Lawrence Berkeley National Laboratory
Recent Work

Title
LOCAL NETWORKING WITH INTELLIGENT NODES

Permalink
https://escholarship.org/uc/item/48297637

Author
Meng, J.

Publication Date
1983-06-01

eScholarship.org Powered by the California Digital Library
University of California

LBL-16042 c.2

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

RECEIVED
LAWRENCE
BERKELEY LABORATORY
JUL 21 1983
LIBRARY AND
DOCUMENTS SECTION

Engineering & Technical Services Division

Presented at the ISMM International Symposium on Mini and Microcomputers and Their Applications, San Francisco, CA, May 16-17, 1983; and to be published in the Proceedings

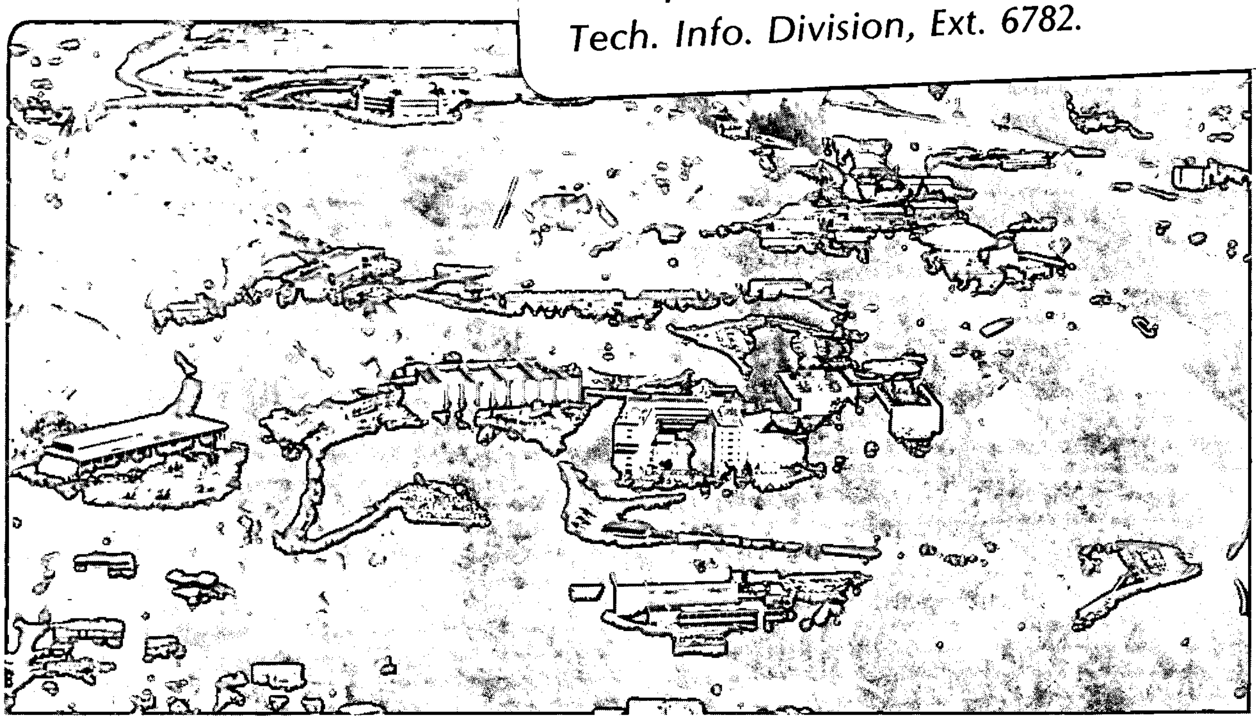
LOCAL NETWORKING WITH INTELLIGENT NODES

J. Meng

June 1983

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 6782.



LBL-16042 c.2

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LOCAL NETWORKING WITH INTELLIGENT NODES

John Meng

Department of Instrument Science & Engineering
Lawrence Berkeley Laboratory
University of California
Berkeley, California U.S.A.

ABSTRACT

In order to lengthen the useful life of existing office-oriented data processing equipment, and ultimately to ease transitions to newer devices, it is desirable to install the capability of connecting the existing equipment to a resource base which, among other things, permits easy movement of data files from older systems onto newer ones and which provides an archiving capability independent of any of the user systems. To achieve this economically, we propose a low-grade RS232-based communications network based on a common intelligent node. Node designs, network interconnections and predicted performance are discussed.

INTRODUCTION

Faced with an accumulation of aging equipment, most of which still serves its purpose adequately, the untraumatic movement of an office into a more coherent, more automated future requires a systematic, consistent evolutionary scheme. The specific problems this scheme must address are:

1) Records, both archival and currently-used, must be provided a path for easy transferral to any newer system.

2) Operations of exisitng devices and systems should not be disrupted. It may not be possible economically or desirable operationally to make changes in any existing unit.

3) The evolutionary path must not be prohibitively expensive to implement.

Our proposed evolutionary pathway is a communications network based on a microcomputer module programmed in one of three contexts--a user context, a communications context or a resource control context. Communications are effected via RS232 linkages. The result of implementing this communications scheme is the ability of any user or users to communicate with a resource base, and incidentally, with each other. This connection enables intermediate storage of data to be transferred from an older system into a newer one. It provides the means to reformat such data at the resource base. Operating via RS232 linkages, most devices are readily able to tap this transitional resource without disrupting normal modes of operating.

HARDWARE

Hardware is similar for each of the three operational contexts (Fig. 1). The microprocessor module includes the central processor (CPU), random access memory (RAM) and read-only memory (ROM). With the future in view, it also contains electrically-erasable programmable read-only memory (EE PROM). Stored programs and/or parameters may be changed from the resource base when user devices or network configurations change. Each module contains 128 K bytes of attached RAM to be used as buffer storage for communications or for local functions. An example of a local function is local editing capability if the user device is a 'dumb' terminal.

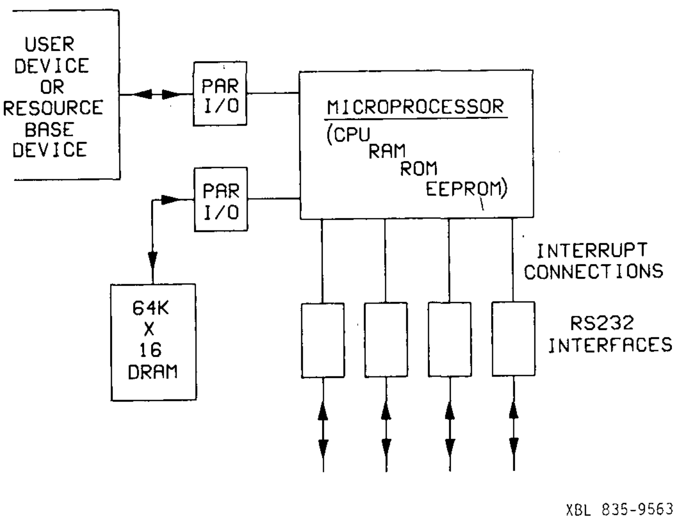


Fig. 1 Hardware module with full complement of connections.

The module supports up to four full-duplex RS232 ports. All are internally used in the modules dedicated to network communications. At a user station, only one (for the network) or two (if the user device is an RS232 device, such as a terminal) are used. At the resource base, only one will normally be used, connecting to the network.

TABLE I: NETWORK CONVENTIONS
Transmission is by variable-length blocks, maximum 16 K bytes per block. Each block contains a header consisting of: 1) BYTES in block (2 bytes) 2) SOURCE (1 byte) 3) DESTINATION (1 byte) 4) BLOCK CHECKSUM (2 bytes) 5) UNALLOCATED STATUS (2 bytes) Each block transmission is followed by a single-byte return transmission signaling 'ok' or 'retransmit'. Handshaking between modules insures no collisions. Ten successive failures in data transmissions cause the link to be closed and a signal to be illuminated. Time out signals, assuming hardware failure after two seconds of 'port busy'.

CONTEXT ALGORITHMS

User Modules (Fig. 2) Must convert a vast diversity into standard flows into and out of the network. User modules span the range from simple devices whose only purpose is conversion of data into human-readable form to a stand-alone computer-aided design system. An electric typewriter connection may require special hardware to drive magnets and special software to translate incoming codes into signals to drive magnet drivers. Standard terminal devices may communicate via RS232 connections. Such devices require buffering and conversion to protocols for communicating with the network. It may be desirable to have standard text editing functions programmed into the user module. Connection to the computer aided design system (or to a similar device, such as a word-processor) may involve passing data through the module, converting to a slightly different format or setting up a different transmission rate. In general, user module software (and hardware in some instances) interfaces the diversity of existing user equipment into standard network conventions.

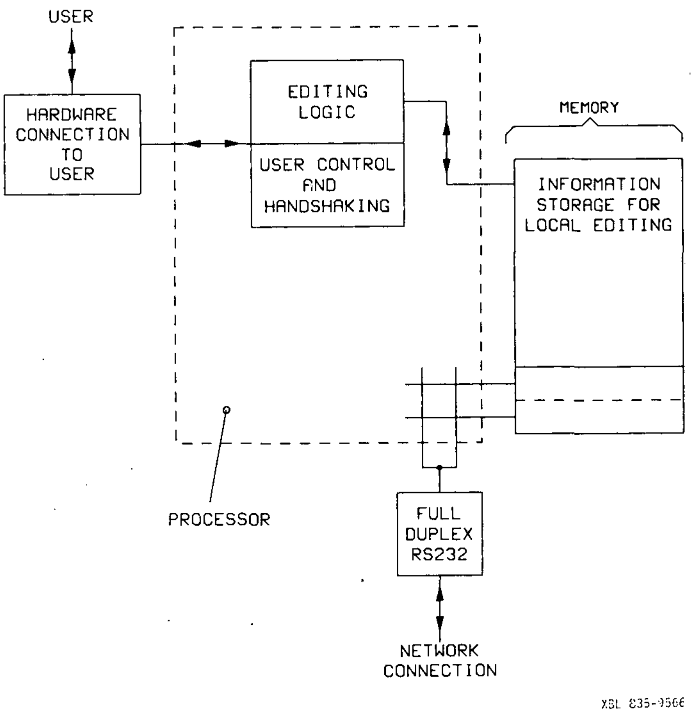


Fig. 2 Software for the user module may include a local editor. It handles control and handshaking for the connection to user hardware, and controls the network connection. Special software may be developed for each user and/or user device. Special hardware may be required to connect to a user device.

Communications Modules (Fig. 3) are service processors for the four active bi-directional RS232 interfaces. Each RS232 interface interrupts the processor when a transmission is complete or when a reception is complete and servicing these interruptions is the primary function of the communications module. Interrupt routines must detect and handle block beginnings, block endings, error detection, return-transmission bytes, and handshaking controls. (See Table) The main program in the module assigns empty buffers to input ports. Output buffers are assigned to output ports depending on the destination byte in the header.

The first byte in any transmission is the most significant byte of a 14-bit number (16 K bytes maximum per block). If the most significant two bits of this byte are set, this byte is assumed to be the return transmission from the last output on this port, and the least significant six bits being reset indicate the previous transmission arrived correctly. This frees the buffer and the link for reuse. If the six bits are not all reset, the buffer must be retransmitted.

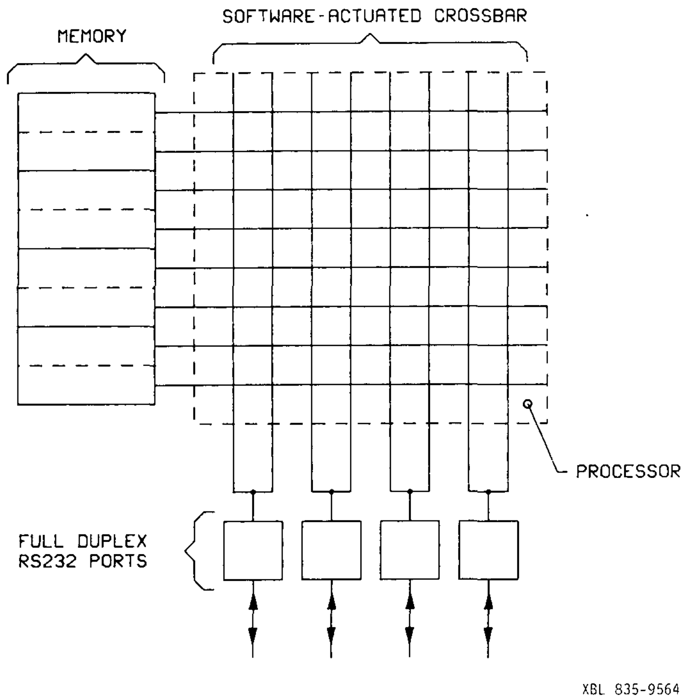


Fig. 3 Software for the network module effectively forms a crossbar switch between the RS232 hardware and the buffer memory.

If the first byte received has both top two bits reset, a block, not just a single byte, is assumed to be arriving. As bytes arrive, they are counted and the checksum is tallied. Any errors in checksum or transmission/reception cause a reverse byte transmission to be queued so the data can be retransmitted. If ten tries fail, a light is lit on the module and the handshaking line is held false until repairs are made.

When an incoming data block is successfully received, the destination byte is used to select an output port. If the best possible output port is in use, an alternate output is set up. If no acceptable alternate is available for two seconds, the module sends the data on the next available output port under the assumption of a local hardware failure.

Resource Modules (Fig. 4) handle the communications network connection to a pool of available resources which may include punched card equipment, paper tape equipment, magnetic tape equipment, disc storage systems, microprocessor development systems, a local batch processor and a link to a remote processing facility. These modules are equally as diverse as the user modules. Each resource module must accept and deliver data from the network and precondition it for delivery to or acceptance from the resources for which it accepts responsibility. Resource modules in some instances must handle possible conflicts among users. It is not generally desirable to interleave printouts from two users, for example.

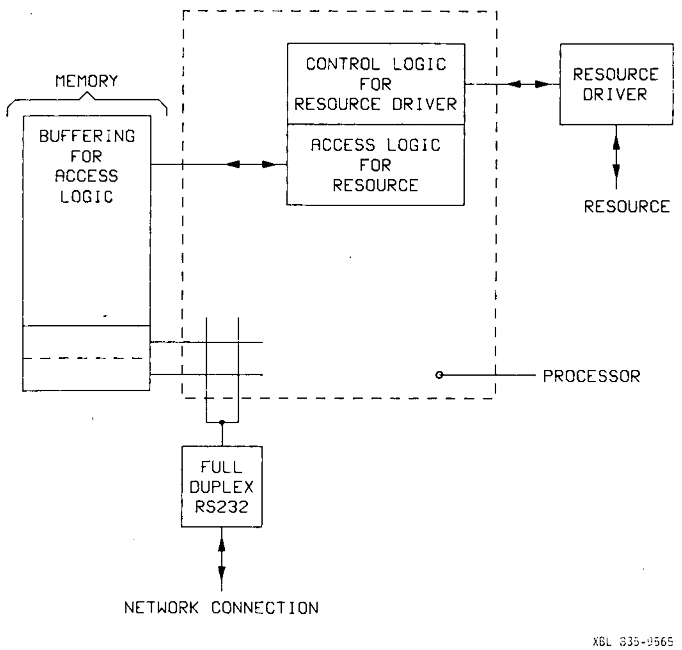


Fig. 4 Resource control nodes use attached memory for front-end buffering.

Our problems are not as complex as they would be in the case of a perfectly general network, however. Our network is designed primarily to economically provide the flexibility to ease orderly transitions from what is currently operational into what will perhaps become a more general network. In this instance, the scheduling problem is more an occasional inconvenience rather than a continuing problem demanding a general solution.

INTERCONNECTING THE MODULES

Module interconnections must handle not only data transfers into and out of the resource base, but transfers among users. We propose a tree configuration with cross-connections. The tree is truncated where the number of branches equals the number of desired connections to resources. (Fig. 5)

Each node in the tree can be thought of as lying on a coordinate grid, and the node is given its coordinate value. When a message arrives, its routing depends on the difference between node coordinates and the destination coordinates arriving with the data block. First routing choice for the block is into the link representing movement in the direction of the maximum coordinate difference. Second choice is movement in any direction which will decrease the coordinate difference. Timeout choice is any possible movement.

PERFORMANCE

Our objective is to provide linkage for occasional use among users and between users and a resource base. Our only valid performance demand is that the connections exist and that they not be so slow that users cannot effectively interact. The fundamental limitation is our choice of RS232 communications at 9600 BAUD. With no additional delays, this transmission/reception rate (about one character per millisecond) is acceptable to a user interacting via text. Enhancements occur by virtue of each user having a smart module and buffering through

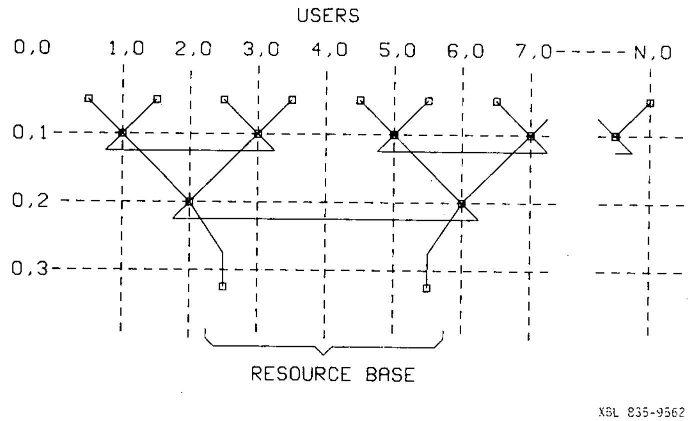


Fig. 5 The tree interconnection is truncated at its base where the number of bottom branches equals the desired number of connections into the resource base.

which local editing may occur. If the network degrades communications, the added delays will only be noticed during block transfers. Assuming no delays caused by conflicting user demands or faulty hardware, the network delays will correspond to the number of nodes through which the data must pass multiplied by the block transmission time into or out of the node.

If the block is assumed to be a line of text, averaging 50 characters, each node represents approximately 50 ms of delay. Transfers to or from the resource base are on diagonals as well as on horizontal linkages, and so are more efficient than transfers among users, all of whom share one coordinate value. So although getting from one user to another requires at least as many delay times as the difference in coordinate values, getting from a user to the resource base will require at least as many transfers as the difference in vertical coordinates, but many require no additional transfers for differences in horizontal coordinates.

If we set our criterion of acceptability at one second turn-around time, ten delays (50 ms x 10 delays) is acceptable. The tree should be less than ten levels high. If the resource base is represented by one node (thereby producing the tallest tree), ten levels above this provides for $2^{10} = 1024$ users. In our case, this is about two orders of magnitude greater than necessary. Translated into our reality, our transfer time to or from our resource base will be very acceptable.

Considering user-to-user transfers, every time the number of users doubles, another vertical node is required to effect a user-to-user connection. For up to 16 users, four vertical levels are required, and a transfer between the two most distant users will require at least 16 node transfers (the difference in horizontal coordinates), and depending on the efficiency of getting from one vertical level to another, may require one or two more. The resulting bi-directional response time may be 3-1/2 seconds. This is long, but the interaction is person-to-person, not person-to-machine, and the turn around time of the responding person is likely to be much longer than the data transfer time.

CONCLUSIONS

Using the same basic microcomputer hardware module in three programmed contexts, it is practical to build a usable transitional network in an office environment using RS232 communications at 9600 BAUD. A truncated tree network with cross linkages provides acceptable interactive response times between any user and the resource base, and among users.

ACKNOWLEDGMENTS

I wish to acknowledge the dedicated efforts of F. Gin in generating the hardware module for use in this proposed system.

This work was supported by the Director's Office of Energy Research, Office of High Energy and Nuclear Physics, Division of Nuclear Physics, and by Nuclear Sciences of the Basic Energy Program of the U. S. Dept. of Energy under Contract Number DE-AC03-76SF00098.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720