Network Working Group
RFC #647
NIC #31117

M. A. Padlipsky
MITRE-TIP
November 12, 1974

A PROPOSED PROTOCOL FOR CONNECTING HOST COMPUTERS TO ARPA-LIKE NETWORKS VIA DIRECTLY-CONNECTED FRONT END PROCESSORS

by Michael Padlipsky

with the advice and general agreement of

Jon Postel and Jim White (SRI-ARC), Virginia Strazisar (MITRE)

and the general agreement of

Tom Boynton (ISI), Frank Brignoli (NSRDC), John Day (CAC), Bob Fink (LBL), Carl Ellison (UTAH), Eric Harslem (RAND), Wayne Hathaway (AMES-67), John McConnell (I4-TENEX), Ari Ollikainen (UCLA), Dave Retz (SCRL), Al Rosenfeld (CASE), Stan Taylor (BRL), Doug Wells (MIT-MULTICS)

All affiliations are shown only for identifying ARPANET involvement, and do not necessarily denote organizational approval of the positions taken here.

****
No -- repeat NO -- expression of general agreement should be taken to apply to Appendix 2, which is exclusively a personal viewpoint.
****

CAVIUM-1019
Cavium, Inc. v. Alacritech, Inc.
Page 001

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

INTRODUCTION

As this proposal is written, in the fall of 1974, the ARPA network has achieved sufficient acceptance that a rather large number of organizations are currently planning either to attach their general purpose computer systems directly to the ARPANET or to interconnect their systems employing "ARPANET technology". The authors have been in touch with efforts sponsored by the Air Force Systems Command, the Naval Ship Research and Development Center, the Defense Communications Agency ("PWIN" -- the Prototype World-wide Military Command and Control System Intercomputer Network), ARPA (the National Software Works), the AEC, and other government agencies. A common characteristic of these networks and sub-networks is the presence of a number of systems which have no counterparts on the current ARPANET; thus, hardware "special interfaces" (between the Host and the network Interface Message Processor) and -- more important -- Network Control Programs cannot simply be copied from working versions. (Systems include CDC 6600's, XDS Sigma 9's, Univac 494's, 1107's, 1108's, and 1110's, and IBM 370's running operating systems with no current ARPANET counterparts). Because it is also widely accepted that the design and implementation of an NCP for a "new" system is a major undertaking, an immediate area of concern for all involved is to develop an approach for attaching systems to networks which employs as much off-the-shelf hardware and software as is practicable. This paper addresses two such approaches, one which apparently is popularly assumed as of now to be the way to go and another which the authors feel is superior to the more widely known alternative.

"FRONT-ENDING"

In what might be thought of as the greater network community, the consensus is so broad that front-ending is desirable that the topic needs almost no discussion here. Basically, a small machine (a PDP-11 is widely held to be most suitable) is interposed between the IMP and the Host in order to shield the Host from the complexities of the NCP. The advantages of this fundamental approach are apparent: It is more economic to develop a single NCP. "Outward" (User Telnet) network access is also furnished by the front end acting as a mini-Host. The potentiality exists for file manipulations on the mini-Host. Two operating systems are in advanced stages of development on the ARPANET for PDP-11's which will clearly serve well as bases for network front ends; thus, the hardware and software are copiable. So if we consider a model along the following lines

Host *** Front End --- IMP --- Network

everything to the right of the asterisks may almost be taken as given.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

(Caveat: Note the "almost" well in the last sentence; neither ANTS nor ELF -- the two systems alluded to above -- is a completely finished product in the estimation of either their respective developers or of the knowledgeable ARPANET workers who have contributed to this report. Both are capable of being brought to fruition, though, and in a reasonable amount of time. We will assume ELF as the actual front-end system here for two reasons: apparent consensus, and current activity level of the development team. However, we have no reason to believe that readers who prefer ANTS would encounter substantive difficulties in implementing our proposal on it.)

(Explanatory notes: ANTS is an acronym for ARPA Network Terminal Support system; it was developed at the Center for Advanced Computation (CAC), University of Illinois. ELF is not an acronym (it is said to be German for "eleven"); it was designed at the Speech Communications Research Lab (SCRL), Santa Barbara, California.)

THE RIGID FRONT-END ALTERNATIVE

Referring back to the model above, the popular view of the asterisks is to have the front-end system simulate a well known device for each Host (typically a remote job entry station alon# the lines of the 200UT on the CDC 6600), effectively requiring no software changes on the Host system. We characterize this approach as "rigid" because an immediate implication is that the Host system is constrained to handle data to and from the network only in fashions which its system already provides. (E.g., if you simulate a card reader, your data will necessarily be treated as batch input; if a terminal, necessarily as time-sharing input.) Now, it may be argued that Host software changes are only being shunned in order to "get on the air" quickly, and may be introduced at a later date in order to allow unconstrained channelling of network data within the Host; but this reasoning may surely be refuted if it can be shown that an alternatve exists which is essentially as quick to implement and does not require the waste motion of constructing known-device simulation hardware and software for each new Host, only to eventually avoid the simulation in the Host.

The major advantage which might be claimed for the rigid front-end approach other than quickness to implement would be embarrassing if true. That is, the possibility exists that either the "new" Hosts' operating systems or system programming staffs are so intractable that avoiding Host software changes is a necessity rather than a desire. We certainly hope neither is the case and have no reason to believe it to be so, but we must acknowledge that such possibilities exist as meta-issues to this report.

CAVIUM-1019
Cavium, Inc. v. Alacritech, Inc.
Page 003

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

DISADVANTAGES OF THE RIGID FRONT-END ALTERNATIVE

The rigidity argument sketched above merits some amplification. The major disadvantage of interfacing with the Host only in fixed ways lies in a loss of functionality. Granted that "Telnet" and "RJE" functions can be performed (though we have deep reservations about file transfer) by simulating a known device there are more things in practice and in theory than just using the Hosts' time-sharing and batch monitors. "Teleconferencing" is an instance which comes immediately to mind. Graphics is another. Neither fits naturally into the setting a typical operating system is likely to assume for a Telnet or RJE connection. Further, the ARPANET is just beginning to evolve a view of "process-to-process" protocols where cooperating programs on dissimilar systems communicate over network sockets in a true use of sockets as interprocess communication media. It is difficult to conceive of viewing a (simulated) line printer as an abstract "port" without considerable contortion of the extant operating system. To attempt to summarize this cluster of objections, a simulation of a known device may be cheaper than a large enough number of phone calls, but it's not networking.

For that matter, it is by no means clear that the goal of no Host software changes can even be met. In the case of one particular system on the ARPANET where a PDP-15 was employed as a front end to a PDP-10, one of the authors discovered that on attempting to login over the net he was confronted by an interrogation as to the type of terminal he was at -- the front end having been attached at the wrong point in the PDP-10's terminal handling code. (Being a battle-scarred veteran of Telnet protocol development, he gave suitable answers for describing a "Network Virtual Terminal". Unfortunately, however, the NVT apparently had no counterpart in the Hosts' normal complement of local terminals. And when he tried such Telnet control functions as "don't echo, I'm at a physically half-duplex terminal" things really got confused). As it happens, he later found himself in the neighborhood of the Host in question, and found himself spending an afternoon attempting to explain the philosophy and importance to the Telnet protocol of the NVT. The site personnel were both appreciative and cooperative, and although we have not had occasion to verify it, we assume that the site is probably now usable from the ARPANET. The important point, though, is that operating systems tend to make extensive, often unconscious, assumptions about their operating environments. This observation is particularly true when it comes to terminal types, and the problem is that there is simply no guarantee that the several systems in question could even "do the right thing" if they were front-ended by simulating a known device -- unless, of course, the simulation of the device in the mini were so painstaking that all we'd get would be an expensive way of adding an RJE station, period.

CAVIUM-1019
Cavium, Inc. v. Alacritech, Inc.
Page 004

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

Less abstract considerations also apply. For one thing, a mini-computer -- even with "third-generation" software -- is not as free and convenient an environment to program in as a full-scale Host; therefore, implementing the several simulations will not be trivial pieces of software engineering. Further, if the simulation software is prepared by front-end experts, they will encounter repeated start up transients in learning enough about the expectations of the several Hosts in order to perform their tasks. For that matter, it is clear that if personnel from the several Hosts are barred from active participation in attaching to the network there will be natural (and understandable) grounds for resentment of the "intrusion" the network will appear to be; systems programmers also have territorial emotions, it may safely be assumed.

On a still more practical level, it should be noted that the potential need to simulate more than one known device -- and even the potential complexity of any single device simulation -- may well lead to a requirement for a larger PDP-11 configuration than would otherwise be reasonable. And although there are other reasons for arguing that each front-end processor ought to be as big a configuration as possible, we must acknowledge that dollars do matter. Also on the topic of numbers, it should be further noted that the line speeds available for known-device simulations can be quite low. The 200UT, for example, is on a 4800 baud line, which is rather a mismatch with a 50,000 baud communications subnet. (Of course, there's always the 40,800 baud line into the 6600 -- but it isn't expected to have interactive devices on it, so the extant software won't send the data to the "right place"....) And no experienced ARPANET protocol designer would be willing to overlook the possibility that there will probably have to be a flow control discipline between the Host and the front-end processor anyway, so the no change to Host software goal becomes rather dubious of fulfillment.

After all that, it is perhaps gratuitously cruel to point out still another level of difficulty, but we feel quite strongly that it should be addressed. For, it must be admitted, the question must be asked as to who will do the front-end implementations. This sort of thing is scarcely within the purview of CAC or SCRL. But, as will be urged in Appendix 2, it is of the utmost importance that whoever performs the task already have ARPANET expertise, for we know of no case where "outsiders" have successfully come aboard without having become "insiders" in the process, which is neither an easy nor a cost effective way to proceed.

In light of the above, it is at least reasonable to consider an alternative to the rigid front-end approach, for regardless of the weight the reader may attach to any particular cited disadvantage, in total they at least suggest that the known-device simulation tactic is not a panacea.

CAVIUM-1019
Cavium, Inc. v. Alacritech, Inc.
Page 005

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.