AT&T Bell Laboratories Technical Journal
Vol. 63, No. 8, October 1984
Printed in U.S.A.

A Packet/Circuit Switch

By Z. L. BUDRIKIS* and A. N. NETRAVALI†

(Manuscript received July 29, 1983)

We propose a switch, suitable for an integrated local communications network, that will support packet switching and circuit switching, with a wide range of bit rates. Key components are two serial memories; a multiplicity of access units, each capable of writing and reading uniformly formatted, addressed information; and a programmed controller. Circuit switching is achieved when the controller repeatedly allocates memory slots, following call setup. Data communications can proceed concurrently without setup, competing for unused slots. We give an example of a 10,000-telephone-line switch carrying a similar load of other traffic. The switch would delay voice by less than 5 ms and could be interfaced to the existing telephone system. We indicate a method of fault detection and isolation that will limit the impact of a failure on a serial memory to an arbitrarily small group of connected lines. We define an index for measuring failure impact and use it to derive most-favorable fault-isolating partitions.

I. INTRODUCTION

The telephone system is by far the world's largest communications network. It was primarily designed for voice, but its role widens continuously, as it adapts to new requirements. Presently it is changing to accommodate data communications.

Already the network extensively caters to data communications, but not yet as well as it might. Although internally the telephone system

* AT&T Bell Laboratories. On leave from the Department of Electrical and Electronic Engineering, University of Western Australia, Nedlands. † AT&T Bell Laboratories.

is rapidly becoming a vast interconnected computer system, proffered data are still largely carried as analog signals externally. That will change, however, as special provisions for data come on-line. As more of the plant, including switches, becomes digital, it will be possible to offer, on a selective basis, switched digital telephone channels usable for 56-kb/s data throughput. Also, packet-switched data services will widen in scope and access. Packet-switched data services are overlay networks that use the digital transmission facilities of the telephone network but bypass its switches, of which many are still analog. The packet networks eventually may become totally interconnected, just as the voice network, and also may become integrated with it.

In-house, or proprietary, telephone networks can benefit from the changing character of the overall network more immediately. Already available are switches and other components that permit an all-digital network that will accommodate on one facility both voice and data. As good as this already is, we are proposing a switch that could make the private network even better. Eventually it might even influence the entire system.

Currently available switches provide only circuit-switched connections. This gives fixed-capacity channels on a continuous basis, whereas much of data comes in bursts. Thus, computer communications are characterized by very long call durations with only low average, but in many instances very high, peak rates. Given the option, direct memory transfers could proceed in some instances at rates of many megabits per second. This is far too high for a switched and continuously held circuit.

It is true that the needs of bursty traffic can be catered to by what already is available, namely by some packet-switched networks. But that introduces a separate communications network for data, with the consequences of proliferating wiring plans, divided responsibilities, and probable long-term dyseconomies. It is better for one facility to serve all communications, and to do so without imposing mismatches.

We propose a switch and, more generally, a new switch architecture that support within one switching fabric both circuit- and packet-switched connections. This would largely avoid mismatches in respect to bursty data traffic, while preserving unity in communications.

The cardinal components of the switch (see Fig. 1) are a pair of Serial Memories (SMs), a Central Controller (CC), and Accessing Units (AUs). The memories do not recirculate and both ends (head and tail) of each terminate on the central controller. The AUs are connected to read-and-write taps along the SMs, an AU having one connecting tap to each memory. The two taps of an AU form a symmetrical pair: the tap to the second memory is as many places from the tail end as that to the first is from the head. Thus, each AU

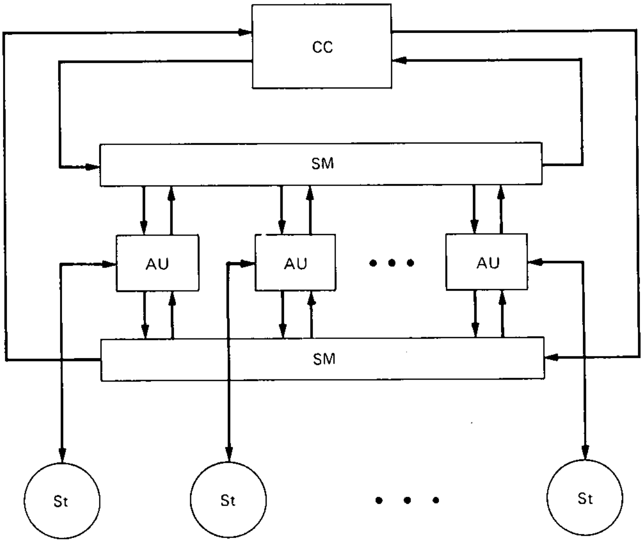


Fig. 1—Block schematic of switch. Access Units (AUs) communicate via two Serial Memories (SMs) on behalf of client Stations (Sts). Central Controller (CC) reserves slots for circuit-switched communications.

can reach every other AU by either one, or the other, memory. It can reach, and be reached by, the central controller by either memory. All writing is logical OR.

An AU acts as an agent of a client station (St) (e.g., telephone, facsimile terminal, computer) and mediates communications between it and other stations by way of corresponding AUs. Communications are carried on by write-and-reads in memory/time slots of uniform length and format. Each slot consists of a data field and several control fields. Collectively, the control fields provide synchronization, "Slot busy" indication, source and destination addressing, and slot pleading. A circuit-switched communication is carried on in regularly recurring slots, which are appropriately premarked by the central controller. For a packet-switched communication an AU simply uses the next available slot.

The capacity of a connected circuit can vary over a wide range, from a small fraction of a single (64-kb/s) telephone channel up to a large multiple of that capacity. It is settled by negotiation with the CC at the time of circuit setup, and need not be the same on different occasions. The capacity available to a packet-switched communication depends on the prevailing competition and can be any portion of the total switch capacity. The latter is a function of size and would be just several megabits per second for a 100-line switch and several hundred megabits per second for a switch that supports 10,000 lines.

A simple realization of the serial memories would be by clocked shift registers. The shift registers can be bit-paralleled to any degree needed to keep the clock rate low. The memories and all access units can be located centrally at the controller, with all connections to the switch then forming a single star. But it is also possible to segment the memories and form the network in clusters. The segments of memory would be connected serially to each other and to the central controller by transmission lines, forming two contrary rings, and the clusters again would form star topologies.

All elements of our proposal are well established and tried. Central, or stored program, control in circuit switching is over twenty years old.[1] The idea of switching by time slot interchanges is even older,[2] followed shortly by its realization through read-and-writes in computer memory.[3,4] Packet switching is more recent,[5] but is also well established both in local and wide-area networks.[6-9]

In essence, our scheme is an adaptation of seemingly diverse procedures, so that they may coexist. Time division slots are enlarged from what is usual in circuit switching, so that they can carry the control information essential to packet switching. Unlike normal packet-switched schemes, packets are of a single fixed length so that they can also be circuit-switched. Instead of separate time and space division stages, common in current telephone switches, we have a combined space/time fabric, abstracted from ring and bus networks, with a particular debt to Fasnet.[9] This makes packet switching possible without controller intervention. Finally, the controller maintains circuit connections by repetitive slot allocations, which is only marginally different from what takes place in a time division stage of a standard switch.

Also, our proposal is not first in its suggestion that voice and data be integrated on a common network.[10-14] But it appears to be first in suggesting a common switch for circuits and packets as the basis for that integration. With few exceptions,[11] prior suggestions have been to treat voice as data and to packet-switch it both in local and wide-area networks. However, these proposals have attendant delays that have to be addressed.

The point is important since, in the global telephone system, transmission delays can limit the quality of many possible connections. In the case of our switch, the delay of voice signals can be kept to less than 5 ms. It depends only on the clock rate, the size of switch, and the size of slots. Since delay considerations have an overriding sway on system choices, we discuss them in Section II. In Section III we give further details of our proposal.

In Section IV we address the question of reliability in our switch. We do this because our proposal may be seen as being particularly vulnerable, since all its communications are to take place via two serial memories to which all AUs have writing privileges. We introduce a scheme for sectional detection and isolation of faults applicable to our switch. We show that this would limit the impacts of faults in our case to those that would prevail in switches that have much more dispersed and/or redundant architectures.

II. DELAY CONSTRAINTS AND RATE REQUIREMENTS

A communication system is expected to deliver messages to the destination in a timely fashion. The permitted delay is different in character for data and for real-time signals. We review the two cases separately.

2.1 Data transmission

Within limits, the exact times of arrival at the destination of the different parts in a data stream generally are unimportant. It is usually required that the sequence in the stream be preserved and that the average delay does not exceed some specified value. When data are presented for transmission at a fluctuating rate and there is not sufficient transmission capacity to cope with the peaks, the flow is smoothed by buffering. Waiting times in buffer stores are the predominant cause of delay.[15]

2.2 Real-time signals

In the transmission of real-time signals, the delay should be a constant and not greater than a specified value. Given fluctuation in transmission rate, there will be a time-varying delay $W_s(t)$ in a buffer at the sending end. A further delay $W_r(t)$ must be deliberately introduced in a buffer at the receiver,[16] so that the total delay could stay constant:

$$W_s + W_r = D. \tag{1}$$

D is the fixed buffer delay with which the system has been designed.

If, at some time, W_s exceeds D, then, at the same time, the buffer at the receiver will become empty and there will be a break in the received signal. Hence, there is no point in storing more at the transmitter than the amount of data that represents the total designed delay. If the rate λ of the real-time data is constant, as in Pulse Code Modulation (PCM) voice, then waiting times are directly related to amounts of stored data. The buffer-store capacities, N_s and N_r, that need to be provided at the two ends are equal, given by

$$N_r = N_s = \lambda D. \tag{2}$$

If λ is not constant, then the required capacities are still equal and are found by substituting the maximum value of λ in eq. (2).

It is important to note that, given fluctuations in data and/or transmission rate(s) and buffer stores to smooth them, the relevant delay for real-time signals is the maximum, i.e., designed, value, not the statistical average. How much larger that designed delay is to be than the average depends on the actual fluctuations in rate(s) and the relative tolerance to lost quality by signal discontinuities and by delay.

III. DESCRIPTION OF SWITCH

We now detail several aspects of the proposed switch. We give an indication of architectural options, describe protocols, and suggest suitable parameters for a 10,000-line switch.

3.1 Architecture

The basic configuration of the switch was shown in Fig. 1 and outlined in the Introduction. The functions of the AUs and the CC will be defined in more detail when we discuss protocols in the next subsection. It will be seen that there are considerable differences in the tasks of an AU that is mediating a circuit-switched, as compared to packet-switched, communication. Further differences in speed and buffer requirements may be identified between, and within, those two categories.

Clearly, there is a choice between designing a number of special-purpose AUs and designing a single universal AU. Further choices concern sharing of, and multitasking by, access units. Should AUs be placed in a common pool and shared by a larger group of stations? That would entail further switching outside the main switch to mediate connections between AUs and stations. Should an AU be multitasked, serving simultaneously different stations? That would make the AU a more complex device. Figure 2 illustrates a switch that incorporates both sharing and multitasking.

Our inclination is towards universal AUs, one to each station, and towards neither sharing nor multitasking. True, this calls for the largest number of AUs, and not the least complex, at that. But it has the advantage of uniformity and, in the light of technology trends, of likely overall economy.

The next choice concerns the serial memories. They may be active, made in semiconductor, or also, reverting to earlier technologies, passive, e.g., acoustic or electromagnetic delay lines. Passive components are attractive because they promise more reliability. However, our purpose would be better served by clocked shift register memories in an arrangement as, say, shown in Fig. 3. This makes for easier synchronization and permits bit-paralleling to hold down clock rates.

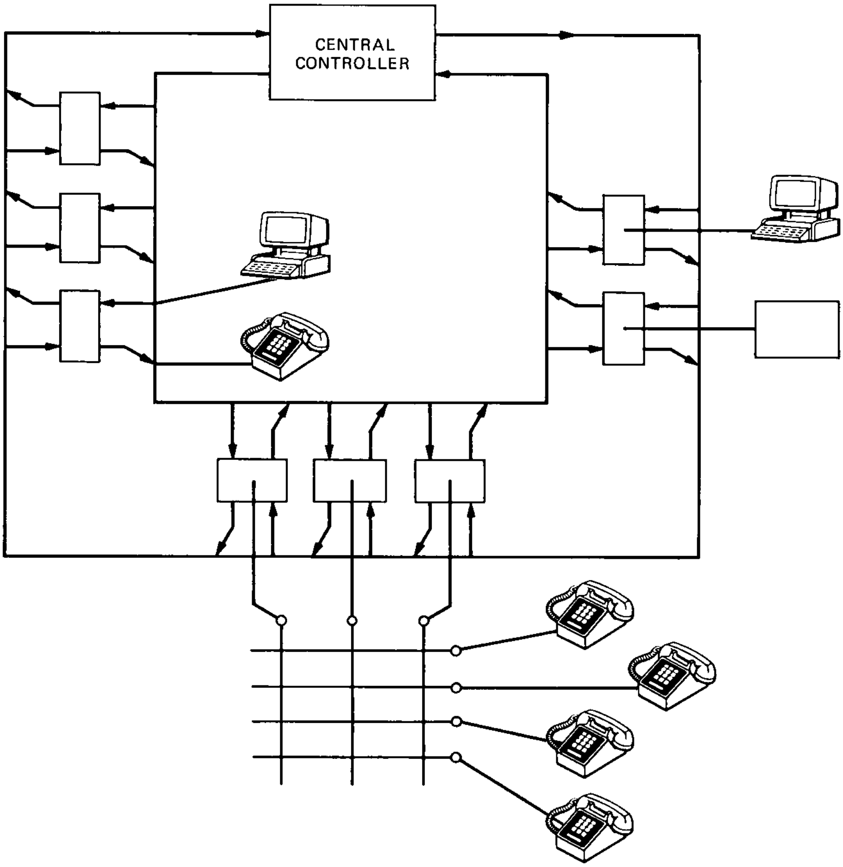


Fig. 2—Different options in AU tasking. All AUs could be of one type, each serving a single station (right); AUs could be shared by a larger group of stations requiring selector switching outside the main switch (center); or an AU could be multitasked, serving more than one type of station (left).

Reliability is a matter of overall design and implementation. In Section IV we discuss an architecture-related aspect of reliability, namely isolation of faults to limited sections.

Finally, we have the question of overall network topology. Three different arrangements are shown in Fig. 4. Figure 4a shows the traditional topology of a central switch and star network. In Fig. 4b a completely distributed arrangement is shown in which the serial memories wend their way past every station. This would make it similar to a local area ring network and would be possible only with passive lines as the memories. A compromise between the above two, and an interesting topology for a PBX that has to serve an extended area, is shown in Fig. 4c. The serial memories are cut into sections, and each section is placed close to the group of stations that it serves.

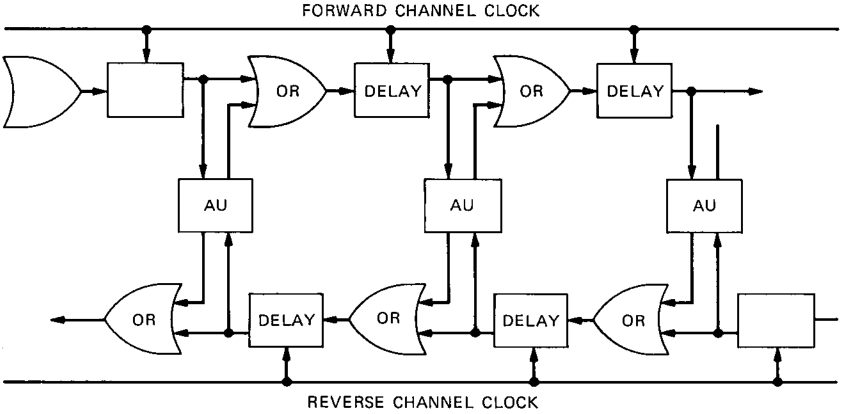


Fig. 3—Shift register realization of serial memories. Access points are at the inputs of clocked unit delays; the writing is through OR gates.

The lines connecting the individual sections and the central controller could be optic fibers, which would carry the total information streams serially even in a large switch.

3.2 Protocols

In the context of our proposal, a packet used in packet-switched communication is made up of five control fields and data, as shown in Fig. 5. The same format could be used in circuit-switched packets. But for these, at least one of the two address fields is unnecessary. Its space may either be added to the data field, or it could be used as a separate channel, a companion to the main channel.

The six fields marked in Fig. 5 are:

1. BUSY—a single bit to indicate slot occupancy
2. RQST—a single bit, common channel used for slot pleading
3. SNDR—address or password of AU sending packet
4. RCVR—address or password of AU intended to receive packet
5. DATA—data field
6. SYNC—synchronization field.

The roles of all the fields, except RQST and SYNC, are self-evident. RQST is used by packet-switching AUs and we will see its function presently when we discuss data communications. The SYNC field is written by the central controller to ensure slot and frame synchronization. Although both synchronizations could be achieved with just one bit per slot, a field of two bits will make them more secure. Altogether, the following numbers would be of the right order: BUSY and RQST one bit each, SYNC two bits, the addresses 14 bits each, and DATA 192 bits, for a total packet of 224 bits, or 28 bytes.

Corresponding to a packet, one may think of a time slot and of a

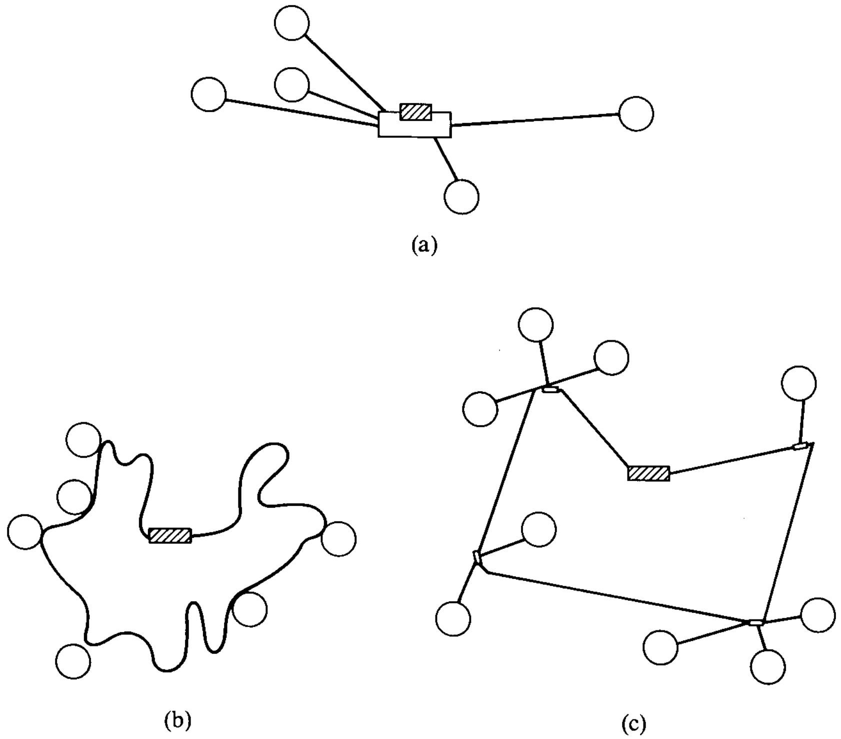
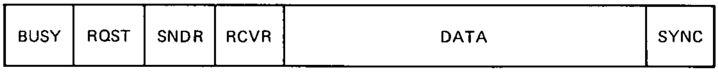


Fig. 4—Network topology options: (a) central switch, stations connected by lines; (b) switch completely distributed with AUs at individual stations and connected by the serial memories realized as buses; (c) switch distributed in clusters, the seral memories within clusters realized by shift registers and between clusters by transmission lines.

propagated memory block as consisting of the same number of bits and divided among respective fields. Note, however, that it is not necessary that the different parts of a packet be placed into a single slot or block. There may be interleaving of packet parts to any extent that is desirable.

Thus, it is conceivable that in order to alleviate the pressure of time for the signaling from receiver to transmitter within the AU, the



BUSY – SLOT BUSY FIELD
RQST – SLOT REQUEST FIELD
SNDR – SENDER ADDRESS
RCVR – RECEIVER ADDRESS
DATA – DATA FIELD
SYNC – SYNCHRONIZATION FIELD

Fig. 5—Slot format. Typically, BUSY, RQST and SYNC would be one-bit fields, the address fields could be two bytes each and the data field 24 bytes.

BUSY field could refer to the state of occupancy—not of the slot that it is riding in, but of the following slot. Similarly, other fields could be advanced or retarded, and not necessarily by only one slot, nor, indeed, just as a complete field. Thus, the DATA field could be broken into single bytes or even bits, and the fragments made to follow the header as an arbitrarily dispersed tail, provided only that all packets are fragmented and dispersed identically.

The extreme fragmentation of packets, as just alluded to, may seem an attractive way of restoring smoothness to data flow for circuit-switched communications. Indeed, almost complete smoothness is possible for any one chosen rate. But it would be at the expense of considerable complication for all other communications, particularly the packet-switched and the circuit-switched that have higher rates than the one singled out for favorable treatment. We will dismiss it from further consideration and turn to describing procedures.

3.2.1 Data communication

Assume that AU addresses are in numerical order along the two memories, ascending in the direction of propagation along one and descending along the other. We will call the memory with ascending addresses the forward channel, and hence the other the reverse channel.

Suppose that an AU has to communicate to another AU of higher address. It must send a message, or packet(s), on the forward channel. To do so, the dispatch processor of the AU will follow the data dispatch routine of Fig. 6. This can be understood more easily with the help of the state diagram of Fig. 7. For the sake of description, this diagram relates to an exclusive forward channel dispatcher, although in practice a single dispatcher would service both directions.

When idle, the dispatcher is normally in the "Go" state and monitors the sending buffer (for the forward channel), checking whether it contains a packet for transmission. If it does, it reads the BUSY field of the next block on the forward channel and at the same time writes a "ONE" in that field so as to seize the slot, should it be available. If it is not, i.e., BUSY was already "ONE," then it will write "ONE" in the next RQST field on the reverse channel and wait for the next BUSY field on the forward channel. It will repeat reading and writing of BUSY on the forward channel and sending RQSTs on the reverse channel until a "ZERO" BUSY occurs. It will then write in the related SNDR, RCVR, and DATA fields, so dispatching a packet.

Having sent a packet, the dispatcher moves to the 'One packet sent' state. If the sending buffer has at that moment one or more further packets for dispatch, then the dispatcher will behave exactly as in the "Go" state and send off the next packet, thereby moving to the "Two

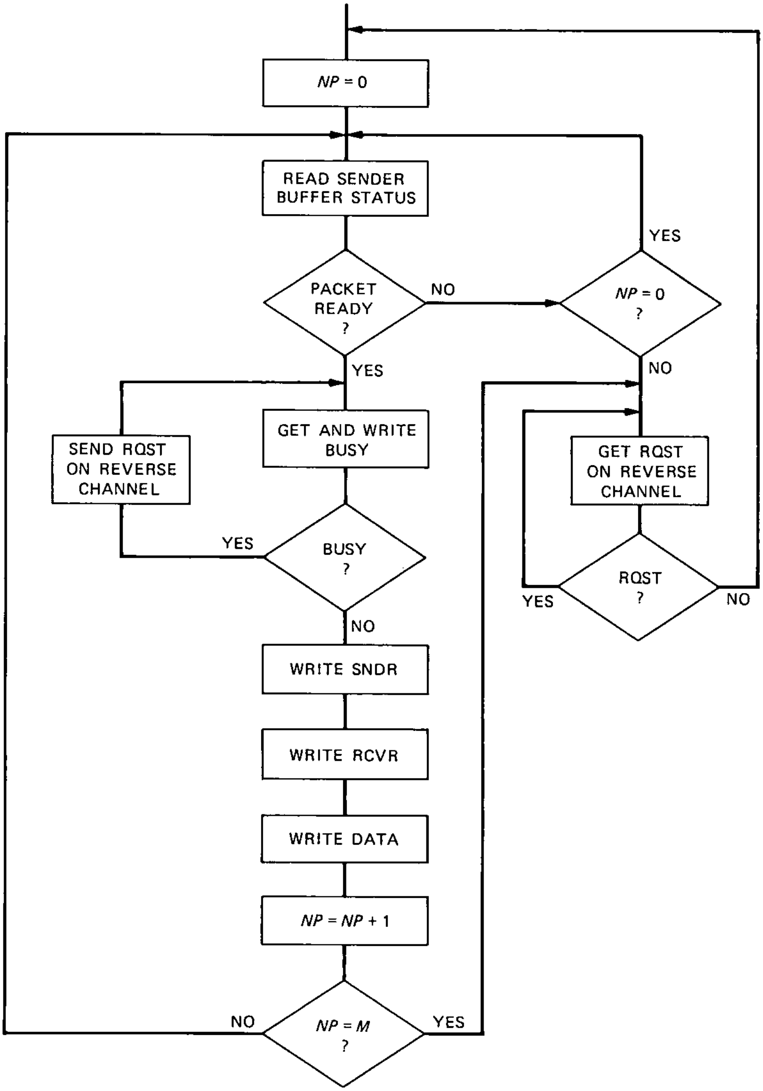


Fig. 6—Flowchart of forward channel data dispatch routine.

packets sent" state. But if there is no packet in the sending buffer on entry to the "One packet sent" state, then the dispatcher will proceed to the "Halt" state. It will remain there until the next "ZERO" is written in the RQST fields on the reverse channel, whereupon it will revert to the "Go" state. Similar conditions apply on entry to the "Two packets sent" and further states, until the dispatcher has sent in a contiguous sequence *M* packets and entered the "*M* packets sent" state. From this it must proceed unconditionally to "Halt."

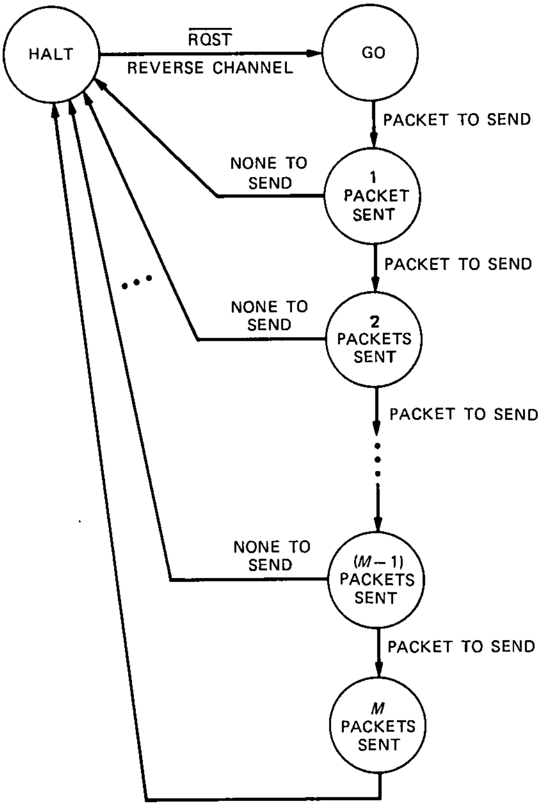


Fig. 7—State diagram of data dispatcher. The dispatcher goes temporarily into HALT state whenever it has no more packets to send or has already sent M packets since the last HALT. It goes from HALT to GO as soon as the RQST bit on the reverse channel is ZERO.

M is a parameter that may vary with AU. It represents priority standing: The larger its value, the less sensitive the AU is to pleadings for slots by other AUs that are downstream from it. It is normally set in relation to the rate of the station that the AU serves.

The task of receiving is less involved but no less time consuming, and an AU will have a separate processor for it. A routine that it could follow is given in Fig. 8. This is set out on the assumption that the SNDR and RCVR fields of a packet would precede the DATA field by one slot.

3.2.2 Real-time signal transmission

An AU serving a real-time device has to act in two distinct modes, one in setting up or tearing down a circuit and the other in transmitting and receiving the real-time signals when the circuit is set up. We

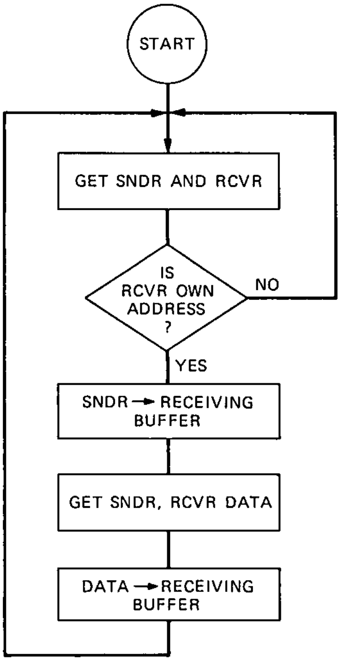


Fig. 8—Flowchart of data receiver routine.

outline the procedure, limiting our attention to telephony. Other devices requiring circuit connections would be served similarly.

Looked at from the telphone, the AU would appear as the line selector of the standard switch. When the telephone is taken off hook, the AU would supply dial tone. As the number is dialed, it would be stored by the AU, which, on completion, would assemble a packet for transmission to the CC. The DATA field of that packet would disclose the fact that a telephone link is being sought, and the numbers of the calling and called stations. The sending procedure for the packet could follow the routine of Fig. 6, even though a simpler routine is possible since no "Halt" state is necessary.

The CC would process received requests using a routine that could be as in Fig. 9. First, the CC would check the total switch capacity already committed to circuit traffic, and from this it would decide whether the setting up of the further circuit is permitted. If it is not permitted, then the CC would inform the originating AU, and that would terminate the processing. If setting up the circuit is permitted, then the CC would determine which AU serves the called station and check whether it is engaged. If it is engaged, then the CC would inform the originating AU accordingly. If it is not engaged, then the CC would tag both AUs as engaged and send messages to both AUs and inform

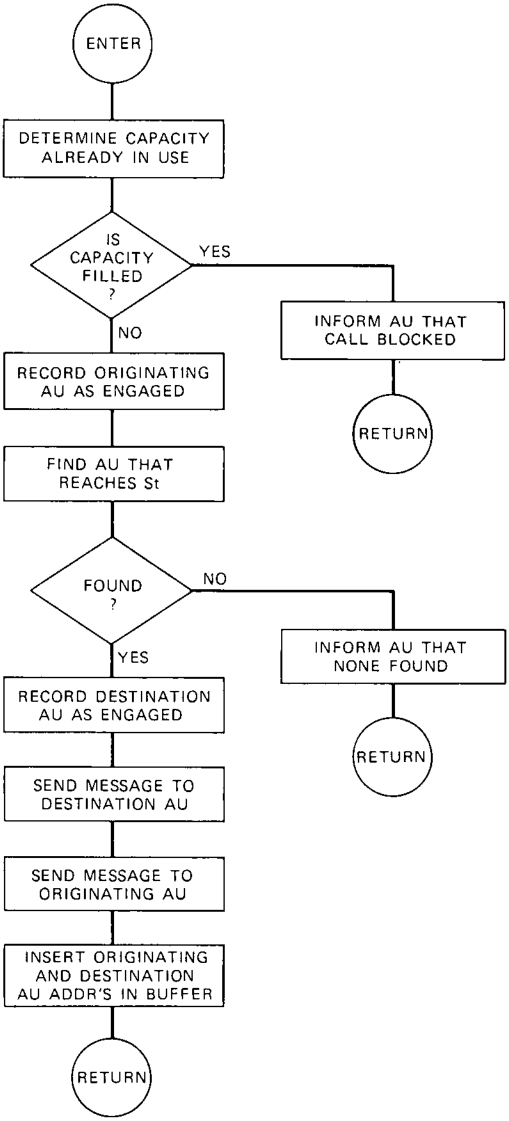

ENTER
DETERMINE CAPACITY ALREADY IN USE
IS CAPACITY FILLED ?
YES
NO
INFORM AU THAT CALL BLOCKED
RETURN
RECORD ORIGINATING AU AS ENGAGED
FIND AU THAT REACHES St
FOUND ?
NO
YES
INFORM AU THAT NONE FOUND
RETURN
RECORD DESTINATION AU AS ENGAGED
SEND MESSAGE TO DESTINATION AU
SEND MESSAGE TO ORIGINATING AU
INSERT ORIGINATING AND DESTINATION AU ADDR'S IN BUFFER
RETURN

Fig. 9—Flowchart of circuit setup routine in central controller.

them of each other's addresses. The two addresses would also be inserted at appropriate places in ring buffers to cause the necessary premarking of slots by writing of BUSY and SNDR on the correct channels at the right frequencies. This would complete the setting up of the two-way circuit. Given a setup circuit, the dispatch and reception of the real-time data would follow the routines of Fig. 10.

Note that only the SNDR address is used in circuit-switched trans-

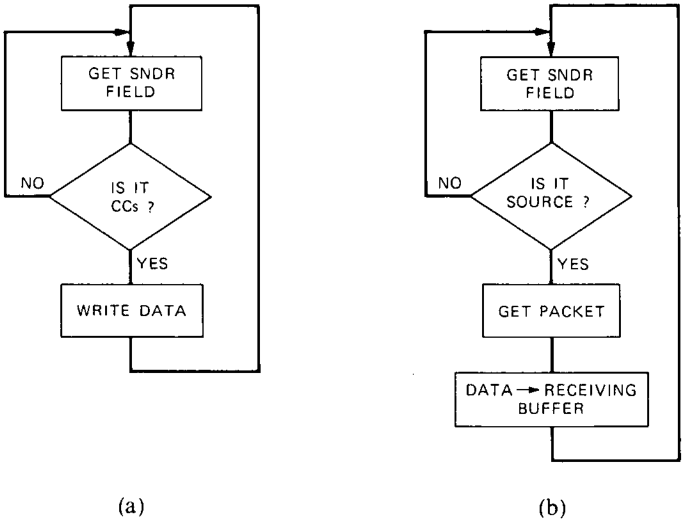


Fig. 10—Flowchart of (a) real-time signal dispatch, and (b) real-time signal reception.

missions: The receiver is given the sender's address and recognizes it for the duration of the call. Apart from saving one address field for other use, there is a further bonus in that more than one receiver can be given the same SNDR address and simultaneously receive the same real-time signal. This leads to the possibility of a simple arrangement for broadcasting to designated outlets for, say, a public address system. If, furthermore, the AU's receiver capability was enlarged to noting several SNDR addresses and taking in packets with those markings, then a telephone conference facility, with voice signal summation at each receiver, would be possible.

The setting up, tearing down, and maintaining of calls to subscribers outside the switch's own area would have to interwork with equipment in other offices. But there is no particular problem about this. The AU serving a trunk would interface with the outside system, sending and responding to signals in conformity with existing specifications. But, in other respects, it would not be different from an AU serving a local subscriber. Data out of, and into, the local switch area could also be carried by circuit-switched trunks, with suitable interfacing to a wider-area data network. The role of an AU providing that interfacing would then amount to that of a gateway processor.

3.3 Packet size and clock rates

The bit rate required along the SMs is related to the total peak load for which the system is designed, multiplied by a factor that accounts for efficiency. Assuming a telephone voice signal sampled at 8 kHz, represented by 8 bits per sample and an allowed delay due to packetization of 3 ms, a packet may contain 24 samples or 192 bits of data. The overheads are mainly in the addresses: Assuming a 10,000-voice-line switch and a total number of AUs not exceeding 16K, the SNDR and RCVR fields could be 14 bits each. As we already noted, BUSY and RQST need be only 1 bit each, and SYNC 2 bits. The total overhead will then be 32 bits and the packet length will be 224 bits.

Another criterion by which the overall size of a packet can be decided is efficiency. Since the allowable delay for voice is binding, the best size indicated for maximum efficiency will be of interest only if it is smaller than that already decided.

It is a reasonable simplification to suppose that all offered traffic divides into two categories: very short bursts, and prolonged streams. Furthermore, it is reasonable to assume that the number of packets-per-second from the very short burst will be independent of packet size. Thus, such very short bursts would be produced by single ASCII characters from, and echoed to, computer terminals, when carried in individual packets. On the other hand, circuit-switched traffic and data file transfers are examples of streamed flow.

Consider the total bit rate, R, that results from traffic consisting of b, short bursts per second, and an aggregated stream flow of S bits per second. If the packet has h bits of header and x bits of DATA, then

$$R = [b(h + x)] + [(S/x)(h + x)]. \qquad (3)$$

This will be a minimum when

$$x = \sqrt{(S \cdot h)/b}. \qquad (4)$$

In a system serving a business, one might provide for a busy-hour voice traffic of 10 ccs (hundred call seconds) per telephone. In the switch, this will divide equally between the two memories. With 10,000 telephones, the aggregate stream S_v on each SM due to voice would then be

$$S_v = 10{,}000 \cdot 5 \cdot 64{,}000/36 = 89 \text{ Mb/s}.$$

A reasonable assumption for the present is that all other traffic would amount to 20 percent of the total, or in our example it would be a further 22 Mb/s.

For the sake of illustration, assume that the very short burst rate, b, is 20,000 packets per second. If each of these carries only one 8-bit byte, then the net traffic from them is 160 kb/s, a negligible amount within the assumed 22 M/bs. But the gross traffic may be much larger, depending on packet size. Hence, the decision for best size of DATA field, which, with the numbers already invoked, follows from eq. (4):

$$x \quad = \sqrt{(111{,}000{,}000 \cdot 32)/20{,}000} = 421 \text{ bits}.$$

For x_{opt} to be less than 192 bits, decided by delay considerations, the very short burst traffic would have to be 4.8 times larger than was assumed. But the assumed rate is already large, and therefore it is unlikely that efficiency considerations would indicate a smaller packet than given from delay.

Given packets of 224 bits and the numbers cited above, the rate, R, in each memory follows from eq. (3) as 134 Mb/s. If 8-bit bytes are propagated in parallel, then the required clock rate is 16.75 MHz, a none too demanding frequency for present technology. The packet rate, which is of greater relevance to AU and CC speeds, would be 598 kHz.

A switch would be designed for a given ultimate size and given an appropriate clock rate from the start. But it would not be necessary to give it immediately the full complement of AUs, nor, indeed, full lengths of memories. AUs could be added without any disruption and memory sections with only a minor pause.

IV. RELIABILITY

Availability of communications services is extremely important and has prompted switch designers to adopt the very highest standards of reliability.[18,19] Thus, it is accepted practice to have two identical central controllers, one being a "hot" standby that can take over at any instant. This and other common practices would also apply to our switch. The features by which our switch is rendered most vulnerable in respect to reliability are its serial memories, which carry all messages and are accessed by all AUs. Below we consider the general question of disruptive impact by failures and suggest a measure for it. Then we introduce a fault detection and isolation scheme that would make the robustness of switching by serial memories with multiple read-and-write taps comparable to that of much more redundant architectures.

4.1 Failure impact

In switching equipment, including ours, failures are unequal in likelihood and in disruptive consequence. We introduce the notion of expected failure impact. Let π_{ik} be the probability that component C_i will fail during the course of one year; let the expected repair time for it be τ_{ik}; and the number of potential communication connections that are unavailable while C_i is in the failed state be ν_{ik}. We define U_i, the expected per annum failure impact (EPAFI) of C_i, as

$$U_i = \sum_k \pi_{ik} \cdot \tau_{ik} \cdot \nu_{ik}. \tag{5}$$

We assume that failures are statistically independent and disregard the probability of another component failing during the repair time of an existing failure. EPAFI values then are additive, and U, with respect to an assembly of N components, is

$$U = \sum_{i=1}^{n} U_i. \tag{6}$$

We consider the expected failure of the SMs and all the AUs connected to them. Suppose that there are altogether N AUs, each with (1) a per annum rate π_1 of failing in a way that affects only one subscriber and takes time τ_1 to repair, and (2) a rate π_2, which disrupts communications on the memory past the failed AU and takes τ_2 to repair. Also, let the memories have a rate π_3 of failing at each of the $2N$ connecting points, with expected repair times τ_3.

With N mutually communicating AUs in the system, the number of potential two-way communication links is $N(N - 1)/2$. If an AU failure of the first kind occurs, then $\nu_1 = (N - 1)$ of these are disrupted. When the failure of the AU is of the second kind, or when a memory fails, then the number of disrupted links is much larger and depends on the actual location of the failure. One can calculate an average number on the assumption that all potential failure locations are equally likely. It is found to be approximately $(N^2)/3$. Hence the total expected impact due to failures of AUs and memory links is

$$U = [N\cdot(N - 1)\cdot\mu_1] + [(N/3)\cdot N^2\cdot\mu_2] + [2\cdot(N/3)\cdot N^2\cdot\mu_3], \tag{7}$$

where $\pi_i\nu_i$ has been contracted to μ_i.

4.2 Failure detection and isolation

We propose to divide the memories into sections and have a fault detector at the end of each section. Further, each section would have a bypass and, in case of a detected failure, a switch would be actuated to pass on to the next section the data stream at the output of the bypass (Fig. 11). Thus, effectively, the consequence of the fault is isolated to one section. A possible realization of the switch is shown in Fig. 12.

A Fault Detector (FD) would compare the data streams at the outputs of the memory section and the bypass, and it would decide that failure has occurred when the evident modification to the stream in passage through the memory section violates existing constraints. The particular constraint of the several that exist in our case and which we use is the following: There may never be a change of any field that is already nonzero. Detecting any such illegal changes will catch failures both in AUs and the memories. The detection will, of course, rely on the output from the bypass being a flawless replica of what entered that section. If necessary, redundancies and error control could be implemented on the bypasses to make that more sure.

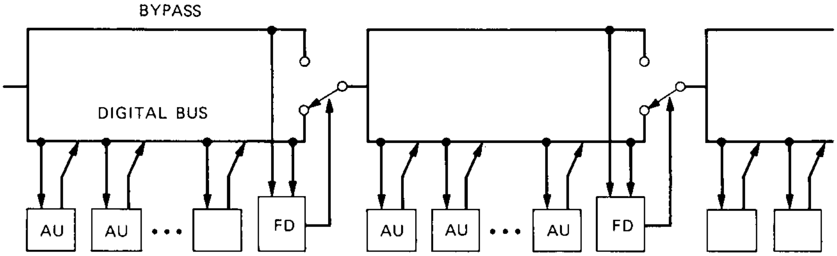


Fig. 11—Fault isolation using bypasses. Whenever the Fault Detector (FD) detects constraint violations, it switches input to next section to output of the bypass from the previous section.

With the memories divided into sections and with fault detection and isolation in place, failure impacts will be reduced. Unless the fault is in the fault detector itself, then if each memory is divided into m sections, only the N/m stations within a section will be affected by a fault. The disruption depends on which section and which point within the section is involved. Again assuming equal likelihoods for locations, we can derive the average number of potential connections that are disrupted and find this, to a good approximation, amounting to 0.75 (N^2/m).

Suppose that fault detectors have their own failure rate π_4 for failures that produce an open line and π_5 for switching off a section when it should not be. Further, suppose that these have expected repair times τ_4 and τ_5, or $\mu_4 = \pi_4\tau_4$ and $\mu_5 = \pi_5\tau_5$. On average, these events disrupt, respectively, $N^2/3$ and $N^2/2m$ potential connections.

The total EPAFI value, with division into m sections and fault detection and isolation, is then

$$U_m = [N\cdot(N-1)\cdot\mu_1] + [0.75\cdot N^3\cdot\mu_2/m] + [1.5 \cdot N^3\cdot\mu_3/m] + [2\cdot m\cdot N^2\cdot\mu_4/3] + [N^2\cdot\mu_5]. \quad (8)$$

The first and last terms in eq. (8) are much smaller than the others and may be neglected. The value of m that results in minimum U_m is

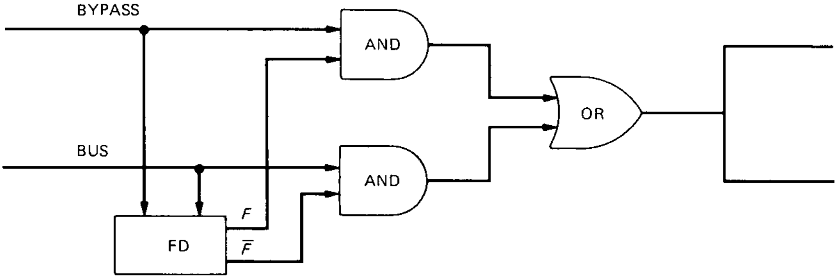


Fig. 12—Realization of two-way switch for fault isolation. FD outputs F and $\bar{\text{F}}$ are complementary, signifying FAULT and NO FAULT.

$$m = (3/2)\cdot\sqrt{N\cdot(\mu_2 + 2\cdot\mu_3)/2\cdot\mu_4}, \tag{9}$$

and the failure impact, with the first and last terms in eq. (8) neglected, comes to

$$U_{\text{opt}} = (4/3)\cdot N^{5/2}\cdot\sqrt{\mu_4\cdot(\mu_2 + 2\cdot\mu_3)/2}. \tag{10}$$

This should be compared with the expected failure impact without sectional detection and isolation, as found in eq. (7). The improvement ratio is

$$U/U_{\text{opt}} = (1/3)\cdot\sqrt{N\cdot(\mu_2 + 2\cdot\mu_3)/2\cdot\mu_4}. \tag{11}$$

Further improvement is possible by instituting super sections by which a number of consecutive sections would be bypassed and again fault-tested and isolated, as shown in Fig. 13. Indeed, one can take the hierarchy of protection to any number of levels.

With just one level of protection and, say, $N = 10{,}000$, and the different failure rates and repair times comparable to each other, the optimum number of sections would be around 185, or 55 AUs to one section. The improvement over no protection would be by a factor of 40. A second level of protection would increase the improvement by a further factor of around 5. Asymptotically, as the hierarchy of protection is taken to higher levels, the functional dependence of EPAFI on N becomes quadratic, which is the relationship that applies when the effect of a failure is confined to a single AU.

V. CONCLUSIONS

We have proposed a switch architecture that supports circuit- and packet-switched communications. Both kinds of communications can proceed at widely varying rates: Circuits can be set up with different capacities, selectable as a binary fraction or multiple of a basic capacity, while packet-switched communications share in the pool of the total switch capacity that is not in use at any given time. Thus, the proposed switch could cater efficiently in mediating real-time signals

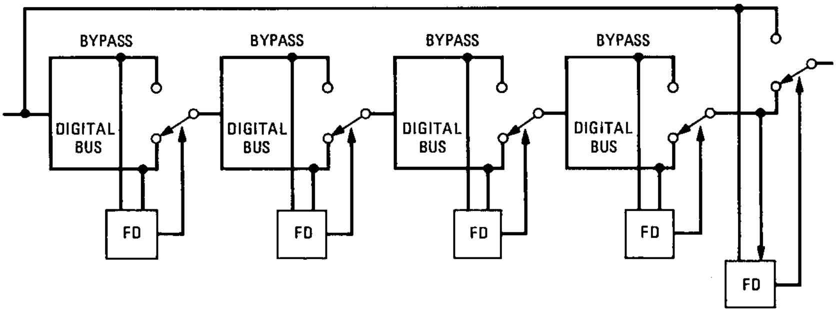


Fig. 13—Fault detection and isolation in super sections.

and data. Specifically, it could be a PBX that, apart from voice, could provide other circuit- and packet-switched services.

The possibility for the two modes is brought about by having enlarged time slots that can include addressing information, and then by making these of fixed length so that they can be made available regularly. Further, the switching is performed by access units that write and read on serial memories on which synchronization can be maintained without interruptions and information transfers can occur without collisions.

We have proposed that data packets be fixed at 192 bits, or 24 samples of pulse-code-modulated voice. This limits the delay due to packetization to 3 ms. The total delay, which includes propagation along the memories, will then be less than 4 ms, even in a very large switch.

It is recognized that a switch used in telephony should conform to very high standards of reliability. We have proposed a scheme of fault detection and isolation applicable to memories as in our switch. This would substantially overcome any added vulnerability due to the serial nature of the signal paths. However, other issues (e.g., overall system reliability) not addressed by us remain to be resolved. In summary, our proposed switch offers the possibility of integrating voice and data in a way that would preserve the quality and reliability of voice communications and therefore, in turn, could be integrated with the telephone system at large. We believe that, provided no compromises need to be made, very real benefits flow from having all communications mediated by a common facility. It is possible that our proposed switch could meet such objectives.

VI. ACKNOWLEDGMENT

The authors would like to thank M. G. Hluchyj for helpful discussions during early phases of this work.

REFERENCES

1. J. Bellamy, *Digital Telephony*, New York: Wiley, 1982.
2. H. E. Vaughan, "Research Model for Time-Separation Integrated Communication," B.S.T.J., *38* (July 1959), pp. 909–32.
3. H. Inose et al., "A Time-Slot Interchange System in Time-Division Electronic Exchanges," IEEE Trans. Commun. Syst., *CS-11* (September 1963), pp. 336–45.
4. Special Issue on No. 4 ESS, B.S.T.J., *56* (September 1977).
5. L. G. Roberts, "The Evolution of Packet Switching," Proc. IEEE, *66* (November 1978), pp. 1307–13.
6. A. G. Fraser, "Datakit—A Modular Network for Synchronous and Asynchronous Traffic," Proc. ICC (June 1979).
7. R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Comm. ACM, *19* (July 1976), pp. 395–404.
8. G. T. Hopkins and P. E. Wagner, "Multiple Access Digital Communications Systems," U.S. Patent 4,210,780, issued July 1, 1980.
9. J. O. Limb and C. Flores, "Description of FASNET—A Unidirectional Local-Area Communications Network," B.S.T.J., *61* (September 1982), pp. 1413–40.

10. T. W. Forgie and A. G. Nemeth, "An Efficient Packetized Voice/Data Network Using Statistical Flow Control," IEEE Commun. Conf. III, 1977, pp. 38.2.44–48.
11. N. F. Maxemchuck, "A Variation on CSMA/CD That Yields Movable TDM Slots in Integrated Voice/Data Local Networks," B.S.T.J., *61* (September 1982), pp. 1527–50.
12. G. J. Coviello et al., "System Design Implications of Packetized Voice," IEEE Commun. Conf. III, 1977, pp. 38.3.48–53.
13. T. Bially et al., "Voice Communication in Integrated Digital Voice and Data Networks," IEEE Trans. Commun., *COM-28* (September 1980), pp. 1478–90.
14. D. H. Johnson and G. C. O'Leary, "A Local Access Network for Packetized Digital Voice Communication," IEEE Trans. Commun., *COM-29* (May 1981), pp. 679–88.
15. L. Kleinrock, *Queuing Systems, Vol. 1: Theory*, New York: Wiley-Interscience, 1975, *Vol 2: Computer Applications*, New York: Wiley-Interscience, 1976.
16. Z. L. Budrikis, J. L. Hullet, and D. Q. Phiet, "Transient-Mode Buffer Stores for Nonuniform Code TV," IEEE Trans. Commun. Technol., *COM-19* (December 1971), pp. 913–22.
17. E. T. Klemmer, "Subjective Evaluation of Transmission Delay in Telephone Conversations," B.S.T.J., *46* (July–August 1967), pp. 1141–7.
18. R. W. Downing, J. S. Nowak, and L. S. Tuomenoksa, "No. 1 ESS: Maintenance Plan," B.S.T.J., *43* (September 1964), pp. 1961–2019.
19. W. P. Karas, "Reliability and Maintainability Improvements Through Distributed Controls in Communication Systems," NTC Record 1981, pp A4.4.1–7.

AUTHORS

Zigmantas L. Budrikis, B.Sc., 1955, and B.E. (Hons I, Electrical Engineering), 1957, University of Sydney; Ph.D., 1970, University of Western Australia; P.M.G. (now Telecom Australia) Research Laboratories, 1958–1960; Aeronautical Research Laboratories, Fishermen's Bend, 1961; Electrical Engineering Faculty at University of Western Australia, 1962—. Mr. Budrikis has had a number of visiting appointments: University of California at Berkeley, 1968; AT&T Bell Laboratories, 1972, 1973, 1981, 1983, 1984; TU Munich, 1977. He is interested in problems in communications, man-machine interfaces, and foundations of electromagnetism. Fellow, IE Australia; member, IEEE, Optical Society of America, New York Academy of Science.

Arun N. Netravali, B. Tech. (Honors), 1967, Indian Institute of Technology, Bombay, India; M.S., 1969, Ph.D. (Electrical Engineering), 1970, Rice University; Optimal Data Corporation, 1970–1972; AT&T Bell Laboratories, 1972—. Mr. Netravali has worked on problems related to filtering, guidance, and control for the space shuttle. At AT&T Bell Laboratories, he has worked on various aspects of digital processing and computing. He was a Visiting Professor in the Department of Electrical Engineering at Rutgers University and the City College. He is presently Director of the Computer Technology Research Laboratory. Mr. Netravali holds over 20 patents and has had more than 60 papers published. He was the recipient of the Donald Fink Prize Award for the best review paper published in the Proceedings of the IEEE and the journal award for the best paper from the SMPTE. Editorial board, Proceedings of the IEEE; Editor, IEEE Transactions on Communications; senior member, IEEE; member, Tau Beta Pi, Sigma Xi.