AD-A097 998 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/2
NETWORK SPEECH SYSTEMS TECHNOLOGY PROGRAM.(U)
SEP 80 C J WEINSTEIN F19628-80-C-0002
UNCLASSIFIED ESD-TR-80-222 NL

1 OF 1
AD
A097998

END
DATE
FILMED
5-81
DTIC

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

NETWORK SPEECH SYSTEMS TECHNOLOGY PROGRAM

ANNUAL REPORT
TO THE
DEFENSE COMMUNICATIONS AGENCY

1 OCTOBER 1979 – 30 SEPTEMBER 1980

ISSUED 13 MARCH 1981

LEXINGTON MASSACHUSETTS

ABSTRACT

This report documents work performed during FY 1980 on the DCA-sponsored Network Speech Systems Technology Program. The areas of work reported are: (1) communication systems studies in Demand-Assignment Multiple Access (DAMA), voice/data integration, and adaptive routing, in support of the evolving Defense Communications System (DCS) and Defense Switched Network (DSN); (2) a satellite/terrestrial integration design study including the functional design of voice and data interfaces to interconnect terrestrial and satellite network subsystems; and (3) voice-conferencing efforts dealing with support of the Secure Voice and Graphics Conferencing (SVGC) Test and Evaluation Program. Progress in definition and planning of experiments for the Experimental Integrated Switched Network (EISN) is detailed separately in an FY 80 Experiment Plan Supplement.

CONTENTS

NETWORK SPEECH SYSTEMS TECHNOLOGY

I. INTRODUCTION AND SUMMARY

This report documents work performed during FY 1980 on the DCA-sponsored Network Speech Systems Technology Program. The areas of work reported are: (1) communication systems studies in support of the evolving Defense Communications System (DCS) and Defense Switched Network (DSN); (2) a satellite/terrestrial network integration design study including the functional design of voice and data interfaces to interconnect terrestrial and satellite network subsystems; and (3) voice-conferencing efforts dealing with support of the Secure Voice and Graphics Conferencing (SVGC) Test and Evaluation Program. The communication systems studies include investigation in satellite Demand-Assignment Multiple Access (DAMA), voice/data integration, and multi-link talkspurt switching which are continuations of efforts reported in previous Annual Reports.[1,2]

New system study areas reported on for the first time this year are adaptive routing, and network design and performance analysis. Progress in planning of experiments for the Experimental Integrated Switched Network (EISN) is reported in a separate 1980 EISN Experiment Plan Supplement. Another separate document, entitled "Advanced Network Technology for the Defense Switched Network," is being submitted to DCA in Project Report form to complement this report. The purpose of that document is to summarize Lincoln efforts in the Network Speech Systems Technology Program over the past several years, and to place these efforts in context of evolving plans for the DSN.

The FY 80 systems studies efforts are reported in Sec. II. In the satellite DAMA area, a new "minimum-loss" channel allocation algorithm is presented which shows some implementational and performance advantages over the "round-robin" algorithm used in previous studies. In the voice/data integration area, a new delay-vs-throughput formula for data-packet traffic in a hybrid (combined circuit and packet) multiplexer has been obtained and verified successfully against previous simulation results. Efforts reported on multi-link talkspurt switching include queueing results and analysis of signaling requirements for a technique introduced in the previous Annual Report[1] for achieving effective Time-Assignment Speech Interpolation (TASI) operation in multi-link hybrid networks. Efforts in adaptive routing include development of a new sequential search algorithm for assigning paths to incoming calls, and initial investigation and simulation of a distributed algorithm for computing routing tables and modifying these tables to adapt to network changes. In network design and performance analysis, initial efforts have been made to assemble a software facility to support satellite/terrestrial routing and network performance studies.

Satellite/terrestrial network integration is the subject of Sec. III. Functional designs and hardware architectures are described for two types of interfaces between the satellite and terrestrial portions of EISN: Interface Applique C to deal with circuit-switched voice traffic, and Interface Applique D for packet-switched data traffic. The functions of these Appliques include signaling translation, participation in the network routing algorithm, and (for Applique C) conversion from circuit-to-packet format and vice versa. Hardware structures for initial implementation of these Appliques include PDP-11 minicomputers, special microprocessor-based

input/output handlers, and (for Applique C) digital channel banks to multiplex and demultiplex a T-1 carrier.

Voice-conferencing efforts during FY 80 are described in Sec. IV. These efforts have been concerned with consulting support for Naval Ocean Systems Center (NOSC), the lead agency for the SVGC Program, and with preparation of test materials for their work. The work was largely carried out by Bolt Beranek and Newman, on subcontract to Lincoln for human-factors conferencing evaluation support.

II. COMMUNICATION SYSTEMS STUDIES

A. SATELLITE DAMA

The previous Annual Report[1] summarized the results of simulations of a satellite DAMA algorithm which will be called here the "simple round-robin algorithm." Its purpose is to use the prediction of speech activity to achieve a network-wide TASI advantage greater than can be achieved at any single node with a small number of speakers.

The simple round robin operates as follows:

(1) Each node predicts, on the basis of its present speech activity, the number of speakers who will be active one satellite round-trip time later.

(2) It pads this prediction by an amount that has been called the "margin" and broadcasts the padded prediction to all the other nodes during a special reservation time slot.

(3) Each node runs the same algorithm, which assigns the time slots for the upcoming frame on a round-robin basis until either the padded predictions are all satisfied or the time slots are all assigned.

For each configuration of users, an optimum margin producing the fewest lost packets was found by repeated simulation.

The simple round-robin allocation algorithm suffers some unnecessary loss of packets because it does not allocate the capacity left over in those frames when all requests have been satisfied. Occasionally, one of the nodes will have requested too-little capacity. Some form of allocation of the spare capacity might save such a node from losing a packet. In addition, the round-robin algorithm concentrates lost packets more heavily at nodes with large channel capacity requests, during frames where not all requests can be satisfied. From the viewpoint of speech perception, it is probably more disruptive to have few speakers suffer large losses than for the losses to be scattered more evenly. Finally, the round-robin algorithm is difficult to analyze. No simple expression has been found for the expected fraction of packets lost, and simulation has been the only alternative.

The "minimum-loss" algorithm, described below, is an attempt to overcome these drawbacks of the round-robin algorithm. In the minimum-loss algorithm, each node reports to the others the number of off-hook callers M and the number of active speakers j. Using M and j in a formula given in Ref. 1, each node can compute $P_{j,k}$, the probability that there will be k active speakers one prediction time from now, given that there are j active speakers now. The formula computation could be implemented by a simple table lookup. If capacity for a speakers is allocated, the expected number of speakers whose packets will be lost is

$$\ell_j(a) = \sum_{k \geq a} (k - a)\, P_{j,k} \quad .$$

The expected system-wide loss L can then be represented in terms of j_i, the present number of active speakers at the i^{th} of N nodes, and a_i, the allocation of capacity to the i^{th} node one prediction time from now:

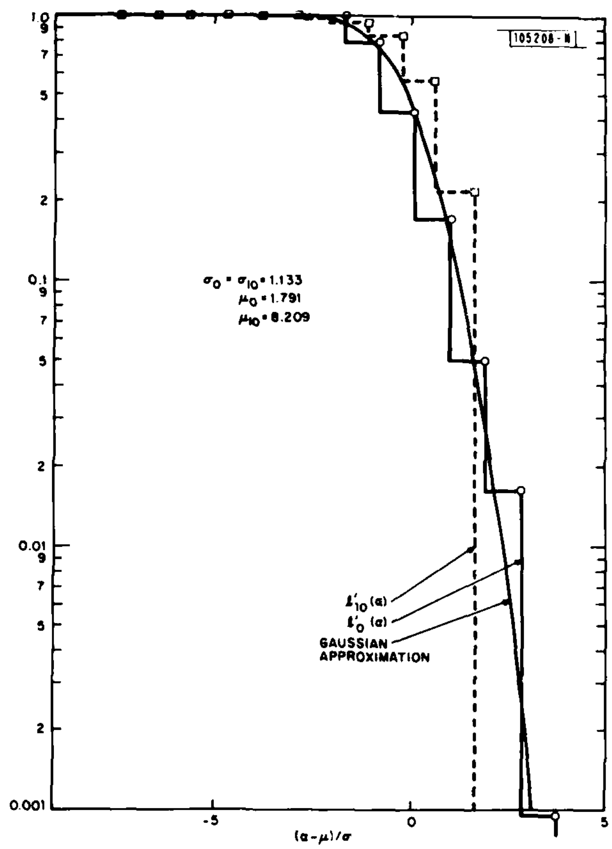


Fig. II-1. Comparison of loss-derivative functions involved in development of minimum-loss algorithm.

$$L = \sum_{i=1}^{N} \ell_{j_i}(a_i) \quad . \qquad \text{(II-1)}$$

In its ideal form, the minimum-loss algorithm chooses the a_i to minimize L, subject to the capacity limitation

$$\sum_{i=1}^{N} a_i = c \quad . \qquad \text{(II-2)}$$

Although the a_i might have to be integers in practice, we can formally consider them to be continuous variables for the purpose of minimization. Later, the appropriate nearby integer can be chosen. Using a Lagrange multiplier b, the minimization conditions become

$$\ell'_{j_i}(a_i) + b = 0 \qquad 1 \leqslant i \leqslant N \quad . \qquad \text{(II-3)}$$

The simultaneous solution of Eqs. (II-2) and (II-3) for the a_i is theoretically intractable and computationally too complex for a node to perform once a frame. However, it becomes very simple under the assumption that $\ell'_j(a)$ can be expressed with sufficient accuracy as $g\,[(a - \mu_j)/\sigma_j]$, where μ_j and σ_j are the mean and standard deviation of the speaker activity one prediction time ahead. Note that the assumption is one of a single function g that works for all values of j. The validity of the assumption is supported by Fig. II-1 which compares, for M = 10, the (magnitudes of the) least-similar functions $\ell'_0(a)$ and $\ell'_{10}(a)$ with each other and with the (magnitude of the) function g that would result from approximating each discrete distribution $P_{j,k}$ by the continuous, Gaussian density function with mean μ_j and standard deviation σ_j. In Fig. II-1, the mean talkspurt and mean silence are each 1.5 s, and the prediction time is 0.25 s.

Under the assumption that the function g exists, one can solve for the a_i without even knowing the form of g or calculating values of $P_{j,k}$.

$$a_i = \mu_{j_i} + \sigma_{j_i} D \quad \text{where } D = \frac{c - \sum_{i=1}^{N} \mu_{j_i}}{\sum_{i=1}^{N} \sigma_{j_i}} \qquad 1 \leqslant i \leqslant N \quad . \qquad \text{(II-4)}$$

The allocation formula (II-4) may be thought of as a two-step process:

(1) Allocate to each node i the capacity equal to its predicted number of speakers, μ_{j_i}.

(2) Distribute the excess capacity

$$c - \sum_{i=1}^{N} \mu_{j_i}$$

among the nodes in proportion to their standard deviations σ_{j_i}. (If

$$c < \sum_{i=1}^{N} \mu_{j_i}$$

then assess the deficit in the same proportion.)

The approximate expression for system-wide loss does depend on the form of the function g.

$$L \approx r(D) \sum_{i=0}^{N} \sigma_{j_i} \qquad \text{(II-5)}$$

where the function r is related to the function g. For the Gaussian approximation,

$$r(D) = \frac{1}{\sqrt{2\pi}} e^{-(D^2/2)} - \frac{D}{\sqrt{2\pi}} \int_D^{\infty} e^{-(y^2/2)} dy \quad . \qquad \text{(II-6)}$$

Values of the r(D) for Gaussian g are given in Table II-1.

TABLE II-1

VALUES OF THE FUNCTION r(D) FOR GAUSSIAN APPROXIMATION TO THE LOSS FUNCTION ℓ'

D	r(D)	D	r(D)
0.0	0.3989	1.6	0.0232
0.1	0.3510	1.7	0.0183
0.2	0.3069	1.8	0.0144
0.3	0.2668	1.9	0.0111
0.4	0.2305	2.0	0.0086
0.5	0.1979	2.1	0.0064
0.6	0.1687	2.2	0.0049
0.7	0.1429	2.3	0.0037
0.8	0.1202	2.4	0.0027
0.9	0.1004	2.5	0.0020
1.0	0.0833	2.6	0.0014
1.1	0.0703	2.7	0.0010
1.2	0.0561	2.8	0.00074
1.3	0.0456	2.9	0.00053
1.4	0.0366	3.0	0.00038
1.5	0.0293		

The system-wide loss L is a function of the N random variables j_i. A measure of the system performance is $\overline{L}$, the expected value of L over the appropriate distrib... ...or the j_i. For comparison with one of last year's simulations of the simpl... ...nd robin, each j_i should have a binomial distribution over M speakers ...ith mean $M\lambda/(\lambda + \mu)$, where $M = 10$, λ^{-1} = average silence = 1.34 s, μ^{-1} ...erage talkspurt = 1.23 s. The following equation gives an accurate approximation to L, which may be obtained by assuming a Gaussian distribution for D:

$$\overline{L} \approx N\overline{\sigma}\sqrt{1 + \sigma_D^2}\; r\left(-\frac{\overline{D}}{\sqrt{1 + \sigma_D^2}}\right) \qquad \text{(II-7)}$$

where $\overline{D}$ and σ_D are, respectively, the mean and standard deviation of the quantity D in Eq. (II-4) and $\overline{\sigma}$ is the rms prediction error at a single node over its joint distribution of speaker activities at the start and end of the prediction interval.

Figure II-2 shows the fraction of lost packets as a function of the number of nodes, each with 10 speakers. The capacity c is 80 speakers, and the predict-ahead interval τ_p is 0.28 s.

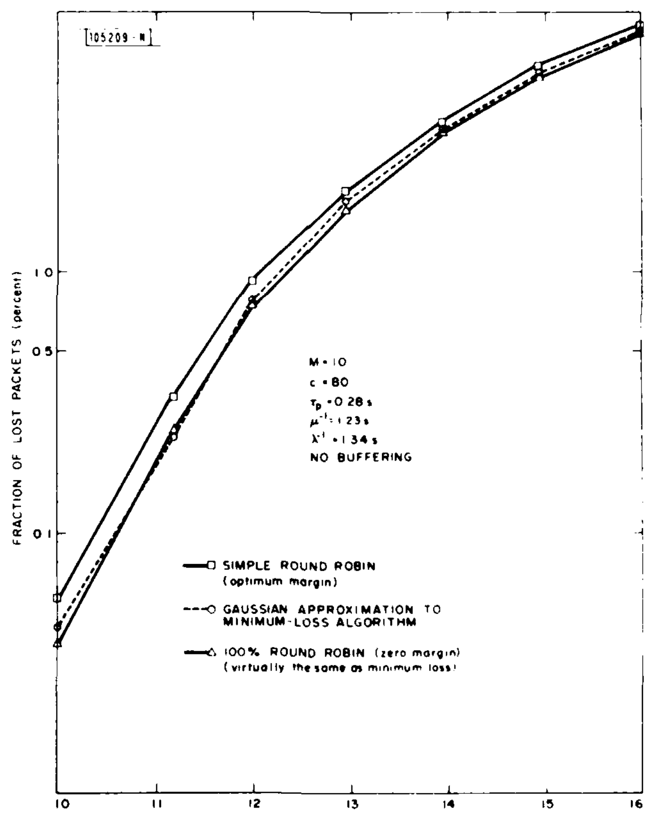


Fig. II-2. Fractional packet loss as a function of number of nodes for simulations of simple round-robin algorithm and "100% round-robin" algorithm, and for minimum-loss algorithm (approximate analytic formula). Note that system TASI advantage equals NM/C so that, for example, N = 12 nodes corresponding to a TASI advantage of 1.5.

The loss fraction, obtained by dividing the expression in (G) by the mean number of active speakers in a frame, is compared with the loss fractions from the simulations using the simple round-robin algorithm with optimum margin and simulations using an algorithm referred to as the "100% round robin." The latter is just the simple round robin with error margin and an extra step that allocates the unreserved slots. For this particular case, it is virtually identical with the minimum-loss algorithm.

In summary, the minimum-loss allocation algorithm, in its approximate form, provides a simple allocation formula [Eq. (II-4)], an equitable spread of lost packets among the speakers, and a simple (but accurate) approximate formula for the average rate of loss of packets, Eq. (II-7). The minimum-loss algorithm has a slightly lower fraction of lost packets when compared with the simple round-robin algorithm with optimum margin.

B. VOICE/DATA INTEGRATION

A simple approximate formula [Eq. (II-5)] for delay vs throughput for a packet speech multiplexer appeared in the previous Annual Report.[1] It was derived from the "lumped-speaker" model, a means of approximating a given number of speakers by a smaller number of fictitious "lumped speakers," to arrive at a simple analytic solution for the queueing behavior. Also in that Annual Report, a suggestion was offered for extending that formula's use to multiplexers combining data with packet speech.

During this past year, it was recognized that the lumped-speaker approximation could be extended even more generally to include cases of combined circuit-switched speech and packet-switched data [e.g., the Slotted Envelope Network (or SENET) system], provided the delays are borne by the data and the data are predominantly short messages, not large file transfers. There are two main steps to this extension.

First the hybrid multiplexer, whose capacity is c bps and whose average data arrival rate is d bps, is replaced (for purposes of analysis) by a fictitious speech-only multiplexer whose capacity is c-d bps. It can be argued plausibly that the average delay (borne by speech) in the fictitious multiplexer is nearly the same as the average delay (borne by data) in the real one.

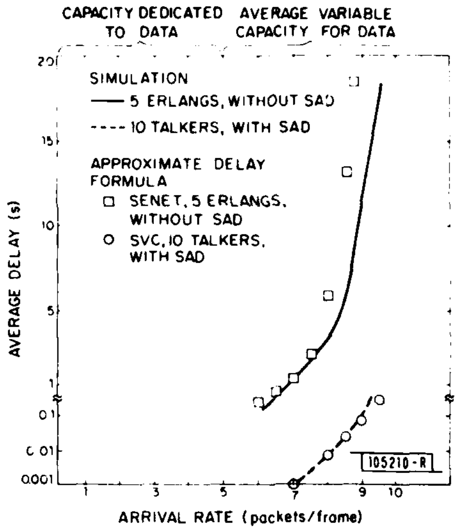
The second step is to postulate that the queueing behavior of the multiplexer is governed primarily by the mean traffic arrival m, the variance of traffic arrival σ^2, and a characteristic correlation time for traffic arrival τ. If valid, this postulate allows us to apply the lumped-speaker delay formula, developed for packetized talkspurts produced by a fixed number of speakers, to other correlated arrival processes. If the traffic were talkspurts, τ would be in the neighborhood of the average talkspurt duration; if it were circuit-switched calls, τ would be close to the average holding time. The average delay would then be given by the lumped-speaker formula, here expressed in terms of m, σ, τ:

$$\text{Average delay} = \tau\rho^{M}\left[\frac{\sigma^2/m^2}{1-\rho} - \frac{1}{M\rho}\right] \tag{II-8}$$

where $\rho = m/(c-d)$, and M is the largest integer such that $(M-1)(1 + M\sigma^2/m^2) \leq M/\rho$.

In order to test both the technique of the fictitious multiplexer and the application of the m, σ, τ characterization to other arrival processes, the delays given by the above formula have been compared with those of simulations of the SENET and SVC schemes reported in the 1978 Annual Report.[2] The smooth curves in Fig. II-3 show those simulation results for a SENET system with 5 Erlangs of voice traffic and an SVC system with a fixed number of calls but with an

Fig. II-3. Comparison of analytical delay-vs-throughput formula with simulation results for SENET and SVC. Note that SVC includes speech activity detection (SAD), but SENET does not.



average of 5 talkers in talkspurt. A total channel capacity of 15 voice slots was chosen, with 5 dedicated to data. The discrete points (labeled "approximate delay formula") were plotted using the above formula. They match quite well the results of the simulations, yet were obtained with far less computation.

The values of m, σ, and τ were computed for the SVC case as follows. The system parameters obtained from Ref. 2 were

N = number of speakers = 10,

P = average speaker activity = 0.5, and

T = average talkspurt length = 1.23 s.

The mean and variance of the speech arrival rate were obtained from the formulas for the binomial distribution:

$$m = N_P$$

$$\sigma^2 = N_P(1-p) \quad .$$

The value chosen for τ was the (unique) exponential delay time for the autocorrelation function of the arrival process. It is the same delay time that appeared in the speech prediction formulas in Ref. 2 and may be expressed in terms of T and p as follows:

$$\tau = T(1-p) \quad .$$

The values of m, σ, and τ for the SENET case were based on the following system parameters from Ref. 2:

N = maximum number of calls accepted = 10,

λ = average call arrival rate = 0.05 s^{-1}, and

μ = reciprocal of average call holding time = 0.01 s^{-1}.

The mean and variance of the speech arrival rate were obtained from the Erlang distribution:

$$m = \sum_{K=0}^{N} p(K)$$

$$\sigma^2 = \sum_{K=0}^{N} K^2 p(K) - m^2 \qquad \text{(II-9)}$$

where

$$p(K) = \frac{\frac{1}{K!}\left(\frac{\lambda}{\mu}\right)^K}{\sum_{j=0}^{N} \frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j} \qquad 0 \leq K \leq N \quad .$$

The autocorrelation function for the arrival process in this case is expressible in terms of many decay times, not a unique time. The value used for τ was an approximation to the dominant decay time, in this case 93.0 s, slightly less than the average holding time.

C. MULTI-LINK TALKSPURT SWITCHING

In the previous Annual Report,[1] there appeared a preliminary discussion of a technique for achieving TASI operation in multi-link, hybrid nets. The technique resembles the SENET and SVC concepts in its use of framed TDMA in which the slots not used for voice in a given frame could carry data. Its chief innovation is the use of the wideband data capability to dynamically reassign a slot within a trunk group to each talkspurt of a conversation. (The choice of trunk groups, which defines the switch-to-switch path of the conversation, is fixed when the call is set up.) This reassignment is controlled by short packets, called activity signals, heralding the start and end of talkspurts, so speech activity detection is unnecessary at intermediate switches. Another feature is the brief buffering, rather than immediate clipping, of a talkspurt arriving at a switch when no slot is available on the outgoing trunk group.

It is worth pointing out that the union of voice and data is more intimate in the multi-link TASI system than in SENET or SVC. In the latter two, the data traffic complemented the voice traffic chiefly by its statistical smoothing effects, in particular the greater tolerance of its users to buffering delays. This smoothing permitted more efficient use of bandwidth. In the multi-link TASI system, the wideband data capability is essential to the control of the voice traffic. This enhanced control permits the multi-link TASI operation, which leads to even greater bandwidth efficiency.

This year, two studies were conducted to help evaluate the practicality of the multi-link TASI idea. One study, done solely for the purpose of putting the signaling load into a familiar perspective, hypothesized that the activity signals would be carried by the Common-Channel Signaling (CCS) paths of a telephone network. The CCITT No. 6 signaling system, very similar to the Bell System's Common-Channel Interoffice Signaling (CCIS), was the assumed message format. Signals are sent as multiples of 28-bit signaling units. The initial signaling unit defines the signal type and the circuit to which the signal refers. Subsequent signaling units contain unrestricted fields for additional information. A talkspurt activity signal is of the form, "A talkspurt belonging to conversation x (for which a switch-to-switch route has previously

been established) is about to begin (end) on circuit y." Such a signal would occupy two signaling units, the first containing y and the second x.

The signaling channel used for CCIS is a 4-kbps subchannel composed of every second framing bit of the T-1 carrier. If it were used for the activity signals, two problems would ensue. First, if one assumes a maximum TASI advantage and an average talkspurt duration of 1.23 s, it would take 14 ms just to clock the signals for the 24 voice channels of a T-1 carrier. The average signaling traffic would be 60 percent of the CCIS channel capacity. Under reasonable assumptions for a queueing model, the average queueing delay for an activity signal would be about 22 ms, with a few percent of the signals queueing for several times this long. The resulting system would be too sluggish for efficient talkspurt switching, even in the absence of other traffic one might normally expect on the CCIS channel.

Therefore, it will probably be necessary to devote a full 64-kbps voice channel to the activity signaling for the other 23 voice channels on a T-1 carrier. Whether this should be done under the recently developed CCITT No. 7 signaling system, or whether it should be done under an independent protocol – more efficient for talkspurt switching – is not clear at this time.

An additional study was done to characterize the expected speech quality and buffer requirements of the multi-link TASI system. A simple model has been proposed for describing the delay at a node. It differs from earlier models of speech multiplexers used at Lincoln because its arrival process for talkspurts depends on the conditions in the queue. Earlier models had postulated an independent arrival process. In particular, it is assumed that a speaker will not produce a new talkspurt before his last talkspurt has been fully transmitted onto the channel. The justification for this assumption is that the two parties to a conversation usually alternate talkspurts.

More precisely, the number of talkspurts in the system (being queued or being transmitted) is described by a birth-death process. The probability per unit time of a new talkspurt entering the system is proportional directly to the number of speakers not having talkspurts in the system, and inversely to the average length of a silence. The probability per unit time of a talkspurt leaving the system is proportional directly to the number of talkspurts being transmitted, and inversely to the average length of a talkspurt.

The model has a simple solution in classical queueing theory. A computer program has been written to calculate that solution and has been run for the cases of 5, 10, and 20 outgoing voice channels with varying numbers of speakers and available buffers. The results shown in Fig. II-4 may be summarized as follows:

(1) The average buffer memory used is one or a few times the number of bits in the average talkspurt, nearly independent of the number of outgoing voice channels (75 kbits for 64-kbps speech).

(2) The average delay, given that delay occurs, is one or a few times the time in which the combined output channels can transmit the number of bits in an average talkspurt. This is of the order of 100 ms for the cases studied.

Figure II-5 shows the probability that a talkspurt will be delayed as a function of TASI advantage. Average talkspurt and silence durations of 1.23 and 1.34 s, respectively, were used.

The above analysis assumed an unlimited amount of available buffer memory. In practice, talkspurts which have been buffered for a certain length of time will have their leading portions

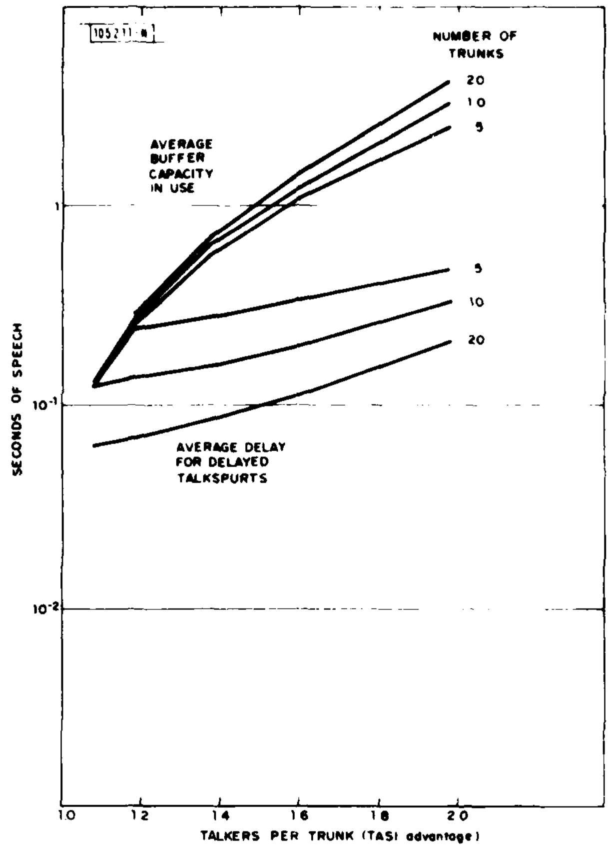


Fig. II-4. Average buffer capacity and average delay for delayed talkspurts as a function of number of talkers per trunk (TASI advantage) in a buffered talkspurt switch.

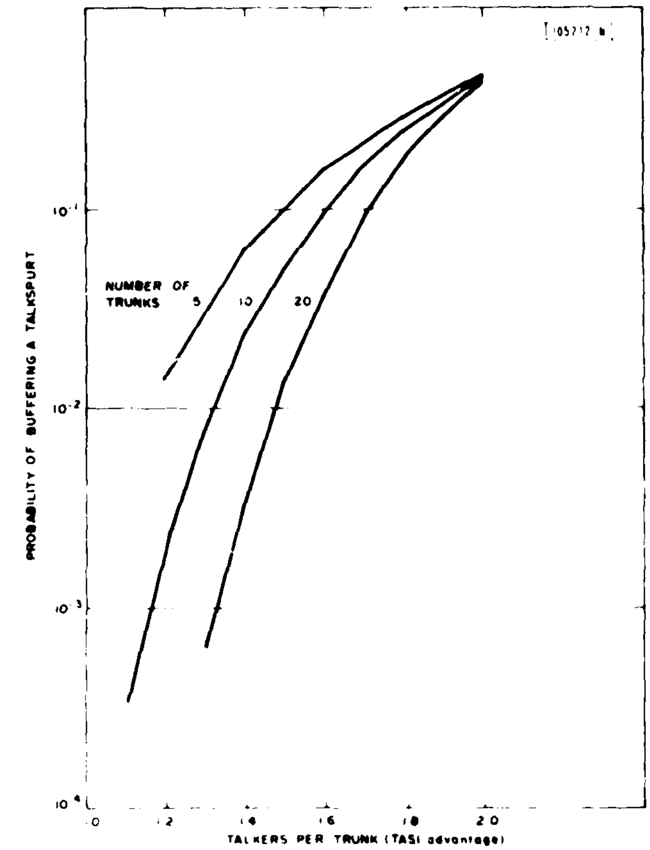


Fig. II-5. Probability that a talkspurt will be buffered as a function of number of talkers per trunk (TASI advantage) in a buffered talkspurt switch.

clipped. The result will be a decrease in average delay (paid for by losses of speech). To examine the trade-off between delay and loss, a computer simulation of a talkspurt switch was done. A typical result is shown in Fig. II-6, for a TASI advantage of 1.6. Included for comparison is a similar result for packet switches obtained by the analytic technique described in Ref. 1.

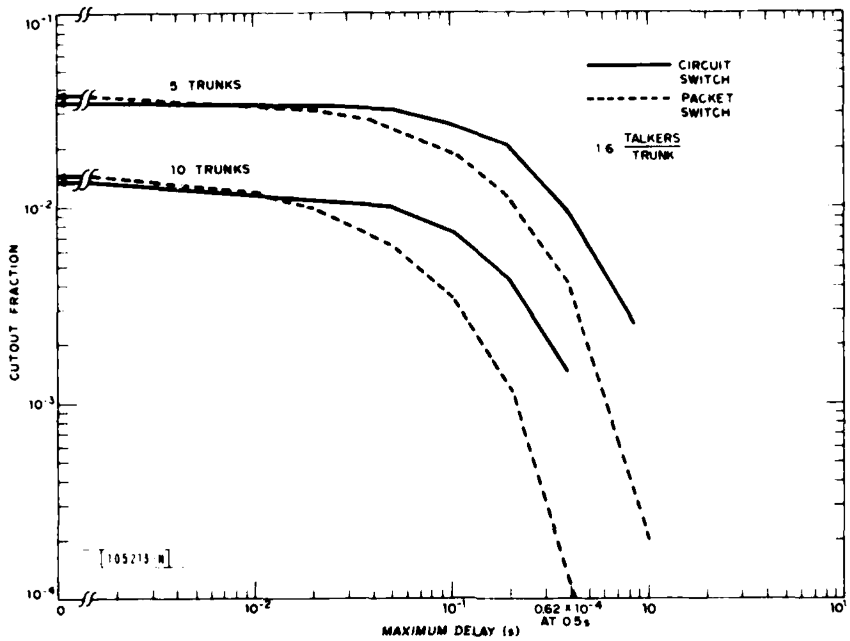


Fig. II-6. Illustration of trade-off between cutout fraction and delay in buffered talkspurt switches and in packet switches.

It should be mentioned that these performance studies considered voice only, whereas the multi-link TASI system is intended for an integrated voice/data network. In the latter, the data traffic (except for the small portion controlling the speech traffic) can be given lower priority than the voice traffic, with the result that much of the delay can be shifted from the voice to the data traffic. Since data users usually tolerate delay better than voice users, there is a fortunate match between these two traffic types.

D. ADAPTIVE ROUTING

A complex and very important aspect of networking is routing, which determines how a path through the network from source to destination is assigned for each call setup request. Routing includes two basic functions: (1) the establishment, and possibly the adaptive modification, of routing tables utilized by the switches to determine paths for calls; and (2) the actual operation of the switches to set up individual call paths based on these tables. The first function is an important element of the more general task of system control for a switched network, which includes all the management functions necessary to maintain survivable and efficient network performance in a highly dynamic operational environment. The second function is an important

element of network operation, which can also have a significant impact on the network grade of service. For military networks, traffic routing techniques have the dual objectives of wartime robustness and peacetime efficiency. Robustness is the overriding consideration; the system must reliably provide essential communication services to critical users in spite of possible severe network damage.

The future DSN is projected as a distributed mixed-media network with control as close as possible to the users. Small, on-base, digital circuit switches will be used, and the potential exists to exploit the computational power of modern switches to adaptively route traffic in the face of network changes. In this context, Lincoln has initiated a network routing study during FY 80 focusing on adaptive routing techniques relevant to the DSN. During FY 80, the two basic routing functions described above have been addressed. Subsection 1 below describes a new sequential algorithm for determination of individual call paths based on a given set of routing tables. With respect to adaptive modification of routing tables, a study and initial simulation of a distributed algorithm for fail-safe routing table adaptation have been initiated. This topic is covered in Subsection 2 below.

1. Sequential Algorithm for Call Routing

A new adaptive scheme called the Sequential Routing Algorithm[3] has been proposed as a possible means for enhancing peacetime efficiency and wartime robustness for a military network of the type projected for the DSN. The algorithm assumes the existence of modest computing capability at each node and a reasonably fast common-channel signaling mechanism joining the node processors. Preliminary analytic results indicate that this algorithm offers a high degree of adaptability in the face of severe congestion or network damage, coupled with fast and effective performance under conditions of light loading.

The essence of the sequential routing algorithm is an efficient tree search of all possible routes for each call, taking full advantage of distributed processing and using a variable-length header field in the call request message as a "traveling path memory." The name of the algorithm reflects its similarity to the powerful tree-search receiver technique called Sequential Decoding[4,5] which is used with convolutional transmitter codes in certain forward-error-correcting systems. A sequential decoder maps all possible message hypotheses onto a tree, and for each evaluated path it computes a "likelihood ratio." This quantity is largest for the best path (that is, the one having the greatest probability of being the message that was actually transmitted). The decoder must incorporate a fairly powerful computer to perform the calculations, as well as a sizable memory to keep track of all the tree branches searched so far in a particular decoding episode. Various techniques are used to achieve a low average processing load by proceeding rapidly to the best answer under normal conditions, while reserving the capability of searching more deeply into the tree whenever necessary to combat an occasional severe burst of noise.

In the sequential routing algorithm the tree information is contained in sets of simple, easily modified routing tables distributed among the N nodes of a network. Each node has (N − 1) tables, one for each of the other nodes in the network. Instead of listing complete routes, each table lists (in order of preference) only the first link of the possible routes. Each intermediate node between a source and a destination for a particular call receives the request message and processes it by determining the uppermost (unblocked) link in its local routing table for the desired destination. It then forwards the request message to the node at the end of that link, and

the sequence is repeated. If all routes out of an intermediate node are blocked, the request is "cranked back" one or more steps to previous nodes. A complete history of the forward and crankback moves of a request is maintained in an expandable header on the request message, with each intermediate node adding an item to the header describing the action currently being taken. Thus, in principle, the request message could make an exhaustive search of all possible routes from the source to the destination. Like the sequential decoder, however, the sequential routing algorithm needs to search only far enough to find the first satisfactory solution. This is fast and efficient under light traffic conditions, but substantially reduces the probability of blockage under heavier loading, as explained below.

The adaptive character of the algorithm lies in the ability of individual node processors to alter their routing tables to reflect changes in the status of links connected to them. Whenever a node discovers that one of its links has become inoperative for some reason (such as an electrical fault or destruction of the link or the equipment at its far end), the node cancels all references to the affected link in all of its routing tables. This automatically causes the system to find alternate routes around the inoperative node or link, if any alternate connectivity still exists.

The sequential routing algorithm is very difficult to analyze mathematically, primarily because of the number and complexity of possible routes through a network when crankback is allowed. Computer programs are in preparation, as described below, for both steady-state evaluation and dynamic performance simulation of the algorithm in general networks. Meanwhile, some interesting preliminary results have been obtained for a class of idealized symmetric tree networks having enough structure and simplicity to permit analysis with reasonable effort. Specifically, we have considered the steady-state performance of network models of the form illustrated in Fig. II-7 composed of identical copies of a simple one-input, two-output node. It is assumed that the conditional probability of link blockage at this node, given that a call attempt is made, is p for the upper output link and q for the lower output link. The protocol for handling a new call entering the node is to try the upper link first; if blocking occurs there, try the lower link; and if blocking occurs again, crank back to the previous node. The probability of successful transmission is thus $(1-p)$ for the upper link and $p(1-q)$ for the lower link, and the probability of blocking at that node is pq. This is illustrated for the leftmost node in Fig. II-7.

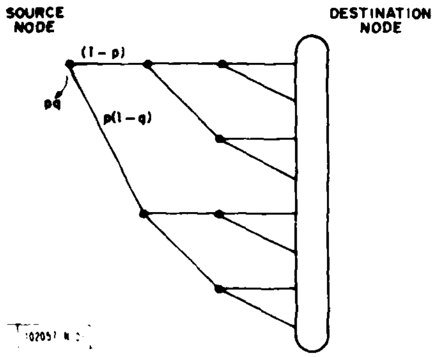


Fig. II-7. Third-order symmetric tree network model.

A k^{th}-order tree of this type has k columns of nodes, and 2^k outputs connected to the destination node; k = 3 for the example shown in Fig. II-7. For any positive integer value of k, exact iterative formulas have been obtained for the blocking probability $(P_b)_k$, the number of distinct forward throughput paths F_k, and the number of distinct reverse (crankback) paths R_k. Exact formulas have also been derived for the blocking probability $(P_{sf})_k$ achieved in the same network with a spill-forward routing algorithm, and it is shown that $(P_{sf})_k$ is higher than $(P_b)_k$ by a factor of k when traffic conditions are moderate to light.

A (k + 1)-order network consists of two k^{th}-order networks fed by the two branches of a node identical to those in Fig. II-7. Figure II-8 illustrates the derivation of the iterative formula for the probability $(P_b)_{k+1}$ that a call request entering the first node of such a network is blocked. The four possible ways that blocking can occur are indicated at the left of the figure. The first occurs with probability pq, when both branches out of the first node happen to be blocked. The second blocking mechanism occurs when the request gets out of the first node via its upper branch [probability $(1-p)$], is blocked by the upper network N_k [probability $(P_b)_k$], cranks back to the first node, and is then blocked when it tries the lower branch (probability q). The joint

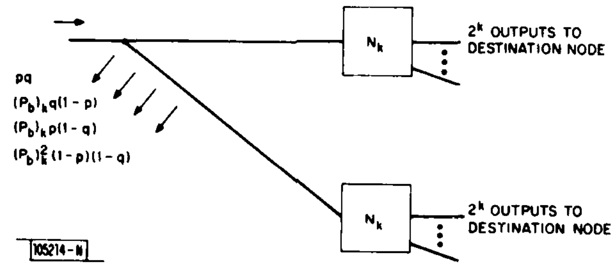


Fig. II-8. (k + 1)-order symmetric tree network.

probability of this event is $(P_b)_k\, q(1-p)$, as indicated. The third blocking mechanism involves a direct trip down the lower branch out of the first node, and blockage there by the network N_k. In the fourth mechanism the call request goes out on both the upper and lower branches in succession, and is blocked by the k^{th}-order network on each of them. The total probability of blocking in the network of Fig. II-8 is the sum of these four terms:

$$(P_b)_{k+1} = pq + (P_b)_k\, q(1-p) + (P_b)_k\, p(1-q) + (P_b)_k^2\, (1-p)\,(1-q)$$

$$= pq + (P_b)_k\, x + (P_b)_k^2\, (1 - pq - x) \qquad \text{(II-10)}$$

where

$$x = p + q - 2pq \quad .$$

Applying Eq. (II-10), we have

$$(P_b)_1 = pq$$

$$(P_b)_2 = pq\,[1 + x + pq(1 - pq - x)]$$

$$(P_b)_3 = pq\,\{1 + pq\,[1 + x + pq(1 - pq - x)] + p^2q^2(1 - pq - x)\,[1 + x + pq(1 - pq - x)]^2\}$$

and so on. Notice that the expressions for $(P_b)_k$ are all of the form $pq(1 + \delta)$, where δ is of first or higher order in p and q.

We may now examine the blocking probability improvement offered by the sequential routing algorithm, compared with a spill-forward scheme in which node output branches are chosen in the same way but crankback is not allowed (i.e., a call request which is blocked at some node is dropped immediately). It is easily shown that the analog of Eq. (II-9) for such a spill-forward network is

$$(P_{sf})_{k+1} = pq + (P_{sf})_k (1 - pq) \tag{II-11}$$

where

$$(P_{sf})_1 = pq \quad .$$

We notice immediately that Eq. (II-11) has the form

$$(P_{sf})_{k+1} = pq(k + 1 + \epsilon)$$

where ϵ is of higher order in p and q. In other words, for a network of order k in which p and q are small (i.e., the grade of service at the individual network nodes is reasonably good), the sequential routing algorithm achieves a blocking probability that is lower by a factor of k than would be achieved by the spill-forward algorithm.

For some purposes, it is of interest to know the number of distinct forward routes F_k through a k^{th}-order symmetric network with sequential routing, and the number R_k of distinct reverse routes by which a call request can be cranked back out of the network. Figure II-9 illustrates the derivation of iterative formulas for these numbers.

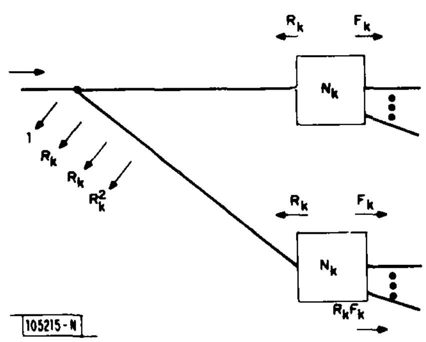


Fig. II-9. Enumeration of routes for a symmetric tree network.

For a single call request entering the first node, there are three throughput mechanisms: F_k different forward routes through the upper k^{th}-order network, F_k different forward routes through the lower network, and a set of R_kF_k routes that get cranked back from the upper network and make it through the lower one. Thus, we have

$$F_{k+1} = F_k(2 + R_k) \quad . \tag{II-12}$$

The four reverse mechanisms shown in Fig. II-9 are: one possibility for immediate blockage at the first node; R_k different crankback routes from the upper branch, which then fail to get out via the lower branch; R_k crankback routes from the lower branch; and R_k^2 routes which get

cranked back on both the upper branch and the lower branch. The total number of possible paths which are available to be cranked back from the (k + 1)-order network is therefore

$$R_{k+1} = 1 + 2R_k + R_k^2 = (1 + R_k)^2 \quad . \tag{II-13}$$

The following table shows the extremely rapid rise in the number of possible routes in these networks:

k	R_k	F_k
1	1	2
2	4	6
3	25	36
4	676	972
5	458,329	659,016

The relationships between these specialized networks and realistic topologies are receiving further study. Clearly, one would not expect to find a real network connected in this special way, and made up of identical nodes with identical link-blocking probabilities. In general, one must resort to computer simulations and numerical techniques to analyze complex practical networks; Lincoln is addressing this also, as described in the following paragraph. On the other hand, the symmetries of the model used here lead to clean, analytically tractable solutions, as we have seen, which appear to offer important insights into the behavior of more general networks and may, indeed, yield approximate solutions applicable to certain classes of more realistic networks.

Work is in progress on the development of a software test bed aimed at further investigation of the sequential routing algorithm as well as other adaptive routing techniques. The initial implementation of the software test bed has focused on a particular example of a distributed adaptive algorithm called the "Failsafe Distributed Routing Protocol,"[6] which has been proven analytically to have the capability to adaptively route around failed portions of a network. This protocol is one of several techniques described in the literature which offer the capabilities of initializing routing tables for a network, and subsequently modifying them in response to changes in the network. The software test bed is designed to permit measurement and comparison of the performance characteristics of such protocols, alone and in combination with sequential routing, spill-forward routing, polygrid routing, and others.

2. Distributed Algorithm for Routing Table Adaptation

As discussed above, network routing can be separated into two basic aspects. The first applies to a network in the steady state. It operates at the individual switch and consists of forwarding routing requests to the appropriate next switch with the aid of a fixed set of tables of preferred routes to the desired destination. It is usually tailored to optimize some measure of network performance, e.g., to minimize overall delay.

The second aspect of routing is the adaptation to changes in network resources, e.g., the loss of some links. Its operation is global and must be implemented either by a central controller or a set of distributed controllers. Its goal is to effect timely updates to the tables controlling the routing at individual switches. In other words, the adaptive routing aspect amounts to dynamic recalculation of the parameters controlling the steady-state aspect.

The sequential routing algorithm described above combines some of both aspects. It is basically static in the sense that it has very limited provisions for altering its routing tables. However, it is global and adaptive in the sense that its traveling path memory permits many switches to participate in a distributed search for alternate routes. Suppose damage occurs and the sequential routing algorithm, after pursuing several dead ends, does find an alternate route. It will have to search through essentially the same set of dead ends to find the same alternate route when the next routing request arrives.

Therefore, the sequential routing algorithm needs to be supported by a basically adaptive routing procedure, one that can update the routing tables in response to network changes. One promising candidate is the "Failsafe Distributed Routing Protocol."[6] It has been shown theoretically that this protocol will maintain a loop-free directed tree (spanning every physically reachable node) of preferred routing links toward a given destination, called the sink. Moreover, if conditions in the network stay the same long enough, the tree will converge to one of "minimum-length" paths. A typical tree is shown in Fig. II-10 by the solid arrows for the case where node 1 is the sink. The tree is updated by means of distributed computation at all the nodes, supported by messages exchanged between nodes and their immediate neighbors. Such a tree can serve as the basis for the tables of the sequential routing algorithm.

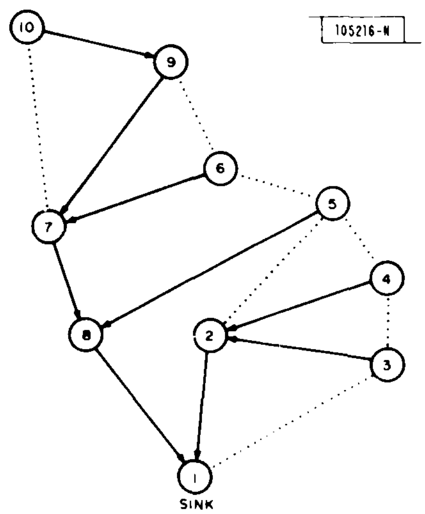


Fig. II-10. Tree of preferred routes for illustration of Failsafe Distributed Routing Protocol. Preferred routes to "sink" node 1 from all other nodes are shown by solid lines with arrows.

Since we know of no simulation of or field experience with the Failsafe Distributed Routing Protocol, we have designed and programmed a computer simulation of it to help answer the following kinds of questions:

(a) Has the theoretical work overlooked some practical aspect that would affect the validity of its results?

(b) What volume of computation and signaling is needed? As the protocol is now written, these volumes will grow much more rapidly than the number of switches grow. Are there modifications to the protocol that will permit its extension to very large networks?

(c) How should the steady-state aspect or routing be coupled to the adaptive aspect? Too rapid rerouting of traffic in response to temporary change can result in instability; yet too sluggish rerouting in the face of major damage can defeat the main purpose of the adaptive protocol.

The basic protocol has been programmed and debugged. A graphical display package has been designed to provide insight into network behavior. Experiments completed with the simulation during FY 80 have shown that the basic failsafe routing algorithm works effectively according to the theory described in Ref. 6. The simulation will be utilized as an important tool in our FY 81 studies on adaptive routing and network control.

E. NETWORK DESIGN AND PERFORMANCE ANALYSIS

The important topic of network routing, discussed in the preceding section, is closely related to the topics of network design and performance analysis. On a long-term basis (e.g., weeks or months), one adapts to network cost or traffic pattern changes by redesigning the network as well as the routing tables. This redesign is evaluated by network performance analysis. During FY 80, Lincoln has requested and obtained from Defense Communication Engineering Center (DCEC) a network design and performance analysis computer program[7] to aid in the overall routing study. The program was originally written to aid in the design of large-scale terrestrial circuit-switched networks. Point-to-point satellite links are easily included, but modifications are required to incorporate broadcast DAMA satellite links. This program has been installed at Lincoln, and modifications have been defined to incorporate DAMA satellite links into the network routing and performance analysis sections of the program.

The intelligence of the program is concentrated in three sections:

(1) "Shortest"-path routing: given a network topology and individual link "lengths" (generalized costs per unit traffic), find a preferred routing from each source to each destination that minimizes overall path "length."

(2) Topological improvement: given a network loaded with traffic according to a certain traffic matrix and a certain preferred routing, seek additions or deletions of links to improve an overall performance measure (in this case to minimize "cost").

(3) Performance: given a network design (including routing-preference tables and a strategy for using them) and a traffic matrix, evaluate in detail the expected grade of service.

During FY 80 the program has been modified for compatibility with the Lincoln central computer facility, and test programs have been run successfully. Modifications to the performance section to include broadcast satellite links have been defined. The intent of these modifications is to make the program serve as an effective tool for our FY 81 routing studies. The modified performance analysis program will allow us to deal with relatively large networks including both point-to-point and broadcast links, so that we can move beyond the simple but unrealistic symmetric tree networks analyzed in Sec. D above.

Although we have been considering the network design and performance analysis program primarily as a tool for our routing studies, it is also possible that such a program would aid in the reconfiguration of a DSN-like network after damage. The design program could assist in

structuring a modified network topology which might include new links (e.g., from the public telephone network) or additional capacity on selected existing links. This strategy essentially amounts to an automatic network redesign after damage, and would accommodate changes in network structure which are beyond the scope of the failsafe algorithm which can adaptively modify routing tables but cannot reallocate network resources.

III. SATELLITE/TERRESTRIAL NETWORK INTEGRATION

The use of flexible demand-assigned satellite communications as an overlay for a terrestrial network offers a number of attractive advantages. Beyond the cost advantages of satellite trunking, flexible DAMA techniques can provide fast response to changes in traffic patterns and rapid reconfiguration in the event of damage. The use of a distributed routing algorithm and system control techniques is indicated to maximize responsiveness and the ability of isolated parts of a damaged network to perform usefully. However, some central control capability is also needed to allow system control policies to be adjusted to meet globally perceived needs.

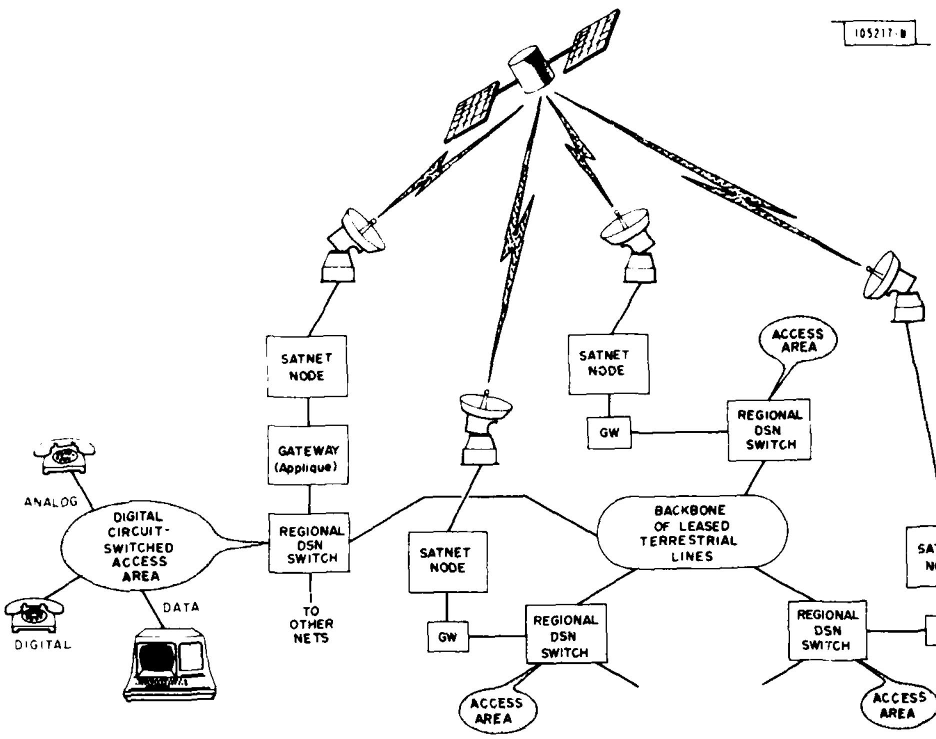


Fig. III-1. Integrated system with satellite network overlaid on switched terrestrial network.

Figure III-1 shows a satellite network overlaid on a terrestrial network whose nodes are DSN regional switches. The figure shows a regional DSN switch servicing integrated speech and data for users in its access area. While the EISN test bed will have some integrated local-access areas, this section will discuss instead the interfacing of EISN to existing circuit-switched speech networks and, separately, to an existing packet data network, the Experimental Data Network (EDN). Therefore, the gateway shown between the regional DSN switch and its associated satellite network node will be described as two distinct parts: Applique C (for "circuit-switched") and Applique D (for "data").

During the past year, Lincoln has developed functional designs and hardware architecture for initial experimental versions of Appliques C and D. Detailed design and construction are

planned for the coming year. The scale of the experiments that can be conducted with these facilities will be limited by the capacity of the EISN satellite channel and the availability of circuit and packet switches. Nevertheless, these experiments will allow a realistic assessment of the implementation requirements for larger-scale gateways, as well as demonstration of the feasibility of a satellite network overlay concept. In particular, EISN will be utilized for experimental validation of functional designs for routing and protocol translation modules in the gateways. The remainder of this section discusses architectural philosophy alternatives for satellite/terrestrial gateways in the EISN context, and explains the specific design choices that have been made for Appliques C and D.

In general, it appears to be true that satellite and terrestrial components can be integrated most effectively in the framework of a single network, rather than as an interconnection of separate and distinct networks. For example, it is preferable for a voice user to dial the number of a party to be called without having to consider whether a terrestrial or satellite route might be involved in the call. The network routing algorithms would decide how best to serve the call request in view of the currently available resources, the caller's precedence, etc. In the other extreme of separate networks, one has the familiar tie-line type of connection, where the caller first dials a number for a gateway and then dials again when in contact with the other network. With the tie-line approach the user has more control of routing decisions, but he requires more knowledge to use the system effectively, and the system is denied much of its potential for optimizing performance. It is therefore preferable that long-range design efforts be directed toward the re 'ization of an integrated network in which satellite/terrestrial interconnections are transparent to users, and the system control algorithms have the greatest potential for optimization.

In practice, however, one must deal with interconnections of networks rather than the design of an overall integrated system. Gateways are required to effect the interconnection, and the best that can be done is to design the gateways to make the interconnection as nearly invisible as possible. For this reason, the Lincoln design for the circuit-to-satellite gateway (Applique C) for the EISN experiments appears as a tandem node in the terrestrial circuit-switched net. As such, it can participate in that network's routing algorithms, and the voice subscribers in the terrestrial net would be unaware of the existence of the satellite links provided by the interconnection except to the extent that they detected the increased delay in the speech path. The same device, in addition to being a gateway between a terrestrial net and a satellite net, is a gateway between a circuit net and a packet net, and must perform the functions of packetization and depacketization so that a continuous bit stream is maintained in the circuit-switched net.

The initial implementation of Applique C will make use of a minicomputer hardware and interface configuration being used at the other wideband satellite network (WB SATNET)* sites for packet voice experiments. Use of this configuration will significantly reduce development costs but will limit the Applique throughput to two or three voice channels. Initial experiments will demonstrate the effectiveness of the circuit-to-packet gateway capabilities in a loop-back mode through the satellite channel. These experiments will exercise in a rudimentary way the routing protocols supported by the Applique. Alternate routing exp ents will be carried out

* The WB SATNET refers to the satellite portion of EISN including the PSATs (demand-assignment processors), Earth Station Interfaces, and the channel itself. The WB SATNET is jointly supported by DCA and DARPA.

at a later time when more than one Applique and real circuit switches with terrestrial connectivity are available.

The Lincoln design for the data-net-to-satellite gateway for the EISN experiments (Applique D) appears as a host computer in both the terrestrial data net and the WB SATNET, so that protocol efficiency and performance measurement experiments can be carried out with the standard internet and transmission control protocols. These protocols place the burden of routing optimization on hosts and gateways rather than on the networks themselves. This strategy is appropriate for an internet environment where the interconnected networks are really independent, but it is not well suited for a situation where a satellite net is simply overlaid on a terrestrial net. In such a situation, the internet routing policy will never choose a path that goes outside the originating network unless special routing is requested by the sender. This standard type of internet datagram service, while less than ideal, is consistent with current data net practices. Data communication experiments in which the satellite net effectively serves as an overlay to the terrestrial packet net can be accomplished by allowing the sender in the terrestrial data network to request routing through the satellite.

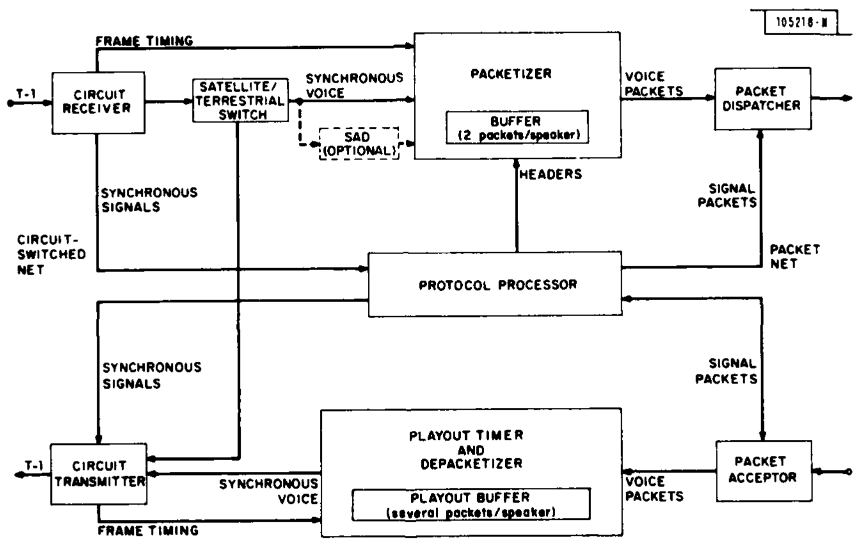


Fig. III-2. Functional block diagram for Interface Applique C for EISN.

Figure III-2 is a functional block diagram for Applique C. Voice and its signaling enter and leave at the circuit-switched side on T-1 digital carriers. While the signaling system is not indicated, the initial implementation of Applique C is intended to support a subset of CCITT Common Channel Signaling System No. 7, as well as one signaling system compatible with commercially available peripheral equipment for digital switches at proposed EISN sites. The circuit receiver block acquires symbol and frame synch with its T-1 carrier and separates the signals from the voice. The circuit transmitter acquires symbol and frame sync with its T-1 carrier and combines the signals with the voice. In the initial implementation of Applique C, only

one to three speakers can be accommodated, so most of the 24 voice channels of the T-1 will be ignored. In the case of CCITT No. 7 signaling, one voice channel will be dedicated to signaling at 64 kbps. Other signaling systems use special subcarriers of the T-1.

The signals for both networks enter the protocol processor, whose main functions are making routing decisions and translating between the protocols of the circuit-switched net and the packet net. In the initial implementation, the routing will be rudimentary. A few local phones will be interfaced to an Applique C at each of two or three EISN sites. The protocol processor will simply decide, from the number being dialed, at which EISN site the other phone resides. If it is a distant site, the only routing available will be through the satellite, and the processor will instruct the satellite/terrestrial switch to route to the packetizer and generate the proper headers to get packets to that site. If the other phone is at the same site, the only sensible routing would be the bypass through the satellite/terrestrial switch to the proper voice slot in the circuit transmitter. However, the option of a loop-back path through the satellite will also be provided to allow experiments to be conducted at a single site.

At a later date, real digital circuit switches with terrestrial connectivity are expected to become available at some EISN sites. Then there will be a nontrivial choice of routing terrestrially or via satellite. There will also be a choice of making the satellite/terrestrial decision in the digital switch or its associated Applique C. Still farther in the future, beyond the EISN experiment, it would make sense to incorporate the gateway function and the DSN regional switch into one hardware package with common control.

When a call from the circuit-switched net is routed via satellite, the synchronous voice stream enters a buffer in the packetizer. There it is accumulated into packets and augmented with a header and a frame time. These voice packets, as well as signal packets from the processor, are sent into the packet net by the packet dispatcher. If the optional Speech Activity Detection (SAD) is used, then those packets deemed to be silence are never dispatched.

Packets arriving from the packet net are separated into signal and voice by the packet acceptor. Voice packets are arranged in the playout buffer according to the relative difference between their recorded frame times and the frame timing derived from the circuit transmitter. The playout buffer must be longer than the packetizing buffer in order to smooth the dispersion in transit times among packets traversing the packet net. Even so, there may come a time to play out speech for which no packet has arrived. This could happen intentionally (through the operation of the SAD) or unintentionally (by packets being lost or unduly delayed). In either case, the playout timer must generate silence to fill the gap.

The hardware being developed for the initial implementation of Applique C is designed to be used in two different experimental configurations. When EISN sites with digital circuit switches become operational, then Configuration I of Fig. III-3 will be used. The Applique C will appear to the digital switch as a tandem switch, connected by one to three interswitch trunks occupying designated time slots of a T-1 carrier. Before real digital switches are available, testing and simple experiments with Applique C will be done using Configuration II of Fig. III-3. Here, a simulator replaces the digital circuit switch. The telephone instruments will be chosen for experimental convenience rather than realism. The four functions of speaking, listening, "dialing," and "ringing" will probably be carried on separate wires. What will be realistic are the signals leaving the simulator and entering Applique C. Coders within the simulator will digitize the telephone speech. A signal translator will convert the "dialing" signals into an interoffice signaling protocol. In other words, the switch simulator will function like a local telephone office, except

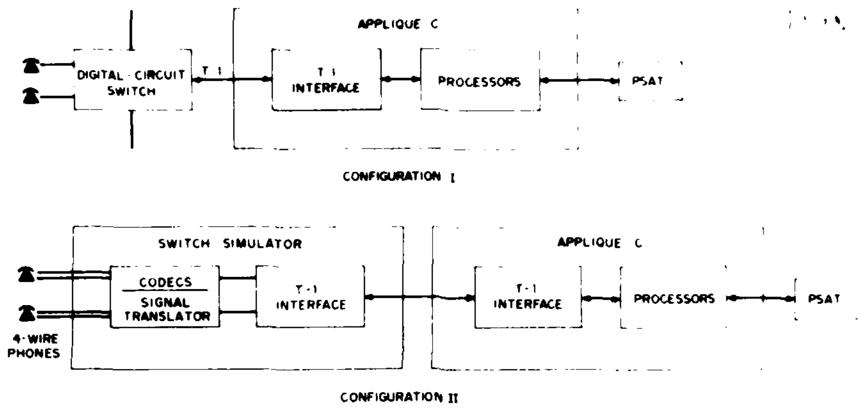


Fig. III-3. Configurations for terrestrial/satellite integration experiments using Interface Applique C.

that it will lack a local switching capability. Even a call from one of its local subscribers to another will be tandemed (on the caller's dedicated trunk) to Applique C for connection.

The initial implementation of Applique C, as well as the switch simulator, will make use of hardware either already built for other purposes in the EISN experiment or available commercially. This approach has been chosen to shorten the design and construction time, even though it is recognized that the initial hardware design probably will not be expandable easily to handle many simultaneous calls. What will carry over to future designs will be the algorithms for protocol translation. However, as the initial implementation of Applique C takes form, it should become clearer how the hardware for a larger version ought to be organized, and design recommendations for such a larger version are considered part of the task of developing the initial version.

Figure III-3 shows Applique C made of two sections, a T-1 interface and processors. The T-1 interface corresponds functionally with the circuit receiver and transmitter in Fig. III-2. The processors perform the remaining functions. The switch simulator is also comprised basically of a T-1 interface and a processor (the signal translator). The hardware to be used for these components will be discussed and illustrated below.

Many of the functions required of the T-1 interfaces resemble functions performed in the digital channel banks sold as peripheral equipment for telephone switches. Today, the most common use of a digital channel bank is the interfacing of a T-1 carrier serving as a trunk group between analog telephone switches. Conceptually, a digital channel bank operates as follows in the analog-to-digital direction:

(a) For each incoming analog trunk, separate the voice from the signal.

(b) Filter, sample, and digitize the voice.

(c) Convert the signals into a digital format suitable for the signaling system being used.

(d) Multiplex the digital voice and signals onto the T-1 carrier with proper carrier and frame synch.

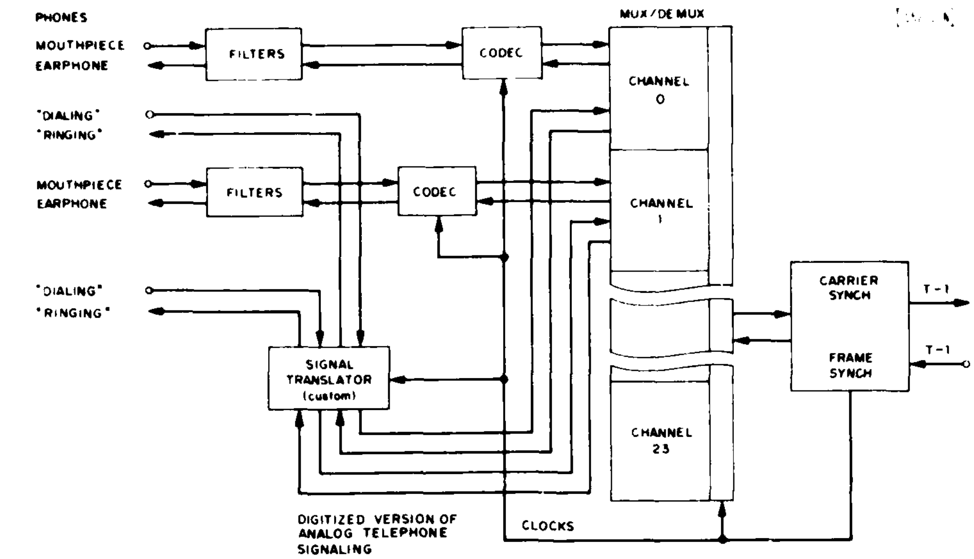


Fig. III-4. A modified digital channel bank serving as a switch simulator. Signals are carried in dedicated subchannels (one per line).

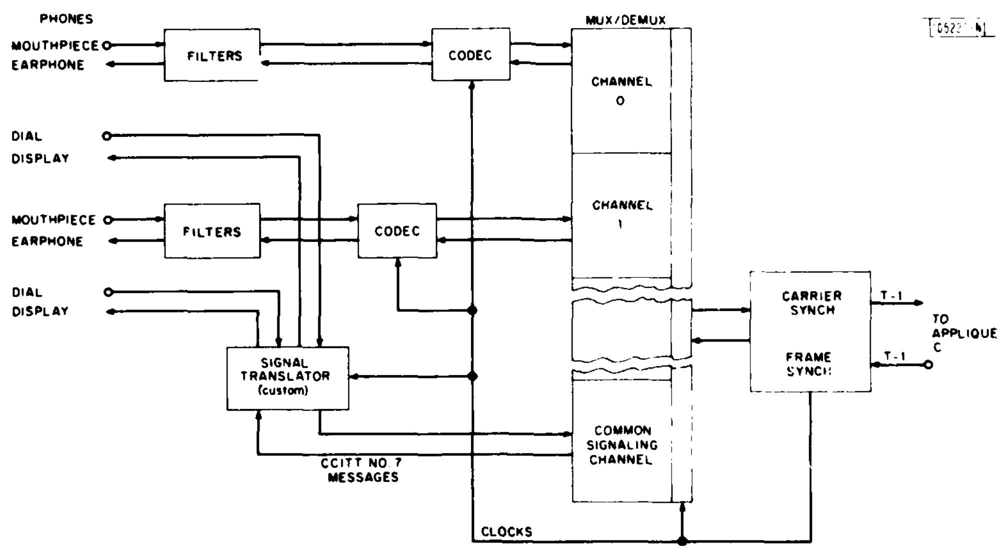


Fig. III-5. A modified digital channel bank serving as a switch simulator. Signals are carried in a common dedicated CCS channel. Assumed signaling protocol is CCITT No. 7.

The actual order of operation of a particular channel bank may differ from the above steps. For example, a channel bank with a common codec would multiplex the voice (into a sampled-analog signal) before digitizing it. Steps (c) and (d) will differ for different signaling systems. Most commercial channel banks implement systems where the signal for each trunk occupies a dedicated subchannel obtained by stealing some or all of the low-order bits of the voice samples. If CCITT No. 7 common channel signaling were being used, one of the 24 voice channels would be dedicated to signaling messages.

Figures III-4 and III-5 are functional diagrams of a digital channel bank modified to serve as the switch simulator. Signaling in dedicated channels (one per line) and CCITT No. 7 common channel signaling are shown in Figs. III-4 and III-5, respectively. Channel bank function (a) is not used, because the voice and signaling are already separated at the phone instruments for two reasons. The first is experimental convenience. Since we are attempting to simulate a switch, not a phone, we might as well avoid the problems of multiplexing signals and speech on the same wires. The second reason is that the channel units (plug-ins that interface individual lines to the channel bank) are made to service interoffice signals, not subscriber-line signals. Therefore, even if a standard phone instrument were used, the channel units would have to be modified to deal with their signals. The filtering and sampling will be done by the channel units, and the digitizing either in channel units or by a common codec, depending on the design of the channel bank. The signals will enter a custom-built, microprocessor-based, signal translator. Different interoffice signaling systems can be simulated with different software in this processor.

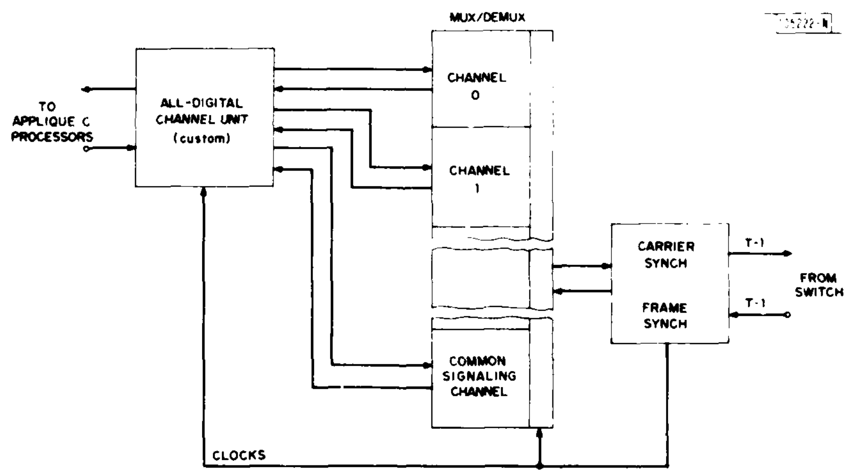


Fig. III-6. A modified digital channel bank serving T-1 interface function of Interface Applique C.

Figure III-6 shows the use of a modified digital channel bank as the T-1 interface of Applique C. Here, the codec function is not needed because both input and output are digital. A custom-built, all-digital "channel unit" simply multiplexes the few voice channels and the common signaling channel (if used) onto a single bus to the Applique C processors. While the channel unit might do some preprocessing of signals to ease the load on the Applique C processors, it would be a simple device compared with the signal translator.

From an examination of the description of a typical manufacturer's channel bank, it seems that the proposed modifications would not be difficult. Since most of the volume of a channel bank is occupied by the 24 channel units, and since Applique C will need at most [illegible] of them, it is hoped that the custom equipment in Figs. III-4 through III-6 can be made to plug into the empty slots and make use of the existing power, signal, and timing buses.

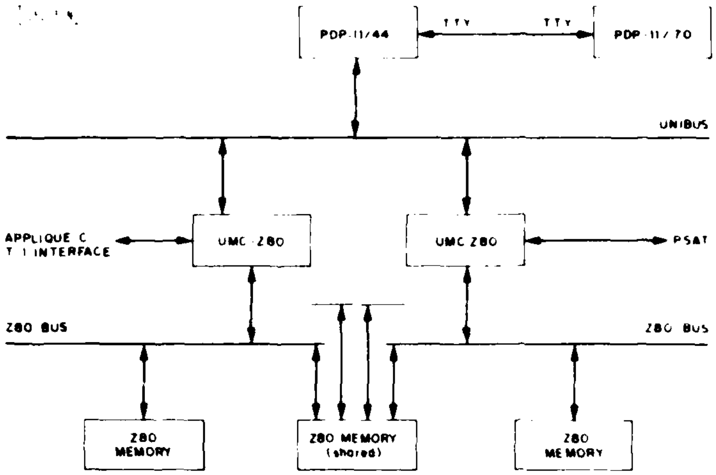


Fig. III-7. Processor configuration for Interface Applique C.

The processor configuration for Applique C is shown in Fig. III-7. It utilizes the same equipment as the speech concentrator: a PDP-11/44 minicomputer; two UMC-Z80's (Z80 microprocessors with UNIBUS interfaces), each with some memory; and a shared memory for the UMC-Z80's. Roughly speaking, the components in Fig. III-7 correspond to the functions in Fig. III-2 as follows. The packetizing and depacketizing are done by the UMC-Z80 communicating with the T-1 interface of Applique C. The buffer storage is supplied by the Z80 shared memory. The packet dispatching and receiving are the tasks of the UMC-Z80 communicating with the PSAT. Protocol processing, including the satellite/terrestrial switching, is done in the PDP-11/44. Software for the processors is developed on the PDP-11/70 and downloaded into the PDP-11/44 and UMC-Z80's via a teletype link.

This same configuration of PDP-11/44 and UMC-Z80's will also be the initial implementation of Applique D, which will interface the EDN at the DCEC site to the EISN satellite network. Instead of the T-1 interface to Applique C shown in Fig. III-5, a card has been built to interface the UMC-Z80 to the 1822 LH-DH data link used for IMP-to-host connection in the EDN. The UMC-Z80 will be programmed to appear as a host to the EDN. Because it is a packet-to-packet gateway, Applique D will be simpler to implement than the circuit-to-packet gateway, Applique C.

It should be noted that, initially, the PDP-11/44 and two UMC-Z80's will be time-shared in their multiple functions of Appliques C and D. Only one of these functions may be experimented with at any given time. Later, a single UMC-Z80 on the PSAT side and multiple UMC-Z80's on the other side may allow more than one kind of experiment simultaneously. Of course, such simultaneous experiments will still be limited in bit rate, since the addition of the extra UMC-Z80's will not increase the throughput of the rest of the processor.

IV. VOICE CONFERENCING

Work on voice conferencing in FY 80 has been concerned with support of the field-test portion of the World-Wide Military Command and Control System (WWMCCS) Secure Voice/Graphics Conferencing (SVGC) Test and Evaluation Program for which the Naval Ocean Systems Center (NOSC) is the lead organization. As originally planned, our work was to include consultation with NOSC on test design, administration, and evaluation, as well as laboratory simulation of field-test configurations and the evaluation of experimental results. As it has developed, laboratory simulation and evaluation of results have not been possible because of delays in the start of the field-test experiments at NOSC which did not get beyond the preliminary experiment stage during FY 80. In particular, the equipment for sharing the broadcast satellite channel which is the essential element in the WWMCCS SVGC approach to conferencing did not become available. Our simulations in FY 79 made certain assumptions about the behavior of that equipment; further simulations planned for FY 80 would have refined the results on the basis of feeding measurements of the actual performance of the equipment back into the simulations. In the absence of such measurements, further simulations would have been pointless.

Because of the delay in the starting of serious testing at NOSC, conferencing work in FY 80 has been concerned only with consultation with NOSC and the preparation of test scenario materials for their use. The work has been largely carried out by Bolt Beranek and Newman, Inc. (BBN), who have provided human-factors support throughout the voice-conferencing evaluation work undertaken at Lincoln Laboratory during the past three years. The primary effort during FY 80 was devoted to preparation of scenario materials for upcoming conferencing tests at NOSC and the evaluation of alternative statistical treatments for data to be collected within that testing environment. A review of the 28 April 1980 NOSC Test Plan was also completed. Each of these activities is described briefly in subsections below.

A. SCENARIO DEVELOPMENT AND TESTING

Following a meeting at NOSC in San Diego in January 1980, it was decided to substitute for the Word-Go-Round (WGR) scenario used during the Lincoln test-bed series, a new scenario that would permit more direct assessment of the ability of noisy channels to support conferencing. In particular, a test was sought that could be employed prior to and immediately after a conferencing session to determine whether or not hardware and software were working properly and to test whether or not subjects were in communication with each other. A simple test, based on words taken from the Modified Rhyme Test (MRT), was devised for this purpose. In this Word Identifiability Test (WIT), each party to the conference reads a set of words over his telephone to listeners who attempt to match the words heard to those appearing on lists in front of them. When one speaker has completed the reading of his words, a second speaker takes over on his telephone. After all parties have spoken, an assessment of the adequacy of the speech links can be made by scoring listeners' response sheets.

Several other scenarios were evaluated informally at Lincoln for possible use at NOSC during the year. The most significant of these was based on a revision of the basic WGR task. In this revision, each member of a conference is given a page containing 50 lines of 4 words each, drawn from the International Phonetic Alphabet and Number sequence. The participant chosen to be the "starter" begins reading aloud the words on his first line while listeners monitor their words for correspondence. When a listener notes a discrepancy between the word he just heard

and the one printed on his list, he immediately begins reading his sequence beginning with the word (or line) following the discrepant word. The original speaker continues until he ascertains that another party has taken over reading of the sequence. If, on reaching the end of the line, no new party is heard, the speaker rereads the line. The task continues in this way until all words have been spoken.

A second version of this task is one in which conferees are provided with segments of text drawn from newspapers, books, etc. into which discrepant words have been introduced. On the basis of our tests, we feel that with further development this second revision of WGR would provide an interesting vehicle for assessment of conferencing networks characterized by satellite-induced delays and distributed control. Because these scenarios focus on the interruptibility of conferencing systems, a factor not currently being stressed in the NOSC tests, further development of scenarios was not pursued.

B. EVALUATION OF STATISTICAL TREATMENTS

One of the most difficult requirements associated with a lengthy evaluation of the type conducted in the Lincoln test bed and envisioned at NOSC is ensuring that either (1) enough test subjects are available that a given subject need serve only once in a given experimental comparison, or (2) each subject is available frequently enough that he/she can serve in all experimental conditions. When the requirements cannot be met by either alternative – that is, when the subject pool is of necessity composed of individuals with varying amounts of experience – considerable information potentially available in the data may be lost. In those circumstances it may be necessary to resort to sophisticated statistical techniques in order to assess the validity of apparent differences among experimental conditions.

In the light of difficulties experienced at Lincoln with test subject availability and in anticipation of similar difficulties at NOSC, BBN undertook a small effort to examine empirically the effects of various methods of normalizing data obtained from a mixed group on levels of statistical significance.

The data base employed in the study was reminiscent of that resulting from the Lincoln test bed, but included frequency distributions that generated different statistical outcomes depending upon the type of normalization used.

Our conclusion from this brief effort was that the basic technique employed to assess the Lincoln data may provide the most satisfactory approach to the handling of data from this kind of research. This method consists essentially of applying nonparametric tests among a set of mean deviations which represent the differences between subjects' ratings for a given conferencing system and test item and their average individual rating over all systems they have experienced. These statistical tests are used to assess the validity of the experimental rating differences among systems. It is expected that the same or a similar technique would be applicable to the NOSC data.

C. TEST SERIES SUPPORT ACTIVITIES

Remaining activities during FY 80 were in direct support of the planned test series. Modified forms of the Consent Questionnaire used at Lincoln and now required in all research employing human subjects were sent to NOSC. A formatted log of critical conferencing events was developed and forwarded for use by test monitors. The goal of this log was to reduce, insofar as possible, the amount of time required to extract timing and event data from the mass of

information collected in a system comparison by acquiring a running record of key items during each experimental session. Finally, the session questionnaires used at Lincoln were redesigned on the basis of our earlier experiences and forwarded to NOSC. The current questionnaire is simpler to use and it samples only those aspects of conferee opinion that appeared salient during the earlier test series.

At the request of NOSC, BBN recently has revived its efforts to produce "consensus" scenarios in order to replace those which have been expended in the course of system shakedown.

D. REVIEW OF NOSC TEST PLAN

Continuing our efforts of last year to support development of a comprehensive NOSC test and evaluation plan, we reviewed the revised plan published in April 1980. Particular attention was given to methods proposed for training of test subjects, administration of experimental conditions, and analysis of results. On the basis of the review, we concluded that most concerns elaborated in connection with the earlier version of the plan had been met, although there remains some doubt that the plan provides for sufficient collection of data to permit generalization of results beyond the college student and National Guard populations identified.

REFERENCES

1. Network Speech Systems Technology Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1979), DDC AD-A086768.
2. Network Speech Processing Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1978), DDC AD-A068318/5.
3. H. M. Heggestad, "A Flexible Routing Algorithm for the Defense Switched Network"; to be presented at the National Telecommunications Conference, Houston, TX, December 1980.
4. J. M. Wozencraft and B. Reiffen, Sequential Decoding (Technology Press and Wiley, New York, 1961).
5. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering (Wiley, New York, 1967).
6. P. M. Merlin and A. Segall, "A Failsafe Distributed Routing Protocol," IEEE Trans. Commun. COM-27, 1280 (1979).
7. M. J. Fischer, D. A. Garbin, T. C. Harris, and J. E. Knepley, "Large Scale Communication Networks – Design and Analysis," OMEGA: Int. J. Mgmnt. Sci. 6, 1 (1978).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-80-222	2. GOVT ACCESSION NO. AD-A097998	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Network Speech Systems Technology Program		5. TYPE OF REPORT & PERIOD COVERED Annual Report 1 Oct[redacted]79 30 Sep[redacted]80
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Clifford J. Weinstein		8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element No. 33126K
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency 8th Street & So. Courthouse Road Arlington, VA 22204		12. REPORT DATE 30 Sep[redacted]
		13. NUMBER OF PAGES 40
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

6 11 9 15 12 47

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

network speech processing	integrated networks	speech encoding
secure voice conferencing	packetized speech	teleconferencing
speech algorithms	human factors	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents work performed during FY 1980 on the DCA-sponsored Network Speech Systems Technology Program. The areas of work reported are: (1) communication systems studies in Demand-Assignment Multiple Access (DAMA), voice/data integration, and adaptive routing, in support of the evolving Defense Communications System (DCS) and Defense Switched Network (DSN); (2) a satellite/terrestrial integration design study including the functional design of voice and data interfaces to interconnect terrestrial and satellite network subsystems; and (3) voice-conferencing efforts dealing with support of the Secure Voice and Graphics Conferencing (SVGC) Test and Evaluation Program. Progress in definition and planning of experiments for the Experimental Integrated Switched Network (EISN) is detailed separately in an FY 80 Experiment Plan Supplement.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

207650

LMED

8