LU6.2 on Unix 4.2BSD

M. Satyanarayanan
27 October 1983

The purpose of this brief document is to summarize recent discussions between and the members of the ITC Transport Group and myself.

The basic question being addressed in these discussions was "How should the the VICE Application Protocol interface with LU6.2?" Initially the application protocol will be built on top of IP/TCP on the Ethernet. When the LU6.2 is usable, the application protocol will use it, but this change will be transparent to the programs using that protocol. It would be nice if the changes that have to be made are not structural in nature, but only involve minor modifications.

Unix (4.2BSD) has a user-level abstraction called a *Socket* that allows two processes on different network nodes to rendezvous and communicate. This abstraction and the associated mechanism provides a clean and uniform interface for programs to dissociate themselves from specific transport protocols. It also supports a client-server communication model that would be simple to map to from the proposed ITC File System design.

It was the consensus of the participants that the Transport Group should make LU6.2 available to programs via the socket abstraction. Put differently, the only changes to the code that should be necessary are the parameters associated with socket creation, and the format of socket addresses.

LU6.2 provides reliable, sequenced, flow-controlled transport of data. This function directly corresponds to a socket type, *Stream Socket*, in Unix. However, it has been the experience of many implementation groups that acceptable performance in distributed systems is possible only with the use of application-specific protocols built directly on

datagrams. Such an approach performs error-checking only end-to-end; if an error is detected, the entire message is retransmitted. In relatively error-free networks, application protocols built on datagrams have repeatedly shown better performance than ones built on reliable byte streams.

Unix provides a socket type, *Datagram Socket*, that provides datagram-level communication. Implementing this socket type on LU6.2, however, poses some difficulties. The primary source of this difficulty is the fact that error-control, flow-control, addressing, and sequencing are spread out through the SNA layers, and obtaining the exact semantics of a datagram socket is nontrivial. It was felt, however, that support for datagrams was vital and that the Transport Group should implement datagram sockets.

Socket support for LU6.2 is being provided in two stages. In the first stage a set of C routines corresponding to Unix socket primitives will be implemented. The purpose of this stage is to check functionality and adherence to socket semantics. The second stage involves transferring this functionality into the kernel and tuning the implementation. Reasonable performance can be expected only at the second stage.

In summary, the VICE Application Protocol will be built on the standard Unix socket abstraction. Initially the sockets will be Internet-domain sockets, using IP/TCP protocols. The Transport Group will develop LU-domain sockets and tune their performance to a point where they can be used. Both stream and datagram sockets will be available.