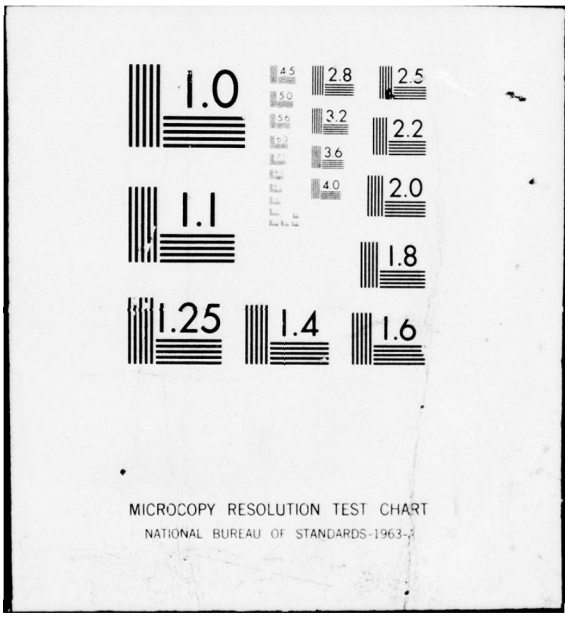
AD-A044 469 COMPUTER CORP OF AMERICA CAMBRIDGE MASS F/G 17/2
PACKET SPEECH MEASUREMENT FACILITY.(U)
JUL 77 D A LOW MDA903-77-C-0156
UNCLASSIFIED CCA-77-07 NL

1 OF 1
AD A044469

END
DATE FILMED
10-77
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

A Packet Speech Measurement Facility: Semi-Annual Technical Report

(12)

Technical Report
CCA-77-07
July 30, 1977

D.A. Low

Computer Corporation of America

Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts 02139

(11) 31 Jul 77

(12) 72 p.

(6) Packet Speech Measurement Facility.

(9) SEMI-ANNUAL TECHNICAL REPORT, 1 no. 1, 11 Dec 76 - 31 June 77

December 11, 1976 to June 30, 1977

(10) D. A/Low

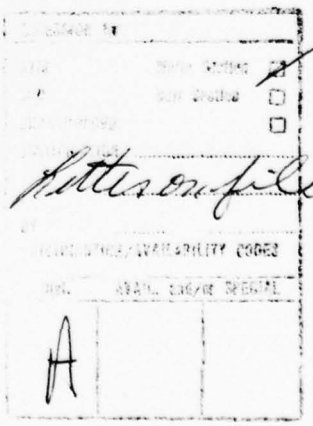
(14) CCA-77-07

(15)

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense under Contract No. MDA 903-77-C-0156. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

387 285

Table of Contents



1. Introduction

The Packet Speech Measurement Facility (PSMF) is a recording, playback, and measurement facility designed to provide members of the Network Secure Communications (NSC) Project with an investigative tool for packetized speech research. PSMF access will facilitate experiments dealing with the effects of network induced perturbations on real-time communications.

This report describes efforts undertaken by the Computer Corporation of America during the initial phase of PSMF development: December 11, 1976 to June 30, 1977. These efforts have culminated in the specification of PSMF design, the development of an access protocol, and the release of a trial facility with operational recording and playback functions.

Chapter 2 reviews the PSMF design philosophy and structure detailed in the Preliminary Design Document of February 28, 1977. Chapter 3 elucidates the approaches CCA has taken to implement this structure. Chapter 4 discusses the work scheduled for completion of this initial phase of the PSMF, while chapter 5 provides a summary of PSMF efforts and plans.

2. PSMF Design Philosophy and Structure

The intent of the PSMF is to provide a facility whereby NSC participants can rigorously examine network effects on real time encoded packet speech streams. Three primary functions have been designed to satisfy this intent:

1. The provision for recording speech segments along with time stamps reflecting network perturbations.

2. The provision for playback of speech segments to enable subjective judgements of network distortions.

3. The provision of statistical functions which analyze recorded speech segments to determine the charactertistics of network distortion.

In section 2.1 we elaborate on the specifications for these functions. Section 2.2 outlines the extension to the Network Voice Protocol by which a user can request these functions, and section 2.3 deals with the PSMF structure designed to accomodate these specifications under the protocol extension.

2.1 PSMF Function Specifications

2.1.1 Recording

The PSMF Recording function is designed to provide a means for storage of:

a. Speech segments consisting of streams of encoded voice messages.

b. Special non-voice control messages, which may be imbedded within a voice stream.

c. Information pertinent to subsequent analysis, such as a time stamp on receipt of each message.

Each recorded segment is identified by name ("filename") and secured with an access password. The typical recording session proceeds as follows:

1. The user initiates a connection with the PSMF over the Arpanet.

2. The user requests the Recording function specifying filename and an optional password.

3. The PSMF acknowledges.

4. The user can send any number of special PSMF non-voice messages for recording.

5. If the user wants to record voice,

   a. He asks to be negotiation master.

   b. The PSMF accedes.

   c. The user suggests negotiation parameters.

   d. The PSMF agrees to all suggestions and records the negotiations.

   e. The user asks if the PSMF is ready.

f. If information sufficient for playback timing has been negotiated, the PSMF signals its readiness.

g. The user sends a stream of voice messages and special PSMF non-voice control messages to be recorded.

6. The user signals the end of the recording session.

2.1.2 Playback

The PSMF provides an access to previously recorded segments that enables selective playback of voice and/or non-voice messages. Voice stream playback permits repeatable tests of vocoding equipment and algorithms. Imbedded control messages can be used to carry textual, graphical, or machine control information. In addition, the results of a PSMF measurement function may be incorporated into files of PSMF control messages, which are then subject to playback. The suggested procedure for playback is as follows:

1. The user initiates a connection with the PSMF over the Arpanet.

2. The user requests the Playback function, specifying filename and password.

3. The PSMF acknowledges success if the filename exists and password is correct.

4. The user specifies the types of messages to be played back, (e.g. voice and textual control messages).

5. If the user has specified voice playback,

   a. The PSMF asks to be negotiation master.

   b. The user accedes.

   c. The PSMF presents the same negotiation parameters encountered in recording.

   d. The user agrees to all suggested negotiations.

e. The PSMF asks if the user is ready.

f. The user assents.

6. The PSMF plays back the requested information. Voice data is timed according to negotiation parameters and message headers. Control data is sent so that only one control message is in transit at a time.

7. The PSMF completes playback, or the user signals an end to the playback session.

2.1.3 Measurement Functions

The PSMF measurement functions provide means for analyzing parameters in PSMF files and creating PSMF files, suitable for playback, containing the results of such analyses. Measurement functions are concerned with:

a. Relative delays inflicted on voice packets by the network.

b. Missing, duplicate, or out of order voice packets.

c. Qualities of the voice stream itself, e.g. periods of silence and speech.

Measurements consist of statistical analyses (mean and standard deviation) and histogram constructions. The standard procedure for invoking a measurement function is as follows:

1. The user initiates a connection with the PSMF over the Arpanet.

2. The user requests the Measurement function, specifying filename and password of the file to be measured.

3. The PSMF acknowledges.

4. The user specifies the nature of the measurement and the filename and password of the file which receives the analyses.

5. The PSMF acknowledges and disconnects to allow other concurrent PSMF access.

6. At a later time the user can play back the measurement file or subject it to further analysis.

2.1.4 File Manipulation

Two provisions for PSMF file "manipulation" are included in the current design: file appendage and file deletion. These functions are available only to the site which created the file.

The PSMF Append function permits a user to add voice and/or non-voice control messages to an extant file of his own creation. The procedure for appending is as follows:

1. The user initiates a connection with the PSMF over the Arpanet.

2. The user requests the Append function, specifying filename and password.

3. The PSMF acknowledges.

4. The user can proceed as in recording, except:

   If the user trys to negotiate voice, and the extant file contains voice data,

a. The PSMF asks to be negotiation master.

b. The user accedes.

c. The PSMF presents the same negotiation parameters encountered in the recording of the original file.

d. The user agrees to all suggested negotiations.

e. The PSMF asks the user to be negotiation master.

f. The user agrees.

g. The user asks if the PSMF is ready.

h. The PSMF agrees.

i. The user sends a stream of voice and/or control message to be recorded.

5. The user signals the end of the append session.

Application of the Delete function causes the named file to be expunged. It is invoked as follows :

1. The user initiates a connection with the PSMF over the Arpanet.

2. The user requests the Delete function, specifying filename and password.

3. The PSMF acknowledges.

2.2 PSMF Protocol Extension (1)

PSMF Protocol

This protocol is an extension to the Network Voice Protocol to allow access to a Packet Speech Measurement Facility. It provides a means to specify recording and playback of speech files, and then to select measurements and statistical analyses of those files.

This protocol is based on the NVP: it uses the same initial connection proceedures (ICP), assignment of links, vocoder negotiation, and ARPAnet message types/subtypes (type-0/0 for control messages and type-0/3 suggested for data). The format of data messages is the same as for normal NVP. The reader is referred to the NVP document for details.

---

1. This section was prepared in conjunction with ISI.

The ICP for NVP is the following sequence of three messages:

On link 377: 1,<WHO>,<WHOM>,K

On link K: 6,L

On link L: 1,<WHO>,<WHOM>,K

For normal NVP, the ANSWERER replies to the second [1] message by initiating vocoder negotiation. Instead, when a PSMF is called, it replies with an "OK, I AM A PSMF" message. After that, the caller can request different operations using this PSMF Protocol.

Summary of Messages

The PSMF Protocol messages are listed below, then described later. All PSMF requests are made with the same message code, with each different request having a unique subcode. Acknowledgement is via a different message code with the subcode returned from the request. The list of message codes and subcodes is as follows:

[60] "OK, I AM A PSMF"

[61],[1] "OPEN A FILE TO RECORD"

[61],[2]	"OPEN A FILE FOR PLAYBACK"
[61],[3]	"OPEN A FILE TO APPEND"
[61],[4]	"RETRIEVE FIELDS"
[61],[5]	"FIELD TO BE RECORDED"/"RETRIEVED FIELD"
[61],[6]	"CLOSE FILE"/"END OF FILE"
[61],[7]	"DELETE FILE"
[61],[8]	"TEXT TO/FROM COMMAND PARSER"
[61],[9]	"DATA TO/FROM MEASUREMENT FACILITY"
[61],[10]	"OPEN A FILE FOR MEASUREMENT"
[62],[n]	"POSITIVE ACKNOWLEDGEMENT"
[63],[n]	"NEGATIVE ACKNOWLEDGEMENT"

Description of Messages

In addition to message codes and subcodes, PSMF messages contain elements labelled <field-name> and <string>. A <field-name> is a single 16-bit word which is the identifier for a field as specified by the PSMF file format. Possible values for <field-name> are listed later in this document. A <string> is a sequence of N+1 16-bit words, where the first word is the count of words to follow, i.e. N. Each word contains two 8-bit ASCII bytes, with the byte in the most significant 8 bits of

the word preceding the byte in the least significant 8 bits of the word. The string also includes either 1 or 2 NUL (binary zero) terminating bytes, depending on whether there are an odd or even number of text bytes contained in the string, respectively. That is,

<string> = N,W(1),W(2),...,W(N)

For example,

"FILENAME" = 5, FI, LE, NA, ME, 00

The other fields in the messages below are 16-bit words, with the values right-justified in the word. Any extra words in the message after those which are defined should simply be ignored by the receiver of the message.

[60] "OK, I AM A PSMF"

This message is used only to answer the second 1 message during ICP. It consists of only one word:

60

[61],[1] "OPEN A FILE TO RECORD"

Only one file may be opened at a time, but different files may be opened in succession (see the "EOF" message).

This message varies in length; it contains two words followed by two strings:

61,1,<filename-string>,<password-string>

The PSMF will create a file with the given name (unless the name is not suitable), and record the password to compare against the password given when the file is opened for playback. This operation will be acknowledged, either positively or negatively, by a [62] or [63] message, respectively (see below).

[61],[2] "OPEN A FILE FOR PLAYBACK"

This message has the same format as "OPEN FOR RECORD":

61,2,<filename-string>,<password-string>

The PSMF will search its directory for the given filename. If it is found, the password will be checked against the one recorded with the file to determine if access is allowed. Again a [62] or [63] is sent as acknowledgement.

[61],[3] "OPEN A FILE TO APPEND"

This message allows the caller to append more information or speech to the end of an extant file. Again, the password must match the one recorded when the file was created. The format is:

61,3,<filename-string>,<password-string>

[61],[4] "RETRIEVE FIELDS"

When a file is open for playback, this message requests that all fields with given field names be extracted from the file and sent to the caller. The message consists of the words:

61,4,n,<field-name 1>,...,<field-name n>

This message is positively acknowledged by the receipt of the requested fields, or negatively acknowledged explicitly by a [63],[4] message. After all fields have been sent, the PSMF sends an "EOF" message (this includes the case where no fields exist).

[61],[5] "FIELD TO BE RECORDED"/"RETRIEVED FIELD"

This message is of variable length, and may contain a string or any collection of words:

61,5,<field-name>,N,W(1),W(2),...,W(N)

Voice files may contain text or binary fields. Messages of this type which are received by the PSMF while a file is open for record will be inserted into the file. Note that voice data messages are not sent using this message, but are sent on the NVP data link instead. This message is also used to return non-voice fields in response to a [61],[4] request.

[61],[6] "CLOSE FILE"/"END OF FILE"

This message is sent by the caller to indicate the end of data to be recorded or to indicate that no more information is desired from a file during playback. For this usage, the message has the consists of only two words:

61,6

This message is also sent by the PSMF to indicate that no more fields of the requested type are left to play back. It does not mean that the PSMF has closed the file. This message contains the message code and subcode followed by the number of messages transmitted:

61,6,<message-count>

The <message-count> is a 32-bit integer.

[61],[7] "DELETE FILE"

This message is sent by the caller to the PSMF. Its format is

61,7,<filename-string>,<password-string>

This message looks up the given filename, and deletes the file if the password matches the one recorded with the file.

[61],[8] "TEXT TO/FROM COMMAND PARSER"

This message provides a means to send text messages to/from the measurement facility. Its format is:

61,8,<string>

These messages might be commands to the PSMF or replies from it. In addition, results of measurements might be reported using this technique. Typically, these messages would simply be typed out on the user's terminal (if any).

[61],[9] "DATA TO/FROM MEASUREMENT FACILITY"

This message has basically the same format as the [61],[8] message, but the data is not necessarily a string of ASCII bytes:

61,9,N,W(1),W(2),...,W(N)

The meaning of these data words is not defined by this document. This message is provided to allow measurement commands or data to be passed between the PSMF and its caller; the sub-protocol which defines the meaning of the data words has not yet been specified.

[61],[10] "OPEN A FILE FOR MEASUREMENT"

This message has the same format as "OPEN FOR PLAYBACK":

61,10,<filename-string>,<password-string>

The procedure and response is the same as for "OPEN FOR PLAYBACK".

[62],[n] "POSITIVE ACKNOWLEDGEMENT"

This message echoes the message which it acknowledges, except that the [61] is replaced by [62], for example:

61,1,"FILE","PASSWORD"

62,1,"FILE","PASSWORD"

Currently, it is used only in this fashion to acknowledge opening of files. If in the future it is used to acknowledge a message with a different format, the format of the [62] message would again follow the format of the message it acknowledges.

[63],[n] "NEGATIVE ACKNOWLEDGEMENT"

Like the [62] message, this message echoes the message it acknowledges. However, in addition to replacing the [61] with [63], this message appends a reason code and string to the end of the acknowledged message:

61,1,"FILE","PASSWORD"

63,1,"FILE","PASSWORD",<reason-code>,<reason-string>

If the sender desires not to include a <reason-string>, he should indicate a null string by giving a zero word count for the string. Therefore, the minimal appendage would be two words of zero, the first indicating an unspecified reason, and the second

indicating a null string. Note that this message is also used with the [61],[4] and [61],[5] messages,, which have a different format:

61,4,<field-name>

63,4,<field-name>,<reason-code>,<reason-string>

Reason codes are defined later in this document.

Vocoder Negotiation

In order to send voice data to/from the measurement facility, the two parties must negotiate vocoder parameters. However, a caller who is incapable of handling voice data may want to access textual data in a voice file, so vocoder negotiation is optional within a PSMF protocol session.

Vocoder negotiation is initiated with an NVP "RENEGOTIATION REQUEST" message, number [12]. This message consists of two words,

12,<I-master>

During recording of a voice file, the caller sets <I-master> to a nonzero value to indicate that he desires to negotiate the vocoder parameters to be stored

with the file. For playback, if the caller specifies voice in the "RETRIEVE FIELDS" command, the PSMF sends a "RENEGOTIATION REQUEST" with non-zero <I-master>. Acceptance by the caller will cause the PSMF to conduct negotiations as per the stored vocoder parameters. For more detail, see the sections on negotiation and renegotiation in the NVP document.

Obviously, vocoder negotiation must occur before transfer of voice data messages is possible. On recording, if the caller sends voice data before negotiating, the PSMF should simply ignore the data.

PSMF File Format Field Names

A field name is recorded as the first word of every record in a PSMF secondary file to specify the field type. The least significant 8 bits of the field name are the field type, and the most significant 8 bits indicate the use of that field, depending upon the field type. The following field names have been defined:

Field Type	Use if Voice	Use if Text

0	voice	(not defined)	1 author
1	control		2 sender
2	text		3 to
3	binary		4 cc
4	graphics		5 subject
			6 date
			7 summary

Reason Codes

The following reason codes have been defined:

0. Unspecified reason
1. No access right
2. File does not exist
3. Wrong password
4. Vocoder negotiation needed first
5. No file has been opened

This list is incomplete and will be augmented as the need arises.

2.3 PSMF Structure Design

The Digital Equipment Corporation RSX-11M operating system was chosen as the vehicle for development and implementation of the PSMF. Reasons for this were:

a. A development machine under RSX-11M was available.

b. RSX-11M has adequate software development tools.

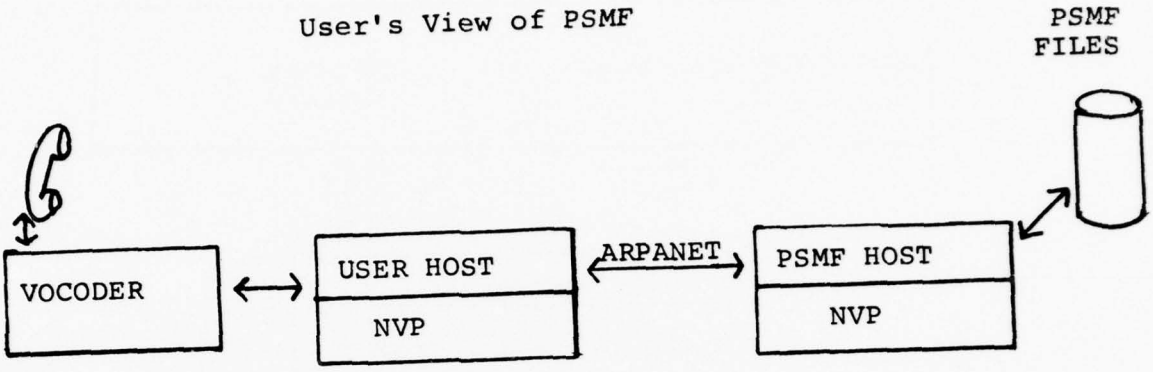
c. A well developed file system is offered.

d. RSX-11M provides executive services for scheduling, inter-process communications, timing, I/O, and dynamic memory allocation and mapping.

With these facilities available, it was decided to design the PSMF as a set of functionally disjoint processes. These processes would communicate via executive directives and a common buffer area, and would share the services of a set of library routines.

The PSMF user is, of course, unaware of these components. His view of the system might be as illustrated in figure 2.1.

Figure 2.1
User's View of PSMF



A more detailed examination of the structure to be found at the PSMF host is diagrammed below in figure 2.2.

Figure 2.2

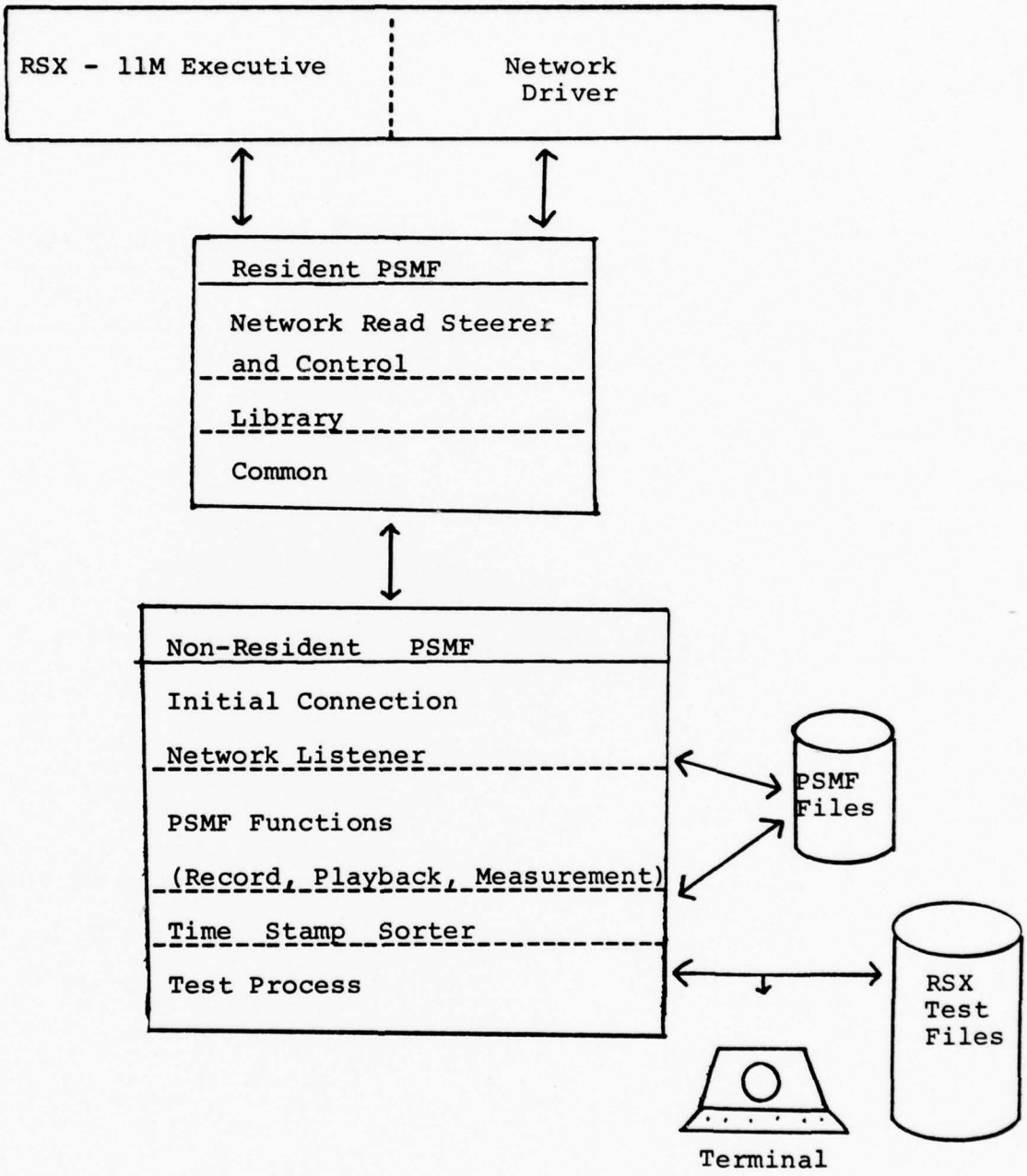


Figure 2.2
Process Structure at PSMF Host

The following sections discuss the design objectives of the PSMF elements illustrated in figure 2.2.

2.3.1 RSX-11M Executive and Network Driver

The executive provides a wide variety of management and communications services. Explicit invocations of executive services are excluded from the PSMF Function process itself (in order to facilitate operating system conversion), but are included in the Library routines, Control process, and Test process.

The network driver provides an interface between PSMF processes and Arpanet. By convention, network reads are requested only by the Network Read Steerer and Control process, via the mediation of Library routines. Network writes can be requested by any process, again through a Library routine. The executive manages I/O queuing.

2.3.2 Library Routines

These are a set of re-entrant and position independent routines which occupy part of the address space of all PSMF processes. They incorporate functions common to all processes, and when structured as a Library, serve to reduce total core space requirements, and to isolate explicit references to executive services. RSX-11M requires that a Library be resident.

2.3.3 Common Area

This is a buffer or data area which occupies part of the address space of all PSMF processes, facilitating interprocess communication. RSX-11M permits such an area to be dynamically created.

2.3.4 Network Read Steerer and Control Process

This process is initiated from an RSX-11M terminal. Its functions are to:

a. Initialize PSMF data structures.

b. Initiate a process which listens to the NVP initial connection link.

c. Make judgements as to the disposition of all network reads.

d. Communicate to processes, via global semaphores, the results of any function requiring a network read. This process will often be referred to as the "Control" process.

2.3.5 Initial Connection Network Listener Process

This process is initiated by the Control process (2.3.4). Its function is to listen to NVP link 377 for a protocol sequence indicating a user's request for a PSMF function. If this sequence is received, the PSMF Function process is initiated.

2.3.6 PSMF Function Process

This process is initiated by the Listener process (2.3.5) when a user requests a PSMF function, and terminates itself on receipt of an NVP "goodbye" message. It incorporates all the NVP and PSMF extensions necessary to communicate with the user.

The principal PSMF functions are Recording, Playback, and Measurement. Each of these requires access to PSMF disk files. Such access is mediated by a group of routines which specialize RSX file manipulation to the PSMF's needs.

It may be possible that the PSMF Function process initiates asynchronous processes on its own. The Recording function, for instance, may initiate a process which performs a sort on user's voice time stamps.

2.3.7 Test Process

This process is initiated from an RSX-11M terminal. Its function is to simulate user access to the PSMF. As a PSMF process in its own right, it conveniently provides a test of the multi-user capabilities of the design.

3. Current Implementation

This chapter describes in detail the current level of PSMF implementation, which includes a working Record and Playback facility.

Section 3.1 discusses the data structure design for PSMF processes. Section 3.2 describes the PSMF file structure. From there, a description of constituent modules ensues. Section 3.3 is concerned with the Network Driver, 3.4 with the Library routines, 3.5 with the Control process, 3.6 with the Initial Connection Network Listener process, 3.7 with the PSMF file accessing routines, 3.8 with the Record junction of the PSMF Function process, 3.9 with the Time Stamp Sorting Process, 3.10 with the Playback junction of the PSMF Function process, and 3.11 with the Test process.

3.1 PSMF Processes Data Structures

It was envisioned from the first that the PSMF would eventually support multi-user activity. This anticipation led to the design of a coherent data structure for all processes. From the standpoint of inter-process communication, it was desirable to have these data structures in the address space of all processes. These considerations led to the following implementation:

a. All processes have associated entries in a "Link Table". For processes which are actually involved in network communications, Link Table positions map into the listening NVP links for that process. For processes which do not need to communicate, pseudo link positions greater than 400 are allocated.

b. A process's Link Table entry points to its "Usertable", which contains the fundamental data structure for the process.(1) The Usertable is allocated from the Common buffer area when the

---

1. With the exception of the Test process, which has unconventional needs for RSX associated structures.

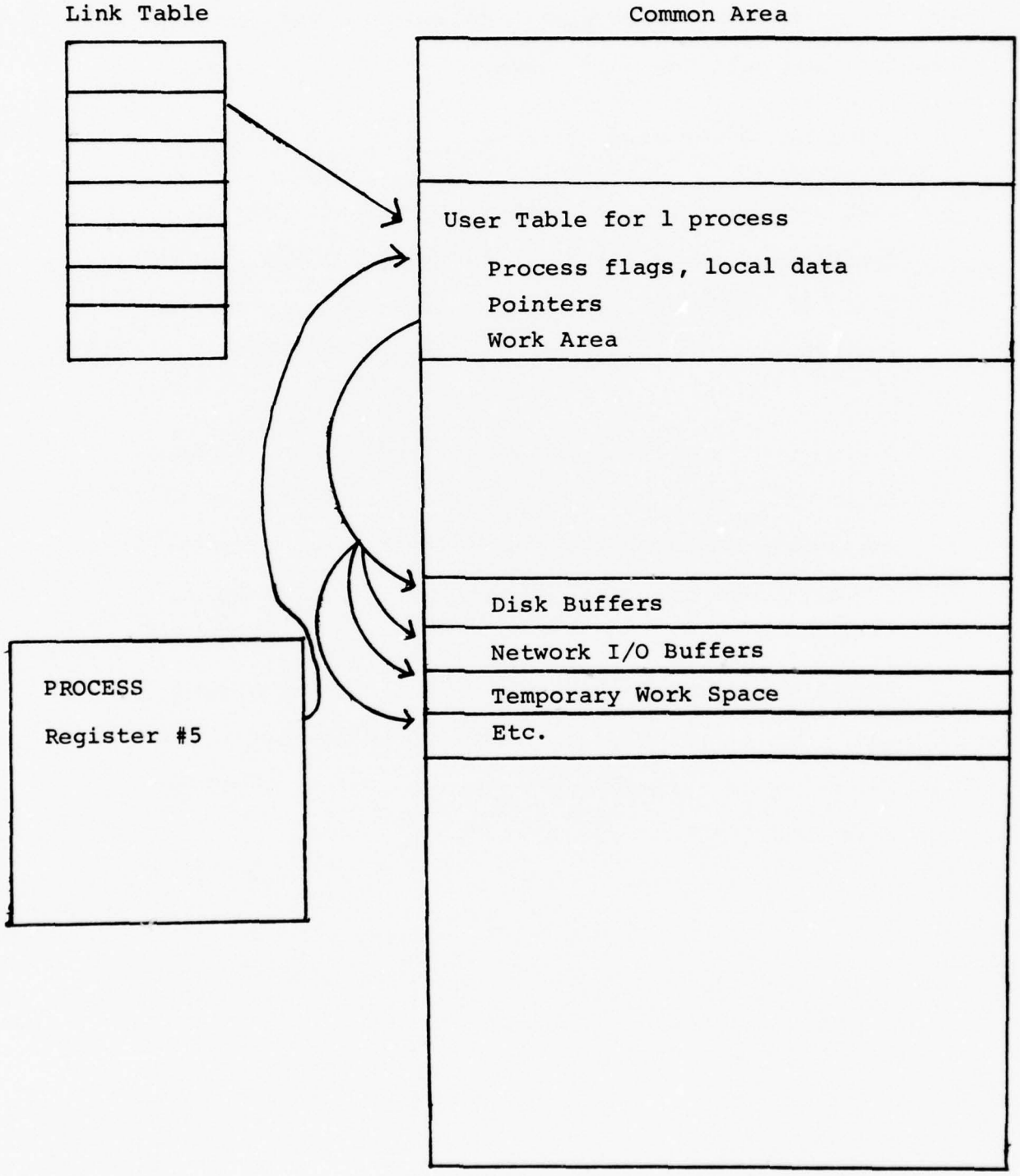
process is initiated, and de-allocated when the process terminates.

c. All other data structure requirements for a process (e.g. disk buffers) are met by allocations from the Common buffer area. Pointers to these areas can ultimately be found in the Usertable.

d. Convention dictates that a process access its Usertable using indices relative to a single dedicated register. This convention facilitates intra-process communications.

Figure 3.1 illustrates these data structures.

Figure 3.1 PSMF Process Data Structure



In addition to the data represented in each Usertable, there is a need for global data which can act as semaphores. This need is met by the RSX "event flags" provision, which includes executive services for setting, clearing, and awaiting such flags.

These semaphores are used to:

a. Lock access to data structures to prevent possibly conflicting alterations. For example, the linked list of buffers in Common have a semaphore associated with allocation and de-allocation.

b. Signal processes that an event has taken place. Each of the PSMF processes has a dedicated semaphore, which is used to indicate that network I/O has been completed, or that a time delay is up.

c. Signal a request for services. The Control process can, for instance, be awakened by its dedicated semaphore to communicate a request for a network read.

3.2 PSMF File Strucutre

Data pertinent to PSMF storage, retrieval, and measurement is of two fundamentaly distinct types. Firstly, there is the actual stream transmitted by the user. Secondly, there is information related to arrival time and sequence. This dichotomy encouraged the following implementation:

a. Any recording session results in two PSMF files: a "primary" file and a "secondary" file.

b. The primary file consists simply of the stream of data transmitted by the user. Any network responses generated by the PSMF during negotiation are also recorded - to serve both as a debugging record and a guide during playback.

c. The secondary file contains the recording date and time, user identification, parcel timing, and file size. In addition, for every message in the primary file, the secondary file contains various flags, PSMF time stamp, and sorted link to next (by user's time stamp) message.

It is intended that the secondary file contain all information needed by the measurement functions, as well as that necessary for ordered playback.

3.3 Network Driver

The network driver is the routine in the RSX-11M executive which provides full duplex access to the Arpanet. This access is achieved via standard RSX "QIO" executive requests. RSX provides the mechanism for queueing requests and communicating parameters to the driver. It is the driver's responsibility, on the other hand, to signal normal or abnormal request completion.

The PSMF driver was designed to treat the interface as two separate devices - one each for input and output. Completion of a type 3 network write (uncontrolled) can be signalled merely by completion of physical transfer to the IMP. Completion of a type 0 network write, however, requires reading a corresponding RFNM from the destination IMP. This means that, in general, processes do not interface directly to the network driver, but are filtered through Library routines which cooperate with the Control process.

3.4 Library Routines

These routines consist of re-entrant and position independent code which is shared by all PSMF processes. While such an entity is not strictly essential, it does serve to reduce core requirements and compartmentalize the use of requests for executive services.

There are many Library routines. They are briefly described below.

3.4.1 Network Read

This routine will suspend a process if a previous network read has not been completed. It then signals the Contral process that data is expected from the network.

3.4.2 Network Write

Type 3 writes and Type 0 writes are treated independently. The Library routines will wait on a write of a given type if a previous write of that same type has not been completed. A QIO is then issued to start the write, and, in the case of a write type 0, the Control process is notified that a RFNM is expected.

In addition, Library routines are available to move data to a temporary buffer before starting the write. This avoids locking the process's own buffers while awaiting completion.

3.4.3 Wait on Network I/O

These routines are called by a PSMF process or by other Library routines to wait until completion of a network read, write type 0, or write type 3. If no I/O is in abeyance, no wait is effected.

3.4.4 Usertable & Process Initiation

If a process is initiated by another process, some communication of data is usually necessary. A PSMF initiating process will allocate and set up a Usertable for the initiated process, start that process, and communicate the Usertable address. Library routines perform all these functions.

3.4.5 Time Interval Computation

A process's initiation time is stored in its Usertable. A Library routine is available to compute, in milliseconds, the difference between the time of call and the time of initiation. This is used as the PSMF time stamp during the PSMF Record function.

3.4.6 Timed Suspension of a Process

This routine will suspend a process for a specificied number of milliseconds. It is used by the PSMF Playback function to time the transmission of encoded voice packets.

3.4.7 Management of Common Buffer Pool

The common pool of fixed length buffers supplies Usertables, network I/O buffers, disk buffers, and temporary work space. A pair of Library routines are used to allocate and free such buffers.

3.4.8 Process Exit

This routine frees all the buffers owned by a process, resets its link table entry, and performs an RSX exit call.

3.5 Control Process

The Control process is initiated at an RSX-11M terminal. Its functions are threefold: it initializes PSMF resources, acts as a network read steerer, and adjudicates network I/O completion.

3.5.1 Initializaton of PSMF Resources

As the first PSMF process to be invoked, it is the Control process's job to initialize resources:

a. Interprocess semaphores are reset.

b. The Link table is cleared.

c. A Common buffer region is dynamically created and a mechanism is set up to allow all future processes to map to it.

d. The IMP is primed with a few NOPS.

e. The Initial Connection Listening Process is initiated.

f. A network read is started.

3.5.2 Network Read Steerer

The initial network read will wake up the Control process when two words (32 bits) have been transferred into a special header buffer. The Control process analyzes this header to effect the disposition of the rest of the message:

a. Unrecognizable headers cause the remaining message to be discarded.

b. Links on type 0 and 3 messages are examined.

   i) If it is not an NVP link, the message is discarded.

   ii) If the link is not owned by a PSMF process, the message is discarded.

iii) If the owning PSMF process had requested a network read, the remaining message is read into the area designated by the process.

iv) Otherwise a free buffer is allocated, the remaining message is read into it, and the process is flagged as having received an unsolicited read. This facility is designed to allow the Recording function to fall a little behind in its retrieval of type 3 messages.

c. A NOP is ignored.

d. A RFNM is used to flag the corresponding network write type 0 as complete.

e. An "incomplete transmission" is ignored. Failure of a type 0 write is noted when no RFNM arrives within five seconds.

f. An "interface reset" forces transmission of a few NOP's and discarding the first input.

g. All other messages are ignored.

3.5.3 Adjudication of Network I/O Completion

The Control process is notified of all requests for network reads.

a. If an unsolicited read has already been undertaken on behalf of the requesting process, its data are moved to the requested area and the unsolicited buffer flushed.

b. Otherwise the next network read with an appropriate link is steered into the requested area.

In addition, the Control process is notified of all completed network I/O. This provides a central point for resolution of PSMF process waits on network I/O. For example, if a process requests a wait until completion of network write type 0:

a. The request is made through a Library routine which checks if such a write has been requested but not finished. If not, an immediate return is effected.

b. The Library routine then clears a global semaphore assigned to the PSMF process, sets a "wait" flag in its Usertable, and issues an RSX executive directive. RSX will resume the process when the global semaphore has been set.

c. When the Network Read Steerer section of the Control process detects the appropriate RFNM, it sees the PSMF process's wait flag, and sets the proper global semaphore. RSX is thus signalled to awaken the PSMF process.

3.6 Initial Connection Network Listener

NVP initial connection protocol requires an opening request on link 377. Since PSMF processes are in general associated with links, it was convenient to assign the initial connection function to a distinct process. This Listener process is initiated by the Control process on PSMF initialization (3.5.1) and performs the following duties:

a. It requests a network read on its link of 377 and waits pending read completion.

b. When a message with the correct format is received, the Link table is searched for the first free entries. Such entries are mapped to a link number, e.g. link 360.

c. The PSMF process associated with that link number is initiated e.g. the process named "NVP360."

d. The Listener process allocates and sets up a Usertable for the new process, sends the Usertable address, and goes back to the listening start. It is the new PSMF process's responsibility to carry on all communications on its own links.

3.7 PSMF File Accessing Routines

These are a series of modules designed to interface between a PSMF process and RSX-11M file-handling primitives. While suitable for Library inclusion, space limitations now dictate that these routines be linked separately with each file handling PSMF process.

Each routine deals with a single file at a time, e.g. with either a primary or secondary file. Access is by record at the PSMF level and by block at the RSX level. Several processes can simultaneously access a single file.

An outline of the routines follows.

3.7.1 File Open Routines

A PSMF file may be opened for writing (record), reading (playback, etc.), or updating (append). In each case, the using process supplies the record size, file name, and the file number (0 thru 3) which will be used for all future accesses.

A file open routine will also allocate Common buffer space for RSX specific file information, and two block buffers for each file.

3.7.2 File Close Routine

The file close routine will initiate an RSX file close and free buffer space used by the file.

3.7.3 Disk Read Routines

The disk read routines require a file number and record number for access. A pointer to the requested record is returned. As an option, the sequential read routine tries to insure that a record's predecessor and successor are in core in anticipation of future reference.

3.7.4 Disk Write Routines

The PSMF disk write routines write only out of the "current" record buffer; in order to change a record, that record must first be read. As an option, the sequential write routine increments the "current" record number and buffer address after each write.

Both the read and write routines employ two block buffers (16 primary file records or 64 secondary file records). No physical disk transfer take place if either of the two block buffers contains the desired record.

3.8 Record Function

The PSMF Function process (Record and Playback) is initiated by the Network Listener process (3.6). The Function process continues the opening protocol and awaits a PSMF command.

The PSMF Record command has provision for a filename and optional password. The Record function is called and:

a. Opens two PSMF files: one with 64-word records and the extension ".PR" (primary file) and the other with 8-word records and the extension ".SE" (secondary file).

b. Sends an acknowledgement to the user.

c. Records the PSMF Record command as the primary file's first record.

d. Records the date, time, and user ID as the first record of the secondary file. Parcel timing and file size information is added to this record at the conclusion of the recording session.

The Record function then goes into a network listen loop in which certain messages are recorded on the disk. These messages which are stored are:

a. PSMF "field to be recorded" messages.

b. All acceptable negotiations for voice transfer. The Record functon will accept all suggestions (except for message size greater than 976 bits.)

c. All voice negotiation replies from the PSMF to the user. These are flagged appropriately and are used as a guide during Playback.

d. When sufficient information for Playback timing is available, the user can terminate negotiation and record voice messages.

Every stored message has an entry in each of the primary and secondary files.

a. The entire message is stored as a single record in the primary file as it is received, preceded by a word containing its length.

b. The corresponding record in the secondary file contains a field name word, a flag word, the PSMF time stamp in milliseconds, words 3 and 4 of the message, and provision for a link to the next record (to be sorted by user's time stamp by the "Sort by Time Stamp" process).

The Record function remains active until receipt of an NVP "goodbye" or PSMF "end of file" message. In either case, the primary and secondary files are completed and closed. The Sort process is initiated in order to link the secondary file by user's time stamp. A "goodbye" will terminate the NVP connection, while an "end of file" will cause control to return to the start of the PSMF Function process.

3.9 Sort Process

The Arpanet does not guarantee the integrity of a stream of type 3 messages. Voice packets may arrive at the PSMF out of sequence, as duplicates, or may not arrive at all.

It is the function of the Sort process to link a PSMF message file so that Playback message sequence will closely approximate that in which they were sent. This is accomplished by

a. Linking the voice messages by the sender's time stamp.

b. Linking imbedded PSMF "field to be recorded" messages as propitiously as possible.

The Sort process is initiated by the record function at the conclusion of a recording session. The Sort process has no requirements for network communication, so is given a link table entry corresponding to a network link greater than 400 (see section 3.1). It opens the secondary file with shared access (to allow concurrent playback) and performs the following functions:

a. Using the parcel timing information from the first record, a one minute "window" is estimated. This is the maximum number of packets, at seven parcels per packet, which could be sent in one minute. No search for a given record's link will exceed the boundaries imposed by plus or minus one window.

   This is a consequence of the one-word length of the user's time stamp. An unrestricted search for a given time stamp could result in linkage to an inappropriate message with the same time stamp modulo 2**16 as that expected.

b. An expected time stamp is computed from the present time stamp and parcel count.

c. A forward search of the secondary file is made up to the window boundary. If a control message (imbedded or preliminary) is encountered immediately, it is linked and the entire search begins again. If the expected time stamp is encountered, it is linked and a new search begins.

d. Otherwise a search in the negative direction up to a window boundary is made. If the expected time stamp is encountered, it is linked and a new search begins.

e. Otherwise a link is made to the record with the time stamp differing least from the expected time stamp, if any.

f. Otherwise the sort terminates. (No more records).

This algorithm permits linkage of files consisting entirely of control messages as well as those with duplicate or missing messages.

3.10 Playback Function

Once a PSMF file has been created by the Record function and linked by the Sort process, it can be accessed by the Playback function of the PSMF Function process.

The PSMF Playback command has provisions for filename and password. When this command is encountered by the PSMF Function process,

a. An attempt is made to open the primary and secondary files associated with the filename. If this attempt is unsuccessful, a negative acknowledgement is sent to the user.

b. The first record of the primary file is read to determine the password. If no password was used for recording, no check is made of that included with the playback command.

c. A password comparison is made. If a bad match is found, a negative acknowledgement is sent to the user.

d. Else a positive acknowledgement is sent and the Playback function awaits user commands specifying the information to be retrieved.

The PSMF extension to the NVP protocol permits selective retrieval of stored information. In particular, the user can request the playback of any combination of:

a. Non-PSMF control messages.

b. PSMF control messages or subsets thereof.

c. Voice messages.

The user specifies a list of his selections in a PSMF "fields to be retrieved" command. The Playback function then performs the following:

a. If the user has specified voice, the PSMF conducts negotiations as per those which were recorded. If the user disagrees with any of the recorded negotiations the playback session is terminated.

b. The secondary file is scanned from the beginning to find any message types on the user's list. If found, the corresponding record in the primary file is transmitted.

c. The interval timing between voice message transmission is computed from the parcel count and parcel timing data.

d. If no further messages are to be retrieved, or if the user sends a PSMF "end of file" command, Playback sends a count of messages transmitted and awaits a further list of selections from the same

file. An NVP "goodbye" elicits the same count and a termination of the Playback session.

3.11 Test Process

The Test process was designed to provide a full testing facility for the PSMF implementation. It essentially consists of highly RSX dependent code, initiated from an RSX terminal, which can read or write RSX files and access the Arpanet as a PSMF process.

When initiated, the Test process:

a. Takes NVP links 375 and 376 for its own read links.

b. Asks for an input RSX file to read data and an output file for results.

The input data can be commands to:

a. Affect network output: i.e. change destination HOST/IMP, output link or message type.

b. Read network input.

In addition, data to be transmitted over the network can consist of strings

N(1), N(2),...,N(K)

where each N(I) is the octal representation of a PDP-11 word or byte, or an ASCII string enclosed in double quotes.

These data strings are converted to an Arpanet message and transmitted, typically through the IMP and back to a PSMF process. Data received from a network read are converted to similar readable strings and sent to the Test process output file. This permits interactive (via a terminal) or high speed (via an RSX disk file) testing of PSMF constituents.

4. Implementation Completion

The schedule included in the Preliminary Design document gives August 15, 1977 as a first release date and September 30, 1977 as a final release date for this implementation of the PSMF.

The first release will incorporate the following improvements and augmentations over the trial release:

a. The Append and Delete functions will be included in the PSMF function process.

b. Improvements will be made in the error recovery procedures.

c. Further discrimination will be possible to enable Playback of various subtypes of PSMF control messages.

The final release will incorporate the Measurement function of the PSMF Function process. This will enable quantitative analyses of PSMF files. These analyses will produce PSMF files consisting of measures of:

a. Relative delays experienced by voice packets.

b. Durations of speech and silent periods.

c. Classifications of packet sequences into duplicates, missing, or out of order.

d. Classifications of packet types.

e. Histograms or statistical properties of the above.

These measurement files will be accessable via the Playback function.

5. Summary

This report has described the work performed at CCA on the Packet Speech Measurement Facility during the initial phase of its development.

A review of PSMF design objectives was followed by a discussion of the structure formulated to meet these objectives. An appropriate protocol extension was described, which was followed by a detailed description of the current implementation and its components. A short summary of work scheduled to complete this phase ensued.

The trial release of the PSMF Record and Playback functions has been successfully employed by members of the NSC community. We anticipate regular experiments once the protocol extensions are incorporated into user NVP's. Plans are currently under consideration for expansion of PSMF capabilities to give participants greater control over and better access to those experimental situations of interest to them. It is in this context that the PSMF will become a valuable network facility.