that the NCP's have buffers for receiving incoming data and that there is some meaningful quantity which they can advertise on a per connection basis in ALL commands indicating the amount of data they can handle. It is noted that the sending side regulates its transmission according to that amount.

When a connection is opened, a connection table entry field called their-allocation-values is set to zero. The receive side will decide how much space it can allocate and send an ALL message specifying that space. The send side will increment their-allocation-values by the allocated space and will then be able to send messages of length less than or equal to their-allocation-values. When messages are transmitted, the length of the message is subtracted from their-allocation-values. When the receive side allocates more buffer space (e.g. when a message is taken by the user, thus freeing some system buffer space), the number of bits newly available is sent to the send side via an ALL message.

Thus, their-allocation-values is never allowed to become negative and no transmission can take place if their-allocation-values equals zero.

Notice that the lengths specified in ALL messages are increments not the absolute size of the receiving buffer. This is necessitated by the asychronous nature of the flow control protocol. The values in the ALL command can be quite large, thus providing the facility for an essentially infinite bit sink, if that may ever be desired.

Closing

Just as two RFC's are required to open a connection, two CLS's are required to close a connection. Closing occurs under various circumstances and serves several purposes. To simplify the analysis of race conditions, four cases are distingusished: aborting, refusing, termination by receiver, and termination by sender

A user aborts a connection when he issues a CONNECT and then a CLOSE before the connection was opened. Typically a user will abort following an extended wait for the acknowledgement; the NCP may also abort for him if he blows up.

A connection is refused when the NCP sends a CLS as a response to an arriving RFC. This may occur if a user has issued a connect and an RFC arrives from some other foreign socket.

After a connection is established, either side may terminate. The required sequence of events suggests that attempts to CLOSE by the receive side should be viewed as requests which are always honored as soon as possible by the send side. Any data which has not yet been passed to the user, or which continues over the network, is discarded. Requests to CLOSE by the send side are honored as soon as all data transmission is complete.

Aborting

Three cases are distinguished:

In the simplest case an RFC is sent followed later by a CLS. The other side responds with a CLS and the attempt to connect ends.

The foreign process may accept the connection concurrently with the local process aborting it. In this case, the foreign process will believe the local process is terminating an open connection.

The foreign process may refuse the connection concurrently with the local process aborting it. In this case, the foreign process will believe the local process is acknowledging its refusal.

Refusing

After an RFC is received, the local host may respond with an RFC or a CLS, or it may fail to respond. (The local host may have already sent its own RFC). If the local host sends a CLS, the local host is said to be refusing the request for connection.

CLS commands must be exchanged to close a connection, so it is necessary for the local host to maintain the connection table entry until an acknowledging CLS is returned.

Termination by the Receiver

When the user on the receive side issues a CLOSE system call, his NCP accepts and sends a CLS command immediately. Data may still arrive, however, and this data should be discarded. The send side, upon receiving the CLS, should immediately terminate the data flow.

Termination by the Sender

When the user on the send side issues a CLOSE system call, his NCP must accept it immediately, but may not send out a CLS command until all the data in the local buffers has been passed to the the foreign host. It is thus necessary to test for both buffer empty and RFNM received before sending the CLS command. The CLS must be acknowledged before the connection table entry can be deleted.

In this presentation several topics have been mentioned which should be further explained, among these are pending call queues, and connection states.

Pending Call Queues

It is essential that some form of queuing for pending RFC's be implemented. A simple way to see this is to examine a typical connection establishment sequence. One side issues a LISTEN, the other a CONNECT. If the LISTEN is issued before the RFC coming from the remote CONNECT arrives, all is fine. However, due to the asynchronous nature of the network, events may not occur in this sequence. If calls are not queued, and the RFC comes before the LISTEN is issued, it will be refused; if it arrives later it will be accepted.

Unless one has infinite queue space, it is desireable to have some mechanism for purging the queues of old RFC's which the user never bothered to examine. An obvious but informal method is to note the time of arrival of each RFC, and then to periodically refuse all RFC's which have been queued longer than some arbitrary limit. Another action which should be included in any purging scheme is for the NCP to send a CLS on any pending connection when a user logs out or blows up.

The following scheme may be used to reduce the number of queued requests. When a CONNECT is issued, the NCP assumes that this local socket wants to talk to the specified foreign socket and to that socket only. It therefore purges from the pending call queue all non-matching RFC's by sending CLS's. Similarly, when the connection is in the RFC SENT state (a CONNECT has been issued and an RFC sent) all non-matching RFC are refused. If a LISTEN is issued and results in an open connection, the remainder of the pending calls are not removed from the queue, in the expectation that the user may wish to accept these requests in the future.

Connection states

Since the sequence of use of a connection involves many events and the legality and interpretation of many of these events is dependent on the preceeding activity, the NCP must remember, for each connection, where it is in the sequence. To keep this knowledge concisely the notion of a state is used. It has often been attempted to construct a state transition diagram to illustrate the possible state sequences of a connection, but to accurately take into account the many possibilities the diagram would be overly complex, thus often a simpler diagram is used that shows only the main lines of the primary sequences.

The states that are typically present are:

NOT ACTIVE

This is not really a state, but the fact that a connection is not in the connection table at all.

LISTENING

The local socket is associated with a process port, and the NCP is waiting for an RFC to this local socket from any foreign socket in any host. When an RFC does arrive for this local socket a matching RFC is sent and the connection is set to the OPEN state.

RFC SENT

This state indicates that the local socket is associated with a process port, an RFC has been sent to a specific foreign socket in a specific foreign host, and no matching RFC has yet been received. This state would be entered if the user process issued a CONNECT call and there was no matching RFC in the pending call queue. When a matching RFC does arrive, the NCP completes the initialization and marks this connection in the OPEN state.

RFC RECEIVED

An RFC has been received for which there was no matching entry in the connection table. This is a pending call. If a user process issues a matching CONNECT or LISTEN it will be satisfied at once. The local NCP will send an RFC and the connection will be marked in the OPEN state.

OPEN

RFC's have been exchanged and the connection is open. Transmission may begin subject to the constraints of the buffer allocation quantities.

ALLOCATION WAIT

To transmit data on a send connection there must be some positive allocation values (i.e. buffer space in the receiving host), if the allocation value (either bits or messages) has fallen to zero then the sender must wait until an ALL command arrives to increment the allocation values.

RFNM WAIT

After sending data on a connection the sender is not permitted to send additional data until the corresponding (IMP to host) RFNM command is received. When a RFNM is received the state changes to either OPEN or ALLOCATION WAIT depending on the allocation values.

CLS SENT

The user program has issued a CLOSE system call and the NCP has sent a CLS command to the foreign host. This cannot be done on a send connection until all the data is sent that the user process has previously output, and until a RFNM has been received for the last message of that data.

CLS RECEIVED

A CLS command has been received from the foreign host. If this is a send connection the NCP notifies the user process at once and answers with a CLS command, moving the connection to the CLOSED state possibly discarding data sent by the user but not yet transmitted by the NCP. If this is a receive connection, the NCP must wait until the user process has read all the received data. The NCP then sends a CLS to the foreign host, and notifies the user process.

CLOSED

The connection has been closed by an exchange of CLS commands. This is a transitory state and the connection should be deleted from the connection table shortly.

### 3.4 Functions of NCP Components

The following are the NCP functional components (program modules) and the tasks they carry out.

IMP Input Routine

Read messages from the IMP, and turn them over to the Network Interpreter Routine.

IMP Output Routine

Write messages to the IMP having received them from the Output Scheduler Routine.

Network Interpreter Routine

Analyze and act on messsages from the network, including maintaining connection table entries, composing replying messages, and exchanging information with the System Call Interpreter Routine, the Error and Statistics Routine, and the Output Scheduler Routine.

Output Scheduler Routine

Queue messages for delivery to the IMP Output Routine and to maintain a sent messages queue in case retransmisssion is called for.

System Call Interpreter Routine

Analyze and act on system calls from the user processes, including maintaining connection table entries, composing messages to foreign hosts, and passing messages to the Output Scheduler Routine.

Error and Statistics Routine

Record and report on detected errors in the program or protocol and the use thereof. Gather and record statistics of interest.

In the following paragraphs each of the functional components is explained. While there are likely to be a number of unusual events not explicitly discussed, the majority of frequent events are described. One comment that is important at this point is that the NCP should be constructed to be resilient in the face of errors. That is to say that when an error is detected the NCP should act to protect itself from any harmful effects, but should also act in a manner consistent with achieving for the user process the most reliable and consistent communications possible.

IMP Input Routine

There must be someplace in the system to handle at the machine instruction level and interrupt level each device attached to the central processing unit. This is true for line printers, disks, terminals, and multiplexor channels as well as for the IMP or rather the IMP-host Special Interface.

The IMP-host communication is a full duplex channel (i.e. simultaneous transmission in both directions) and it is often easier to interface the IMP to the computer as two independent devices.

On input the routine has a buffer available for the longest message that can be received and has a pending instruction to read from the IMP. When an end of input interrupt occurs the routine checks the length and signals the Network Interpreter Routine that a message is ready. The routine then gets a new buffer and starts a new read operation.

The amount of buffering depends on the rate and frenquency at which the IMP Input Routine and the Network Interpreter Routine operate. Two buffers are recommended. The manner in which the routine signals the Network Interpreter Routine varies in various systems, it may be anything from raising a flag to a pseudo-interrupt.

IMP Output Routine

For output from the host to the IMP this routine is supplied by the Output Scheduler Routine with the starting address and the length of a data buffer to move to the IMP. When the transfer completes the routine frees the buffer and signals the Output Scheduler Routine.

Network Interpreter Routine

As a message from the network is processed the leader should be examined to check the link number field. If the value is zero then the message is a host to host control message. If the link number is (curently) 2 through 71 the message is a data message associated with an open connection. If the link number is other that these two catagories the message is either part of another protocol (e.g. Message Switching Protocol <Bressler>) or an error.

If the NCP is aware that another protocol is being used in parallel with the host to host protocol it can turn over any message belonging to that protocol to the appropriate program on the basis of this link number inspection.

A message from the network is processed by examining the type field. The action taken for each type is indicated. The two types expected most frequently are REGULAR and RFNM.

REGULAR

This is a regular message, it is passed

to the next phase in the input message analysis.

ERROR IN LEADER

This message indicates that there has been an error in a previous host to IMP message such that the IMP could not decipher the leader. This is the only response that will be received to one of the unanswered messages on the sent messages queue, it will take careful detective work to determine which message. If the message was related to this response can be determined, the message and leader should be checked for correctness and retransmitted.

IMP GOING DOWN

The IMP is warning of of an impending service outage, parameters in the message leader tell something about how soon and how long, so the user processes can be notified.

NOP

This no operation message is discarded.

RFNM

Ready for next message on this link. This message is used to confirm the transmission of messages from the host to the destination IMP. There should be an associated message on the sent message queue which can now be discarded.

The link number should be used to locate a connection table entry. The state field of that table entry should indicate RFNM WAIT. This state should be changed to either OPEN or ALLOCATION WAIT depending on the amount of data ready to send and the allocation values. If there is data to send and the allocation values are positive the data send subroutine should be called.

DEAD

This is an indication that the destination host or IMP is dead. Normally the corresponding message on the sent message queue is discarded, and this host is marked dead in the host status table.

Note that some very recent work has been done on making the process to process communication more reliable in the face of network and operating system errors. Among the techniques is to treat a DEAD response as a temporary service interruption that will be quickly repaired (e.g. in a few minutes) and thus to retransmit the message associated with the DEAD response.

If the destination host (or IMP) is really dead then the NCP must close all of the connections to that host and notify any processes effected. The connection table must be updated.

ERROR IN DATA

There has been an error in the transmission of a previous host to IMP message, but the leader was preserved, so the link field can be used to attempt to associate this message with a message on the sent message queue. Once the associated message is determined, it is retransmitted.

INCOMPLETE

In this case the destination may be alive but the message was not delivered, the message is retransmitted.

RESET

The IMP has dropped and raised its ready line. If at the time of this occurance the IMP held a message to be transmitted to the host, or if a message transmission was in progress, data has been discarded. Similarly if a message was being transmitted to the IMP at the time of this occurance, it was discarded. The host and the IMP should at this point send each other several NOP messages to clear the line and reestablish the flow of messages. Some of the messages on the sent messages queue may need to be retransmitted.

UNASSIGNED

There are several messages types that are not assigned any meaning currently, these should be treated as NOPs, that is ignored.

The next phase in the analysis of a message from the network is to determine whether this is a host to host command or a data message associated with an open connection.

Suppose the current message has a link number in the range 2 through 71 identifying it as a data message associated with an open connection.

The link number is used to find a connection table entry and find the buffer associated with this connection, then checking the allocation and buffer space available the data is copied from the message into the processs buffer. Of course the proper checking is done to see that the connection is open, etc. and the allocation values are updated as appropriate possibly sending an ALL command. If the process has requested some notification when data arrives then the appropriate notice is given.

If the link number is zero then this is a host to host command, and contains one or more commands. Some of the commands are trivial while others are quite complex and require the mantenance of state information in the connection table.

NOP

This command is discarded.

RTS

This is a receiver to sender request for connection. The connection table is searched for a matching entry (due to a CONNECT or LISTEN system call). If a match is found the connection is opened (sending an STR if it has not been done earlier). If the connection state was LISTENING the matching STR is sent, otherwise the connection state should be RFC SENT. In either case the connection state is set to OPEN.

If no match was found then the information is added to the table, creating a new entry. This is the case of a pending call. The connection state is set to RFC RECEIVED.

STR

This is a sender to receiver request for connection. The connection table is searched for a matching entry. If a matching entry is found the connection is opened (sending an RTS if it has not been done earlier). If the connection state is LISTENING the matching RTS is sent, otherwise the state should be RFC SENT. In either case the state is set to OPEN.

If no match was found then the information is added to the table, creating a new entry. This is the case of the pending call. The connection state is set to RFC RECEIVED.

CLS

This is a command to close the connection. If the connection is in the state CLS SENT, the matching CLS has been sent and the connection state set to CLOSED. Otherwise the state is set to CLS RECEIVED.

If this is a send connection the associated process is informed and any unsent data is discarded. The matching CLS is then sent and the connection state set to CLOSED.

If this is a receive connection there may be data received and buffered which has not yet been read by the associated process. Thus the information that the connection is now closed must be flagged so that the process can be informed as it finishes reading the accumulated data. The matching CLS can be sent as soon as the NCP has flagged the connection entry. The connection state is also set to CLOSED.

At this point the state should be CLOSED, the data buffers should be empty, and the process aware that the connection is closed, thus the connection table entry can be deleted.

ALL

This is an allocation of buffer space for messages and bits that may be sent on the associated connection. The connection table entry fields for their-allocation-values is updated by adding the just received quantities to the values in the table entry. If the connection state was ALLOCATION WAIT it is changed to OPEN. If there is data waiting to be sent the data send subroutine is called.

GVB

This is the give back command, it requires the return (in a RET command) of a portion of the current allocation for the associated connection. This is done by building a RET command and asking the Output Scheduler Routine to send it. The connection state should change to ALLOCATION WAIT if the allocation values have been reduced to zero.

RET

This is the return command, in answer to a give back command associated with this connection. The allocation values are now decremented by the amounts indicated in the return command.

INR

This is a command to interrupt the process associated with the receive connection indicated by this link number.

INS

This is a command to interrupt the process associated with the send connection indicated by this link number.

ECO

This command requires that an echo reply command be sent. The ERP command containing the received parameter is constructed and turned over to the Output Scheduler Routine.

ERP

This is a response to an echo command and the data received should be exactly that which was sent.

ERR

This command indicates that the sender of this command has detected an error. This command and the date and time are turned over to the Error and Statistics Routine for recording. The data portion of the command indicates which of the messages and connections are involved since these connections may have to be resynchronized.

RST

This is a host to host reset command. This indicates that the sending host has cleared all of its tables of information, i.e. all connection are dissolved. Thus, all the tables of information relating to the sending host are cleared, and a RRP command. The RRP is composed and turned over to the Output Scheduler Routine to be sent. Also any user process that may be effected are notified.

RRP

This is a response to a reset command previously sent.

Data Send Subroutine

This subroutine checks to see if there is buffered data and allocation available, and the state is OPEN. If so the subroutine forms a message whose length is the minimum of the data available, the allocation available and the maximum allowed message size. This message is turned over to the Output Scheduler Routine. The allocation values are updated, and the state is set to RFNM WAIT.

Output Scheduler Routine

As messages ready for transmission to the various remote hosts are turned over to the Output Scheduler Routine, they are queued and delivered one at a time to the IMP Output Routine. It may be appropriate to order the queue according to some priority (e.g. host to host commands first), but this is optional. As the messages are sent by the IMP Output Routine they should be placed on a sent messages queue. As RFNMs are received the corresponding messages can be deleted from the sent messages queue. Other responses (e.g. INCOMPLETE) will cause a message on the sent messages queue to be indicated for retransmission.

System Call Interpreter Routine

As a user process issues system calls the NCP (eventually) must be invoked to service these requests. Some calls will be control requests (CONNECT, LISTEN, CLOSE, ...) while others will be data requests (SEND, RECEIVE).

For the control requests the NCP must check the status of table entries referenced in the call or create new entries. Some calls may require composition and sending of NCP commands to other hosts.

For the data flow requests the NCP must check tables and buffers and move data from system buffers to user process buffers or vice versa as the buffer space or data availability permits. This may result in the NCP sending to the other host either a data message or an allocate command depending on the direction of data flow.

The actions taken by the NCP to satisfy each of the system calls is now indicated.

LISTEN

The NCP searches the connection table for a pending call which matches this request. If a match is found the table entry is completed with the information supplied by this call; namely, the process identification and the port identification. A buffer should be assigned and initialized. The state should be RFC RECEIVED, a matching RFC should be sent and the connection moved to the OPEN state. The user process should be notified of the now open connection.

If there is no matching pending call a new table entry is created, filling in the values for local socket, process, and port identification. The state should be set to LISTENING.

CONNECT

The connection table is searched for a pending call. If a pending call is found the state should indicate RFC RECEIVED. A matching RFC is sent and the state updated to OPEN. A buffer should be assigned and initialized, and the remaining table entries filled in. The user process should be notified of the now open connection.

If no matching pending call was found a new table entry is created and filled in with the supplied information. The NCP sends an RFC and sets the table entry state to RFC SENT.

SEND

The indicated data is copied from the users buffer to the NCP buffer (being concatenated to any data already there) associated with this port. The buffer can be found from the the connection table entry associated with this port. The state is checked and if it is OPEN the allocation values are checked if either is zero the state is set to ALLOCATION WAIT. If there is space allocated the data send subroutine is called.

RECEIVE

The NCP moves data from the NCP buffer indicated in the connection table entry associated with this port to the users buffer up to the limit of either the amount specified or the amount available. The amount of data actually moved is indicated to the process. The NCP also checks to see if this frees a sufficient amount of buffer space to send an allocate command. If so an ALL is formatted and turned over to the Output Scheduler Routine.

CLOSE

The NCP will try to close this connection as soon as it can be sure the data flow has stopped. If this is a send connection, the NCP will wait until all the data issued by the user in SEND system calls has been transmitted to the remote host and a RFNM returned from the last message. This condition can be checked by ascertaining that the NCP buffer for this connection is empty and the state is OPEN (or even ALLOCATION WAIT). Once this all-data-transmitted condition has been met the NCP can begin to close the connection. The System Call Interpreter Routine forms a CLS command and turns it over to the Output Scheduler Routine. The connection state is set to CLS SENT.

If this is a receive connection the user process clearly does not want any more data even if there is some it has not read, so any buffered data or any that arrives following the CLOSE call is discarded, and no new allocates are sent. The NCP sends a CLS at once to notify the sending NCP and process to stop their transmission and to close the connection. The connection state is set to CLS SENT.

INTERRUPT

For a send connection the NCP forms a INS command, and for a receive connection the NCP forms a INR command; the command is forwarded to the Output Scheduler Routine.

STATUS

The NCP returns, in the INFO argument, data from the connection table entry associated with this port. This system call has no effect on the state of connections or buffers, and no information is transmitted to the network because of it.

Error and Statistics Routine

In case of any error condition detected a record should be added to a log file indicating the date, time, leader, and other circumstances of the error (e.g. NCP control message, portion of the content of the message).

Errors in use of the host to host protocol should also be reported back to the offending host using the ERR command. Any ERR commands received should be logged.

Certain kinds of errors and errors that occur with high frenquency should be reported to a system operator via an on-line console.

The Connection Table

The entries of the connection table should contain the following fields:

link number

local socket

foreign socket

foreign host

process identification

port identification

buffer address

state of connection

byte size

allocation values

date and time

In summary, the functions of the NCP are restated: to provide to the user processes a form of interprocess communication called connections. In carrying out this function the NCP must implement mechanisms for error control, flow control (allocates), multiplexing (sockets and links), and synchronization (interrupts).

## 4.0 Basis of Survey Comparisons

This section describes the model NCP discussed in the previous section as if were an actual implementation. For purposes of comparison the format is the same as that used in the descriptions of the surveyed systems.

Model system

System

The operating system is a timesharing system which provides for user process, a file system, and interprocess communication. Each user process has an independent (virtual) address space, a set of general registers, a location counter, and a set of open files. The system provides system calls to open, read or write, and close files, terminals or connections. Interprocess communications are supported by a mechanics similar to the network connections. There is also a pseudo interrupt facility such that the system can cause the location counter of a process to be set to the processes interrupt address.

NCP

The network control program is implemented as part of the operating system, though it is programmed in an way that would allow it to be run as a user program (except for the privileged interaction with the IMP input and output handler). The communication between the NCP and the user processes uses the existing interprocess communication system calls.

Points of comparison

System Calls

The system calls Listen, Connect, Send, Receive, Close, Interrupt, and Status are available to users. These call are similar to other input and output service call.

Return Characteristics

These system calls are nonblocking, with the option of blocking until completion.

Programming Languages

The network system calls are available to programmers in every programming language on the system.

RFC Queueing Policy

Requests for connection are queued until either the local process issues a Connect system call or a timeout period elapses. When a Connect system call is issued on a local socket all requests queued on that socket are refused execept the matching request. Requests queued on a local socket opened via a local Listen system call are retained in the queue.

Timeout Policy

Queued RFCs are timed out after ten minutes, a missing CLS is timed out after two minutes.

Connection States

The following states are used: Not Active, Listening, RFC Sent, RFC Received, Open, Allocation Wait, RFNM Wait, CLS Sent, CLS Received, Closed.

Allocation Policy

The allocation policy is to carefully control the flow of data using the bit count of the allocation to allow exactly the buffer space reserved on a per connection basis. In particular the initial allocation is a large number of messages (100) and as many bits as available in the connection buffer. As the data flows, the allocation values are adjusted whenever the values fall to a lower bound expressed as a fraction of the initial allocation (e.g. two-thirds). When the allocation is adjusted it is set to the maximum values then available.

Interrupt Treatment

The network interrupt signals INS and INR cause the user program associated with the connection to begin executing at its interrupt address.

Retransmission Policy

The incomplete transmission reply from the IMP will cause the indicated message to be retransmitted.

Error Treatment

A log file is kept of all unusual occurances, among the things entered into this log are all ERR messages received, all ERR messages sent, and any information about internal errors detected.

Measurement and Status Information

The NCP keeps running totals on the number of times each host to host command is sent or received, on the number of messages sent and received of the number of bits sent and received, and the number of each type of IMP to host and host to IMP message received or sent. The NCP keeps an accounting log in which is recorded the information about each connection when the connection is closed. The data recorded is the user name and account number, the foreign host, the number of messages and bits sent and received, the elapsed time and the time and day when the connection was closed.

The NCP connection table is accessed by a status display program which displays the foreign host, the socket numbers, the link number, the allocation values, and connection state, for each connection.

Operator Interaction

The operator can close any connection or send a reset to any host, can stop, continue, or reinitialize the NCP. The operator is informed of errors by the NCP.

Experimental Protocols

There is a provision to pass messages which arrive on designated links to other programs, to provide for testing of experimental protocols.

## 5.0 The Survey

### TENEX

#### System

The TENEX system was designed and built by Bolt, Beranek, and Newman, Inc. (BBN) to operate on the Digital Equipment Corporation (DEC) PDP-10 computer. The BBN modifications include a special instruction to implement system calls (JSYS), and paging hardware to implement a virtual memory but also to allow the physical core memory to be expanded beyond the 256 kilowords directly addressable by the 18 bit instruction address field.

The TENEX system provides a user environment which is embodied in a job. A job can be made up of several forks arranged in a tree structured hierarchy. Each fork is a program in a virtual machine and corresponds closely to the concept of a process.

The TENEX file naming conventions are organized for consistent naming of all types of devices and files. Network sockets are one type of file name and can be used anywhere a file name argument is called for. The file structure is rather flat in that each user has one directory which contains all his files, though the file names do consist of three parts - name, extension, and version.

NCP

The TENEX NCP consists of five modules: the Interrupt Service Routine, the Message Packaging Routine, the Control Routine, the File System Interface Routine, and The Server-Telnet Routine. The bulk of the work is concentrated in two routines: Message Packaging, and Control.

The Message Packaging Routine not only formats the data into host-IMP messages, it also deals with the IMP-host protocol and RFNM waits. The Control routine takes care of flow control on open connections, connection establishment and termination, and connection table management.

Points of comparison with the general model

System Calls

The network system calls are implemented as regular file system calls. Within this framework the functions suggested in the general model are provided. In addition to the suggested calls are calls to dump the buffer (i.e. transmit now), and to control the maximum allocation size. There is also a call to cause a pair of connections to be treated as a terminal (TTY:).

Return Characteristics

The system calls are blocking, but there are options for pseudo interrupts and immediate return.

Programming Languages

Both LISP and BCPL have features for network communication as well as assembly language.

RFC Queueing Policy

All requests for connection which do not immediately match are queued. The limit for such queued requests is approximately 100 requests. Unmatched requests are timed out.

Timeout Policy

Queued (unmatched) RFCs are timed out after two minutes, local Listenss are not timed out. A missing matching CLS is timed out after two minutes. A missing RFNM is timed out after two minutes.

Connection States

Fourteen connection states are used. The three additional states are a finer level of detail in the sequences of events for opening and closing connections, for example there is a state for "CLS received but waiting for RFNM for last data sent".

Allocation Policy

The initial allocation issued for a connection is two full messages plus a user specified buffer, this usually totals approximately 24,000 bits. The initial message count is two messages. The policy is to maintain the allocation values as close to these initial (maximum) values as possible.

Interrupt Treatment

The network interrupt signal causes a pseudo interrupt to the indicated process.

Retransmission Policy

The incomplete transmission response to a message causes the NCP to retransmit the message. There is no limit on the number of times this may occur.

Error Treatment

An online log is made of detected program errors, detected protocol errors, and ERR commands received. No ERR commands are sent. No statistical information is kept on errors.

Measurement and Status Information

There are no provisions for measurement in the NCP. Status information is maintained by the NCP such that user programs can obtain information on hosts up, connections open, and states of connections. There is a program called Netstat that users can run to display this information at their terminal.

Operator Interaction

The computer operator is informed online about program and protocol errors detected. The operator can suspend and continue network service, and can reinitialize the NCP. A systems programmer can watch the activity on a Telnet connection.

Experimental Protocols

There is a provision for turning over messages with link numbers outside the ranges assigned to the host to host protocol to another program. This allows experimental protocols to be tested in parallel with the host to host protocol.

Multics

System

The Multics (for Multiplexed Information and Computing Service) system was designed and built at project MAC at the Massachusetts Institute of Technology. The system is currently implemented on the Honeywell 6180.

A major emphasis in the Multics system is on the controled sharing of information (programs and data). The directory structure is quite general, allowing a hierarchy of directories with each level containing the names of segments or other directories.

The term file is not used in the Multics environment, rather the term segment is appropriate. A segment is a linear address space which may contain either a program or data. The set of segments known to a process are all directly accessible to (authorized) users. Each segment is subject to an elaborate set of access controls.

In addition to segment access controls there are protection rings which are an extension of the master/slave or supervisor/user state concepts in many systems. In the Multics system, programs executing in one ring are protected from programs executing in higher numbered rings, except the programs running in higher numbered rings are permitted to make calls to prespecified entry points in the programs executing in the lower numbered rings.

System calls are identical to calling any program segment. Some processes, however, can access more deeply into the system or are privedgled to control system resources. The method of interaction with the network is by means of special system calls, different from those that handle other input output operations.

NCP

The Multics network control program is organized in three components: the Network Daemon, the NCP, and the IMP DIM (for IMP Device Interface Module). Taking these in reverse order, the function of the IMP DIM is to manage all the frequent operations of the host to host protocol (e.g. data messages and allocates) and the low level input and output operations with the IMP. The NCP module performs the less frequent operations (e.g. opening connections) and services the user program system calls. The Network Daemon deals with administrative functions, and responds to host to host control commands.

The IMP DIM contains the code which manages the leader portion of IMP messages, issues input and output instructions for the full duplex Asynchronous Bit Serial Interface (ABSI) which connects the IMP to a pair of H6180 IOM (input output controller) common peripheral channels. The IMP DIM manages the assignment of link numbers on messages to be written, and it associates link numbers with socket numbers (and hence processes) on messages which are read. A natural extension of such functions, which was also a practical necessity, is enforcing the host to host protocol flow control discipline, particularly in processing the NCP ALL command. In part because the IMP DIM operates at interrupt time this role avoids the need to wakeup the Network Daemon every time a small recipient host is willing to allow a few more characters to be transmitted.

The NCP interfaces with the IMP DIM on one hand and with the user processes on the other. It must manage the socket space of the host to host protocol on behalf of any and all user (and system) processes which deal with the network on Multics and process the bulk of the NCP commands (such as interrupt the process associated with a given socket, reset all table entries associated with a given host, etc.). The socket management is based on the notion of a socket being in a particular state, and when that state changes the NCP directs a wakeup to the process controlling the socket.

The Network Daemon is a process in the user ring which processes the host to host control commands, sometimes calling on the NCP to complete the processing. The IMP DIM wakes up the Network Daemon to process the control commands in the same way it wakes up user processes to process data messages.

Points of comparison with the general model

System Calls

Multics has two additional system calls: activate and deactivate. A socket must be activated before it can be used in any other system call, in a sense the activate call indicates to the NCP that the user program is interested in this local socket and reserves it to the user process. Deactivate then releases the local socket and frees any associated resources. Also the echo function is reserved to privileged processes only.

The NCP primitives (the discrete entry points of the NCP and several free standing subroutines) allow for precise management of sockets, from activation through establishing byte sizes and sending or accepting host to host protocol requests for connection to causing host to host protocol CLSs to be sent. The subroutines provide for performing the initial connection protocol or sending or accepting requests for connection, the managing a process socket space, and performing the 9-to-8 and 8-to-9 bit conversions necessary to go to and from Multics' internal representation of ASCII from and to the networks representation.

Return Characteristics

The system calls are nonblocking and a

wakeup signal occurs when the requested action is completed. System call routines also post status information which can be inspected by the process.

Programming Languages

PL/1, Lisp, and BCPL have interfaces to the NCP primitive functions. Fortran programs can access the network function via special subroutines. Assembly language programs can invoke the network functions.

RFC Queueing Policy

Requests for Connection are not queued.

Timeout Policy

The Multics NCP does not timeout an step or state.

Connection States

Multics uses a few more connection states than the example description, but these additional states primarily make explicit the fine structure of events during connection establishment and termination. For example there is a "cls-read" state to indicate the connection is closed on the foreign side but that the local process is still reading buffered data. Two other states are the "active" state corresponding to socket selected by an "activate call, but not used as yet; and the "broken" state. The broken state indicates that this socket has been involved with some anomaly occuring within the Multics network software, any previous connection state has been destroyed.

Allocation Policy

Multics issues large allocation quantities relying on a large input buffer used in a pooled fashion. The total input buffer space is a segment set to allow 225 full sized (8000 bit) messages. The initial allocation for each receive connection is 25 messages and 65000 bits. The allocation is updated only when one of the values falls to an established lower bound, and then it is raised to the upper bound.

The IMP DIM's buffering strategy on input is such that Multics permits quite large allocations (in the NCP sense), by means of placing input on receipt from the read channel of the ABSI as rapidly as possible into pageable buffers managed by the IMP DIM code which runs at "call time" in the user's process. (The actual servicing of the ABSI interrupts is, of course, performed at "interrupt time" and involves wired down buffers.) The buffering strategy for writing also involves a mixture of pageable and wired buffers, although at the present time the wired buffer strategy does not utilize maximum IMP message-size buffers as does the read side.

Interrupt Treatment

An incoming interrupt signal (INS or INR) is first read by the IMP DIM which passes it to the Network Daemon process. The Network Daemon calls in the NCP to process the command. The NCP determines the relevant process and sends it an "interprocess signal" (ips). This is basically a PL/1 on condition which is raised, the particular condition being "QUIT".

Retransmission Policy

On incomplete transmission Multics attempts to retransmit the unfortunate message up to three times.

Error Treatment

Multics does not send ERR commands when it detects protocol violations. Any ERR commands received are merely counted.

Measurement and Status Information

Few measurement features are included in the network software beyond the meters in the basic Multics software.

The operator and processes can access a data base which indicates which hosts are currently communicating, a user can determine the status of his own connections.

Operator Interaction

The NCP reports online about program reinitializations, IMP going down messages and IMP crashes. The system operator can force a Telnet user connections closed, suspend and continue, or reinitialize the NCP operation.

Experimental Protocols

Multics does allow for experimental protocols by allowing for messages with a given link number in the leader to be diverted to another protocol process.

IBM 360/75 MVT

System

The University of California, Santa Barbara Computer Center operates an IBM 360/75 with the MVT operating system. This system is primarily batch oriented, however at Santa Barbara it supports the Culler-Fried On Line Mathematical System.

NCP

The network control program was designed with several objectives. First and most predominant was to keep the software independent of and entirely sepatate from the operating system. Second was to make the services of the NCP available to any task in the system. Implicit in this requirement is the need for some sort of interprocess communication mechanism. Although such a facility exists in most multiprogrammed systems, it is conspicuously absent from OS MVT; thus the NCP must itself provide whatever portion of such a mechanism it requires to communicate with tasks that use its services. Third, the NCP must provide a means for subordinating network activity in the host to the operational requirements of a multiprogrammed systems. This means that the operations staff must be able to control the NCP's activity, and obtain status information from the NCP. Finally the NCP must provide a record of its activity for debugging and tuning the NCP and for statistical information on network use <White2>.

The NCP is inserted into the system as a normal job. Once in execution the NCP assumes control of selected portions of the operating system. All modifications made to the system by the NCP are made dynamically and are transparent to the operating system. The NCP terminates execution only at the operators request, and as it does so, it extracts itself from the system, undoing the modifications it made at initiation. The NCP employs the operating system specify-task-abnormal-exit macro instruction to assure the extraction process is performed even if the NCP terminates abnormally.

THE NCP operates in supervisor state and with a protection key of zero. Obtaining this status is part of the initialization process and is accomplished with the aid of an installation provided supervisor call.

Tasks in the system communicate with the NCP by means of a supervisor call (SVC). So that the NCP can detect the issuance of its SVC the NCP instates itself as the systems SVC first level interrupt handler (FLIH). In this capacity the NCP examines every SVC interrupt which occurs, intercepting and processing the one of interest to the NCP, while allowing all other SVCs to proceed to the systems FLIH where they are processed normally. The NCPs presence is an overhead of ten instructions per SVC.

The NCP assumes the IMP to be attached to the

360 as two devices, one input and one output. In transfering data to or from the IMp the NCP bypasses the conventional execute channel program facility of the operating system Instead it executes directly the I/O machine instructions. So that the NCP can process the I/O interrupts which occur, the NCP instates itself as the systems I/O FLIH. In this capacity the NCP examines every I/O interrupt which occurs, intercepting and processing those generated by the IMP interface devices, while allowing the other interrupts to proceed to the systems FLIH and by processed normally. The NCPs presence is an overhead of ten instructions per I/O interrupt.

Points of comparison with the general model

System Calls

In addition to the normal system calls there is a call to perform the host name to host number mapping, and a call to find out the status of a host. The echo function is not available as a system call.

Return Characteristics

The system calls are nonblocking and use a wakeup type return. The mechanism is called Wait and Post.

Programming Languages

There are provisions for network system calls in PL/1, Fortran, and of course assembly language.

RFC Queueing Policy

Requests are queued up to ten for each local socket, then succeding requests are refused.

Timeout Policy

There are no timeout periods for request for connection either locally or remotely initiated, An echo reply is timed out after 4 minutes, and a matching close is timed out after 15 seconds,

Connection States

The only additional state is called "I/O pending",

Allocation Policy

The NCP reserves a 2048 bit buffer for each open connection and allocates strictly according to the space available in that buffer plus any addition al space set by the user program, The message allocation is set very large (65,000) and therefore is not a factor in controling message flow, A give back command might be sent if the user issues a receive system call with the length allowed to be variable and a large maximum length. Such a call would allocate some buffer space in the users address space. If the call was satisfied with less than the maximum length the left over space in the users address space buffer would be deallocated by means of a give back command,

Interrupt Treatment

When a network interrupt command is received a note is made, and the return values of the next user send or receive system call will carry this note, Thus the network interrupt will not be effective if the user process is stuck in a computation loop,

Retransmission Policy

No retransmission is attempted,

Error Treatment

While received error commands are logged, no error commands are sent when a protocol error is detected, No statistics on errors are kept, Measurement and Status Information

The computer operator can find out which hosts are currently communicating, and see the connections currently open. Both the users and the operator can determine the states of active connections, but neither can find out the allocation values, though they can determine the number of bytes pending input or output.

There are two host status tables, the first is maintained by the NCP by sending ECHO commands to each host once every four minutes, the second table (network accessible on socket 15) is maintained by attempting a connection to the Server-Telnet socket of each host onec every 15 minutes.

Operator Interaction

The NCP reports to the operator changes in the IMP status. The operator may close all connections, or send a reset to a specific site, as well as suspend and continue or renitialize the NCP operation.

Experimental Protocols

There is no provision for experimental protocols.

IBM 360/91 MVT

System

The University of California, Los Angeles, Campus Computing Network (CCN) operates an IBM 360/91 computer using the OS/MVT operating system. This is primarilly a batch oriented system. Hovever, at CCN there are several interactive subsystems. Among these are URSA and TSO.

URSA is a locally developed interactive system using display terminals which allows users to compose program and data data sets (files) and to submit data sets to run as jobs in the batch job stream. The URSA system provides users (and the computer operator) extensive job and system monitoring capabilities.

TSO is the IBM provided Time Sharing Option for its OS/MVT systems. This provides the facilities generally found in an interactive timesharing system.

CCN has developed a general purpose and powerful interprocess communication facility called the Exchange <Braden>. This facility is used to implement the network software.

NCP

The network related software is organized into three catagories: the NCP, the protocol routines, and the user level programs. The user level program interact with the protocol routines via the Exchange facility. The protocol routines obtain services from the NCP by making subroutine calls on the NCP.

The NCP is organized into three sections: the IMP input and output section, the NCP section, and the Logger section. The logger section performs the Initial Connection Protocol (ICP).

There are protocol routines for several function oriented protocols, for example User-Telnet, Server-Telnet, File Transfer, and Remote Job Service.

Points of comparison with the general model

System Calls

The user program makes request via the Exchange facility to protocol routines. These requests are generally in terms of higher level things than the system calls suggested in the general model. For example the user level requests are for such things as "open a Telnet connection pair", or "send this line of EBCDIC characters". The protocol routine to which such a request is addressed of course has the facilities to deal with the NCP in the terms of the suggested calls.

Return Characteristics

The Exchange provides options for either blocking or nonblocking behavior, and for a wakeup or interrupt type of return.

Programming Languages

Only assembly language currently allows convenient access to the network (through the Exchange macro), but work is being done to make network access from PL/1 and Fortran possible.

RFC Queueing Policy

Requests for connection are refused at once unless the local socket matches an active instance of a protocol routine, or the request is for an ICP soocket. Requests to open sockets are refused, except for requests to open ICP sockets, which are queued.

Timeout Policy

Requests by local processes (protocol routines) are not timed out by the NCP. A queued RFC from a remote host is timed out after one minute. A missing matching CLS is timed out after one minute. A missing RRP or ERP is timed out after 30 seconds. The initial connection protocol is timed out if the 32 bit number is not sent or received within one minute. Of course the protocol routines can time out any operation they request.

Connection States

The suggested connection states are used with a slight elaboration in the connection establishment and termination phases.

Allocation Policy

The allocation is handled by the protocol routines, which may each have a different strategy. Generally the existing protocol routines follow strict dedicated space policies using circular buffers.

Interrupt Treatment

The NCP notifies the appropriate protocol routine on a network interrupt. The protocol routine can notify the user level process via the Exchange.

Retransmission Policy

If an incomplete transmission response is received the NCP attempts retransmission up to five times.

Error Treatment

When a protocol error is detected the NCP composes an ERR message and sends it to the other host. An online report is made of both the ERR messages sent and the ERR messages received. Any errors detected in the operation of the NCP are reported online.

Measurement and Status Information

There are some facilities for measuring network usage in an accounting sense, but no facilities in the NCP to measure particular protocol functions. The things that can be messured are: per connection the number of bits and messages, the user identification, the protocol routine used, the time of day and the elapsed time the connection was open. There is status information on hosts currently communicating, and connections currently open.

Operator Interaction

The operator is informed online about

error conditions and program or network crashes. The operator is informed if a message is received from a host not in the host table. The operator may force a Telnet connection pair closed, may reinitialize the NCP, or may stop and remove the NCP from the system. It may be possible for the operator or a systems programmer to spy or advise on a connection but this depends on the protocol routine being used.

Experimental Protocols

While at the time of the survey there were no experimental protocols facilities, such facilities would be easy to add as a new protocol routine.

IBM 370/158 OS/MVT

System

The RAND Corporation operates an IBM 370/158 computer. The operating system at the time of this survey was the OS/MVT system with HASP, there were plans to change to the VS2 Release 2 system. In any case the information in this report is on the NCP implementation with the OS/MVT system.

NCP

The network control program is the program written at UCSB. The network is interfaced in a way that allows all programs running under OS/MVT to access and be accessed by the network.

The system at RAND is much the same as at UCSB. There are some differences from the UCSB situation at the next level of software, however. At RAND there is sustantial use of the Wylber and Milten software packages to provide interactive computer services to local users. The NCP communicates with Milten via the operating system to provide network access to this interactive facility. There is also a Network Access Program (NAP) through which local users can access the network.

Points of comparison with the general model

System Calls

Same as UCSB.

Return Characteristics

Same as UCSB.

Programming Languages

Assembly language.

RFC Queueing Policy

Same as UCSB.

Timeout Policy

Same as UCSB.

Connection States

Same as UCSB.

Allocation Policy

Same as UCSB.

Interrupt Treatment

Same as UCSB.

Retransmission Policy

Same as UCSB.

Error Treatment

Same as UCSB.

Measurement and Status Information

The status information available is basically the same as that as the UCSB implementation, however the status information is not available on socket 15 nor is the ICP polling done.

Operator Interaction

Same as UCSB, except that the operator can spy on a connection.

Experimental Protocols

Same as UCSB.

IBM 370/145 VM

System

The Systems Development Corporation has two computers connected to the ARPANET, an IBM 370/158 and an IBM 370/145. The model 158 uses the same software as the RAND Corporation computer so this section will discuss the model 145 only. The operating system being used Virtial Machine or VM system, which is much like the CP system used on the IBM 360/67.

NCP

The NCP for the 145 at SDC, although still in design at the time this information was obtained, is presented to the extent that the design exists. The general idea is to use as much of the UCSB NCP as possible and thereby to reduce the work required to get the 145 on the network. The approach is to dedicate a virtual machine to the NCP and have that virtual machine own the IMP. Other virtual machines will communicate with the NCP by associating their (virtual) card readers and line printers with the NCP virtual machine's line printer and card reader. This design is similar to the design used at Lincoln Laboratories for a 360/67 CP/CMS system <Winett>.

Points of comparison with the general model

System Calls

The system call interface will be different than the general model due to the communications between the users virtual machine and the NCP virtual machine being an association between one machine's virtual card reader and the other machine's virtual line printer. Further the intention is to allow the user to control only Initial Connection Protocol connection pairs of sockets (i.e. Telnet data streams).

Return Characteristics

The calls can be constructed to be either blocking or nonblocking and pseudo interrupt returns are possible.

Programming Languages

Since the calls are communicated as data to (virtual) line printers and from (virtual) card readers, any language that has provision for input and output to such devices can be used to communicate with the NCP.

RFC Queueing Policy

Same as UCSB.

Timeout Policy

Same as UCSB.

Connection States

Same as UCSB.

Allocation Policy

Basically the same as UCSB, but there may be some adjustment of the buffer sizes.

Interrupt Treatment

The treatment of the interrupt is not defined at this time, and may be more difficult that in other systems due to the virtual machine structure.

Retransmission Policy

Same as UCSB.

Error Treatment

Same as UCSB.

Measurement and Status Information

Same as UCSB.

Operator Interaction

The communication between the computer operator and the NCP has not been designed. It is planned to be similar the the UCSB implementation.

Experimental Protocols

Same as UCSB.

Burroughs 6700 MCP

System

The University of California, San Diego Computer Center operates a Burroughs 6700 computer system. The operating system is the Burroughs supplied Master Control Program (MCP). The system is a dual processor system.

The Burroughs machine provides hardware segmentation mechanism. The operating system is written in a version of Algol.

The system supports both batch and interactive modes of processing, and many users find it convenient to mix these modes. The interactive executive program is called CANDE (for Command and Edit).

NCP

The network software is principally contained in a program called the Network Message Control System (NETMCS).

The flow of data from and to the network is as follows: messages from the IMP first are read by a Micro 820 minicomputer which is the IMP-host special interface, the data is reformatted and passed to the DCP, the DCP then passes the data to the NETMCS program where it is examined and acted on. If the data were to be intended for a service program (e.g. CANDE) the data would be reformatted and passed back to the DCP which would then route it to the service program. In the oposite direction the path is exactly reversed.

Points of comparison with the general model

System Calls

There is a system call to perform the Initial Connection Protocol. There are system calls to establish a character code to be used. There is a system call to set the mode of interaction to character at a time or line at a time. There are a pair of calls for sending and receiving data. Basically the calls are set up to provide a higher level interface than that described in the model. Here the user is presented with a mechanism where the elments are the ICP established Telnet connection pairs.

Return Characteristics

The system calls may be either blocking or non blocking and may wakeup or interrupt the process on completion. The interrupt return feature is not normally used however. The primary mechanism for process to process or process-system communication is placing messages on a queue and reading messages from a queue.

Programming Languages

The programmers best interface to network functions is in the Algol language, but the network functions can also be accessed from Fortran, Cobol, Basic, and PL/1.

RFC Queueing Policy

All foreign RFCs are queued and if not

referenced by a local process in three minutes are answered by a CLS. RFCs for local sockets in the range 0 through 255 are queued forever.

Timeout Policy

Requests for connection are refused after three minutes, except requests to sockets which are Listened to.

Connection States

The states RFNM WAIT and ALLOCATION WAIT are not used because these conditions are implemented using the event mechanism. There is a state for "socket owned by process but not in use".

Allocation Policy

The allocation policy is essentially to allocate an infinite space and message count. In the past the largest values that would fit in the allocate command fields were used, but recently this has been changed to 10 messages and 32,000 bits due to problems with some hosts which seemingly were not able to deal correctly with the large values. The system would send a give back command if too much data were to be queued. The space for this buffering is drawn from a pool of buffers commomly shared by several of the system modules, these buffers may be backed off to the disk if they become large. This would be done by the buffer management module without the awareness of the NCP.

Interrupt Treatment

The network interrupt signal is ignored.

Retransmission Policy

There is no retransmission of messages by the NCP. If an incomplete transmission response is received it is acted on as if it were a RFNM, except that an error indication is displayed to the computer operator.

Error Treatment

When a protocol error is detected both a log entry is made and a ERR message is sent to the other NCP. When an ERR command is received it is both logged and reported online to the operator.

Measurement and Status Information

There is available to processes a data base which shows the host currently communicating and the connections currently open. There are no special facilities in the NCP for measuring the network performance, however, programs which use the network are charged on a per packet basis.

Operator Interaction

The computer operator may force a connection (Telnet pair actualy) closed, or the operator may spy on the traffic flowing on a connection. The operator can stop, start ,and reinitialize the NCP. The NCP will report online about some types of errors detected.

Experimental Protocols

There is no provision for experimental protocols.

TIP

System

The Terminal Interface Message Processor (TIP) is a special case of a host computer in the ARPANET. The TIP is an extension of an IMP to perform the host to host and Telnet protocols for a set of up to 63 terminals. This is done in the Honeywell 316 computer with a total of 28 kilowords of core memory, 16 kilowords for the IMP and 12 kilowords for the TIP.

NCP

points of comparison with the general model

System Calls

Because of the specialization of the TIP to the User-Telnet function it is not clear that this is a relevant question. The human user has commands available which are reflected by program actions to listen (RFH, RFS), connect (STH, STS, ICP), send and receive (implicit), close (C), and interrupt (SS). In addition the reset host to host command can be sent.

Return Characteristics

The internal mechanism for communication between program modules is to set a flag and give up.

Programming Languages

The only programming language used in the TIP is an assembly language.

RFC Queueing Policy

No queueing of RFCs is done by the TIP.

Timeout Policy

Local requests for connection are timed out after 45 seconds. A matching CLS is timed out after 45 seconds. A RFNM is timed out after 30 seconds.

Connection States

The TIP uses a eight state description of a connection: 0) try to open, 1) RFC sent, 2) RFC received -- try to reply, 3) solid connection, 4) try to close, 5) CLS sent, 6) CLS received -- try to reply, 7) no connection. The information about allocation wait and RFNM wait is managed with flags.

Allocation Policy

The TIP follows a strict policy of allocating exactly the space available in per terminal dedicated buffers, The initial allocation is one message and the buffer size number of bits,

The total buffer space is 4000 words of 16 bits each, For each terminal there is one buffer for data from the terminal to the IMP and two equal sized buffers for data from the IMP to the terminal, These buffers can be tailored to the set of terminals specifically for each TIP,

Interrupt Treatment

The interrupt does not serve any great purpose in this special environment since the only "process" to be interrupted is the human user, However the TIP does process the INS command,

Retransmission Policy

The TIP does retransmit a message if it receives an incomplete transmission response from the IMP, The TIP will attempt retransmission indefinitely,

Error Treatment

The TIP neither keeps a log of ERR messages received or sends ERR messages when it detects protocol errors, However some protocol violations and program traps cause information to be sent to the Network Control Center (NCC), In these cases the TIP program keeps going generally by faking the missing event or ignoring the extra event,

Measurement and Status Information

There is no provision for measurement or status information in the TIP itself, The NCC and the Tenex supported RSEXEC program however do provide status information, Operator Interaction

The provision for operator interaction is via a debugging program resident in the IMP. This program can be accessed either from a dedicated local teletype or from the NCC via the network. This allows the operators at the NCC to examine and change any memory cell in the TIP. This is then a quite powerful tool, and allows the NCC operator to close a connection, force an allocate, examine the data buffers, and many other functions.

The operators at the NCC can suspend and continue operation of the TIP, or they can cause the program to be reloaded over the network from the NCC.

Experimental Protocols

There is no provision for experimental protocols in the TIP.

ANTS

System

The ARPA Network Terminal System (ANTS) was conceived at the University of Illinois Center for Advanced Computation as a flexible means to interface a wide range of interactive terminals and perpherial devices to the ARPA Network. The system has been constructed with this goal in mind, and communication between program modules receives careful attention thruout the system design.

This is a small system and it is not designed to permit users to run programs on it, but rather to enable users to reach larger computers in the ARPANET.

The system is implemented on a Digital Equipment Corporation PDP 11/50, but it is designed to run on a range of the PDP 11 series. There are several instances of the system running on the 11/45.

The version of the system surveyed is the ANTS MARK II system, a redesigned and reimplemented system based on the "quick and dirty" ANTS MARK I.

NCP

The NCP is the central part of this system, the system is tailored to the needs of the host to host protocol. Communication between program modules is by means of data paths which use a flow control mechanism very similar to the host to host protocol.

Points of comparison with the general model

System Calls

All of the suggested system calls are available to programs which call on the NCP.

Return Characteristics

The system call is a form of interprocess communication which can be best thought of as message and reply. The call (message) is either blocking or nonblocking at the callers request. The reply will wakeup a blocked process.

Programming Languages

The system programming language is PEESPOL, an ALGOL like language. The programs which call on the NCP are also written in PEESPOL.

RFC Queueing Policy

Requests for connection are queued if the socket is assigned, otherwise the request is refused. The only limit to queuing of requests is the lack of memory space.

Timeout Policy

There are no timeouts.

Connection States

The suggested states are used with the addition of several states in the close sequence for conditions such as "CLS received but data to be read by local process". Also the states RFNM WAIT and ALLOCATION WAIT are represented by a separate bit and the allocation value entries respectively.

Allocation Policy

The message allocation is set to either 100 messages and kept near that value or set to the largest value, in either case the message allocation plays no important role in flow control. The bit allocation is an exact transform of the data path allocation set up between the NCP and the calling program.

Interrupt Treatment

There is no pseudo interrupt feature in the system communication scheme.

Retransmission Policy

The NCP does retransmit a message when the IMP responds with either an incomplete transmission or a data error. The same message will be retransmitted a unlimited number of times.

Error Treatment

When protocol errors are detected ERR messages are sent, and both ERR messages sent and received are logged on the operators console.

Measurement and Status Information

The NCP does maintain a table of "hosts up" on the basis of recent communication and resets. It is also possible for the operator to determine which connections are open. No measurement facilities are provided.

Operator Interaction

The operator is informed of every Telnet connection pair established. There are no current provisions for operator intervention with the operation of the NCP.

Experimental protocols

There is a simple way to route messages to and from program modules that are using experimental protocols.

DEC PDP10 Monitor

System

The Center for Computer-based Behavioral Studies (CCBS) at the University of California, Los Angeles operates a DEC PDP-10 Computer with a modified version of a DEC supplied operating system. The system is configured with a PDP-15 processor to act as a terminal concentrator. The PDP-10 and the PDP-15 communicate via shared memory.

NCP

The network control program is implemented in the PDP-15. The PDP-15 supports local user access to the network with a User-Telnet program, as well as allowing remote users to access the PDP-10 via Server-Telnet, or Server-FTP.

Points of comparison with the general model

System Calls

At the time of the survey there was no means for user written programs to access the network.

Programming Languages

No programming languages have provisions for calling on network functions. The NCP itself is written in assembly language.

RFC Queueing Policy

Unmatched RFCs are not queued.

Timeout Policy

The listen of the Server-Telnet "process" is never timed out. Requests for connection from other hosts that do not match local requests are refused at once. A missing matching CLS is timed out after 30 seconds, as is a missing matching ERP.

Connection States

The connections states are the same as those in the general model, except that the closed state is not explicitly represented since at that stage the table entry can be deleted. The allocation wait state is not explicity a state since this information is checked by consulting the allocation value.

Allocation Policy

The allocation policy is to allocate the largest possible values with the expectation that data will be processed faster than it can be sent. Additional allocate messages are sent when the message value falls to 16 messages, then the allocation is incremented to 100 messages and the maximum possible number of bits.

Interrupt Treatment

For Telnet connections (the only type presently supported) all data is forwarded to the PDP-10 as if it were from a local terminal. The PDP-10 either processes the data or discards it if there is buffer overflow, but all "important" characters (e.g. control C) are examined and are effective. In this situation the processing of the network interrupt signal is irrelevant.

Retransmission Policy

No retransmission of messages occurs.

Error Treatment

When a protocol error is detected an ERR command is sent. When an ERR command is received it is ignored. As a debugging tool two circular buffers (one for input one for output) are kept of the most recent network traffic.

Measurement and Status Information

There is thus far no measurement of the NCP performance. The status data is limited to the currently connected (and logged in) Telnet users, but there is consideration being given expanding this to have status data on the current connection state and allocation values.

Operator Interaction

The operator is informed of most NCP program failures, and of network failures. The operator can reinitialize the NCP, and can remove the NCP from the system. The operator can inspect the running NCP using a DDT debuging program in the PDP-15.

Experimental Protocols

There is no provision for experimental protocols.

BKY

System

The Lawrence Berkeley Laboratories operate a system composed of three principal processor and several peripheral processors. The major machines in the complex are Control Data

Corporation products. There is a CDC 6400, a CDC 6600, and a CDC 7600. These machines are interconnected with 12 megabit per second channels and share a commom disk file. There is a front end computer connected to both the 6400 and 6600 for supporting access by interactive users. The front end can handle up to 128 lines at up to 9600 bits per second. Also connected to the 6600 is an IBM photostore of 10^12 bits capacity.

The operating system for this complex is necessarily distributed, but the major portion is on the 6600. The system actually executes in the set of perpheral processors on the 6600 with tables and data in the 6600 memory.

NCP

The NCP as part of the operating system executes in a perpheral processor of the 6600. The NCP has table and data space in the 6600 memory.

Points of comparison with the general model

System Calls

The user programs have available system calls for all the suggested functions except echo and status, in addition there are calls for assigning and releasing a socket, for determining a unique socket number.

Return Characteristics

The system calls are nonblocking and wakeup the program when completed. A program can determine the outcome of a system call by examining information posted in the users space.

Programming Languages

Programs written in Fortran can access the network through special subroutine calls.

RFC Queueing Policy

All requests to assigned sockets are queued.

Timeout Policy

An unmatched request for connection will be timed out in 4 minutes. An unmatched echo or reset or close will be timed out in 2 minutes.

Connection States

The states used are the same as those in the general model (a few have different names) with a few elaborations in the close sequence. The states for ALLOCATION WAIT and RFNM WAIT are not used as distinct states, but separate bit flags are used for these conditions.

Allocation Policy

The allocation policy is to allocate exactly the available portion of a circular buffer in the users address space on a per connection basis. The initial size of this buffer and hence the allocation is chosen by the user. The message count part of the allocation is kept positive by the NCP.

Interrupt Treatment

The NCP increments an interrupt counter in the users argument and result area upon receipt of a host to host interrupt signal. This is effective in the case of Server-Telnet since the program receives a wakeup from the system every half second and when the INS is received to check for such conditions.

Retransmission Policy

The NCP retries the transmission of a message up to 5 times if the incomplete transmission response is received.

Error Treatment

The NCP does not send error messages, but it does log a wide variety of events including detected errors and ERR messages received.

Measurement and Status Information

The log (called the NCP dayfile) receives information on each connection as it is terminated including usage statistics. As the NCP was just completed at the time of this survey no measurements have been done. There is quite a bit of information that could be extracted from the log (dayfile) to provide measurements of various aspects of the NCP performance and usage, but this has not been done.

Operator Interaction

The operator can examine the log (dayfile) and can control the execution of the NCP (start or stop, reinitialize or remove), the operator can also kill the job associated with a particular Telnet connection pair, or examine the core of any job (including the NCP).

Experimental Protocols

The NCP does not now have any special provisions for experimental protocols.

6.0 Physical Characteristics

TENEX

Programming and Maintenance Costs

The NCP was written by two top systems programmers working together for two months for a total of 16 man-weeks, including the time spent debugging. The machine time used is estimated to be about 40 connect-time hours.

Program maintenance is estimated to be about three man-days per month.

Program Size

code size

The program is about 4K 36 bit words or 144,000 bits. This includes a portion of the Server-Telnet function.

table size

The space set aside for tables is 1512 words of 36 bits or 54,432 bits.

buffer size

The total buffer space reserved to the NCP is 4K words of 36 bits or 144,000 bits.

total size

The total of the above figures is 9.5K words of 36 bits or 342,432 bits.

Performance Measurements

A measurement of the transmission and CPU bandwidth was made on an unloaded TENEX system in November 1972, the TENEX system has been improved since that time especially by reducing the input and output system overhead.

Data was sent from a user program through the NCP to the IMP and then back the reverse path to the same user program. When sending short blocks (50 to 200 bytes) of 8 bit bytes a throughput of 20 to 30 Kilobits per second (KBS) was achieved. Sending large blocks (400 to 800 bytes) of 36 bit bytes a throughput of 60 KBS was achieved <Murray>.

Multics

Programming and Maintenance Costs

The programming and debugging effort required two man-years. The program was operational for most of the development period yet not complete. The continuing maintenance requires about 2 man-days per month.

Program Size

code size

The program size is 22K 36 bit words or 792,000 bits.

table size

The tables are included in the preceeding program size.

buffer size

The buffer space is 3K 36 bit words or 110,592 bits.

total size

The total size is then 25K 36 bit words or 902,592 bits.

It should be noted that of this total only 5K words of program and 3K words of buffers are wired down, the remainder is paged.

Performance Measurements

Very little performance measurement has been done, but it is known that typical throughput using the file transfer protocol has been 12KBS with Tenex and 8KBS Multics to Multics. (A reason for the low value Multics to Multics is that Multics restricts the maximum output message size to a one packet message, this to avoid an old IMP or IMP-host interface bug.)

Another measure of interest is system overhead: input or output with the network cost (when last measured some time ago) 1.2 times the same input or output operations to local devices. This value may have decreased due to changes since the time of the last measurement.

IBM 360/75 MVT

Programming and Maintenance Costs

The NCP was written by one top systems programmer in 26 weeks, including debugging time. The maintenance of the NCP requires about

4 man-days per month. About $3500.00 worth of computer time was used for testing the NCP.

Program Size

code size

The code of the NCP is 52K 8 bit bytes or 416,000 bits.

table size

The total size allocated for tables is 53K bytes or 424,000 bits.

buffer size

The buffers are included in the tables.

total size

The total size is then 105K bytes or 840,000 bits.

Performance Measurements

A throughput of 70KBS has been achieved in sending a large core resident block of data to the IMP and back. This transfer used 2 percent of the CPU bandwidth.

It takes one tenth of a second to send a full 8000 bit message from a user program to the IMP and back to the user program.

IBM 360/91 MVT

Programming and Maintenance Costs

The NCP programming required 1.5 man-years, plus additional time creating the environment for the protocol programs (called the ICT environment). Maintenance requires about 1 man-day per month.

Program Size

code size

The program is 15K bytes or 120,000 bits.

table size

The tables are dynamically allocated, and consist of the following sections: for each activly communicating foreign host there is a 32 byte pointer block and a 256 byte link table, for each connection there is an 80 byte table. At a time of typical use this might require 5K bytes of table space or 40,000 bits.

buffer size

The buffer space is also dynamically allocated, typical allocations being 256 bytes per connection. At a time of typical use this might require a total of 5K bytes or 40,000 bits.

total size

The total size in typical use is 25K bytes or 200,000 bits. (However, it should be noted that the NCP is run in a 140K byte region 1,120,000 bits).

Performance Measurements

No careful measurement of NCP performance has been done but it is known that on a typical day there may be on the average 3 Network RJS users and 6 Network TSO users and that over the 8 hour prime shift on such a day about one and one half percent of the CPU usage can be charged to the NCP.

IBM 370/158 OS/MVT

Programming and Maintenance Costs

The NCP is based on the UCSB program. Modification to adapt the NCP to this environment took four man-weeks. Maintenance of the NCP requires about 1 man-day per month.

Program Size

code size

The size of the code is 35K 8 bit bytes or 280,000 bits.

table size

The size of the tables is 12K 8 bit bytes or 96,000 bits.

buffer size

The buffers are included in the tables.

total size

The total size is then 47K 8 bit bytes or 376,000 bits.

Performance Measurements

NCP performance measurements are not part of the normal procedures, but one day of NCP usage was analyzed. On a typical day in the 12 hours from 6 am to 6 pm there were 118 network sessions which involved a total data transfer of 5.96 million bytes. During this same period the NCP job can be charged for 484 CPU seconds (including interrupt handling). These facts suggest the following conclusions: the NCP consumes 1.1 per cent of the CPU, and the NCP overhead cost of transfering 1000 bytes is about .088 seconds.

IBM 370/145 VM

programming and Maintenance Costs

The NCP is under development still but the estimated time to convert the UCSB supplied program to this environment is six man-months.

Program Size

The program is not complete, so the finished size is unknown but it is estimated to be about 20 per cent larger than the UCSB 360/75 MVT implementation but less than 100K bytes (800,000 bits). This appears to be impossible but perhaps the code will be 20 per cent larger but the tables and buffers will be much smaller.

Performance Measurements

No performance measurements have been done.

Burroughs 6700 MCP

Programming and Maintenance Costs

The NCP was written in three man-months. About three man-days per month are required for program maintenance.

Program Size

code size

The NCP code is about 15K 48 bit words or 720,000 bits. The NCP, Telnet, FTP and RJE programs are combined together in one module (the NETMCS) which is 25K words in size.

table size

The table space is typically 6K words or 288,000 bits.

buffer size

The buffers are included in the tables.

total size

The total size is then 21K words for the NCP or 1,008,000 bits. It should be noted that the entire NETMCS requires 31K words but can run in 10K to 16K of core due to the segmentation structure.

Performance Measurements

On one typical day the NCP used 3% of the CPU of one of the dual processors over the eight hour prime shift.

TIP

Programming and Maintenance Costs

The TIP program was written in about two months by one person. There has been significant modification and improvement since the program was first operational however. The current maintenance requires about one to one and a half man-days per month per site, with currently 20 sites, this requires one person full time.

Program Size

code size

The program is about 6K 16 bit words or 96,000 bits.

table size

The tables require about 2K words or 32,000 bits.

buffer size

The buffer space is 4K words or 64,000 bits.

total size

The total size for the above is then 12K words or 196,000 bits.

Performance Measurements

Because the TIP is oriented to terminal use the throughput measurements made with larger host are not applicable. The interest in the case of the TIP is the total or maximum combined throughput. BBN has arrived at a formula that indicates the throughput constraints on the TIP. Recalling that the TIP is an IMP too, the competition for CPU use is between traffic due to IMP to IMP phone lines (L), hosts (H) and the collection of terminals (T). The formula is L + H + 15T < 600 KBS. Where H, L, and T are full duplex rates, that is, a 50KBS full duplex line counts as 50 in the formula. The maximum total terminal traffic is about 80KBS, for example eight 9600 baud display terminals doing output only (assuming sufficient buffer space is available).

ANTS

Programming and Maintenance Costs

The ANTS MARK II NCP was designed in two man weeks, coded in six man weeks, and debugged in seven man weeks, for a total of fifteen man weeks. Note that the designers had the longer experience with ANTS Mark I, however.

Program Size

code size

The program is 4K words of 16 bits or 64,000 bits.

table size

Tables are dynamically assigned from the system free memory pool, but a typical load might require 1000 bytes of 8 bits each or 8000 bits.

buffer size

There are IMP input and output buffers of 256 bytes each for 4096 bits total.

total size

Thus the total size is about 76,000 bits.

Performance Measurements

There have been no performance measurements as such, but it is noted that the system can support 9600 bit per second terminals with no apparent degradation of the terminal speed.

DEC PDP10 Monitor

Programming and Maintenance Costs

The NCP and Server-Telnet programs together were written in six man-months. The User-Telnet and Server-FTP are included in the code in the PDP-15. The program was only recently completed and a normal maintenance level has not been established.

Program Size

code size

The NCP program is 12K 18 bit Words or 216,000 bits.

table size

The table size is 12K words or 216,000 bits.

buffer size

The buffers are included in the tables.

total size

The total size is then 24K words or 432,000 bits.

Performance Measurements

No performance measurements have been performmed.

BKY

Programming and Maintenance Costs

The NCP coding effort took one man-year including a month of debugging time.

Program Size

code size

The code is about 8K words of 12 bits each, or 96,000 bits.

table size

The tables are reserved 2K words of 60 bits each, or 120,000 bits.

buffer size

The NCP also has an input and output buffer to the IMP, each capable of containing one full message, or a total buffer of 16,000 bits.

total size

The total number of bits then is 232,000.

Performance Measurements

Since the NCP was completed just at the time of the survey no performance measurement is available.

Summary Chart

The following chart summarizes, by system, the size of the program, tables, buffers, and total of these, for each NCP.

| | Code | Table | Buffer | Total |
|---|---|---|---|---|
| TENEX | 144 | 54 | 144 | 342 |
| Multics | 792 | - | 110 | 902 |
| 360/75 416 | 424 | - | 840 | |
| 360/91 120 | 40 | 40 | 200 | |
| 370/158 | 280 | 96 | - | 376 |
| 370/145 | - | - | - | 800 |
| B6700 | 720 | 288 | - | 1008 |
| TIP | 96 | 32 | 64 | 196 |
| ANTS | 64 | 8 | 4 | 76 |
| DEC10 | 216 | 216 | - | 432 |
| BKY | 96 | 120 | 16 | 232 |

Note: The table entries are thousands of bits.

7.0 Summary

The findings of the survey of network control programs are presented in this section. These remarks might best be labeled opinion rather than conclusions because of the sparseness and inconsistency of the evidience.

First note that the ARPANET works. The NCPs in the host do communicate among themselves, and user level programs in the various hosts do communicate with other user level programs in other hosts.

This has been accomplished by a uncoordinated group of systems programmers working for diverse organizations geographically distributed. These programmers established a working group and used a series of technical memos (called Request for Comments) to exchange information and viewpoints on the developing protocols. Several meetings were held to gather concensus on protocol issues and adopt proposals for implementation. The specifications produced by this process are loose in several respects, the two most important areas being the functional specification of the user process to NCP interface (system calls) and the requirement for queueing of requests for connection.

The current host to host protocol has several flaws in addition to the weak positions on the user interface and queueing cited above. Error control is present in a very limited sense, the ERR command is useful for reporting detected protocol violations, but such violations should arise only due to program bugs, and the ERR command is not employed by many of the implementations. The hosts should have a means of ensuring that the data transmitted is received correctly and that the messages transmitted all arrive. The first problem could be attacked by an end to end checksum, and the second by a message sequence number.

Another problem is in the flow control aspect, while the allocate mechanism is constructed to allow quite flexible buffer management, many of the implementations have chosen to use a very simple strategy, often one that requires an allocate for each message, thus insuring a host to host round trip delay between messages of the same conversation. Similarly the host to host protocol requires a host to destination IMP round trip delay between each message of the same conversation by requiring the NCP to wait for a RFNM to each message on a logical link before sending another message on the same logical link. These constraints limit the throughput achievable on any particular connection.

The performance measurements of network control programs has been very spotty and informal. There should be some consistency and regularity to performance measurements. There needs to be a standard set of experiments defined and these experments should be performed regularly.

Generally network control programs implemented as core resident modules are capable of higher throughput rates and cause smaller delays, this is not necessairly due to better coding but usually due to avoidance of the paging or swapping overhead incurred with a nonresident program, On the other hand the nonresident program may have the advantage of not tying up core memory when not in use and may be able to have only the active subroutines in core thus using a smaller portion of core even when in use.

There is no authority to designate network control programs complete or correct. There should be a mechanism for a third party to review and certify NCPs, as the situation stands each implementation is correct only on the word of its implementer.

The documentation of network control programs is spotty, often there is no documentation (other than the code) of the program, however see <BBN91>, <White1>, <Winett>, <Wong>. Further it is sometimes difficult to find documentation on the user program interface (system calls). This latter problem is serious in that it tends to prevent users from constructing inovative applications of the network facilities.

## A.1 Glossary

### Abbreviations

AEN
another eightbit number

ALL
A host to host protocol command to allocate buffer space to the sending NCP in the receiving NCP.

ANTS
ARPA Network Terminal System

ARPA
Advanced Research Projects Agency of the Department of Defense

ARPANET
Advanced Research Projects Agency Computer Network

ASCII
American Standard Code for Information Interchange. The character encoding used in the network.

BBN
Bolt, Beranek, and Newman, Inc. Cambridge, Massachusetts

BKY
The operating system used at Lawrence Berkeley Laboratories for the CDC 6600 computer.

CCBS
Center for Computer-based Behavorial Studies at University of California, Los Angeles.

CDC
Control Data Corporation

CLS
A host to host protocol command to close the connection.

DEC
Digital Equipment Corporation

DMS
Dynamic Modeling System. A host computer on the ARPANET at MIT.

EBCDIC
Extended Binary Coded Decimal Interchange Code. The character encoding used primarily by IBM computer systems.

FCP
File Control Program

FTP
File Transfer Protocol

IBM
   International Business Machines
ICP
   Initial Connection Protocol
IPC
   interprocess communication
IMP
   Interface Message Processor
LBL
   Lawrence Berkeley Laboratory
MCP
   The operating system for the Burroughs 6700.
MIT
   Massachusetts Institute of Technology
Multics
   Multiplexed Information and Computing Service, the operating system for the Honeywell 6180 computer designed and implemented at MIT's project MAC.
NCC
   Network Control Center at BBN.
NCP
   Network Control Program
NIC
   Network Information Center at the Augumentation Research Center of Stanford Research Institute, Menlo Park, California.
OS/MVT
   An IBM operating system for the 360 series of computers.
PDP
   Programmed Digital Processor
RAND
   The RAND Corporation
RFNM
   Request For Next Message
RFC
   request for connection
RTS
   Receiver to Sender request for connection. A host to host protocol command.
SDC
   System Development Corporation
STR
   Sender to Receiver request for connection. A host to host protocol command.
TCP
   Terminal Control Program
TENEX

The operating system designed and implemented by BBN for the DEC PDP10 computer.

TIP
: Terminal Interface Processor

UCLA
: University of California, Los Angeles

UCSB
: University of California, Santa Barbara

UCSD
: University of California, San Diego

UI
: University of Illinois

VM
: The IBM operating system for the 370 series of computers.

Terms

another eightbit number
: The user program specified portion of the socket number.

ARPA Network Terminal System
: A particular small host system designed to interface a wide variety of terminals and peripherals to the ARPA network. This system was designed and implemented by the Center for Advanced Computation at the University of Illinois. The system operates on a DEC PDP11 computer.

connection
: The form of interprocess communication provided to the user level processes by the NCPs in the host computers. A connection is a logical simplex stream of data from one port of one process to another port of another process in the network.

control message
: A message (of the regular type) that contains host to host commands.

File Control Program
: That module in the operating system that controls the access to files by the user processes.

File Transfer Protocol
: The protocol that specifies the communication interaction required to move blocks of data (files) between host computers in the network.

full duplex

A channel in which data can flow in both directions simultaneously.

half duplex

A channel in which data can flow in both directions, but may only flow in one direction at a time.

header

The control information at the begining of a packet.

host

A computer attached to an IMP. A host does not necessarily offer services to other computers in the network.

Initial Connection Protocol

The sequence of actions taken by user level programs to establish a pair of connections between a user program and a service program.

Interface Message Processor

The packet routing computers which are the nodes of the ARPA network. An IMP is connected to between 1 and 5 other IMPs and to between 0 and 4 hosts.

interprocess communication

The facility for one process to communicate with another process.

leader

The first 32 bits of a message, containing address and control information. The most important fields in the leader are: the message type, the link number, and the host address.

link number

A parameter in the leader that selects a logical communication channel between the source and destination hosts.

message

The unit of transmission between a host and an IMP, up to 8096 bits.

Network Control Program

The program module added to the operating system that interfaces the user processes to the IMP and controls the communication between hosts by implementing the host to host protocol.

packet

The unit of transmission between IMPs, up to 1008 bits.

port

The input or output identifier associated with a particular data stream of a process. For example a Fortran logical unit number or a data set reference number, or an assembly languague data control block.

prefix
: A 40 bit block immeadiately following the leader and containing the byte size and number of bytes of following text.

process
: A program in execution with its associated address space, registers and location counter.

protocols
: The rules of behavior, in particular, the allowed formats and sequences of communication between two processes.

regular message
: A message from the host to the IMP or from the IMP to the host that is the normal data carrying type. When following the host to host protocol a regular message may carry either a set of control messages or a users data.

request for connection
: Either of the host to host protocol commands STR or RTS.

Request For Next Message
: A message from the IMP to the host indicating that the previously sent message on the same link number as this RFNM was received by the destination IMP and has begun transmission into the destination host.

socket
: The terminus of a connection. The network wide name of an input or output port associated with a process.

Telnet
: The protocol (or the programs that implement it) that specifies the communication interaction such that a user on one system gains access to the services of a second system as if he were a local user of the second system.

Terminal Interface Processor
: An extension of the IMP to allow a variety of terminals to access the ARPA network. The TIP contains the NCP and User-Telnet programs as well as the terminal handling code in the same processor as the IMP. In addition there is a BBN constructed multi-line controller for up to 63 terminal.

simplex
: A channel in which data can flow in one direction only.

Terminal Control Program
    The program module in the operating system that controls the flow of data between the interactive terminals and the user processes.

virtual
    Being something in effect, but not in actuality. For example a virtual memory might be one that a user process accesses as if it were a large linear core resident set of memory words, when in actuality the memory is managed by the operating system using paging and maping such that only a small portion of the users set of memory words are in core at any particular time.

A.2 References:

<BBN1822>
Bolt, Beranek, and Newman, Inc. "Specifications for the Interconnection of a HOST and an IMP," Report No. 1822, Cambridge, Massachusetts, Revised March 1974. Also contained in NIC 7104.
<BBN91>
Bolt, Beranek, and Newman, Inc. "The Terminal Interface Message Processor Program," Technical Information Report No. 91, Cambridge, Massachusetts, Revised November 1973.
<Braden>
Braden, R.T., and S.C. Feigin, "Programmers Guide to the Exchange," Campus Computing Network, University of California, Los Angeles, California, March 1972.
<Bressler>
Bressler, R., and D. Walden, "A Proposed Experiment with a Message Switching Protocol," RFC 333, NIC 9926, Augmentation Research Center, Stanford Research Institute, Menlo Park, California, May 1972.
<Crocker>
Crocker, S.D. et.al. "Function-Oriented Protocols for the ARPA Computer Network," AFIPS Conference Proceedings, 40:271-279, SJCC, 1972.
<Heart>
Heart, F.E. et.al. "The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, 36:551-567, SJCC, 1970.
<McKenzie>
McKenzie, A.A. "Host/Host Protocol for the ARPA Network," NIC 8246, January 1972. Also contained in NIC 7104.
<Murray>
Murray, H. "TENEX Bandwidth," RFC 415, NIC 12407, Augmentation Research Center, Stanford Research Institute, Menlo Park, California, November 1972.
<Newkirk>
Newkirk, J. et.al. "A Prototypical Implementation of the NCP," RFC 55, NIC 4757, Augmentation Research Center, Stanford Research Institute, Menlo Park, California, 19 June 1970.
<NIC7104>
Network Information Center. "Current Network Protocols," NIC 7104, Augmentation Research Center,

Stanford Research Institute, Menlo Park, California, Revised June 1973.

<Ornstein>
Ornstein, S.M., et.al. "The Terminal IMP for the ARPA Computer Network," AFIPS Conference Proceedings, 40:243-254, SJCC, 1972.

<Roberts>
Roberts, L.G. and B.D. Wessler. "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, 36:543-549, SJCC, 1970.

<White1>
White, J.E. "An NCP for the ARPA Network," Computer Research Laboratory, University of California, Santa Barbara, California, December 1970. Also available as NIC 5480.

<White2>
White, J.E. "Dynamic Extension of OS/360 for a Network Environment," Computer Research Laboratory, University of California, Santa Barbara, California.

<Winett>
Winett, J.W., and A.J. Sammes. "An Interface to the ARPA Network for the CP/CMS Time-Sharing System," Technical Note 1973-50, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts, November 1973.

<Wong>
Wong, J. "Network Control Program (NCP)," SEX Notebook Section 25.3, SPADE Group, Computer Networks Research Project, Computer Science Department, University of California, Los Angeles, California, January 1972.

MTR6722



(J24295) 24-OCT-74 12:41;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JBP( [ ACTION ] ) ; Sub-Collections: SRI-ARC; Clerk: JBP;