A SIMPLE MESSAGE GENERATOR

IEN 172

David Flood Page

9 March 1981

Bolt, Beranek and Newman Inc.

50 Moulton Street

Cambridge, Massachusetts 02238

TABLE OF CONTENTS

1 Introduction

We want to have a controllable message generator for the Catenet that will be simple to implement. There currently exists the TRAF system which was built to do performance measures for another project; this document describes a subset of the TRAF facilities aimed at producing a simple message generator that can be controlled by the existing TRAF control system. A full specification and description of TRAF will be issued later.

Because we want it to be controllable by the existing TRAF control system, the server will have to include a TCP. This conflicts slightly with the idea of a minimal server, but will allow us to get things working faster. At some point in the future we will define a simpler, XNET-like protocol for control of the server; the message contents will be the same as those described here in any case. The messages will be transmitted via UDP.

2 Overview of TRAF and the subset described here

TRAF consists of two programs: a controlling program, called the user program, which runs on a Unix system, and a server program, which runs on a PDP-11 (or LSI-11) under MOS. One user program can control many instances of the server. The user program communicates with the servers via TCP connections, so the servers do not have to be on the same net as the user.

The user program provides three basic services:

1. It accepts and executes the user's commands, entered at a terminal. Often these commands cause the user program to forward commands to a server.

2. It displays the results of its actions, messages received from the servers, and the status of the message generation activity it controls.

3. It performs various file handling functions, such as storing job definitions and command files.

The Server program provides three basic services:

1. Packet echo delay or TCP connection open/close delay jobs. These jobs generate messages.

2. Echo jobs. These just echo packets after a TCP connection has been established.

3. Special TCP and UDP ports which act as raw echoes or sinks (depending on the port number), without the need to open any connections.

A full TRAF server will perform these functions and also gather data about the delays it sees. This data is then sent back to the user program. This specification deals with message generation but not with data gathering. It describes a subset of the message generation commands which will allow control by the existing user program.

3 The User-Server Interaction

When the server starts up it listens on TCP port 51 (decimal). The user establishes a connection on this port and this causes the server to set up a process called the session, which then accepts commands from the user program, and sends responses back to it, over the TCP connection.

Servers without a TCP will listen on UDP port 51 (decimal) for the messages from the user program. The first message from the user program will be a "Set time and date" message (see below).

In addition, the user and server send echo requests, and the appropriate responses, to each other every 30 seconds, to check that the connection is still working. If either end times out an echo request, then it closes the connection. The timeout time is 2 minutes.

A session can be in a number of states, and state transitions are caused by user commands. The states used in this specification are:

- Defined
- Started
- Aborted
- Ended
- Active

A session is active if it has active jobs running (see below). The following is the state transition table:

SESSION STATE

COMMAND	Defined	Started	Aborted	Active
Start	started	error s3	started	error s3
Abort	aborted	aborted	error s9	aborted
End	ended	ended	ended	ended

Where an error is indicated, no state transition takes place. A session enters the Defined state when the TCP connection is established. A session that is ended goes away completely, so there are no transitions from the Ended state.

Details of command formats are given in section 4. If an active session is aborted, all jobs within that session are ended. (See below.)

A session creates processes called jobs to do the message generation. The full TRAF server allows multiple jobs to be running simultaneously within a session, but this is not

essential to the operation of the system; one only will do.

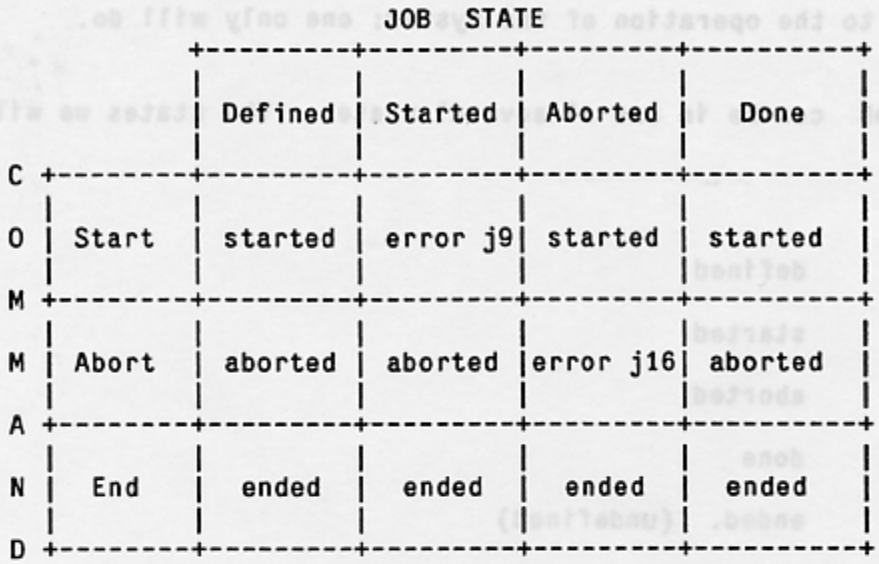
A job can be in one of several states. The states we will use are:

Job defined
started
aborted
done
ended. (undefined)

As with sessions, job state transitions are caused by user commands. The commands we will use are:

- o job start
- o job abort
- o job end

The state transition table is given below. A job enters the Done state when it has run to completion, and when a job is ended it goes away altogether, i.e. its resources are freed; so, as with sessions, there are no transitions from the Ended state.



A job is initially defined by an "assign and define job" command. The argument list for this command includes a job number, called the user job number, assigned to it by the user program. The server assigns its own job number to the job and returns that job number to the user program in its acknowledgement; all further references to that job in user-server or server-user messages use the job number assigned by the server.

4 Message formats

Bits within fields are numbered according to the PDP-11 convention, i.e. bit 0 is the least significant bit.

All communication between user and server programs takes

place over a TCP connection. The server listens on port 51 (decimal) and the user host initiates the connection.

The first octet of every message is the opcode. This opcode defines the operation required and determines the structure of the rest of the message. The next two sections list the opcodes, their meanings, and the argument structure (if any) of the messages. All numbers are in decimal.

4.1 User to Server Messages

Opcode	Meaning	Argument Structure
0	echo request	octet to be echoed
1	echo response	echoed octet
2	set time and date	version number, 16 bits Unix format time and date, 32 bits
5	return job state information	job number, 1 octet
6	return session state	
7	start session	
11	abort session	
12	end session	
13	assign and define job	job definition block, see below.
14	start job	job number, 1 octet
18	abort job	job number, 1 octet
19	end job	job number, 1 octet
20	error reset	job number, 1 octet

4.2 Server to user Messages

Opcode	Meaning	Argument Structure
0	echo request	octet to be echoed
1	echo response	echoed octet
2	set time and date	
5	job state information	job number, 1 octet job state vector, see below.
6	session state	session state vector, see below.
7	session started	
11	session aborted	
12	session ended	
13	job assigned	server job number, 1 octet user job number, 16 bits
14	job started	job number, 1 octet
18	job aborted	job number, 1 octet
19	job ended	job number, 1 octet
21	job complete	job number, 1 octet
22	error	job number, 1 octet number of characters in error string, 16 bits error string.

Note that server-user opcode 19 (job ended) is a response to user-server opcode 19 (end job), while opcode 21 is an asynchronous message indicating that the job has run to its end.

The response to any opcode not included here is an error message for that opcode with the string "Limited server only". See error messages, below.

4.3 Argument Structures

These are the formats for the arguments whose format is not fully specified in the preceding sections.

4.3.1 Job Definition Block

Field Name	Size (Octets)	See Note
job option word	2	1
job duration, low order	2	2
job duration, high order	2	2
destination net	1	
destination host	1	
destination tcp id	1	
destination logical host	1	
number of outstanding packets	2	
unused in this specification	2	
lower limit of packet size	2	
upper limit of packet size	2	
unused in this specification	10	
user job number	2	3
job comment	80	4

Notes:

1. The job option word is a 16 bit field. The bit settings relevant here are: (bit 0 is the least significant)

0	must be set. Indicates an echo delay job.
1	If set, the server will send UDP datagrams; if not, TCP datagrams.
2	If set, indicates that messages are to be sent to the sink port on the destination host; if unset, to the echo port.
3	Not used in this specification.
4	Not used in this specification.

2. Job duration is in units of 0.1 second.

3. The user job number is sent back with the "job

assigned" message, along with the server job number; thereafter, only the server job number is used.

4. The job comment is not used by this limited server.

4.3.2 Job State Vector

Field Name	Size(Octets)	See Note
Job state word	2	1
TCP state word	2	2
(Internal information)	26	
low order of time left to run	2	3
high order of time left to run	2	3
unused in this specification	2	
echo delay data queue	4	4
unused in this specification	8	

Notes:

1. The job state word has the following bit settings:

0	job defined
1	job started
3	job aborted
5	job done
10	job is an echo delay job
11	job is currently active
13	error sent (waiting for reset)
14	restart process after tcp connection closes

Bits not specified are not part of this specification subset.

2. The TCP state word has the following bit settings.

0	open wait
1	send wait
2	read wait
3	close wait
4	connection open
5	abort connection
6	close connection after write
7	data successfully read at some time on connection
8	data successfully sent at some time on connection

3. The time left to run is a 32 bit quantity in units of 0.1 second.

4. Data queues consist of a pointer and an entry count. The pointer is for internal use only and need not be set; the entry count is just the number of items on the queue.

4.3.3 Session State Vector

A session state vector consists of 2 16-bit fields. The first is the session state, with bit settings as follows:

Bit	
0	session is defined (tcp connection is open)
1	session is started and idle (no active jobs)
2	session is started and active
3	
4	session is aborted.

The second field is the TCP status word, with settings as shown in 2. above. (Job status vector)

4.4 Error Messages

The high order 2 bits in the server to user upcode are used to signal an error condition (bit 7) or message condition (bit 6). When either bit is set, a string is returned with the opcode. This string has the format:

```
number of characters in string, 1 octet
string, ASCII octets
null octet
```

If the opcode is normally followed by a job number then that precedes the string. The error bit is set in the asynchronous error message (opcode 22).

If this limited server receives an opcode that it does not understand, then it will send back a message containing the opcode with the error bit set on, and an error string saying "Limited server only".

4.4.1 Session Related Errors and Messages

s1 Server identifying message. For this limited server it will be "TRAF Limited Server. Version n (<date>)".

This is sent when the server receives a time set command from the user.

s3 Session already started
s4 Session active, must be aborted
s9 Session already aborted

Errors s3, s4, s9 are sent in response to illegal state transition requests.

s12 No new jobs currently available.

This is sent when the server has no new jobs with which to satisfy an assign request.

s13 Illegal job number

This is sent if the server receives a request containing an undefined job number.

s14 Server using a newer version of the TRAF protocol.
s15 Server using an older version of the TRAF protocol

The time set command contains a version number; if this doesn't match the server's version number, s14 or s15 is returned and the TCP connection is closed.

s16 Echo request timeout, connection closed.

If the server doesn't get a response to one or its periodic echo requests within a 2 minutes, it sends this message and closes the connection.

4.4.2 Job Related Errors and Messages

j1 Connection error
j2 Dead foreign host
j3 TCP connection refused
j4 Unexpected connection close
j5 TCP/UDP buffer unavailable
j6 TCP send error

These are messages relating to errors in the job's TCP connection, if any.

j9 Job already started
j15 Job was aborted
j16 Job already aborted
j19 Job done

These are messages relating to job state transitions.

5 Echo/Sink Servers

The server will echo any TCP or UDP packets sent to port 7 by swapping the source and destination addresses and ports, and sending the packet out again. This means that, for TCP echo, no TCP connection is established to the server; in effect, the connection is made by the source of the packets to itself.

Port 9 is the sink port. UDP packets sent to port 9 should be discarded with no acknowledgement. For TCP the case is different, because even discarded packets must be acknowledged,

so it will be necessary to open a TCP connection to port 9 to use it as a sink. Servers which do not have a TCP will therefore not offer this facility.