SPECIFICATIONS FOR THE

NETWORK VOICE PROTOCOL (NVP)

and

Appendix 1: The Definition of Tables-Set-#1 (for LPC)

Appendix 2: Implementation Recommendations

NSC NOTE 68

(Revision of NSC Notes 26, 40, and 43)

Danny Cohen, ISI

January 29, 1976

CONTENTS

PREFACE

The major objective of ARPA’s Network Secure Communications (NSC) project is to develop and demonstrate the feasibility of secure, high-quality, low-bandwidth, real-time, full-duplex (two-way) digital voice communications over packet-switched computer communications networks. This kind of communication is a very high priority military goal for all levels of command and control activities. ARPA’s NSC projrct will supply digitized speech which can be secured by existing encryption devices. The major goal of this research is to demonstrate a digital high-quality, low-bandwidth, secure voice handling capability as part of the general military requirement for worldwide secure voice communication. The development at ISI of the Network Voice Protocol described herein is an important part of the total effort.

ACKNOWLEDGMENTS

The Network Voice Protocol (NVP), implemented first in December 1973, and has been in use since then for local and transnet real-time voice communication over the ARPANET at the following sites:

- o Information Sciences Institute, for LPC and CVSD, with a PDP-11/45 and an SPS-41.
- o Lincoln Laboratory, for LPC and CVSD, with a TX2 and the Lincoln FDP, and with a PDP-11/45 and the LDVT.
- o Culler-Harrison, Inc., for LPC, with the Culler-Harrison MP32A and AP-90.
- o Stanford Research Institute, for LPC, with a PDP-11/40 and an SPS-41.

The NVP’s success in bridging the differences between the above systems is due mainly to the cooperation of many people in the ARPA-NSC community, including Jim Forgie (Lincoln Laboratory), Mike McCammon (Culler-Harrison), Steve Casner (ISI) and Paul Raveling (ISI), who participated heavily in the definition of the control protocol; and John Markel (Speech Communications Research Laboratory), John Makhoul (Bolt Beranek & Newman, Inc.) and Randy Cole (ISI), who participated in the definition of the data protocol. Many other people have contributed to the NVP-based effort, in both software and hardware support.

1. INTRODUCTION

Currently, computer communication networks are designed for data transfer. Since there is a growing need for communication of real-time interactive voice over computer networks, new communication discipline must be developed. The current HOST-to-HOST protocol of the ARPANET, which was designed (and optimized) for data transfer, was found unsuitable for real-time network voice communication. Therefore this Network Voice Protocol (NVP) was designed and implemented.

Important design objectives of the NVP are:

- Recovery of loss of any message without catastrophic effects. Therefore all answers have to be unambiguous, in the sense that it must be clear to which inquiry a reply refers.

- Design such that no system can tie up the resources of another system unnecessarily.

- Avoidance of end-to-end retransmission.

- Separation of control signals from data traffic.

- Separation of vocoding-dependent parts from vocoding-independent parts.

- Adaptation to the dynamic network performance.

- Optimal performance, i.e. guaranteed required bandwidth, and minimized maximum delay.

- Independence from lower level protocols.

The protocol consists of two parts:

(1) The control protocol,

(2) The data protocol.

Control messages are sent as controlled (TYPE 0/0) messages, and data messages may be sent as either controlled (TYPE 0/0) or uncontrolled (TYPE 0/3) messages (see BBN Report 1822 for definition of MESSAGE-TYPE).

Throughout this document a "word" means a "16-bit quantity".

2. THE CONTROL PROTOCOL

Throughout this document the 12-bit MESSAGE-ID (see BBN Report 1822) is referred to as LINK (its 8 MSBs) and SUB-LINK (its 4 LSBs).

The control protocol starts with an initial connection phase on link 377 and continues on other links assigned at run time.

Four links are used for each voice communication:

```
Link L    will be used for control, from CALLER to ANSWERER.
Link K    will be used for control, from ANSWERER to CALLER.
Link L+1  will be used for data,    from CALLER to ANSWERER.
Link K+1  will be used for data,    from ANSWERER to CALLER.
```

Both L and K should be between 340 and 375 (octal). L and K need not differ.

The first message (CALLER to ANSWERER) on link 377 indicates which user wants to talk to whom and specifies K. As a response (on K), the ANSWERER either refuses the call or accepts it and assigns L.

The CALLER then calls again (this time on link L). The ANSWERER initiates a negotiation session to verify the compatibility of the two parties.

The negotiation consists of suggestions put forth by one of the parties, which are either accepted or rejected by the other party. The suggesting party in the negotiation is called the NEGOTIATION MASTER. The other party is called the NEGOTIATION SLAVE. Usually the ANSWERER is the negotiation master, unless agreed otherwise by the method described later.

If the negotiation fails, either party may terminate the call by sending a "GOODBYE". If the negotiation is successfully ended, the ANSWERER rings bells to draw human attention and sends "RINGING" to the CALLER. When the call is answered (by a human), a "READY" is sent to the CALLER and the data starts flowing (on L+1 and K+1). However, a "READY" can be sent without a preceeding "RINGING".

This bell ringing occurs only after the initial call (not after renegotiation).

The assignment of L and K cannot be changed after the initial connection phase.

Only one control message can be sent in a network-message. Extra bits needed to fill the network-message are ignored.

The length of control messages should never exceed a single-packet (i.e., 1,007 data bits).

Control messages not recognized by their receiver should be ignored and should not cause any error condition resuting in termination of the connection. These messages may result from differences in implementation level between systems.

SUMMARY OF THE CONTROL MESSAGES

#1	"1,<WHO>,<WHOM>,K"
#2	"2,<CODE>" or only "2"
#3	"3,<WHAT>,<N>,<HOW(1),...HOW(N)>"
#4	"4,<WHAT>,<HOW>"
#5	"5,<WHAT>,<HOW>" or only "5,<WHAT>"
#6	"6,L" or only "6"
#7	"7"
#8	"8"
#9	"9"
#10	"10,<ID>"
#11	"11,<ID>"
#12	"12,<IM>"
#13	"13,<YM>,<OK>"

DEFINITION OF THE CONTROL MESSAGES

#1 CALLING (on 377 and L)

This call is issued first on link 377 and later on link L. Its format is "1,<WHO>,<WHOM>,K", where <WHO> and <WHOM> are words which identify respectively the calling party and the party that is being called, and K is as defined above. The format of the <WHO> and <WHOM> is:

(HHIIIIIIXXXXXXXX)

where HH are 2 bits identifying the HOST, followed by 6 bits identifying the IMP, followed by 8 bits identifying the extension (needed because there may be more than one communication unit on the same HOST).

The system which sends this message is defined as the CALLER, and the other system is defined as the ANSWERER.

#2 GOODBYE (TERMINATION, on L or K)

This message has the purpose of terminating calls at any stage.

ICP can be terminated (on K) either negatively by sending either a single word "2" ("GOODBYE") or the two words "2,<CODE>", or positively by sending the two words "6,L", as described later.

After the initial connection phase, calls can be terminated by either the CALLER (on L) or the ANSWERER (on K). This termination has two words: "2,<CODE>", where <CODE> is the reason for the termination, as specified here:

0. Other than the following.

1. I am busy.

2. I am not authorized to talk with you.

3. Request of my user.

4. We believe you are down.

5. Systems incompatibility (NEGOTIATION failure).

6. We have problems.

7. I am in a conference now.

8. You made a protocol error.

#3 NEGOTIATION INQUIRY (on L or K)

Sent by the NEGOTIATION MASTER for compatibility verification. The format is:

"3,<WHAT>,<LIST-LENGTH>,<HOW-LIST>", meaning

"CAN-YOU-DO,<WHAT>,<LIST-LENGTH>,<HOW-LIST>".

The <HOW-LIST> is a list of pointers into agreed-upon tables, as shown below.

#4 POSITIVE NEGOTIATION RESPONSE (on L or K)

Sent by the NEGOTIATION SLAVE in response to a NEGOTIATION INQUIRY. The format is:

"4,<WHAT>,<HOW>", meaning: "I-CAN-DO,<WHAT>,<HOW>".

#5 NEGATIVE NEGOTIATION RESPONSE (on L or K)

Sent by the NEGOTIATION SLAVE in response to a NEGOTIATION INQUIRY. The format is either:

"5,<WHAT>,0", meaning "I-CAN’T-DO-<WHAT>-IN-ANY-OF-THESE-WAYS",

or: "5,<WHAT>,N", meaning inability to accept any of the options offered in the INQUIRY, but using "N" as a suggestion to the ANSWERER about another possibility. Examples are presented later in this report.

#6 READY (on L or K)

Sent by either party to indicate readiness to accept data. Its format is "6,L" in the reply to the initial call, and "6" thereafter.

#7 NOT READY (on L or K)

Sent by either party to indicate unreadiness to accept data. It is always a single word: "7".

#8 INQUIRY (on L or K)

Sent by either party to inquire about the status of the other. It is always a single word: "8". It is answered by #6, #7, or #9.

#9 RINGING (on K)

Sent by the ANSWERER after the negotiations have been successfully terminated and human permission is needed to proceed further. The ringing will continue for 10 seconds, and then stop, UNLESS a #8 is received. This message is always a single word: "9".

#10 ECHO REQUEST (on L or K)

Sent by whichever party is interested in measuring the network delays. Its only purpose is to be echoed immediately. The format is "10,<ID>", where <ID> is any word used to identify the ECHO.

#11 ECHO (on L or K)

Sent in response to ECHO REQUEST. The format is "11,<ID>", where <ID> is the word specified by #10. The implementation of this feature is not compulsory, and no connection should be terminated due to lack of response to ECHO-REQUEST.

#12 RENEGOTIATION REQUEST (on L or K)

Can be sent by either party at ANY stage after LINKS are agreed upon. This message consists of the two words "12,<IM>". If the word <IM> (for I MASTER) is non-zero, the sender of this message requests to be the NEGOTIATION MASTER. If it is zero, the receiver of this message is requested to be the NEGOTIATION MASTER. Renegotiation is described later.

#13 RENEGOTIATION APPROVAL (on L or K)

This message may be sent by either party in response to RENEGOTIATION REQUEST. It consists of the three words "13,<YM>,<OK>". If <OK> is non-zero, this is a positive acknowledgment (approval). If it is zero, this is a negative acknowledgment (i.e., refusal). <YM> is set to be equal to the <IM> of #12, for identification purposes.

Messages #7, #8, and #9 are always a single word. Messages #1, #3, #4, and #5 are several words long. Messages #2 and #6 are either a single word or two words long. #10, #11 and #12 are always 2 words long. Message #13 is always 3 words long. Message #1 is always 4 words long.

Message #1 is sent only by the CALLER, #3 only by the NEGOTIATION MASTER, and #4 and #5 only by the NEGOTIATION SLAVE. Message #9 is sent only by the ANSWERER. All the other control messages may be sent by either party.

The last <HOW> which was both suggested by the NEGOTIATION MASTER (in #3) and accepted by the NEGOTIATION SLAVE (in #4) for each <WHAT> is assumed to be in use.

DEFINITION OF THE <WHAT> AND <HOW> NEGOTIATION TABLES:

<WHAT>	<HOW>
1. VOCODING	* 1. LPC + 2. CVSD 3. RELP 4. DELCO
2. SAMPLE PERIOD (in microseconds)	N. N (*150) (+62)
3. VERSION	* 1. V1 (see definition below) + 2. V2 (see definition below)
4. MAX MSG LENGTH (in bits) NVP header included (32 bits) but not HOST/IMP leader and not HOST/IMP padding	N. N (*976 and +976)
5. If LPC: Degree	N. For N coefficients (*10)
If CVSD: Time Constant (in milliseconds)	N. N (+50)
6. Samples per Parcel	N. N (*128) (+224)
7. If LPC: Acoustic Coding	* 1. SIMPLE (see below) 2. OPTIMIZED
8. If LPC: Info Coding	* 1. SIMPLE (see below) 2. OPTIMIZED

```
9. If LPC:

   Pre-emphasis                 N. N (*58, for
   1 - mu x [Z**-1]               mu = 58/64 = 0.90625)
   N = 64 x mu

10. If LPC:

   Table-set                    N. N (*1)
                                   See definition of Set #1
                                   in Appendix 1
```

(* indicates recommended options for LPC)
(+ indicates recommended options for CVSD)

No parameter (<WHAT>) should be inquired about by the NEGOTIATION MASTER if some option (<HOW>) for it has been previously accepted by the NEGOTIATION SLAVE implicitly in the "VERSION". The purpose of this restriction is to avoid a possible conflict between individual parameters and the VERSION-option.

Version 1 (V1) is defined as:

1-1	LPC
2-150	150 microseconds sampling
3-1	V1
5-10	10 coefficients
6-128	128 samples per parcel
7-1	SIMPLE acoustic coding
8-1	SIMPLE information coding
9-58	mu = 58/64 = 0.90625
10-1	Tables set #1

Version 2 (V2) is defined as:

1-2	CVSD
2-62	62 microseconds sampling (16 KHz sampling)
3-2	V2
5-50	50 msec time constant
6-192	192 samples per parcel

Note that this defines every negotiated parameter, except MAX MSG LENGTH.

SIMPLE and OPTIMIZED codings will be described below in Section 3.

All the negotiation is managed by the NEGOTIATION MASTER, who decides how much negotiation is needed, and what to do in case some discrepancy (incompatibility) is discovered: either to try alternative options or to abort the connection. Upon completion of successful negotiation, the NEGOTIATION MASTER sends either #9 (RINGING) only if it is the ANSWERER and if this is an initial connection, else it sends #6 (READY-FOR-DATA), and probably inquires with #8 about the readiness of the other party. The inquiries (#8) before the successful completion of the negotiation are ignored. However, these inquiries after the first RINGING (#9) and before the first READY (#6) are needed to keep the ANSWERER ringing.

Note that the negotiation process can be shortened by using the VERSION option, as shown in the examples that follow.

ON RENEGOTIATION

At any stage after links are agreed upon, either party might request a RENEGOTIATION. If the request is approved by the other party, either party might become the NEGOTIATION MASTER, depending on the type of renegotiation request. When renegotiation starts, no previously negotiated agreements (except LINK numbers) hold, and all items have to be renegotiated from scratch. Note that renegotiation may entirely replace the negotiation phase and allows the CALLER to be the NEGOTIATION MASTER.

Upon issuance (or reception) of RENEGOTIATION REQUEST, all data messages are ignored until the positive indication of the successful completion of the renegotiation (#6).

After the completion of renegotiation, the frame-count (see the section on MESSAGE-HEADER) may be reset to zero.

THE HEADER OF DATA MESSAGES

Data messages are the messages which contain vocoded speech. The first 32 bits of each data message is the MESSAGE-HEADER, which carries sequence and timing information as described below.

For each vocoding scheme a "FRAME" is defined as the transmission interval (as agreed upon at the negotiation stage in <WHAT#6>). Since this interval is defined by the number of samples, its duration can be found by multiplying the sampling period <WHAT#2> by the interval length (in samples) <WHAT#6>. For example, in V1 the sampling period is 150 microseconds and the transmission interval is 128 samples, which yields:

128*150 microseconds = 19.2 milliseconds.

The data describing a FRAME is called a PARCEL. Each parcel has a serial number. The first parcel created after the completion of the negotiation (or every RENEGOTIATION) has the serial number zero. Each message contains an integral number of parcels.

The serial number of the first parcel in the message is put in the first 16 bits of the message and is referred to as the MESSAGE-TIME-STAMP. Note that this time stamp is synchronized with the data stream. Note also that these 16 bits are actually the third word of the message, following the 2 words used as IMP-to-HOST leader (see BBN Report 1822).

The next bit in the header is the WE-SKIPPED-PARCELS bit, which is described later. The next 7 bits tell how many parcels there are in the message; this number is called the COUNT, or the PARCEL-COUNT.

Note that if message number N has the time stamp T(N) and the count C(N), then T(N+1) must be greater than or equal to T(N)+C(N). Usually T(N+1) = T(N)+C(N), unless the XMTR decided not to send some parcels due to silence. If this happens then the WE-SKIPPED-PARCELS bit is set to ONE, else it is set to ZERO. Hence, if T(N+1) is found by the RCVR to be greater than T(N)+C(N) and the WE-SKIPPED-PARCELS is zero, some message must be lost.

Note that by definition the time stamps on messages monotonically increase, except for wrap-around.

The message header structure is illustrated by the following diagram:

```
       WORD 1           WORD 2           WORD 3          WORD 4
!................!................!................!................!...
!P000TTTTHHIIIIII!LLLLLLLLZZZZZZZZ!TTTTTTTTTTTTTTTT!WCCCCCCCSSSSSSSS!DDD
!................!................!................!^...............!...
!<--HOST/IMP-OR-IMP/HOST-LEADER-->!<--TIME-STAMP-->!^<COUNT><-SAVE->!<-D
                                                    ^
                                           WE-SKIPPED-PARCELS
```

```
P = PRIORITY (one bit = 1)
T = MESSAGE TYPE (4 bits = 0011)
L = link ("L" OR "K", 8 bits, greater than 337 octal)
D = data bits (from here to the end of the message)

ZZZZZZZZ = 8 ZERO bits
HHIIIIII = HOST (8 bits, destination or source)
CCCCCCC = parcel COUNT (7 bits)
SSSSSSSS = 8 bits saved for future applications
TTTTTTTTTTTTTTTT = TIME STAMP (16 bits)
```

The first parcel sent by either party after the NEGOTIATION or RENEGOTIATION should have the serial number set to zero.

During silence periods, the XMTR might send a "6" or "7" message periodically. If it does not do so, the RCVR might interrogate the livelihood of the XMTR by sending periodically "8" ("ARE-YOU-THERE?") or #10 (ECHO-REQUEST) messages.

3. THE LPC DATA PROTOCOL

The DATA sent at each transmission interval is called a PARCEL.

Network messages always contain an integral number of PARCELs.

There are two independent issues in the coding. One is, obviously, the acoustic coding, i.e., which parameters have to be transmitted. SIMPLE acoustic coding is sending all the parameters at every transmission interval. OPTIMIZED acoustic coding sends only as little as acoustically needed. DELCO is an example of OPTIMIZED acoustic coding.

In this document only the format of the SIMPLE acoustic coding is defined.

All the transmitted parameters are sent as pointers into agreed-upon tables. These tables are defined as two lists of values. The transmitter table {X(J)} is used in the following way: The value V is coded as the code J if X(J-1) < V =< X(J). The receiver table {R(J) is used to retrieve the value R(J) if the code J was received. X(-1) is implicitly defined as minus-infinity, and X(Jmax) is explicitly defined as plus-infinity.

For each parameter, {X(J)} and {R(J)} may be defined independently.

The second coding issue is the information coding technique. The SIMPLE (information-wise) way of sending the information is to use binary coding for the codes representing the parameters. The OPTIMIZED way is to compute distributions for each parameter and to define the appropriate coding. It is very probable that the PITCH and GAIN will be decoded absolutely in the first PARCEL of each message, and incrementally thereafter.

At present, only the SIMPLE (information-wise) coding is used.

The details of the LPC data protocol and its Tables-Set-#1 can be found in Appendix 1.

Following is the definition for the format of the SIMPLE-SIMPLE coding, according to Tables-Set-#1:

For each parcel:

PITCH	6 bits	(PITCH=0 for UNVOICED)
GAIN	5 bits	
I(1)	7 bits	
I(2)	7 bits	
I(3)	6 bits	
I(4)	6 bits	
I(5)	5 bits	
I(6)	5 bits	
I(7)	5 bits	
I(8)	5 bits	
I(9)	5 bits	
I(10)	5 bits	

where each of the I(j) is an index for inverse sine coding. If K(j)=arcsin(Theta(j)) and N bits are assigned for its transmission, then I(j)=(Theta(j)/Pi)*2**N.

Hence at each transmission interval (128 samples times 150 microseconds) 67 bits are sent, which results in a data rate of 3490 bps. Since this bandwidth is well within the capabilities of the network, SIMPLE-SIMPLE coding is used, which requires the least computation by the hosts. Note that this data rate is a peak rate, without the use of silence.

4. EXAMPLES FOR THE CONTROL PROTOCOL

Here is an example for a connection:

```
(377)  C: 1,<WHO>,<WHOM>,340    Please talk to me on 340/341.

(340)  A: 2,1                   I refuse, since I’m busy.
```

Another example:

```
(377)  C: 1,<WHO>,<WHOM>,360    Please talk to me on 360/361.

(360)  A: 6,350                 OK.  You talk to me on 350/351.

(350)  C: 1,<WHO>,<WHOM>        I want to talk to you.

(360)  A: 3,1,1,2               Can you do CVSD?  (ANSWERER tries
                                to be the NEGOTIATION MASTER)

(350)  C: 12,1                  I want to be it.

(360)  A: 13,1                  That’s OK with me.

(350)  C: 3,1,1,2               Can you do CVSD?

(360)  A: 5,1,1                 No, but I can do LPC.

(350)  C: 3,1,1,3               Can you do RELP?

(360)  A: 5,1,1                 No, but I can do LPC.

(350)  C: 3,1,1,1               How about LPC?

(360)  A: 4,1,1                 LPC is fine with me.

(350)  C: 3,2,1,150             Can you use 150 microseconds
                                sampling?

(360)  A: 4,2,150               I can use 150 microseconds.

(350)  C: 3,4,3,976,1040,2016   Can you use 976, 1040, or 2016
                                bits/msg?

(360)  A: 4,4,976               I can use 976.

(350)  C: 3,5,1,10              Can you send 10 coefficients?

(360)  A: 4,5,10                I can send 10.
```

```
      (350)  C: 3,6,1,64              Can you use a 64 sample
                                      transmission?

      (360)  A: 4,6,64                I can use 64.

      (350)  C: 3,7,2,1,2             SIMPLE or OPTIMIZED acoustic
                                      coding?

      (360)  A: 4,7,2                 OPTIMIZED!

      (350)  C: 3,8,1,1               Can you do SIMPLE info coding?

      (360)  A: 4,8,1                 I can do SIMPLE.

      (350)  C: 3,9,1,58              mu = 0.90625?

      (360)  A: 4,9,58                Fine with me.

      (350)  C: 3,10,1                Table set #1?

      (360)  A: 4,10,1                Of course!

      (350)  C: 6                     I am ready.  (Note:  No "RINGING"
                                      sent)

      (350)  C: 8                     And you?

      (360)  A: 6                     I am ready, too.

         .......                      Data is exchanged now,

         .......                      on 351 and 361.

      (350)  C: 10,1234               Echo it, please.

      (360)  A: 11,1234               Here it comes!

         .......

      (360)  A: 10,3333               Now ANSWERER wants to measure

      (350)  C: 11,3333               ...the delays, too.

         .......

      (???)    X: 2,3                 Termination by either user.
```

Another example:

```
   (377)  C: 1,<WHO>,<WHOM>,360    Please talk to me on 360/361.

   (360)  A: 6,340                 Fine.  You send on 340/341.

   (340)  C: 1,<WHO>,<WHOM>        I want to talk to you.

   (360)  A: 3,3,1,1               Can you use V1?

   (340)  C: 4,3,1                 Yes, V1 is OK.

   (360)  A: 3,4,1,1984            Can you use up to 1984 bits/msg?

   (340)  C: 5,4,976               No, but I can use 976.

   (360)  A: 3,4,1,976             Can you use up to 976 bits/msg?

   (340)  C: 4,4,976               I can use 976.

   (360)  A: 9                     Ringing (note how short this
                                   negotiation is!!).

      .......

   (340)  C: 8                     Still there?

   (360)  A: 9                     Still ringing.

      .......

   (340)  C: 8                     Still there?

   (360)  A: 9                     Still ringing.

      .......

   (340)  C: 8                     How about it?

   (360)  A: 9                     Still ringing.

   (340)  C: 2                     Forget it!  (No reason given.)
```

APPENDIX 1

THE DEFINITION OF:

TABLES-SET-#1

by

John D. Markel

Speech Communication Research Laboratory

Santa Barbara, California

TABLES-SET-#1

This set includes tables for:

PITCH -	64 values, PITCH table
GAIN -	32 values, GAIN table
I(1) -	128 values, INDEX7 table
I(2) -	128 values, INDEX7 table
I(3) -	64 values, INDEX6 table
I(4) -	64 values, INDEX6 table
I(5) -	32 values, INDEX5 table
I(6) -	32 values, INDEX5 table
I(7) -	32 values, INDEX5 table
I(8) -	32 values, INDEX5 table
I(9) -	32 values, INDEX5 table
I(10) -	32 values, INDEX5 table

These tables are defined specifically for a sampling period of 150 microseconds.

GENERAL COMMENTS

The following tables are arranged in three columns, {X(j)}, {j}, and {R(j)}. Note that the entries in the {X(j)} column are half a step off the other columns. This is to indicate that INTERVALS from X-domain (pitch, gain, and the Ks) are mapped into CODES {j}, which are transmitted over the network, to be translated by the receiver into the {R(j)}. These intervals are defined as OPEN-CLOSE intervals. For example, the PITCH value (at the transmitter) of 4131 belongs to the interval "(4024,4131]", hence it is coded as j=6 which is mapped by the receiver to the value 21. Similarly, the value of 2400 for INDEX7 is found to belong to the interval "(2009,2811]", coded into the CODE 3 and mapped back into 2411.

Note that if N bits are used by a certain CODE, then there are 2**N+1 entries in the X-table, but only 2**N entries in the R-table.

The transformation values used for PITCH, GAIN, and the K-parameters (in the X- and R-tables) are as defined in NSC Note 42.

Values above and below the range of the X-table are mapped into the maximum and minimum table indices, respectively.

Note that R(J) of INDEX5 is identical to R(2J) of INDEX6, and that R(J) of INDEX6 is identical to R(2J) of INDEX7. Therefore, it is possible to store only the R-table of INDEX7, without the R-tables of INDEX5 and INDEX6.

In the SPS-41 implementation there is no need to store any R-table for the K-parameters. The transmitted index can be used directly (with the appropriate scaling) as an index into the SPS built-in TRIG tables.

COMMENTS ON THE PITCH TABLE

The level J=0 defines the UNVOICED condition. The receiver maps it into the number of samples per frame (here 128).

This PITCH table differs significantly from previous tables and supersedes the table published in NSC Note 36. Details of the calculation of the table can be found in NSC Note 42. Immediate questions should be referred to John Markel.

COMMENTS ON THE GAIN TABLE

The level J=0 defines absolute silence.

This table is designed for a maximum of 12-bit A/D input, and allows for a dynamic range of 43.5 dB.

NSC Notes 36, 45, 56 and 58 supply background for the GAIN table. Gain is the energy of the pre-emphasized, windowed signal.

This table is the NEW GAIN table. NSC Notes 56 and 58 explain the reasoning behind the NEW GAIN.

COMMENTS ON THE INDEX7 TABLE

Positive values are coded into the range [0-63, decimal]. Negative values are coded into the 7-bits two’s complement of the codes of their absolute value [65-127, decimal].

Note that all values -403 < V < 403 are coded as (and mapped into) 0. Note also that the code -64 (100 octal) is never used.

In SPS-41 implementation, the R-table is not needed, since TRIG(2J) is the needed value R(J).

COMMENTS ON THE INDEX6 TABLE

Positive values are coded into the range [0-31, decimal]. Negative values are coded into the 6-bits two’s complement of the codes of their absolute values [33-63, decimal].

Note that all values -805 < V < 805 are coded as (and mapped into) 0. Note also that the code -32 (40 octal) is never used.

In SPS-41 implementation, the R-table is not needed, since TRIG(4J) is the needed value R(J).

COMMENTS ON THE INDEX5 TABLE

Positive numbers are coded into the range [0-15, decimal]. Negative numbers are coded into the 5-bits two’s complement of their absolute values, i.e., [17-31, decimal].

Note that all values -1609 < V < 1609 are coded as (and mapped into) 0. Note also that the code -16 (20 octal) is never used.

In SPS-41 implementation, the R-table is not needed, since TRIG(8J) is the needed value R(J).

THE PITCH TABLE (as of 10-29-74)

X(J)	J	R(J)	X(J)	J	R(J)	X(J)	J	R(J)
0			6002			10770		
	0	128*		21	33		42	61
0			6168			11080		
	1	18		22	34		43	63
3630			6338			11399		
	2	19		23	35		44	65
3724			6515			11728		
	3	19		24	36		45	67
3821			6696			12067		
	4	20		25	37		46	69
3921			6883			12417		
	5	20		26	38		47	71
4024			7075			12776		
	6	21		27	39		48	73
4131			7274			13147		
	7	22		28	40		49	75
4240			7478			13529		
	8	22		29	41		50	77
4353			7689			13922		
	9	23		30	43		51	80
4469			7905			14327		
	10	24		31	44		52	82
4588			8129			14745		
	11	24		32	45		53	85
4711			8359			15175		
	12	25		33	47		54	87
4838			8596			15618		
	13	26		34	48		55	90
4969			8840			16075		
	14	27		35	50		56	93
5104			9092			16545		
	15	27		36	51		57	95
5242			9351			17029		
	16	28		37	53		58	98
5385			9618			17529		
	17	29		38	54		59	101
5533			9894			18043		
	18	30		39	56		60	104
5684			10177			18572		
	19	31		40	57		61	107
5841			10469			19118		
	20	32		41	59		62	111
6002			10770			19681		
							63	114
						infinity		

Note: This table has only 58 different intervals defined, since 5 values are repeated in the R(j) table.

* This value is the "Transmission Interval" (measured in samples) as defined in item #6 of the NEGOTIATION.

THE GAIN TABLE (as of 9-17-75)

X(J)	J	R(J)	X(J)	J	R(J)
0			225		
	0	0		16	245
20			266		
	1	20		17	289
22			315		
	2	24		18	342
26			372		
	3	28		19	404
30			439		
	4	33		20	478
36			519		
	5	39		21	565
42			614		
	6	46		22	667
50			725		
	7	54		23	789
59			857		
	8	64		24	932
70			1013		
	9	76		25	1101
83			1197		
	10	90		26	1301
98			1415		
	11	106		27	1538
116			1672		
	12	126		28	1818
137			1976		
	13	148		29	2148
161			2335		
	14	175		30	2539
191			2760		
	15	207		31	3000
255			infinity		

INDEX7 TABLE (as of 9-23-74)

X(J)	J	R(J)	X(J)	J	R(J)	X(J)	J	R(J)
0			15800			27897		
	0	0		21	16151		42	28106
402			16500			28311		
	1	804		22	16846		43	28511
1206			17190			28707		
	2	1608		23	17531		44	28899
2009			17869			29086		
	3	2411		24	18205		45	29269
2811			18538			29448		
	4	3212		25	18868		46	29622
3612			19195			29792		
	5	4011		26	19520		47	29957
4410			19841			30118		
	6	4808		27	20160		48	30274
5205			20475			30425		
	7	5602		28	20788		49	30572
5998			21097			30715		
	8	6393		29	21403		50	30853
6787			21706			30986		
	9	7180		30	22006		51	31114
7571			22302			31238		
	10	7962		31	22595		52	31357
8351			22884			31471		
	11	8740		32	23170		53	31581
9127			23453			31686		
	12	9512		33	23732		54	31786
9896			24008			31881		
	13	10279		34	24279		55	31972
10660			24548			32058		
	14	11039		35	24812		56	32138
11417			25073			32214		
	15	11793		36	25330		57	32286
12167			25583			32352		
	16	12540		37	25833		58	32413
12910			26078			32470		
	17	13279		38	26320		59	32522
13646			26557			32568		
	18	14010		39	26791		60	32610
14373			27020			32647		
	19	14733		40	27246		61	32679
15091			27467			32706		
	20	15447		41	27684		62	32729
15800			27897			32746		
							63	32758
						infinity		

INDEX6 TABLE (as of 9-23-74)

X(J)	J	R(J)	X(J)	J	R(J)
0			22595		
	0	0		16	23170
804			23732		
	1	1608		17	24279
2411			24812		
	2	3212		18	25330
4011			25833		
	3	4808		19	26320
5602			26791		
	4	6393		20	27246
7180			27684		
	5	7962		21	28106
8740			28511		
	6	9512		22	28899
10279			29269		
	7	11039		23	29622
11793			29957		
	8	12540		24	30274
13279			30572		
	9	14010		25	30853
14733			31114		
	10	15447		26	31357
16151			31581		
	11	16846		27	31786
17531			31972		
	12	18205		28	32138
18868			32286		
	13	19520		29	32413
20160			32522		
	14	20788		30	32610
21403			32679		
	15	22006		31	32729
22595			infinity		

INDEX5 TABLE (as of 9-23-74)

X(J)	J	R(J)	X(J)	J	R(J)
0			22006		
	0	0		8	23170
1608			24279		
	1	3212		9	25330
4808			26320		
	2	6393		10	27246
7962			28106		
	3	9512		11	28899
11039			29622		
	4	12540		12	30274
14010			30853		
	5	15447		13	31357
16846			31786		
	6	18205		14	32138
19520			32413		
	7	20788		15	32610
22006			infinity		

APPENDIX 2

IMPLEMENTATION RECOMMENDATIONS

(1) It is recommended that the priority-bit be turned ON in the HOST/IMP header.

(2) It is recommended that in all abbreviations, "R" be used for Receiver and "X" for Transmitter.

(3) The following identifiers and values are recommended for implementations:

SLNCTH 30 SILENCE-THRESHOLD.

Used for LONG-SILENCE definition. See below. Measured in the same units as GAIN, in its X-table.

TBS 1.000 sec TIME-BEGIN-SILENCE.

LONG-SILENCE is declared if GAIN<SLNCTH for more than TBS.

TAS 0.500 sec TIME-AFTER-SILENCE.

A delay introduced by the receiver after the end of LONG-SILENCE, before restarting the playback.

TES 0.150 sec TIME-END-SILENCE.

The amount of time the transmitter backs up at the end of a LONG-SILENCE in order to ensure a smooth transition back to speech.

TRI 2.000 sec TIME-RESPONSE-INITIAL.

Time for waiting for response for an initial call (#1 and #3). The initial call is repeated every TRI until an answer arrives, or until TRIGU expires.

TRIGU 20.000 sec TIME-RESPONSE-INITIAL-GIVEUP.

If no response to an initial call is received within TRIGU after the FIRST initial call, the system gives up, assuming the other system is down.

TRQ 1.000 sec TIME-RESPONSE-INQUIRY.

If no response to an inquiry (#8) is received within TRQ, the inquiry is repeated.

```
      TRQGU   10.000 sec  TIME-RESPONSE-INQUIRY-GIVEUP.
```

If no response to an inquiry is received within TRQGU from the FIRST inquiry, the system gives up, assuming the other system is down.

```
      TBDA     3.000 sec  TIME-BETWEEN-DATA-ARRIVAL.
```

If no data arrives within TBDA, an INQUIRY (#8) is sent. This repeats every TBDA.

```
      TNR      2.000 sec  TIME-NOT-READY.
```

If the other system is in the NOT-READY (#7) state for more than TNR, an INQUIRY (#8) is sent. This repeats every TNR.

```
      TNRGU   10.000 sec  TIME-NOT-READY-GIVEUP.
```

If the other system is in the NOT-READY (#7) state for more than TNRGU, then the system gives up, assuming the other system is down.

```
      TBIN     3.000 sec  TIME-BUFFER-IN.
```

The input buffer size is equivalent to the time period TBIN (and its size is the DATA-RATE multiplied by the period TBIN). If the INPUT QUEUE ever gets to be longer than TBIN, data is discarded.

```
      TBOUT    3.000 sec  TIME-BUFFER-OUT.
```

The output buffer size is equivalent to the time period TBOUT (and its size is the DATA-RATE multiplied by the period TBOUT). If the OUTPUT QUEUE ever gets to be longer than TBOUT, data is discarded.

REFERENCES

Bolt Beranek & Newman, Inc., Report No. 1822, Interface Message Processor: Specifications for the Interconnection of a Host and an IMP.

NSC Note 42 (in progress).

NSC Note 36, Proposal for NSC-LPC Coding/Decoding Tables, by J. D. Markel, Speech Communications Research Laboratory, Inc., July 20, 1974.

NSC Note 45, Everything You Always Wanted to Know about Gain, by E. Randolph Cole, USC/Information Sciences Institute, October 11, 1974.

NSC Note 56, Nothing to Lose, but Lots to Gain, by John Makhoul and Lynn Cosell, Bolt Beranek & Newman, Inc., March 10, 1975.

NSC Note 58, Gain Again, by Randy Cole, USC/Information Sciences Institute, March 12, 1975.