IEN: 102
Comparison of AUTODIN FTP with NI FTP
4 May 1979

H. Forsdick
A. McKenzie
BBN

see IEN 100-101

Comparison of AUTODIN FTP with NI FTP

With so many FTP designs already in existence, it is reasonable to ask why it is necessary to design a new one, rather than using an existing one. Since we claim that the AUTODIN FTP has been heavily influenced by the design of the NI-FTP it is especially reasonable to ask why we couldn't use that protocol.

In fact, the AUTODIN FTP design draws heavily on the NI-FTP design; we have attempted to incorporate most of the features of the NI-FTP into the AUTODIN FTP. (We have also omitted a number of features which did not seem to us sufficiently general.) In the end, however, we were driven to choose to produce a new design rather than an adaptation for the following reasons.

1) We felt that one of the problems with previous FTPs was the need for understanding two somewhat different models of essentially the same logical operation: local file copying versus inter-host file copying. The NI-FTP has taken a step at attempting to develop standardized models of the elements of file transfer with their "conceptual file store." One of the goals set for AUTODIN FTP was to provide an integrated facility for inter-host file copying including, but not limited to access control, file data typing, resource accounting, and scheduling precedence. This goal required reconsideration from the start of some issues associated with file transfer.

2) Since AUTODIN FTP implementations will be used by a community that requires long term stability and maintainability, as well as the potential for adding protocol extensions, we felt that the benefits of protocol uniformity could and should be exploited. In several different areas we tried to develop a single mechanism where other protocols utilized two or more, depending on the nature and placement of a particular task. For example, in AUTODIN FTP, there is:

   * One integrated mechanism for handling the data transfer needs of files of data as well as the information exchanged in pre-file transfer negotiation.

* One unified mechanism for handling commands and responses,
   each of which may reference parameters. The same
   mechanism is used for command extensions as well as
   parameter extensions.

   Developing such unified approaches has required rethinking
   most issues addressed by previous FTPs.

3) The NI-FTP chose a very restricted capability for
    intermixing commands and data during the transfer phase.
    The rigid restriction to two byte command sequences made it
    difficult or impossible to provide for functions known to be
    desirable such as partial transfer or Tabstop setting, or to
    allow future protocol extensions. These abilities were
    traded for minimization of overhead - the NI-FTP may
    transfer 63 data bytes with only one header byte. By using
    a much more general structure the AUTODIN FTP incurs a fixed
    overhead of 3 bytes rather than one, but can send as many as
    255 bytes per segment rather than 63. However, we believe
    that even if the normal transmitted segment is shorter than
    the breakeven point (189 bytes), the increased flexibility
    will be worth the cost.

4) We are convinced that in the future a major use of FTP will
    be to handle the task of updating distributed data bases,
    and that performing this function economically will require
    partial file transfers and/or multiple file transfers. We
    felt that it would be difficult or impossible to graft such
    functions onto the NI-FTP, which is oriented to the transfer
    of exactly one complete file per session.

5) Based on our experience with ARPANET FTP, we feel strongly
    that a single-host operating system view of access control
    which requires explicit login to each host involved in a
    file transfer, is a major impediment to effective file
    sharing. We wanted to produce an FTP design which supported
    and encouraged the construction of a file transfer system
    which incorporated a more flexible access control system
    which could be controlled by anyone who owned a file and
    wanted to share it. While access control and the underlying
    file transport mechanism seem orthogonal, in an effective
    system they need to be addressed from within one integrated
    framework, not added together as an afterthought. The
    approach to access control so central to our thinking was
    far from the orientation of the NI-FTP. A different
    underlying design seemed appropriate.

6) We believe that the problem of moving data around in a
    network environment is not the unique province of FTP, and
    are suspicious of the utility of indefinite layering of even
    higher level protocols on top of FTP. For example, many
    individuals in the ARPANET community feel that "mail" should
    be independent of FTP. From this viewpoint, we do believe
    in the utility of semi-independent, standardized mechanisms
    to support data movement. Accordingly we wished to separate
    the details of the data movement mechanisms from the
    mechanisms used to describe file handling; this we have
    done with the specification of DTP. We envision that a
    library of DTP routines might be incorporated into other
    protocols as well as FTP. As a side benefit, we are able to
    use the DTP mechanisms for transporting FTP parameters,
    replacing the ad-hoc structure of other FTPs (including
    NI-FTP) with a uniform and more general structure.

7) Although many FTP designs, including the ARPANET FTP,
    include the concept of three-party transfer, the mechanisms
    generally appear to be added as an afterthought. We believe
    that when access control problems are solved, as we have
    tried to provide for in our design, three party transfers
    will become a significant FTP use. Accordingly our design
    treats the three party model as the normal case, which can
    reduce easily to the currently-common two party case.

8) It is almost inevitable that with the passage of time
    various groups will wish to extend any protocol in one
    direction or another. Some extensions will be useful to
    everyone; others will be useful to only a subset of the
    systems involved. However, with the exception of the
    ARPANET "new" TELNET protocol, the provision of an extension
    mechanism is weak, if not totally ignored in most high level
    protocols. The AUTODIN FTP explicitly addresses the need
    for protocol extension mechanisms.