Laboratory for Computer Science
Computer Systems Research Division

July 2, 1976
Request for Comments No. 117

CSR SECTION OF ARPA/ONR CONTINUATION PROPOSAL

from J. H. Saltzer

Enclosed is a draft of the CSR section of the continuation proposal that the laboratory will shortly submit to ARPA/ONR for 1977 calendar year funding. Our section of the final proposal submitted will undoubtedly differ in detail from this, but should be similar in intent. This draft was prepared with lots of help from several different division members, which help is much appreciated.

COMPUTER SYSTEMS RESEARCH

The Computer Systems Research Division of the laboratory proposes four activities in the coming year, two of which are expansions of recently started activities, and two wrap up long-standing activities:

1) Study of the potential for distributed computing (expanding)
2) Local network implementation (expanding)
3) National Software Works participation (wrapup)
4) ARPANET technology transfer (wrapup)

1) Study of the potential for distributed computing

The Computer Systems Research Division has as its major interest the discovery, pragmatically, of ways of systematically engineering useful computer systems. In the past, this interest led to the development of time-sharing through the vehicle of the Compatible Time-Sharing System, and the development of the information sharing computer utility, through the vehicle of Multics. More recently, the division has been working on protection and security in information-sharing systems, again using Multics as a vehicle. Today, the major engineering issues in creating useful computer systems seem to revolve around the integration of data communication and the exploitation of modern (LSI) hardware technology. Together, these two approaches lead to a new view as to the proper way of modularizing complex systems. Rather than highly multiplexed general-purpose modules, the "components" of tomorrow's systems appear to be specialized free-standing computers interlinked with communications networks. The term "distributed computing" has been commonly applied to this phenomenon.

In the Spring of 1976, faculty members of the Division led a semester-long graduate seminar in a review of the current wide range of activities being labelled "distributed computing" in other universities and research centers. In current technical material, the term "distributed computing" has started appearing with alarming frequency, often referring to such common computer organizations and operating system strategies as multiprocessing, array processing, time-sharing with remote terminals, etc. There exists a considerable confusion as to what is distributed computing. This confusion frequently results in one of the two attitudes:

1. Distributed computing is seen as the "wave of the future" that will solve all current problems in large-scale computing and information processing.
2. Distributed computing is dismissed as just a new catchword for old techniques that do not work very well.

Perhaps the main conclusion of this review is that only superficial attention is being paid to the question of whether or not there are new, fundamentally important problems to be solved. Superficially, justifications of higher reliability and increased speed (of parallel operation) are often used, but these justifications usually turn out, on more careful analysis, to be based on the economic advantages of modern mini- and micro-computers, rather than on any intrinsic limitations of the functional capability of traditional system organizations. The economic justification is obviously a very strong one, though probably not quite so overwhelming as some claim, since programming of the newly proposed architectures seems intellectually difficult, and therefore is probably expensive.

One very significant problem stems from the apparent economic advantage of dedicating a small computer to each small task: organizations are deciding at a low adminsitrative level to purchase local, dedicated computers, rather than participate with other organizations in using a shared computer. By itself, each such decision seems quite reasonable, but insufficient attention is paid to possible future need for these systems to communicate with other locally justified small computer systems. Clearly, for any single special case, one can string communication lines and invent protocols for the output of one operation to be used as the input to another, but a need is certain to develop for a higher level of "coherence" across such operations. It is unlikely that "patching on" of coherence across systems that are not prepared for it will be very cheap; it is more likely to be impossible. Many computer users are just now involved in unsuccessfully trying to "patch on" security in systems that were not prepared for it; the issues seem quite parallel.

Because of the slippery quality of arguments surrounding distributed computing, we propose, in the coming year, to move cautiously in this area, with two specific projects:

1. Continue to study the combination of computing with communications to try to develop more firmly the arguments that such combinations provide capabilities unachievable with centralized, tightly coupled, widely shared hardware. We currently suspect that the strongest argument by far is connected with the inherent vulnerability of a centralized system to accident, sabotage, or attack.

If a policy of decentralization were technically feasible, there would be the potential for reducing this vulnerability. Decentralization such that geographically distributed subsystems can continue functioning even if the rest of the system is physically destroyed would underly one way of realizing this potential. Also, physical separation of user programs from vital system functions (e.g., information protection) can reduce system vulnerability to a more subtle form of harm, that of a user modifying the system software, causing it to fail.
We want to investigate some other potential capabilities also, namely the potential to grow in information storage size and in information access rate without encountering important bounds.

2. Begin analysis, on paper, of system organizations that support information sharing on distributed hardware resources that are dedicated to individual users and system functions. To focus on this goal, we are thinking in terms of a system for the L.C.S. community using the L.C.S. network (described in the next section) as its underlying backbone. The idea would be to arrange the file system of a local (personal) computer so that the user can coherently use his own files, those stored in other concurrently communicating (and consenting) personal computers, and those stored in a community file system. The basic issue to be explored here is how to allow independent file systems to act in a coordinated way, and to proceed robustly in the face of unavailability of other community members.

During the proposed period, the output of both of these projects is expected to be reports and papers rather than working systems.

2) Local Network Implementation

This project is basically one of developing a tool for carrying out other research missions of the laboratory, though it has some novel aspects. The purpose of a local network is to provide for intercommunication among existing and future computers at the laboratory, to provide a "gateway" through which all L.C.S. computers can access the ARPANET, to provide an an integrated file system for the laboratory, to allow efficient, varied use of terminals and other peripheral devices within the laboratory, and to provide a base for research on distributed computing systems. Currently, these purposes are weakly served by attaching each laboratory computer as a distinct host on the ARPANET. That strategy is increasingly unsatisfactory as the number of L.C.S. computers grows and the volume of purely local data traffic grows; some recently acquired computers in the laboratory cannot be attached to the ARPANET because all of the host ports on one IMP and one TIP are now in use. Similarly, since the TIP has no capacity for more high speed lines many laboratory terminals and peripherals are inflexibly attached to some particular host.

During the current year, a network design has been completed, based on a buffered version of the Ethernet, developed at XEROX PARC, and we hope that initial operation of a few PDP-11 hosts will occur before the end of the year. In the coming year, the following activities are proposed:

1. Implement a minicomputer gateway between the ARPANET and the L.C.S. local network. This step will make the laboratory as a whole appear to be a single host on the ARPANET, with consequent simplification in our appearance to the outside world for purposes of sending net mail, etc. It will also provide a connection between the ARPANET and the currently unattached L.C.S. computers. In the future it may allow us to remove some of the direct connections between L.C.S. hosts and the ARPANET.
2. Implement a front-end system that operates on a PDP-11 computer, and that allows attachment of a PDP-10 system to the local network. This step will allow attachment of two or three of the laboratory PDP-10 computers, and provide what may be the initial or only connection point of the proposed new KL-10 computer. Ideally, this same front-end computer can be used to attach terminals that use other local or ARPANET computers, without bothering the associated host.
3. Implement a subnetwork to interconnect the two more distant Multics hosts to a gateway to the local network. For this subnetwork, we propose to use the ARPA packet radio system, to experiment with the problems encountered in creating a medium size network out of subnetworks of radically differing design. The problems to be dealt with here are not just those of interconnecting the hardware, but of establishing that protocols for internetwork transmission operate as anticipated when faced with usage peaks, transmission errors, and other real-life situations.

4. Implement a version of the "Transmission Control Protocol" (TCP) for Multics, so that Multics can communicate via the packet radio system to the L.C.S. network, thence to the ARPANET, and through the ARPANET to other TCP-implementing hosts.

3) <u>National Software Works Participation</u>

We anticipate that by December, 1976, most of the significant work required to make Multics a tool-bearing host participating in the National Software Works will be complete, and that it will be possible to edit NSW files, translate them, and use the Multics GCOS simulator, remotely through the facilities of NSW. In addition, the NSW interface for Multics has been developed in a way that many other Multics commands should be either directly usable, or usable with only minor changes. For this reason, we propose to reduce our level of activity on this project to that needed to finish up loose ends, and provide technology transfer support. We are actively looking for some organization that is interested in taking over support of the special NSW packages created for Multics.

4) <u>ARPANET Technology Transfer</u>

Although the current year was scheduled to be the last in which we actively work with Honeywell to arrange for its adoption of the Multics/ARPANET software as a standard product option, progress in fueling Honeywell's interest has been slower than expected. Nevertheless, we still hope that this effort will be successful, as we have recently observed a strengthening of corporate commitment to networks at several levels in the Honeywell organization, and also continued attempts, by Honeywell, to identify government interest in the Multics ARPANET software. Our activity has been at a low level, restricted to occasional maintenance of the ARPANET software and travel to discuss technology transfer. We currently expect this activity to continue at a low level for some time into the coming year. We also intend to finish off some of the user and system documentation required to support the transfer.