COMMENTS ON RFC #141 (A FILE TRANSFER PROTOCOL)

1. A file transfer protocol is needed. Bushan's proposal would
satisfy a particular current need that we have, as well as short-term
envisioned needs.

2. Bushan's protocol would apear to be straight-forward in
implementation, and extensible as claimed.

3. We would like to see implementations of such protocol be
accomplished such that the file transfer program has general and
complete access to the local file storage. That is, it should be
able to access a file that it did not create. For example, if a
program or user creates a file at site X (completely independent of
the file transfer program), it would then be desirable to be able to
retrieve the file via the file transfer program. This is not a
requirement of RFC #114 but we would like to see it implemented where
possible.

4. Since implementation of a subset of transaction types is
specifically permitted, we suggest inclusion of the following
commands (in addition to append).

    insert records   within a file
    delete records   from within a file
    replace records   within a file

Although these operations are not directly supported under IBM
OS/360, we have used them with a non-standard file subsystem under
IBM OS/360 and find them quite useful.

5. In addition to retrieve and lookup, get names of files under my
access control would be useful.

6. The absence of status requests and responses is apparent.
Although this is typically a function associated with a remote job
entry (RJE) system, since the execute request is present it would
seem appropriate to inquire about the status of the process created
by the execute command. This becomes increasingly more important
where the execute is implemented as an RJE-like operation and
scheduling time of the job might be prolonged.

7. When requesting execute, the using host sends parameters upon
receipt of the rr response. Executing a task can be implemented in
several ways. The options our 360 affords are RJE at job level and
the attach macro. Our preference would be the attach macro which
immediately initiates an independent OS task within the partition of
the program issuing the attach (presumably the File Service). Such a
task normally receives parameters upon initiation and can thereafter
receive parameters from a program via some mechanism such as an event
control block. The second method requires special modifications to
the program being executed; hence, it is not desirable. Therefore,
we either need the parameters included in the execute command or will
not actually start execution until parameters are received.

8. Upon abnormal termination, one should include part or all of the
spurious request as well as an identify- ing code to facilitate
precise error recognition.

9. We would be interested in the outcome of the MIT/ Harvard
experiments with the RFC #114 protocol. What were the pitfalls,
etc.?

        [This RFC was put into machine readable form for entry]
         [into the online RFC archives by Simone Demmel 4/97]