論 文

음성과 데이타가 집적된 패킷통신망을 위한 시뮬레이터개발

正會員 朴 淳* 正會員 殷 鍾 官**

A Simulator for Integrated Voice/Data Packet Communication Networks

Soon PARK* *and* Chong Kwan UN**, *Regular Members*

要 約 音聲과 데이타가 集積된 패킷 通信網의 性能을 豫測하고 시스템 파라메터를 最適化하기 위한 시뮬레이터의 개발에 관하여 記述하였다. 具現된 시뮬레이터는 CCITT의 勸告事項에 따라 運用되는 데이타 터미널이나 host는 물론 패킷 音聲터미널도 연결가능한 音聲및 데이타集積通信網의 性能을 여러 상황에서 豫測할 수 있다. 시뮬레이션 技法으로는 지금까지 알려진 세가지 discrete event 시뮬레이션技法 중 process interaction 方法이 사용되었는데 이 方法을 사용하면 실제 시스템과 가장 비슷한 시뮬레이터를 具現할 수 있다. 시뮬레이터는 약 4,000 line의 GPSS 시뮬레이션 언어와 PL/I으로 具現되었다. 시뮬레이터의 컴퓨터 run time을 줄이기 위하여 GPSS의 LINK block을 사용함으로써 條件的 event의 數를 줄이는 方法을 사용하였다. 구현된 시뮬레이터를 사용하여 7-node 通信網의 性能을 豫測하였다. 또 개발된 시뮬레이터의 妥當性을 檢證하기 위하여 간단한 音聲과 데이타 multiplexer를 시뮬레이션 모델로 구성한 뒤 그 시뮬레이션 결과를 解釋的 방법에 依한 결과와 比較하였다.

ABSTRACT In this paper, the development of a simulator for the performance estimation and parameter optimization of an integrated voice/data packet communication network is described. The simulator implemented is capable of simulating the integrated voice/data network that handles packet voice terminals as well as data terminals and hosts operating under standard CCITT protocols. Of the three discrete event simulation approaches presently known, the process interaction method has been chosen. With this approach one can implement a simulator that is related most Closely with the real system. The simulator has been implemented in PL/I and GPSS simulation languages, resulting in a software package of about 4,000 lines. To reduce the computer run time of the simulator, we have used a method of reducing conditional events based on a GPSS LINK block. We describe various aspects of the simulation model developed. We then investigate the performance of a 7-node network using the simulator, and present the results. For validation of the simulator developed, we construct a simulation model for a simple voice/data multiplexer, and compare the results of simulation with those of an analytical model.

I. INTRODUCTION

As a first step toward the development of an integrated services digital network (ISDN), the integration of voice and data signals in a common network has attracted many researcher's interest [1]-[4]. In 1975, Coviello and Vena proposed an inegrated circuit/packet switching system which realized the circuit switching concept used in the traditional telephone networks

*, ** 韓國科學技術院 通信工學研究室
Communications Research Laboratory, KAIST
論文番號86-12 (接受1986. 3. 18)

and the packet switching concept used in the ARPANET in a common carrier [3]. Moreover, Forgie and Nemeth proposed a virtual circuit switching system in which two kinds of data are processed in the same manner [4].

It is well known that voice and data signals have inherently different statistical characteristics. As a result, they have several conflicting requirements in communication protocols [4]. For example, data signal requires error-free transmission, although it can allow some transit delay. On the other hand, voice has a strict requirement in transit delay (normally up to 200 ms [5]), but it is tolerant of small amount of errors without significant degradation in its quality. Hence, the way to process voice packets should be different from that of processing data packets. In addition, the algorithms for the conventional data network (e.g., flow control) have been found unsuitable for real time vice, and should be amended to meet the above conflicting requirements. Therefore, in the design phase of an integrated network, the choice of system parameters and network algorithms should be examined carefully so that the implemented system may yield the best performance.

It is very difficult, if not impossible, to predict the performance of a large, complex system such as the integrated voice/data packet communication network by mathematical analysis. Simulation offers a convenient tool for the performance analysis and parameter optimization of a system when the mathematical analysis is not possible, and real world tests are not feasible [6]. Generalized computer network simulators have been proposed for data signals only by many researchers [7]-[9]. Those simulators are modularized in software structure such that the user can easily modify some parts of the modules and simulate the performance of a network for a given protocols or algorithms. However, they model the exchange of data packets among host computers and ordinary data terminals, and thus, one cannot simulate the performance of an integrated network in which statistically different signals (e.g., voice, data, picture, facsimile, etc.) coexist.

In this paper, we present a simulator for an integrated voice/data packet communication network. We first model the voice traffic from real conversations. Then, we design the software structure for given communication protocols. With the simulation package implemented, we simulate the performance of a 7-node network, and discuss the results. To reduce the computer run time of the simulator, we study the event scanning method of the simulator. The network modeled by the simulator is composed of several node processors interconnected via high speed links and a number of asynchronous terminals, several hosts and packet voice terminals attached to each node processor. Asynchronous terminals and hosts have been assumed to operate under standard CCITT protocols (X.3, X.28, X.29, and X.25), whereas packet voice terminals follow a simplified X.25 for voice.

A network protocol normally takes the form of layered structure. This layered architecture facilitates the implementation of the entire protocol, since it offers an excellent isolation between two adjacent layers, and requires only correct interfaces [10]-[11]. In our simulator, only three layers (the link and network layers, and a part of the transport layer) have been implemented. The user can select the values of certain system parameters, such as the number of nodes, topological configuration, line speed, packet size, mean message length, bit error rate of a channel, and so on. Besides, the simulator has been implemented in such a way that the user can modify parts of protocols or network algorithms, such as routing, flow control and virtual circuit management, thereby enabling to have simulations for various situations.

Following this introduction, in Section II we specify the integrated voice/data packet network. This includes the organization and the characteristics of the integrated network. In Section III we describe the implementation of the simulator, including the traffic model, the software structure, and the event scanning method that is used for reduction of the computer run time of the simulator. In addition, we check the validity of the simulator using the simple analytical model of a voice/data multiplexing system. In Section IV, we investigate the performance of a 7-node network for various situations using the simulator, and present the results. Finally, in Section V we draw conclusions.

II. SPECIFICATIONS OF AN INTEGRATED VOICE/DATA NETWORK

A packet communication network consists of four groups of basic elements; network node processors (NNP's), communication links, hosts and terminals attached to each NNP, and the network management center (NMC). In an integrated voice/data packet network, the third group also includes packet voice terminals. Fig. 1 shows a configuration of the model for an integrated voice/data packet communication network which is capable of handling both voice and data signal. A virtual circuit (VC) switching system is modeled in this simulator, since it is most appropriate to deal with voice traffic as well as data [12]-[13]. In the VC switching system, the routes established at the call set-up time remain unchanged until the VC is closed by the user.

Of the four groups of network elements, the NNP plays the most important role in determining the network performance, since most of the network programs are contained in the NNP[1] The NNP's interconnected via high speed (56 kbits/s or more) communication links exchange voice and data packets by the store and forward method. Furthermore, they manage hosts and

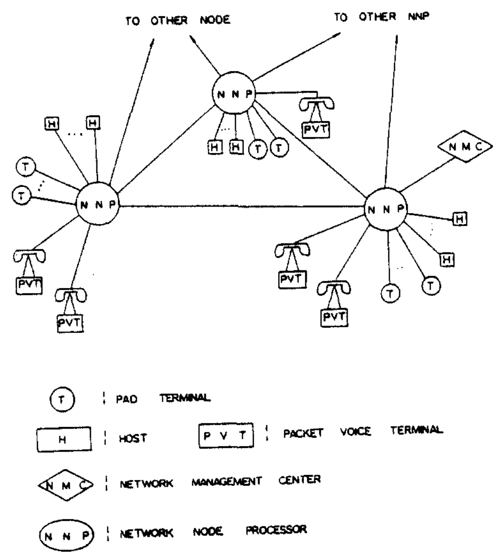

Fig. 1 Integrated voice / data packet communication network.

terminals. The exchange of data packets between two NNP's is assumed to follow the CCITT recommendation X.25.

Several hosts can be connected to each node. These hosts are responsible partly for the packet mode data terminal equipment (PDTE) services specified by the network, such as file transfer, virtual terminal agent, mailing and data transmission, and partly for the asynchronous packet assembly and disassembly (PAD) service. The host interface for the synchronous channel is in the packet mode using the CCITT X.25 protocol, as is the case for communication between two NNP's. Asynchronous data terminals can access a local or remote host through a PAD implemented in the NNP. The access protocol for the asynchronous terminal and the host for the PAD service is assumed to follow the CCITT recommendation X.3, X.28 and X.29.

Voice terminal interface to the NNP is also in the packet mode, but does not follow the X.25. When the X.25 protocol is adopted for handling voice packets, a certain portion of voice packets

[1] In our simulation model we do not include the NMC.

will suffer unacceptably long delay due to the flow control and retransmission strategy used for data. Therefore, for voice transmission one must modify the X.25 protocol such that flow control for the conventional data network, and retransmission procedures are eliminated. In our simulator, we use a system model in which voice and data packets are distinguished at the link layer of the layered protocol.

For voice packetization, speech is digitized in the range of 2.4 to 16 kbits/s using a medium or low rate voice encoder such as adaptive differential pulse code modulation (ADPCM) [14], adaptive delta modulation (ADM) [15], residual-excited linear prediction (RELP) vocoder [16] and linear predictive coder (LPC) [17]. Speech is represented as a series of talk spurts and silence. To achieve the time assignment speech interpolation (TASI) advantage in channel utilization, only talkspurts are packetized using a speech activity detector (SAD), and transmitted [18].

In this work, we use three performance measures to investigate the performance of a packet network. They are throughput, packet delay and the effect of errorneous or lost packets. Throughput is dependent on the packet length, protocol overhead, link speed, bit error rate and the size of a flow control window. Packet delay includes processing delay, queueing delay and transmission delay. Hence, it is a function of packet length, line speed, bit error rate, and network algorithms.

In voice transmission, errorneous packets and packet slips cause degradation of speech quality. When a bit error occurs in a voice packet, repeating the packet in the previous frame is known to yield better quality of speech than using the errorneous packet [3].

Consequently, the relevant factors that govern the network performance are the packet length, the bit rate of a speech coder, bit error rate, the size of a flow control window, the processing power of a switching equipment and the reconstruction strategy for voice packets. In our simulator, all these factors are taken into account.

III. IMPLEMENTATION OF THE SIMULATOR

A. Network Model

A network model used in our simulator is shown in Fig. 2. In our simulation model, data packets are generated from asynchronous term-

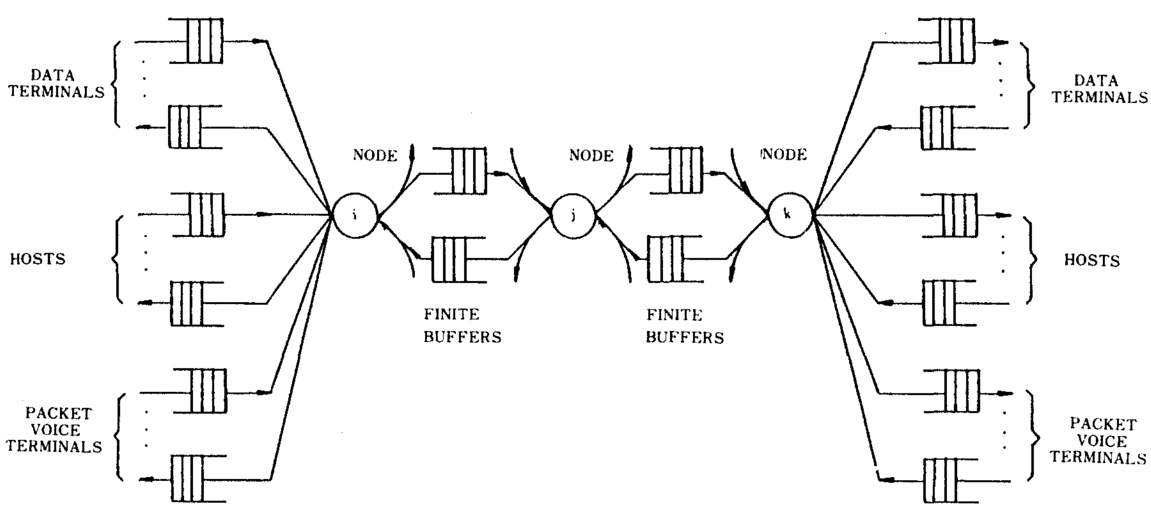

Fig. 2 Network model.

inals, and voice packets from packet voice terminals. Data packets generated are transmitted to a remote host, and the destination host sends messages to the source terminal. As for voice, two packet voice terminals exchange voice packets interactively through the network. Hence, since the simulator models a VC switching system, each packet generated from the source travels on a fixed route that has been established at the call set-up time.

Due to the limit of the link capacity, voice calls are restricted to no more than four for each NNP[2]. Arriving voice calls are discarded if remaing channel capacity is not sufficient for voice traffic. Once a voice call is established, voice traffic has priority over data traffic.

B. Traffic Model

In our simulator the data traffic is modeled as a Poisson process. In this model, the interarrival time of generated bursts is exponentially distributed. Thus, the probability that the next arrival of a burst will occur in time t is given by

$$F(t) = 1 - e^{-\lambda t}, \quad t \geqq 0$$

where $1/\lambda$ is the mean interarrival time. The Poisson process can easily be implemented using the GPSS V function FN$EXPON as the second argument of an ADVANCE block. For the burst length distribution, the results obtained for terminal bursts by Martin have been used [19]. For the length distribution of the response bursts from hosts, a normal distribution is assumed.

For the voice traffic, we use an interactive two-way conversation model in which there are A-TALK, B-TALK, PAUSE-A and PAUSE-B states as shown in Fig. 3 [20]. Talkspurts and silence are generated according to a state table. The lengths of talkspurts and silence are determined by the cumulative functions FNS$ALKS and

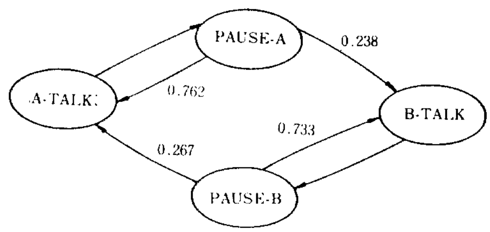

Fig. 3 State transition diagram for interactive talkers.

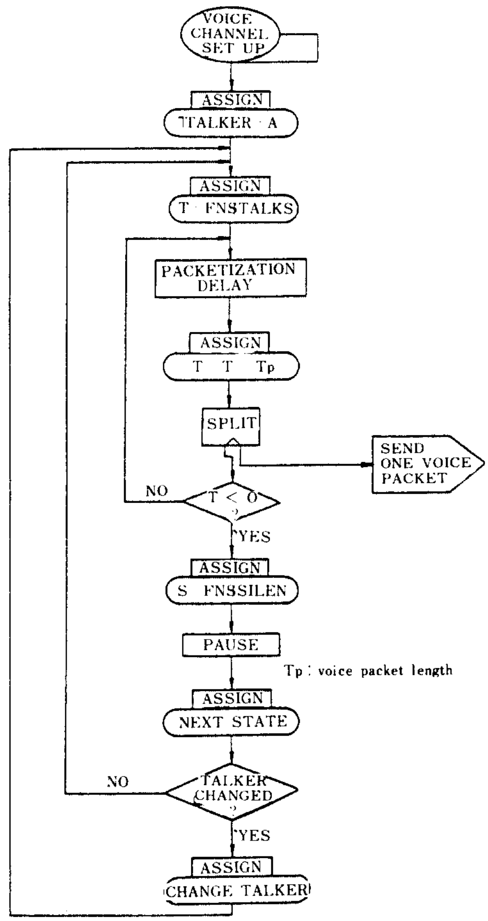

Fig. 4 Procedure for generation of voice traffic.

FNS$ILEN that have been obtained from the on-off statistics of real conversations that is 50 min long.

Fig. 4 shows the procedure for the generation

[2]We assume that a 16 kbits/s speech coder is used.

of voice traffic. In the figure, once a voice call is set up, the talker A is allowed to talk. During this period, a talkspurt is generated with the length determined by the cumulative function FN$TALKS. With the elapse of the talkspurt duration, the talker A enters the PAUSE-A state in which he pauses for a duration of time and the talker B is in silence, and then the talker may be changed by means of the state transition probability (see Fig. 3). Therefore, the actual silence duration for each talker is given by parameters assigned from the FN$TALKS plus those from the FN$SILEN. The on-off characteristics of speech by means of the cumulative functions FN$TALKS and FN$SILEN are shown in Fig. 5. It is seen that the on-off patterns generated are fairly close to the results obtained by Gruber [21].

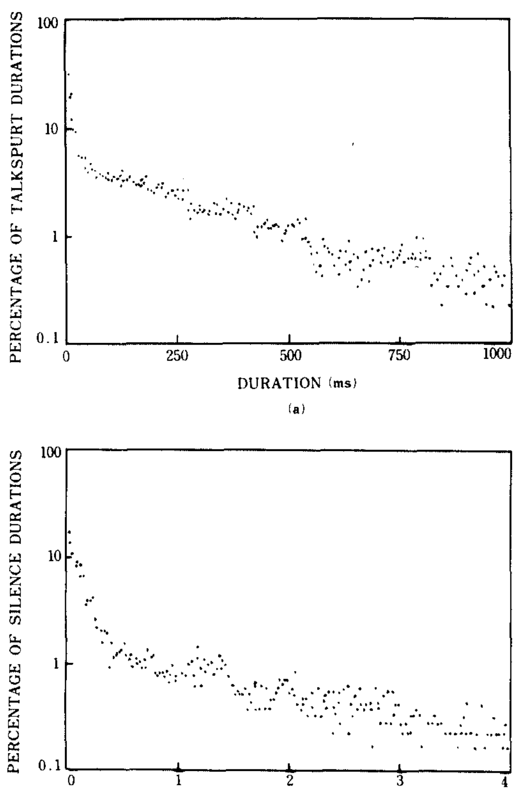

Fig. 5 Distributions of (a) talkspurt and (b) silice durations.

C. Software Structure

Our simulator simulates among others the access protocols for interactions of NNP-NNP, NNP-host, NNP-asynchronous terminal, and NNP-packet voice terminal. To account for those protocols, the simulator is made up of a number of modules. The software structure of the simulator developed is shown in Fig. 6. The X.25 interface is depicted by DMAIN which controls transitions of the state machine and processing of the synchronous channel data. The module VMAIN performs voice packet processing, and manages packet voice terminals. To discriminate a frame between voice and data frames the module ANAL is used. The module DRVER is a model of the hardware driver routine. For the network and transport layer protocols, the modules NETRX. NETTX and TRANS have been implemented. As for the access protocols for an asynchronous terminal, ARTRX and ARTTX which perform the X.28 have been implemented. Moreover, the PAD which follows the CCITT recommendation X.29 has also been implemented. In addition, routines for buffer interrupt, host processing, traffic generation and report generation have been realized.

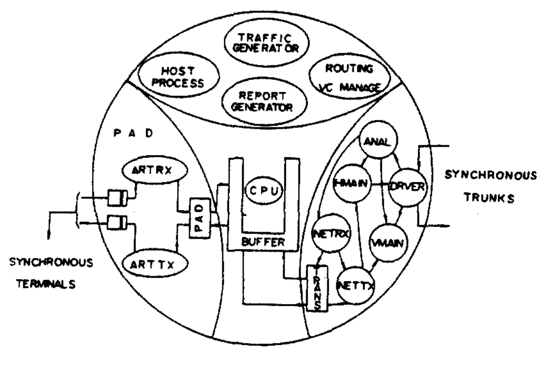

Fig. 6 Software structure of the simulator developed.

For voice, error recovery is not done at all layers. In the link layer, using the address field of a frame, voice frames are classified from data frames. Furthermore, voice frames are excluded from send and receive sequencing like the unnumbered frames described in the X.25. In this way, the flow control and the retransmission strategy for voice packets can be eliminated. The link layer protocol for voice packets described above are modeled in VMAIN. In the network layer, using bits in the general format identifier (GFI), voice packets are discriminated from data packets, and are stamped with a time information. Unlike the link layer, however, the send sequencing is done for the detection of packet slips of voice packets. The modules implemented in our simulator are summarized in table I.

Table 1 Modules implemented in the simulator

Module	Function
DRVER	Hardware driver routine for synchronous channel
ANAL	Analysis routine for input packets
DMAIN	Main routine of the link layer for data packets (X.25)
VMAIN	Main routine of the link layer for voice packets
NETRX	Receive routine of the network larer (X.25)
NETTX	Send routine of the network layer (X.25)
TRANS	Routine for transport layer
PAD	Packet assembly/disassembly routine (X.29)
ARTRX	Receive routine for asynchronous channel (X.28)
ARTTX	Send routine for asynchronous channel (X.28)
RHOST	Remote host process (X.25)
VCMGR	Routine for VC manager
DAGEN	Data traffic generation
SPEAK	Voice traffic generation

The simulator implemented in this work is executed in five steps as shown in Fig. 7. In the first step, speech is digitized by a 12-bit A/D converter, and stored in a magnetic tape. From this speech data file, speech statistics are obtained. These include the distributions of lengths of talkspurts and silence, and the transition probability of the interactive two-way conversation talker model. The default speech parameter

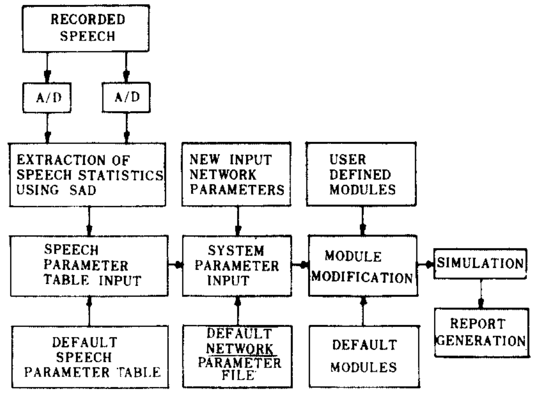

Fig. 7 Steps of simulation.

table that has been obtained from real speech conversations is provided. In the second step, new parameters which are not equal to the values provided for the default network parameters are accepted. Parameter lists that can be changed by the user are tabulated in Table II. The software for the first and second steps has been implemented in PL/I language for the interactability. In the third step, protocol modules are

Table 2 List of input parameters

Symbol	Function
XH1	Number of nodes
XH3	Unit buffer size in byte
XH4	Data packet length
XH13	Voice packet service priority
XH14	Voice packet length
XH17	Flow control window size
XH20	Total number of node buffer avaliable
X17	Encoding rate of speech
MESS1	Mean length of terminal message
MESS2	Mean length of respose messages from host
MLINE	Connectivity matrix for nodes
MX1-MX10	Line type and capacity
TBLRO	Routing table
FN1-FN10	Traffic matrix for data
CYCLE	One instruction cycle time of a node processor
HOSTD	Response delay for hosts
BERR	Bit error rate for lines
SIMTM	Total simulation time
COST	Cost table
RSEED	Initial seeds for 8-random number generators

updated from the user-defined modules and default protocol modules. Most of the user-defined modules have been programmded in GPSS V, but part of them have been programmed in PL/I. In the fourth step, the main program is executed. It runs for the time specified by the user. Finally, in the fifth step we obtain the simulation results.

D. Event Scanning

In a large simulation program, the sequencing of simultaneous events should be carefully considered. The simultaneous events make it difficult to interprete the real actions of a simulation model. For this reason, we have used a priority scheme to control the ordering of simultaneous events. For example, higher priority is given to voice packets in seizing the transmission channel than data packets. And for CPU, the buffer interrupt processing to allocate or free buffers has higher priority than the packet level processing or the acknowledgement processing.

Another aspect of consideration in handling the events is conditional events. The conditional events are a set of pending events for which further execution is blocked, because a certain condition has not been met. Hence, there is a chain effect, such that the execution of a certain event activates one or more events which in turn activate other conditional events at the same clock [6]. Therefore, the scaning of the conditional event list may be performed many times in the same clock time. However, this would increase computer run time considerably, especially with the GPSS simulation language [22]. Also, there may be the case that many events simultaneously try to seize one facility in vain. It is only time consuming to check the status of a facility when the facility has already been seized or preempted by other event. To alleviate these problems, we use a method in which once a frequently used facility is seized, every transaction pending to get the facility is placed in a user chain, but not to in active event list. When the facility becomes free, the next event that has been placed in a user chain seizes it according to a priority scheme or the first-come first-serving rule for equal priority. This method that has been implemented using the GPSS V LINK block reduces significantly the number of scans of the conditional event list in the simulator model. When the method is adopted, the simulation results have been proved to be almost equal to the one obtained from the conventional simulation model

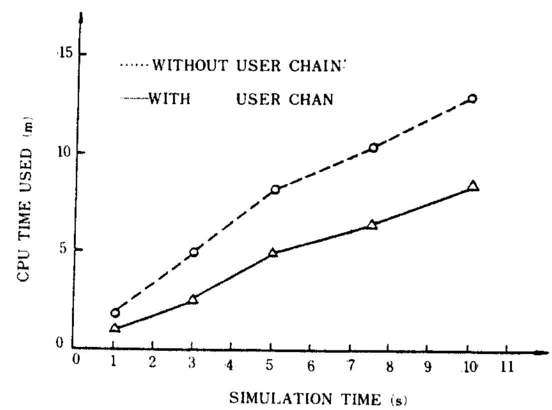

Fig. 8 Comparison of CPU time for two event scan methods when traffic is light ($\rho = 0.2$).

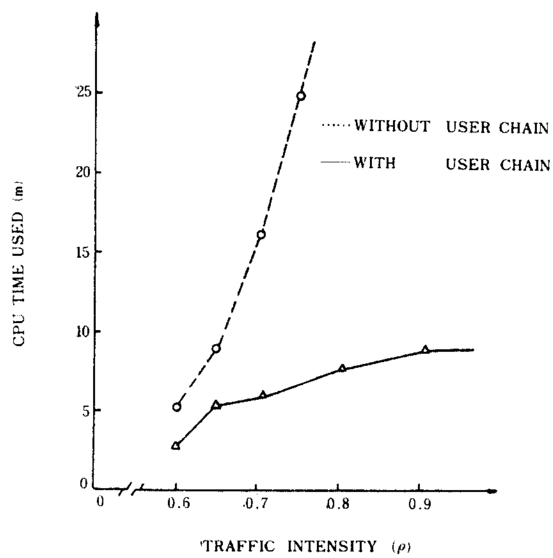

Fig. 9 Comparison of CPU time vs. traffic intensity for the two event scan methods. (simulation time = 3s).

without the user chain. With the method, the computer run time has been reduced by one fifth that normally required for the IBM/370 system when the traffic is heavy. The results are shown in Figs. 8 and 9. For a light traffic (i.e., traffic intensity (ρ)=0.2), the total CPU time used by the simulator was slightly reduced when the conditional events were reduced (see Fig. 8). For a heavy traffic, however, there was a large reduction in the total CPU time used (see Fig. 9). It is seen in the figure that, as the traffic intensity becomes higher (i.e., as the number of events becomes larger), the use of a user chain improves the speed of the simulator considerably.

E. Validation of the Simulator

To determine the validity of the simulator developed[3], we compare the performance results of a simple voice/data multiplexer obtained by the simulator and by analysis. As an analytical model for the integrated voice/data multiplexing system, a simple nonpreemptive priority queue has been used. The system model is depicted in Fig. 10. The time delay analysis of the non-

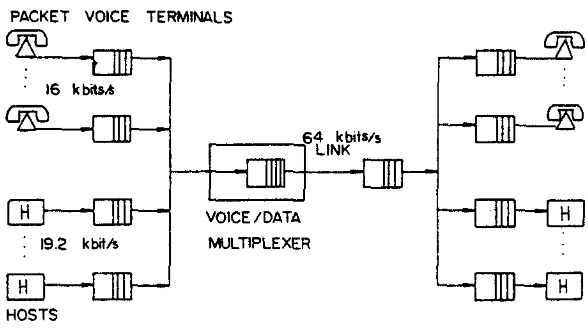

Fig. 10 Integrated voice / data multiplexing system.

preemptive priority queue can be done with the use of an imbedded Markov chain. Here, the packet arrival pattern for voice has been assumed to be Poisson. We have used the following analytical formula for average waiting time of voice and data obtained by Schwartz [23]:

$$E(W_v) = \frac{(\rho_v/\mu_v) + (\rho_d/\mu_d)}{1-\rho_v}$$

and

$$E(W_d) = \frac{(\rho_v/\mu_v) + (\rho_d/\mu_d)}{(1-\rho_v)(1-\rho_v-\rho_d)}$$

where E(w), ρ, $1/\mu$ are average wait time, traffic intensity and average service time, respectively; and the subscripts 'v' and 'd' indicate voice and data, respectively. The total end-to-end delay can be obtained by summing waiting time, service time, processing delay and packetization delay for voice. We assumed that the processing delay for voice and data packets were 5 and 15 ms, respectively. The packet sizes for voice and data were assumed to be 512 and 1024 bits, and the transmission speed for voice and data were 16 and 19.2 kbits/s, respectively. In addition, 10 ms was counted for modem delay.

In Fig. 11 the results for various traffic intensity obtained from analytical and simulation

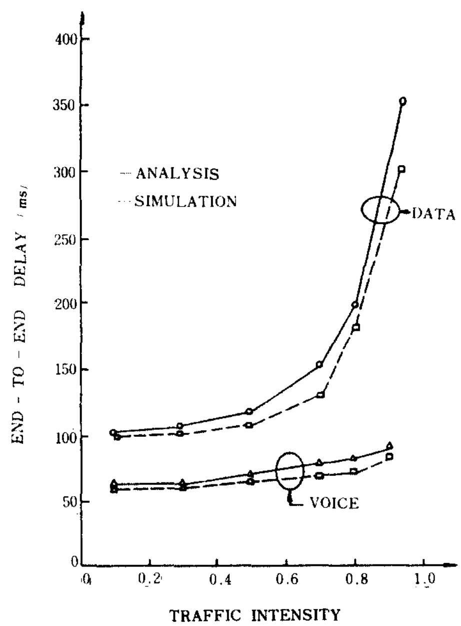

Fig. 11 End-to-end delay vs. traffic intensity.

[3]Nevertheless, this validation is by no means exhaustive.

models are compared. The simulation time for the model was 60 s. The results obtained by analysis and simulation agree closely, especially when the number of voice calls are increased under the equal offered traffic load, thus validating partially the simulator developed.

IV. RESULTS OF NETWORK SIMULATION

The simulator developed has been used to test the performance of an integrated voice/data packet network. The network model is shown in Fig. 12. Throughout the simulation, the numbers of nodes and links were fixed to 7 and 12, respectively. In addition, the following parameter values were used:

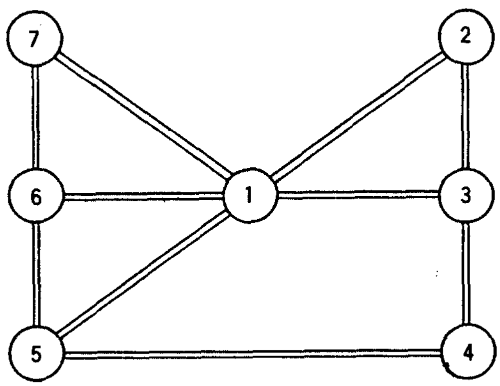

Fig. 12 Network configuration used in simulation.

N_v (Number of voice circuits) = 12,
N_d (Number of data circuits) = 100,
C (Link capacity) = 64 kbits/s,
B_v (Voice coding rate) = 16 kbits/s,
B_d (Bit rate of data terminals) = 2400 bits/s,
L_h (Packet header length) = 100 bits.

In the performance evaluation of a communication network, packet delay is one of the most important factors that must be considered. It is mainly affected by the offered traffic load, processing power of the node processor and other system parameters. Fig. 13 shows the effect of processing power of the NNP on the packet delay

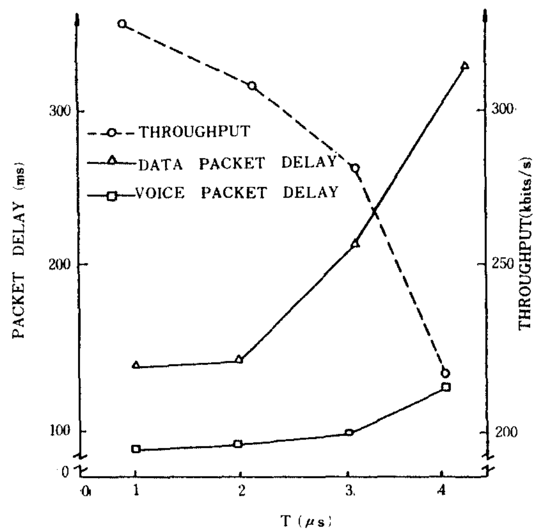

Fig. 13 Packet delay and throughput vs. time (T) for one instruction cycle of CPU.

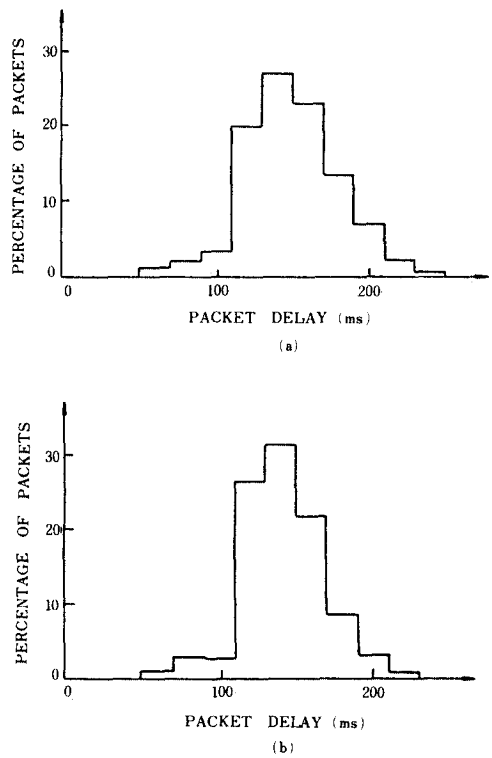

Fig. 14 Delay distributions of data packets for different traffics. (a) $\rho = 0.52$ (b) $\rho = 0.34$

and the throughput under the fixed traffic condition. In the figure, with the decrease of the processing power, the rate of increase for data packet delay is larger than that of a voice packet. This is because higher priority has been given to the voice packets in seizing the CPU for packet processing. In addition, Fig. 14 shows the effect of the offered traffic load (ρ=0.34, 0.52) on the data packet delay. In this figure, one can see that the spread of the packet delay distribution becomes wider with the increasing traffic.

As mentioned previously, since packet voice communication requires a stringent restriction in packet delay, priority is generally given to voice packets. We compared the following two cases. The first case is that higher priority is given to voice packets, and the second case is that prority is the same for data and voice packets. The result is shown in Fig. 15. When the priority is given to voice packets, the packet delay for voice can be maintained nearly constant at the cost of some increase of data packet delay. When the traffic load is small, the results of the two cases appear to be almost the same. In the simulation, the processing time for voice packets was assumed to be one third of that for data packets.

Another factor that affects the packet delay is the size of a flow control window. In our simulator, an entry-to-exit layer flow control has been implemented so that no more than k (k is the size of a flow control window) packets can be transmitted without an acknowledgement from the exit node to the entry node for each data circuit [24]. Thus, the total traffic in the network is maintained not to exceed a certain level. The data packet delay in the uncontrolled (i.e., without flow control) system is much larger than that in the controlled system. Also, it has been found that voice packet delay appears to be little affected by having flow control of data signal. But, the portion of voice packets that are delayed more than 300 ms increases, when the flow control for data signal is not done. This result indicates that the flow control for data packets is slightly helpful for voice packets as well. Fig. 16 shows the power (throughput/delay) for

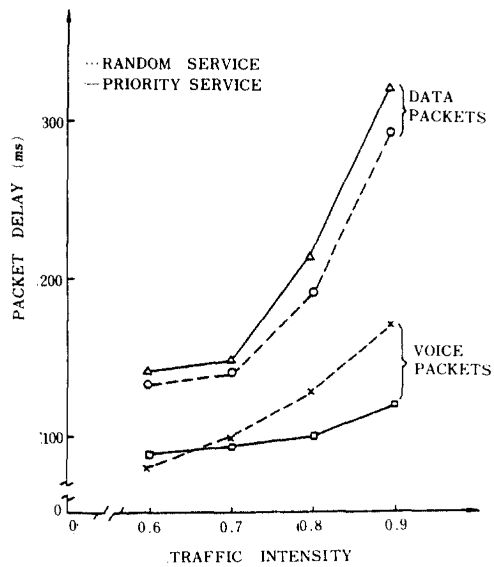

Fig. 15 Packetdelay vs. traffic intensity for voice and data packets.

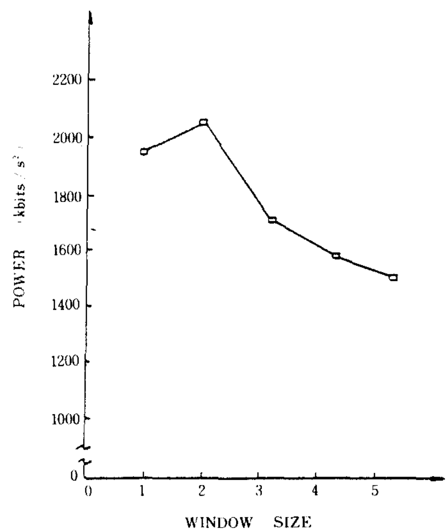

Fig. 16 Power vs. window size.

various window sizes. It is seen in the figure that, when the window size is 3, the power becomes maximum. In this case the mean hops for packets were 1.8. According to the result of the GMDNET simulation experiments, it is known that the optimal window size which depends on packet length and other system parameters is given by the mean hops of packets plus 1 [24]. In the simulated system, however, because of the existence of voice packets, the optimal size of window is slightly smaller than the value obtained by the GMD simulation. In Table III, we summarize the simulation results of throughput, delay and power for various sizes of the window.

Table 3 Throughput, delay and power for various window sizes

Window size	Throughput (kbits/s)	Data packet Delay (ms)	Voice packet Delay (ms)	Power (kbits/s²)
1	320	162.5	78.3	1969
2	360	178.9	80.5	2012
3	387	227.0	83.9	1704
4	394	251.0	87.1	1570
5	422	281.3	92.2	1500

Another parameter that affects the network performance is the packet size. In general, the smaller the packet size, the smaller the delay. Also, the throughput becomes smaller due to the packet overhead. Hence, in determining the packet size, various parameter values including the average message length and the desired throughput and delay should be considered. Fig. 17 shows the performance of a 7-node network for different packet sizes.

The voice packet delay is commonly expressed as the sum of packetization delay and link delay due to the packet overhead [25]. Hence, the voice packet size should be selected such that the sum of the two delay elements is minimized. Fig. 18 shows the end-to-end delay of voice packets for different sizes of a voice packet. In getting the result, the bit rate of voice signal was assumed to be 16 kbits/s, and the traffic intensity was 0.7. In this case, the optimal size of a voice packet appears to be about 500 bits.

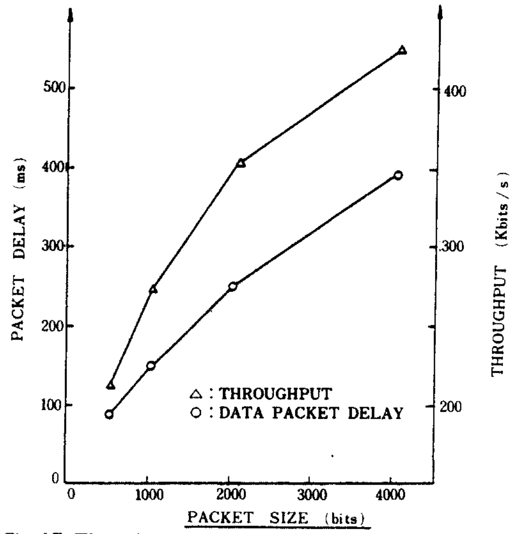

Fig. 17 Throughput and delay for different data packet sizes.

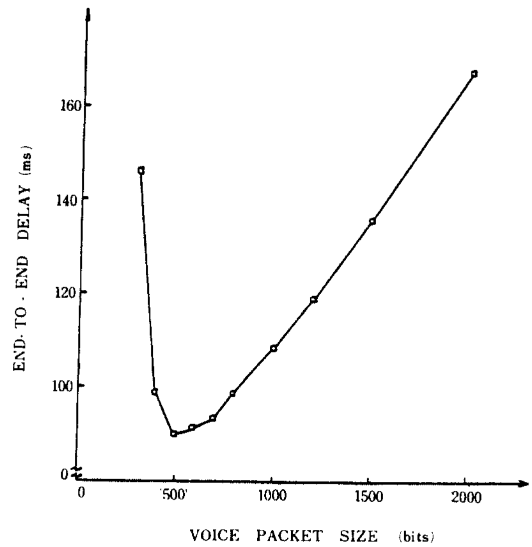

Fig. 18 End-to-end delay vs. voice packet size (ρ = 0.7)

V. CONCLUSIONS

We have examined various aspects of developing a simulator for an integrated voice/data packet communications network. The simulator developed has modeled packet voice protocols and voice traffic generation as well as data transmission protocols. Thus, it can simulate a network that includes packet voice terminals as well as data terminals and hosts.

In our simulator, we have chosen the process interaction approach because, by using the method, we could implement the simulator that is related most closely with real systems. The method chosen requires fairly long execution time, however. To reduce the computer run time, we have used the method of reducing the conditional events on the basis of GPSS V LINK block. The resulting run time could be reduced by a factor of one fifth for the case of heavy traffic. The simulator has been implemented mostly in GPSS V simulation language and partly in PL/I, resulting in a software package of about 4,000 lines.

The voice signal used in this simulator was generated in the form of on-off patterns using the two cumulative functions which have been obtained from the on-off statistics of real conversation that is 50 min long. By using the simulator on the IBM/370, the performance of a network with 7 nodes and 12 links has been studied for various circumstances and parameter values.

For validation of the simulator implemented, we constructed a simulation model for a voice/data multiplexer and compared the simulation results with those obtained from the analytical model.

PEFERENCES

(1) C. J. Weinstein and J. W. Forgie, "Experiences with speech communication in packet networks," IEEE J. select. Areas Commun., vol. SAC-1, No. 6, pp. 963-980, Dec. 1983.

(2) T. Bially, A. J. McLaughlin, and C. J. Weinstein, "Voice communication in integrated digital voice and data networks," IEEE Trans. Commun., vol. COM-28, pp. 1478-1490, Sept. 1980.

(3) G. J. Coviello and P. A. Vena, "Integration of circuit / packet switching by a SENET concept," Proc. National Telecommunications Conference, pp. 42-12-42-17, 1975.

(4) J. W. Forgie and A. G. Nemeth, "An efficient packetized voice / data network using statistical flow control," Conf. Rec., Int. Conf. on Commun., pp. 38.2.44-38.2.58, June 1977.

(5) D. Minoli, "Issues in packet voice communication," Proc. IEE, vol. 126, No. 8, pp. 729-740, Aug. 1979.

(6) G. Gordon, System Simulation, Englewood Cliffs, NJ.: Prentice-Hall, 1977.

(7) G. M. Scheider, "A modeling package for simulation of computer networks," Simulation, vol. 30, No. 12, pp. 181-192, Dec. 1978.

(8) I. Chlamtac and W. R. Franta, "A generalized simulator for computer networks," Simulation, vol. 34, No. 10, pp. 123-132, Oct. 1982.

(9) C. H. Sauer, E. A. MacNair and J. F. Kurose "Queueing network simulations of computer communiactions," IEEE J. Select. Areas Commun., vol. SAC-2, No. 1, pp. 203-219, Jan. 1984.

(10) P. E. Green, Jr., "An introduction to network architectures and protocols," IEEE Trans. Commun., vol. COM-28, No. 4, pp. 413-424, Apr. 1980.

(11) H. Zimmermann, "OSI reference model-the ISO mdel of architecture for open·systems interconnection," IEEE Trans. Commun., vol. COM-28, No. 4, pp. 425-432, Apr. 1980.

(12) G. S. Bhusri, "Considerations for ISDN planning and implementation," IEEE Communications Magazine, vol. 22, No. 1, pp. 18-32, Jan. 1984.

(13) H. M. Heggestad and C. J. Weinstein, "Voice and data communications experiments on a wideband satellite / terrestrial internetwork system, " Proc. Int. Conf. Commun., vol. 1, pp. A.1.1.1-A.1.1.8, Boston, MA, June 1983.

(14) P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive quantization in differential PCM coding of speech," Bell Syst. Tech. J. vol. 52, pp. 1105-1118, Sept. 1973.

(15) C. K. Un and H. S. Lee, "A study of the comparative performance of adaptive delta modulation systems,"IEEE Trans. Commun., vol. COM-28, No. 1, pp. 96-101, Jan. 1980.

(16) C. K. Un and D. T. Magill, "The residual-excited linear prediction vocoder with transmission rate below 9.6 kbits / s," IEEE Trans. Commun., vol. COM-23, No. 12, pp. 1466-1474, Dec. 1975.

(17) J. D. Markel and A. H. Gray, Jr., Linear Prediction of Speech. Berlin: Springer-Verlag, 1976.

(18) K. Bullington and J. Fraser, "Engineering aspects of TASI," Bell Syst. Tech. J., vol. 38, pp. 353-363, Mar. 1959.

(19) J. Martin, Systems Analysis for Data Transmission, Englewood Cliffs, NJ.: Prentice-Hall, 1972.

(20) P. T. Brady, "A model for generating on-off speech patterns in two-way conversation," Bell Syst. Tech. J., vol. 48, No. 7, pp. 2445-2472, Sept. 1969.

(21) J. G. Gruber, "A comparison of measured and calculated speech temporal parameters relevant to speech activity

detection," IEEE Trans. Commun., vol. COM-30, No. 4, pp. 728-738, Apr. 1982.

(22) General Purpose Simulation System V, IBM Corp., 1977.

(23) M. Schwartz, Computer Communication Network Design and Analysis, Englewood-Cliffs, NJ : Prentice-Hall, 1977.

(24) M. Gerla and L. Kleinrock, "Flow control : a comparative survey," IEEE Trans. Commun., vol. COM-28, No, 4, pp. 553-574, Apr. 1980.

(25) D. Minoli, "Optimal packet length for packet voice communication," IEEE Trans. Commun., vol. COM-27, No. 3, pp. 607-611, Mar. 1979.

朴 淳(Soon PARK) 正會員
1959년 8월 4일생
1982년 : 서울대학교 전자공학과 졸업(B.S)
1984년 : 한국과학기술원 전기및 전자공학과 졸업(M.S)
1984년～현재 : 한국과학기술원 전기및 전자공학과 박사과정
1984년～현재 : 금성전기(주) 연구소 연구원
관심분야 : computer networking, Data communication, Military communication.

殷 鍾 官(Chong Kwan UN) 正會員
1940年 8月25日生
1964. 6 : 미국University of Delaware 電子工學科卒業, 電子工學學士 學位
1966. 6 : 同大學院卒業, 電子工學碩士學位
1969. 6 : 同大學院卒業, 電子工學博士學位
1969. 9～1973. 5 : 美國University of Maine電子工學助教授
1973. 5～1977. 6 : 美國스텐포드研究所(SRI) 責任研究員
1977. 6～現在 : 韓國科學技術院電氣및電子工學科 教授, 本學會 理事
研究分野 : 디지탈通信, 디지탈信號處理