

THE NETWORK OPERATING SYSTEM CONCEPT FOR FUTURE SERVICES

Wu-Hon F. Leung and Gottfried W. R. Luderer

Wu-Hon F. Leung is supervisor of the Network Operating System group in the Advanced Switching Network Department of AT&T Bell Laboratories, Indian Hill Park, Naperville, Illinois. He and his group do applied research on software technologies supporting new network services. They have developed an experimental testbed, based on fast packet technology, to investigate interface and control software for interactive multimedia conferencing applications, as well as protocols and network management software for wide area packet switching networks. Mr. Leung holds a B. S. in electrical engineering from California State University, Northridge, and M.S. and Ph.D. degrees in computer science from the University of California, Berkeley. He joined AT&T in 1978. **Gottfried W. R. Luderer** is head of the (continued on page 35)

In this paper, we present some recent results towards developing the notion of a network operating system. We consider this an approach to support the evolution of the network offering more advanced services. We first speculate on the nature of future services by expanding on the attributes of the Universal Information Services marketing vision. Then, we introduce the network operating system concept by extending the computer operating system concept from only managing and providing data services to include interactive communications using multiple media across a larger geographical area. Finally, we describe some results of an experimental system implemented on a prototype fast packet network.

Introduction

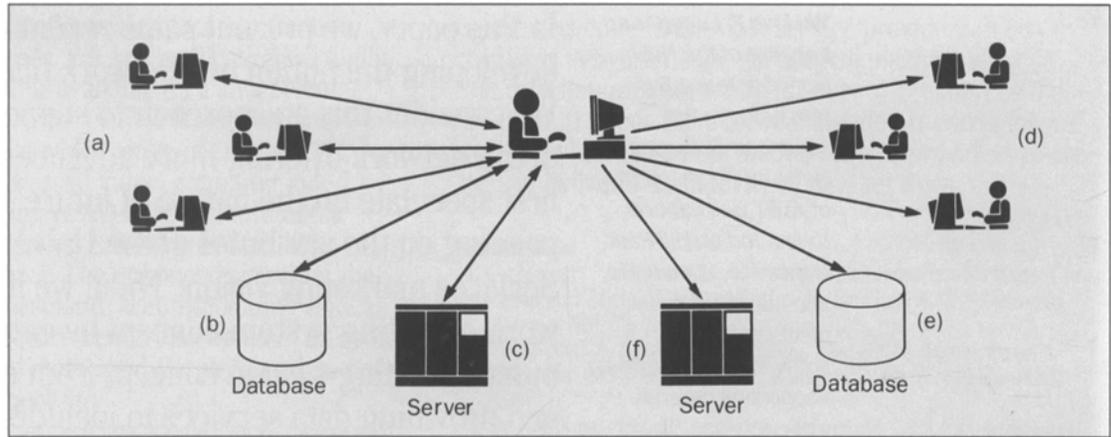
It is our belief that the network of the future has the potential to provide a vast array of new services that can offer enormous benefits for the work life and private life of the average citizen. To realize this potential, we must lay the groundwork for a new network architecture. We must then define new concepts that guide the evolution from today's network to the envisioned network of the future. The approach we have taken toward this goal is to extend the computer operating system concepts to a network operating system.

We begin by presenting our view of future—or futuristic—network services. We then proceed to suggest how the network operating system concept can guide us towards the realization of these services.

The Needs of the Knowledge Worker

Sophisticated high-tech telecommunications users of the future are sometimes referred to as *information workers* or, more appropriately, as *knowledge workers*. We use the term *information movement and management* to describe the way the telecommunications network would serve these users.

Figure 1. The job of the knowledge worker.



The basic job scenario of knowledge workers is shown in Figure 1 as a *decision-making process*. This is an input-output process, usually with many iterations. As input for their decisions, knowledge workers need to have facts and opinions, i.e., data that exist both in data banks and in the minds of other people. The network must allow easy access to both sources. Because facts and data must be presented in the most appropriate way, we need tools for visualization, graphic or animated presentation, conversion, combination, and correlation. Multimedia conferencing—aided by computing tools that provide the ability to record the conference as a multimedia document—with extensive recording capability is another aspect of this environment.

The job of knowledge workers is to make decisions that involve information movement and management according to a basic information flow shown in Figure 1.

1. The knowledge worker interacts with other people (a) using facts stored in information systems (b) and processing aids in servers (c).
2. The decisions arrived at in this first part of the process are communicated to other people (d), stored as new facts (e) and/or changes to programs (f).

Once the users arrive at decisions, through many iterations and through consensus-seeking with those whose concurrence is essential, the information resulting from those decisions must be communicated to the appropriate recipients in its most effective and efficient form so they can act on it. We can, for example, see

Panel 1. Terms and Acronyms in This Paper

ANSI	American National Standards Institute
CCITT	International Telephone & Telegraph Consultative Committee
CCS7	Common Channel Signaling 7 network (ANSI standard)
FPT	fast packet-switching technology
IEEE	Institute of Electrical and Electronics Engineers
ISDN	integrated services digital network
LEC	Local Exchange Carrier
OSI	open systems interconnection basic reference model
PTT	Postal, Telegraph & Telephone (European equivalent of an LEC)
Q.931	CCITT network layer protocol
UIS	Universal Information Services concept

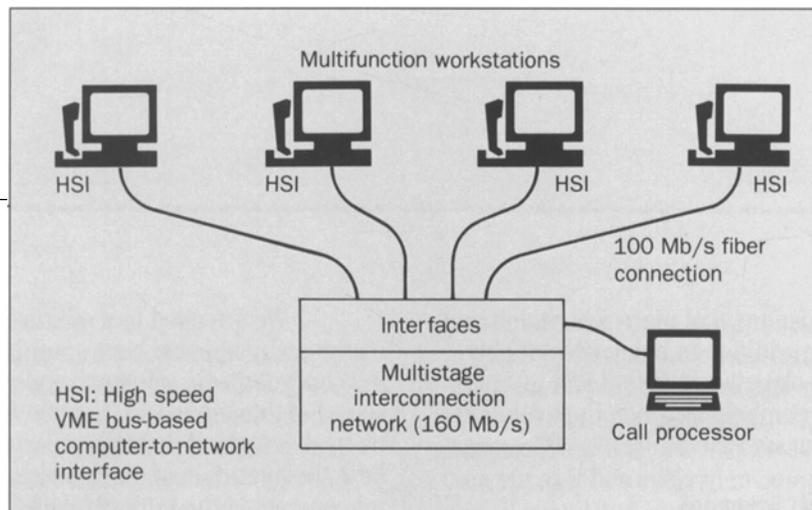


Figure 2. A knowledge working using a multimedia workstation.

needs for encryption, authentication, verified acknowledgment, tracking, and possibly even legally authorized recording. Different types of information such as voice, data, text, graphics, and video need to be transported, converted, stored, and retrieved through arrangements involving many parties.

The best tool for the knowledge worker seems to be a multifunction terminal with a large high-resolution color screen enabling multiple windows that—besides the conventional computer text—display graphical data, facsimile images, and video conference pictures. Voice and sound messages can be dealt with by an integrated speakerphone or alternatively, by a lightweight, tetherless headset (as illustrated in Figure 2).

The topology of possible arrangements will be much more general than today's point-to-point, multi-drop, or simple conference arrangements. We expect that complex multiparty connections with controlled multi-directional information flow for multiparty-multimedia conferencing will become possible.

The common term *information productivity* has become part of the vocabulary of the knowledge work environment. Improving information productivity has been chosen as the broadest objective for future network services. Whereas *manufacturing productivity* long has been understood to measure the output of goods and services per cost unit, we must not get trapped into reading the inexpensive generation of new information into the term information productivity. Instead, the term should be understood as the effort to measure improvements in the *whole process* of communication and decision making based on the effort applied.

To achieve high information productivity, it will be necessary to have ergonomically well-designed user interfaces to operate the multifunction terminal. Learning how to operate this terminal should be no more difficult than learning how to type or use a spreadsheet program. Much ergonomics research is needed in the design of such interfaces.

With this notion of the future knowledge worker established, we will now turn to a visionary description of the future network.

A Vision of Future Information Services

While it is relatively easy to assess today's data communication market needs, it is quite a different matter to speculate about tomorrow's markets. Modern technology can provide virtually unlimited bandwidth and keep delay low. The main consideration is one of cost benefit: that is, will it be worthwhile to offer such advanced capabilities? What could these new services be?

Communication service providers such as Local Exchange Carriers (LECs) in the United States and overseas Postal, Telegraph & Telephone organizations (PTTs) have a vital interest in expanding the telecommunications market, as do telecommunications equipment vendors. Thus, the creation of an advanced telecommunications market is a unifying objective for the entire industry. This market must offer products and services that improve a customer's information productivity.

In 1985, AT&T made an effort to give momentum to this unifying objective by presenting a vision of the future called *Universal Information Services* or UIS. With guidance from the general UIS attributes, we will ex-

plore some specific applications that place a high demand on network capabilities in addition to being presumably useful to customers. Our objective is to find and test architectural concepts that can help us guide the evolution of network architecture far into the future.

Some Advanced Application Scenarios

The currently defined integrated services digital network (ISDN) will be capable of delivering a vast set of new services. Even more advanced applications become possible if one assumes an abundance of bandwidth, processing power, and storage capacity. We will give several examples of service types requiring network resources substantially beyond the reach of today's network. These are:

- Multimedia teleconferencing
- Multimedia documents
- Classroom teleconferencing
- High resolution documents
- Viewgraphs and animation.

Multimedia Teleconferencing. The *multimedia teleconferencing* is an epitome of the future telecommunications session. Imagine a conference call involving several parties at several locations, each with a multimedia service terminal. As mentioned earlier, such a terminal has not only voice capabilities, but also a multiple-window, high-resolution color screen that could visually display any group of teleconference partners. Documents or viewgraphs could be displayed in separate windows; or a single-window presentation could be viewed by all. Data could be manipulated differently at different sites, and these various presentations could be synopsized in different windows.

Multimedia Documents. The ability to store not only data but also voice and video opens new opportunities. Documents might have voice annotation in which a screen marker shows that a "voice bubble" can be recovered from storage and played. Someone reviewing a draft paper might choose to add voice comments for the author. Or the author might add voice commands as editing instructions to the typist.

We are used to a mixture of visual and text information; why not also have sound included? Certainly a future encyclopedia will store under the entry *robin* the sound of this bird's call. Certain facts and processes can be best explained through an animated picture sequence, be it the operation of an electrical backplane bus or the movements of the human heart. Eventually, electronic speech recognition will appear and provide yet another human-engineered interface opportunity.

Classroom Teleconferencing. The *classroom teleconferencing* is another example of a multimedia application. Students can see the lecturer in one window and the blackboard or projection screen in other windows. If a student wants to ask a question, the lecturer would have a window to identify the student and decide whether to turn over voice control at that point. Students would be able to get copies of the lecture material, store it in their files, and add their own annotations. A listener might take "voice notes" during a lecture and append them to particular lines on the screen.

High Resolution Documents. For *high resolution documents*, we expect quality equivalent or close to the original. This means letters and memoranda can be transmitted in electronic form with hard-copy as an option. This capability would allow a variety of applications: for example, data banks of all company memoranda, patents, collections of illustrations, and other records.

Viewgraphs and Animation. We are sure there will be uses for this technology that cannot be imagined today. We see an opportunity to move away from "static" viewgraphs to "dynamic" viewgraphs, which improve the communication process by animation.

Before we present our approach to a framework for introducing such new services, we will put this approach into a larger perspective by defining our view of the future network.

Network Architecture

The notion of *architecture* refers to two different aspects of the same concept. Some researchers have divided them into the related categories of *exo-architec-*

ture and *endo-architecture*.

The exo-architecture, or architecture as viewed from the outside (i.e., the user) might be called "the user-perceived structure of the system." That is, it includes interfaces that allow users to bypass the physical system design components and codes, and instead access and use a "logical" system view for their intended purposes.

The endo-architecture refers to the internal structure of the system as envisioned by the designer. It defines the system components, including their internal interactions and interfaces. It is more of a physical view and usually is much more complicated than the exo-architecture.

To propose an architecture of the future network, we must first propose its exo-architecture. We have prepared the ground for this task by discussing, later in this paper, some of its potential services. From a fundamental point of view, this network will provide connectivity to a set of end users who are joined to it, following in general terms the concept outlined as the universal port.¹ In addition to end users, the exo-architecture also supports network servers that may use the universal port or a specialized server port.

For the following discussion of a network operating system approach, we take the viewpoint of the exo-architect.

Approach: The Network Operating System

In the last twenty years, significant influences on the evolution of the network have come from computer science. We have seen digitalization, stored program control, and the introduction of packet switching in signaling and data networking. More is certain to come. One particularly useful guiding principle that computer science offers is the model of the *computer operating system*, especially if we apply two generalizations to adapt it properly to the telecommunications network. The resulting concept we call the *network operating system*, a term that has been used in various other contexts, e.g., to describe the operating system of a multiprocessor system; or for the services provided by the upper layers of the open sys-

tems interconnection (OSI) basic reference model.² We propose to view the evolving future network as one massive general-purpose information movement and management system, under the control of the network operating system. We will first briefly review the operating system concept as it has evolved in computer science over the last 30 years.

A computer *operating system* realizes the system architecture visible to the user. It transforms the hardware by performing two major functions:

1. It gives users a *convenient general-purpose interface*, which allows them to define in functional terms the tasks to be done. This interface includes a command language, a set of system calls for programs, and a collection of building blocks and tools. This constitutes the *exo-architectural* aspect of the system.
2. It implements a *resource management system* that interprets the tasks specified by the user and controls and manages the system resources to perform these tasks. This constitutes the *endo-architectural* aspect of the system.

Before we elaborate on the generalizations, let us show the analogies to the operating system concept. As with the computer, the network needs a *user interface* to define its tasks. This interface is supported by message-oriented signaling of the kind defined in standards such as the International Telephone & Telegraph Consultative Committee (CCITT) Q.931 network layer protocol, or the American National Standards Institute (ANSI) Common Channel Signaling 7 (CCS7) network. The network also needs a *control system* that manages its resources to perform the desired functions. These two basic aspects are only its most general similarities to the computer. At deeper levels, there are many more analogies between today's computer systems and the telecommunication system's needs. These similarities are not accidental. Consider that the most dramatic innovation in today's telephone network has been the introduction of *stored program control*, which has brought with it a wealth of computer concepts.

Rather than use the operating system as we know it today, we will generalize and extrapolate the operating system concept into the future, by introducing multiple media, geographic distribution, and peer-to-peer communication. We can then use it as a guiding model for the evolution of the future network.

The first generalization is *geographic transparency*. The network operating system differs from a computer operating system in that it deals not just with a single computing system but with the entire network.

The second generalization moves us from data-only to *multimedia services*. That is, whereas today's computers deal almost exclusively with data, the network operating system has to deal with voice and image, including video, as well as classical data. And these media are to be handled not just for interactive use but also for storage, conversion, etc.

The third generalization introduces *peer-to-peer communication* by extending the man-machine or user-host communication scenario to include user-to-user or user-host-user scenarios. In such a scenario, the human user of the computer could be replaced by another computer; or, following an increasingly popular trend, we can consider the user to be "augmented" by a personalized work station.

The Operating System Paradigm

A large body of knowledge about computer operating systems has been developed over the last 30 years that can serve as a guide for the network operating system. It can serve, in effect, as the integrated service system paradigm. Several aspects of this model can be transferred from the operating system model to the network operating system model.

The Human Interface. At the *human interface* level, we envision extensive use of windows, menus, and icons to present the end user with a plausible model of the network and its resources. This model will provide a graphic simplification of the real network architecture. Graphical representations will show dynamic changes *as they take*

place. As an example, consider a future multiparty call setup. A window can show a simplified graph under construction with icons of telephones depicting ringing and answering events. The touchtone pad usually will be replaced by a keyboard; many end users may even operate a pointing device such as a mouse or joystick.

Such developments may lead to the demise of the telephone number as we know it today. In computing's infancy, objects could be accessed only by giving their address, even in hexadecimal. This inconvenience was soon overcome by superimposing a name that was much more convenient than the numerical address. Even though telephone numbers have lost part of their address significance (except in area codes and, perhaps, exchange codes), we are still using cumbersome "numeric names," which carry much of the previous address baggage. We hope this will change.

Because computer users have been navigating through hierarchically organized filename spaces for many years, *alphanumeric names* should be much more attractive. Such naming has both a local and a global aspect. Locally, a user's telephone directory is a personalized list of names mapped into network numbers. The list can be kept in the user's terminal, in a central file, or even in an updatable "smart-card." A global name service must be available to find anyone anywhere. In this case, the user might engage in a "dialogue" with the computer system. Of course, at the end of the dialogue the entry may be added automatically to the user's personal directory.

In a time-sharing computing system, users have unique alphanumeric "user ID's." Instead of today's telephone number, it is conceivable that people would get a unique alphanumeric form of their name registered (such as a Social Security Number), as has been common for Telex addresses or tickertape symbols. For this network name, uniqueness would be guaranteed in a local area, be it the town, county or state—certainly not the local exchange. This is similar to certain 800 services that use the U. S. telephone dial's mapping of three letters to the digits 2 through 0 (e.g., 1-800-ATT-MAIL)—a

practice that nevertheless is too restrictive to be satisfactory for future evolution.

On-Line Registration. In time-sharing systems, when new programs or databases are offered for public use, there is some form of *advertisement*, or access to an on-line services directory with manual pages and other user aids. Similar registration facilities will be offered from the network, that will take on the function of a *broker*. The network would offer a "yellow page" entry, measure access from customers, and perform the billing task, all functions performed now in advanced time-sharing systems. In addition, the creditworthiness of the customer requesting service can be checked beforehand.

Command Language and System Calls. Another function offered by operating systems is a *command language* with the capability of storing executable sequences of commands, often called *scripts*. An example is the call-forwarding service that depends on the day of the week and the time. Network customers could be offered a similar capability. Such scripts could be triggered by incoming voice or data calls and take different branches depending on time, origin, and type of call. This feature would provide convenient customer programmability of the end-user interface.

To request execution of privileged functions in a controlled fashion, present operating systems offer *system calls* that are used from within application programs. We can foresee future network counterparts of such capabilities made available to so-called third party network service providers.

Connectives. Finally, operating system command languages offer *connectives* to combine programs and databases. The UNIX® system's *pipe* facility,³ which allows one program to forward its output as input to another program, is a convenient command language notation and powerful operating system service of great utility. Not only does it reduce repetitive programming by fostering re-use of software, it also is very helpful for understanding, modifying and debugging command scripts. With the *pipe* connective, programs are considered as

so-called filters, providing a transformation from an input to an output format.

The same concept can be generalized for multimedia communications, whether the program is a noise filter, voice editor, image enhancer, or simply some data mapping or conversion for better representation. New connectives will allow information streams to be split or joined. For example, incoming voice could be played out and recorded at the same time, a voice commentary could be added to an image sequence, or complex multi-party set-ups could be arranged.

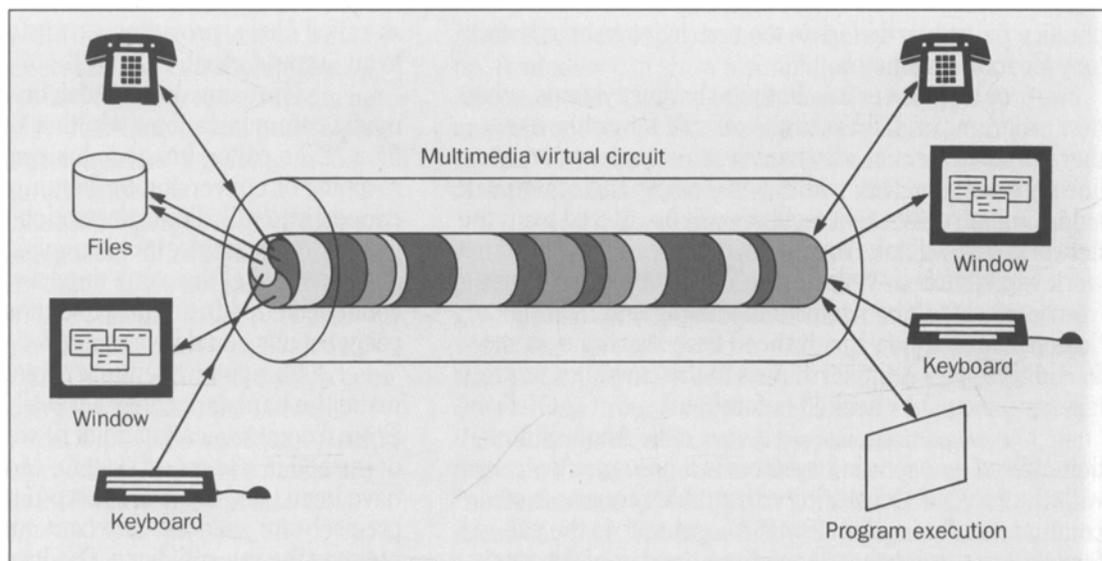
Computer operating systems are supposed to make the hardware easily accessible to human beings. From a communication point of view, what we have described is a *user-host* system. More recently, computers have been used for *user-user* communication, or more precisely, for *user-host-user* communication, with the computer as the intermediary. The increasingly popular *electronic mail* or *bulletin board systems* are examples of such use. In addition, the UNIX system provides a command called *write* that allows interactive text communication by enabling a logged-on user to type text that appears on the terminal of another logged-on user.

These are just the beginnings of user-user communication. For the network operating system, we expect this so-called *peer-to-peer communication* to play an increasingly important role. We have applied some of these ideas to build an experimental system in the laboratory. This is the subject of the following sections.

Elements of an Experimental Network Operating System

Over the last few years, we have developed a prototype of an experimental network operating system. It is implemented on a fast packet-switching system testbed⁴ and currently supports interactive multimedia communications.⁵ It provides a variety of applications, including the ability to record a multimedia conference in real time and allow existing programs running under the UNIX system to be used in a conference situation without modification.⁶ The network supports switched virtual cir-

Figure 3. The configuration of an experimental wide-band packet network.



30

cuits and is capable of transporting variable-length packets coming from access links running at a data rate of greater than 100 megabits per second (Mb/s) each. Thus, voice, data, and even video can be transported on a single network. Several multifunction workstations have been equipped with high-speed fiber interfaces to the network. Figure 3 shows several multimedia workstations connected to the network. These multimedia workstations are equipped with peripherals such as a voice handset and image scanner. The fast packet network is controlled by a call processor that controls the connectivity. That is, it sets up, routes, and tears down connections, and manages other resources of the network such as bandwidth. As part of our studies, we implemented several routing and resource management algorithms in this testbed.

The Use of Fast Packet Technology (FPT). There are several reasons for our choice of fast packet technology as the basis for our work. We will discuss a few of those reasons in this section.

The proven ability of FPT to transport voice,

data, and images in a single network is important for multimedia applications. During interactive multimedia communication, there is a *temporal relationship* among the various information streams of voice, data, and images. If these media are transported by different networks, maintaining this temporal relationship can only be by probability, not control. Later, we will show our approach to maintaining the temporal relationship in multimedia communication.

A unit of information in a packet-switching network is a *packet*. Each packet is identified by a *header* that contains information such as the packet sender and receivers, and allows special processing. Our prototype uses this property to allow the user to discriminate different information streams at any particular time. For example, during a multiparty conference, the user may choose to listen to a selected person or talk only to a subset of the people in the conference. The network can use the information in the packet header to filter out the information to be sent to this user and to direct the information from the user to the proper destinations.

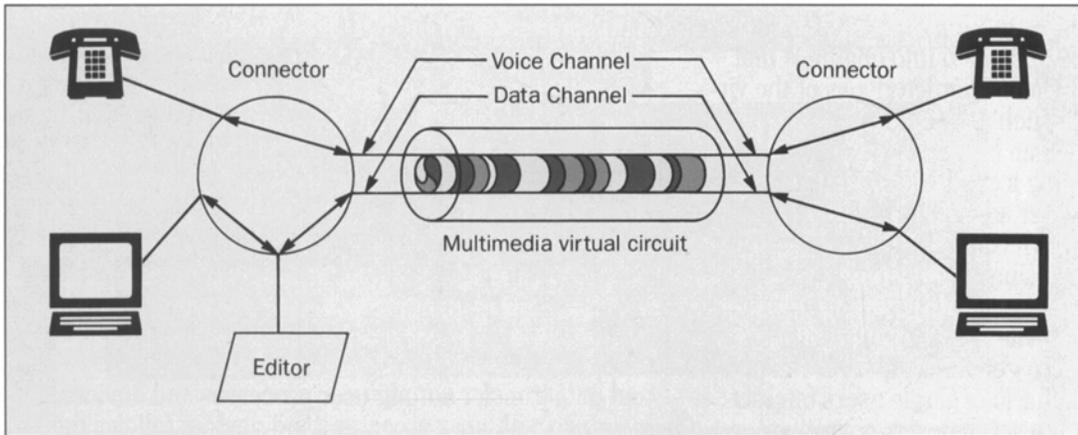


Figure 4. Multimedia virtual circuit.

A communication connection between two users is implemented as a “virtual circuit” in an FPT network. A virtual circuit is established by agreement between the network and users. Once it is set up, a “virtual circuit ID” is assigned to the connection. This identifier is contained in the header of every packet exchanged by users; it is used by the network to route the packet to its destination. Since the virtual circuit ID is a logical entity, theoretically a user can be involved in an unconstrained number of connections and information streams.

Scope of Investigation

In an interactive multimedia communication application, multiple information streams of voice, data, and signaling are involved. To build such an application, it is important that these information streams be conveniently and efficiently controlled. Thus, our investigation covers two areas:

- Communication software to support multiple information streams in a telecommunication connection
- Operating system mechanisms to manipulate (e.g. direct, store, split, and join) voice, data, and image information streams in workstations connected to packet switching networks.

Our multimedia workstations and call processor

use the UNIX operating system. Our software is embedded into the UNIX system environment, enabling us to take advantage of existing UNIX system utilities and application programs. In the following sections, we highlight some of our results.

The Multimedia Virtual Circuit. Interactive multimedia communication has more stringent requirements than multimedia document and messaging systems. For example, when one participant says: “*I am moving the cursor down one line,*” the statement should be heard at the same time the cursor moves. We solve this *temporal synchronization* problem using the notion of a *multimedia virtual circuit*.

A multimedia virtual circuit, as illustrated in Figure 4, while it provides a multiple information channel interface to upper layer software, multiplexes packets of different information streams into a sequence of packets transported by the network in a single virtual circuit. As a result, the ordering of the packets, and hence their temporal ordering, is preserved.

Efficient multiplexing/demultiplexing and buffer management algorithms are necessary to maintain temporal synchronization. The algorithms used in our prototype have been reported by Leung, et al.⁷ The transport layer interface of the multimedia virtual circuit

allows a user to create and then clear a virtual circuit. The virtual circuit can be subdivided into channels that can then be added and deleted. Characteristics of the virtual circuit and channels—such as priorities and flow/error control policy—can be specified when creating a virtual circuit or adding a channel. They also can be modified at run time. Another advantage of this arrangement is that it relates different information streams belonging to the same call using the network layer virtual circuit ID.

The Concepts of Connector and Active Device. As we suggested earlier, we want to generalize from the computer operating system scenario of single users interacting with a computing system to a network operating system in which multiple users interact, not only to communicate with each other, but also to share in the operation of computer programs. In other words, we are interested in applications in which people in different offices can talk to each other while they are, for example, editing the same file, debugging the same program or working on some other shared-access applications. In addition to writing new editors, debuggers, or spreadsheet programs for such applications, we would like to be able to use existing programs such as the `vi` screen editor and even the unmodified UNIX system `shell`. In addition, the user may want to change the information flow of an ongoing application. As an example, a multimedia conference participant at some time may elect to record the conference in progress.

These applications require some mechanism that can be used to:

- Multicast control and data information written by a user process to several terminals
- Funnel data and control information from several terminals to the same user process
- Change in real time the information transfer path.

We implemented a software abstraction called `connector` to support these requirements. A `connector` supports bi-directional and multicast control

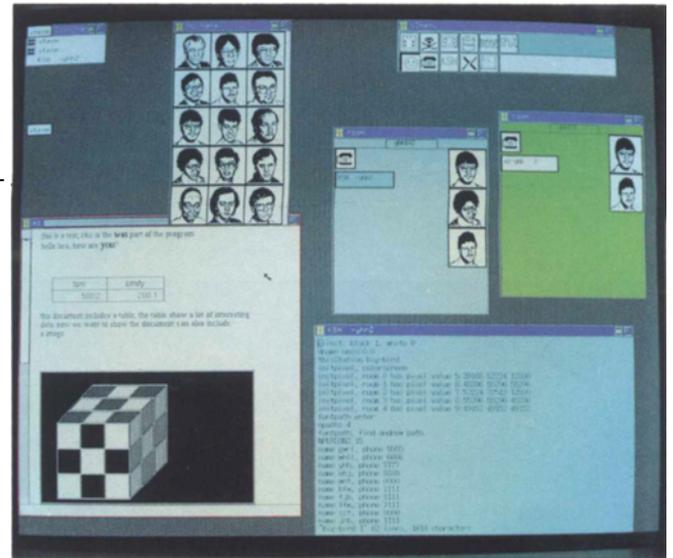


Figure 5. Interactive editing with voice synchronization.

and data transfer among user processes and devices. Its interface with user processes and devices follows the UNIX system uniform device driver interface. The software which controls a peripheral device in our system has the same interface characteristics as a UNIX process and is called an *active device*. Thus, user processes and devices can communicate with each other using a single interconnection mechanism, namely the `connector`. In this framework, to copy a file we would connect a *file active device* to another *file active device* via a pipe or the `connector`. As a result, it is possible to write a simple application program that would first send voice and data from a multimedia virtual circuit to a handset and program, respectively, and later—upon user commands—direct the voice and data to a file.

The `connector` preserves certain properties pertaining to how information is relayed. Information is delivered in the order it is sent. Furthermore, receivers will receive the data sequence from common sources. For example, suppose that source A writes characters “a” and “b” while source B writes “c” and “d.” If receiver C receives the characters in the sequence “a” “c” “d” and “e,” receiver D should receive precisely the same sequence of characters.

Figure 5 shows an interactive editing session with a voice channel to let writers discuss their work, and a data channel to connect the remote user to the single

Existing Calls

Name	Feature	Source	Status
Tom	Voice	Outgoing	Active
Merrill Lynch	Portfolio Update	Outgoing	Hold
555-2000		Incoming	Ringing

HOLD CALL RESUME CALL MODIFY CALL ACCEPT CALL CLEAR CALL MORE INFO?

Figure 6. Window for calls in progress.

33

editor program. The connector distributes both the data and control outputs from the editor to both workstation displays, and funnels commands from both workstations to the editor. This example illustrates that control and data information are similarly relayed.

User Programmability of Multimedia Workstations. A terminal on such an FPT network theoretically could be engaged in an unconstrained number of multimedia calls; and any given call might use many information streams of voice, data, and image. Whether users can take full advantage of these new capabilities depends on:

- How well they can manage the multiple calls and information streams
- How conveniently multimedia teleconference applications can be constructed.

We implemented a window-based user interface with pull-up menus to manage calls and a nonprocedural language to develop multimedia applications. The user interface consists of a set of windows. Each window displays certain

types of events or data the user may need to manage, such as calls in progress. A window is associated with a menu listing the operations that can be applied to items in the database. This interface implements a paradigm by which the user invokes an application in two steps:

1. Select an object from a window
2. Push a function key corresponding to a particular operation in the menu associated with the window.

For example, if the window is for calls in progress and the user points to a particular call and pushes the "hold" function key, the call will be placed on hold (see Figure 6).

The user can also use a nonprocedural language to write new multimedia applications. Hansen et al. discuss some advantages of using nonprocedural languages for call processing applications.⁸ Our new multimedia nonprocedural language supports the *connector* and *active device* abstractions. It is discussed by Baumgartner, et al. in greater detail.⁹

Outlook

In our experimental system, we have implemented the beginnings of a network operating system that allows control of some new forms of communication, such as multimedia multicasting communication. One objective has been to explore the demands of new types of network communication services on the elements of the future networks. We have been guided by the vision of exploiting the synergy between computers and communication technology. Our experience has shown that fast packet-switching technology provides an excellent communication technology base for work of this kind, and that the operating system technology from computer science provides a suitable framework for guiding network evolution.

There is no dearth of forward-looking concepts, several arising from different agendas. For example, the communication industry is working on the introduction of the *intelligent network*. Another recent concept is the so-called *open network architecture*. This will extend the competition for network products and services by opening internal network interfaces for connection of devices and services from different parties. When we look at the history of computer operating systems, we can see that the network operating system concept can provide useful guidelines for furthering the network evolution toward more openness and access to new intelligent capabilities.

Conclusion

The network of the future has the potential to offer many new benefits to its users by helping to introduce a wide variety of new services. In this paper, we have emphasized that the orderly introduction of new network services can be greatly aided by following an overall guiding concept, that of the *network operating system*. We developed this concept by extending and generalizing the well-known and proven concept of the computer operating system. The generalization pertains to the types of information handled by adding sound and video to the data handled by current systems. We extended the

geographic coverage from single systems to network-wide, and we observed that the change from the asymmetric nature of human-host communication to the symmetric peer-to-peer communication of human-host-human configurations will influence the evolution of the network operating system concept. We reported on experiments to demonstrate multimedia conferencing and other multimedia network computer services on an experimental testbed that takes advantage of the communication flexibility offered by wideband packet technology.

Acknowledgments

We wish to acknowledge the contributions of our colleagues who designed and implemented the multimedia testbed and the experimental fast packet switch at AT&T Bell Laboratories' Indian Hill facility: T. J. Baumgartner, J. R. Brouk, Y. H. Hwang, E. H. Jones, A. A. Kapauan, J. P. Lackey, L. E. Morgan, M. J. Morgan, M. A. Pashan, D. W. Peterson, K. T. Teraslinna, W. N. Toy, S. C. Tu, A. K. Vaidya, and B. F. Wong.

References

1. G. P. Austin, B. S. Bosik, and C. J. Capece, "The Universal Port Concept," *AT&T Technical Journal*, Vol. 68, No. 2, March/April 1989, pp. 14-22.
2. *Information Processing Systems—Open System Interconnection—Basic Reference Model*, International Organization for Standardization, Publication No. 7498, October 1984.
3. R. E. Staehler, J. J. Mansell, E. Messerli, G. W. R. Luderer, and A. K. Vaidya, "Wideband Packet Technology for Switching Systems," ISS 87, Phoenix, Arizona Institute of Electrical and Electronics Engineers (IEEE), CH2431-5/87/0000-0448, pp. B6.1.1-7.
4. J. J. Degan, G. W. R. Luderer, and A. K. Vaidya, "Fast Packet Technology for Future Switches," *AT&T Technical Journal*, Vol. 68, No. 2, March/April 1989, pp. 36-50.
5. D. Ritchie, K. Thompson, "The UNIX Time-Sharing System," *Bell System Technical Journal*, Vol. 56, No. 2, Part 2, July/August 1978, pp. 1905-29.
6. D. Ritchie, "A Stream Input-Output System," *Bell System Technical Journal*, Vol. 68, No. 8, October 1984, pp. 1897-1910.
7. W. H. Leung, G. W. R. Luderer, M. J. Morgan, P. R. Roberts, and

-
- S. C. Tu, "A Set of Operating System Mechanisms to Support Multimedia Applications," *Proceedings of 1988 International Zurich Seminar on Digital Communications*, March 1988, pp. R4.1-R4.6.
8. T. L. Hansen, W. E. Hyatt, W. H. Leung, and P. M. Zislis, "A Non-procedural Language for Telecommunication Call Processing Applications," *Proceedings of the 1986 IEEE Workshop on Languages for Automation*, 1986, pp.100-106.
9. T. J. Baumgartner, Y. H. Hwang, M. J. Morgan, and W. H. Leung, "The User Programmability of a Multimedia Workstation for Fast Packet Networks," *Proceedings of IEEE Global Telecommunications Conference (Globecom) '88*, November 1988, pp. 1.1.1-1.1.6.

Biographies (continued)

Advanced Switching Networks Department at AT&T Bell Laboratories' Indian Hill Park Laboratory in Naperville, Illinois, where he is responsible for applied research in next-generation networking products. His accomplishments have

included developing an experimental Wideband Packet Switch and a prototype fast-packet switch capable of high-speed packetized video switching and broadcasting. He led software development for AT&T's ACCUNET® Packet Service, as well as projects involving the Datakit® circuit switch and the UNIX® operating system. He holds Dipl. Ing. and Dr.-Ing. degrees in Electrical Engineering from the Technical University of Braunschweig, FRG. Mr. Luderer joined Bell Laboratories in 1965.

(Manuscript received February 24, 1989)
