

Empowering Low-Bandwidth Users

Larry Press¹

To empower people who are not computer professionals, systems must be conceptually simple and easy to use. Unfortunately, non-professional users are likely to have low-bandwidth connectivity, particularly in less-industrialized nations and regions. One strategy in such cases is asynchronous operation, where a bi-directional transfer of messages occurs when the user logs onto the Net. Seamless integration with the local file system and other tools is also necessary. Three examples of such systems are mentioned, but none are in widespread use on the Internet today.

This session is about empowering users, but users in less-industrialized nations (and many parts of industrialized nations) are often on a tight budget. This is likely to result in low-bandwidth connectivity. Consider my own situation.

I frequently access the Net by dialing into a timeshared computer from my home, using a computer with a low-speed modem. I have been doing this for nearly 20 years now, and have gotten fairly good at it -- I type a few DOS commands, then execute Kermit, my terminal emulation program. A quick command to Kermit dials the Unix-based host where I log in and read my mail using mm, a mail program. To reply to mail messages I enter emacs, an editor. If I want to transfer a file to another computer, I use ftp, and since I have a project on another host, I often telnet to that VMS-based host where I use EVE and Mail, the VMS editor and mail programs. To send or read Usenet News, I use Pnews and rn respectively. I also often compose documents on my home computer (using WordStar) and upload them to the hosts, and download files from the hosts to my home computer.

I am tired just thinking of all of this. To use the Net I have to be familiar with 13 programs, and that is only to gain access -- it says nothing of knowing about servers I connect to. Perhaps more important than knowing a subset of each of these programs, I need a model of their relationships and the data flows between them. Beginners are often confused as to which machine they are talking to and where a

file is stored.

I am strongly motivated and have been using computers for many years, so acquiring this model and learning a subset of the commands for 13 programs was worthwhile and not too difficult. On the other hand, if I were not a computer professional, and had to pay for Net access, I doubt that I would be willing to make the effort to acquire a model of the Net and learn the details of so many programs.

But it does not have to be so difficult. For example, I am working with students in an inner-city high school in Los Angeles. The school received a grant which provided them with eight NeXT workstations and a T1 connection to the Net. Within ten minutes of turning the system on for the first time, we retrieved a file from Africa, displayed it, and printed it. This was done with a few mouse clicks using a Gopher client with a graphical user interface. It was easy because the African file appeared as if it were any other file in the NeXT file system. The students did not have learn to use a different file system, and they did not have to understand that there was a different host computer somewhere. The Net seemed to be a single system, where accessing remote information is the same as accessing local information. Their local and remote email are also seamlessly integrated as is their editor. Composing and spell checking an email message is the same as writing a term paper.

If everyone had facilities like this school, Net access would be greatly simplified, and we would empower even casual users. With technology improving at the rate it is, a NeXT-equivalent workstation will be affordable to many people in a few years; however, the T1 connectivity will not be ubiquitous or cheap. The situation is of course worse in poorer regions.

While the T1 connectivity is very nice, we could do a lot without it. Some of our Net activity is interactive, for example file browsing and transfer, but other activities, such as email and reading lists are asynchronous. Even some file-browsing and transfer can be done asynchronously if we have suitable query processing software and agents which seek and screen information. If I look at my own Net usage, the great majority is asynchronous communication with people, not information retrieval.

¹ Department of Computer Information Systems, California State University at Dominguez Hills, lpress@isi.edu.

Since terminal equipment is becoming cheap and universally available, and high bandwidth connectivity will come along more slowly, we should pay attention to support of asynchronous users. When I establish my low-speed connection to the Net, I do not want to begin typing interactive commands. As soon as I log onto the net, there should be an automatic bi-directional transfer of all incoming messages from people, agents, lists, news groups, and pending queries and of all outgoing messages I have generated since I last logged on. The state of my workstation database and that of the Net should be automatically reconciled. What I see is what I have, regardless of its location. The connection could then be broken, and I would read incoming messages and compose replies which would be sent in a batch when I next connected. This asynchronous operation should be as rapid, conceptually simple, and seamless as a session on a T1- connected workstation. The user would not even have to be cognizant of the fact that messages were queued rather than sent immediately after being composed.

This mode of operation is well suited to a world where terminal equipment is cheap and bandwidth expensive, which is the case today with wireless communication using portable computers. There are examples of rich support for asynchronous network connectivity in the portable market. One, the PenPoint [1] operating system, has hooks for intermittent network connectivity and automatic transfer of queued messages between systems when a connection is established. The object-oriented PenPoint user interface smooths over any seams between documents on the terminal and host. It is likely that our network terminals will end up looking more like pen-based cellular telephones than desktop computers.

Lotus Notes also supports asynchronous communication while providing a rich user interface [2]. When connections between Notes systems are established, their databases are automatically reconciled. They may then be viewed and altered off-line using flexible, high-level tools.

Communication cost are also relatively high in less- industrialized nations, and another example of a system supporting asynchronous connection with powerful user interface is found in a network being developed for Chilean elementary schools by Pedro Hepp and his colleagues [2]. The Macintosh-based network is being deployed in 100 schools. Cost considerations dictate asynchronous connectivity, and grammar school students and teachers are not interested in learning things like DOS, Unix, emacs, and Kermit. The user interface centers around a picture of a typical Latin American plaza.

The plaza has a "kiosk" (student and teacher newsletters, and articles like comics and stories), a "museum" (educational software with curriculum notes, user's experience, and recommendations), a "cultural house" (computer conferences), and a "post office" (email). The metaphor is continued "inside" the "buildings," and point-and-click interfaces are used throughout. Data and message transfer is done in batches, and Hepp states that the system can be learned in about 30 minutes.

Given the financial pressure we all feel, it may be tempting to criticize a project like Hepp's by saying that he could achieve the same functionality using \$300 PC clones instead of Macintoshes. But workstation cost is only the tip of the iceberg. The costs of communication, training, support staff, documentation, "selling" the system, and so forth far outweigh terminal cost today, and terminal cost is falling rapidly. We should trade terminal complexity for training, support, and bandwidth in such situations.

References

- [1] Carr, Robert and Shafer, Dan, *The Power of PenPoint*, Addison-Wesley, Reading, MA, 1991.
- [2] Hepp, P. K., Alvarez, M. I., Hinostroza, E. S., and Laval, E. M., *'La Plaza,' A Software Design for an Educational Network*, phepp@lascar.puc.cl.
- [3] Kawell, L., Beckhardt, S., Halvorsen, T., Ozzie, R., and Greif, I., Replicated Document Management in a Group Communication System, in Marca, D. and Bock, G., *Groupware*. Los Alamitos, CA, IEEE Press, 1992.

Author Information

Larry Press' interests include network applications in less-industrial nations and inner cities. He teaches and is a contributing editor to the Communications of the ACM.