                          Host Names On-line

Now that we finally have an official list of host names, it seems
about time to put an end to the absurd situation where each site
on the network must maintain a different, generally out-of-date,
host list for the use of its own operating system or user
programs.

For example, each of the TENEX sites to which I have access
(SRI-ARC, BBN-TENEX, USC-ISI, and PARC-MAXC) has a slightly
different mapping between host names and host addresses: none
is complete, and I believe each one differs in some way from
the official List.

Since the NIC has responsibility for maintaining the official
list, lt seems appropriate for them to maintain an on-line file,
accessible to anyone, which Lists names and host addresses (and
certain other information which I will suggest in a moment) in an
easily machine-readable form.

This rules out, in my opinion, providing this information only
in the form of an NLS structured file, since there are no
facilities for accessing such files from the network and since
many sites would not want to accommodate themselves to this
structure even if there were.

The file I have in mind would be devoted principally to that
information needed by programs, as opposed to people, since the ;
former want their information in compact, easily parsed form,
whereas the latter appreciate more verbose expression and more
sophisticated facilities for browsing or querying. Therefore, I
propose that the following information be included in such a file:

Of course, the official name and host address for each host.
This would be the primary content of each entry.

Some information about the options of the various protocols
supported by the host, including (for FTP) the preferred byte
size and (for TELNET) the preferred duplex mode. The former
can have an enormous effect on the efficiency of file
transfers. Since the new TELNET allows negotiation of options,
the list need not be complete or accurate.

The function o f the host vis-a-vis the network (user, server,
TIP, etc.). This may aid NCPs in deciding whether to poll the
host or give useful information for statistical purposes (e.g.
I would like to make my NCP collect statistics on traffic with
TIPs vs. other hosts).
Since the file will be generated centrally by a single program,
but used widely by a variety of programs, it follows that its
format should be organized for ease of interrogation at the
expense of ease of construction. I feel a reasonable way to
achieve this is to store it as an ASCII text file with the logical
structure of a "property list".

In other words, aside from the two basic facts in each entry
(name and address), the information will be expressed in the
form of <attribute, value> pairs rather than having the
attribute be recognized by format, position, etc.

l don't believe it matters a great deal exactly how this file is
formatted, so I will make a suggestion in the hope that no one
cares enough to protest it. (This has never worked before in the
history of the network, but it' s still worth a try) The
following is the proposed syntax of the file.

<host-name-file> ::= <entry> | <host-name-file> <entry>

<entry> ::= <data-part> <end-of-line>

Note that this produces a blank line after the <data-part>.
<data-part> ::= <basic-part> | <data-part> <attribute-item>
<basic-part> ::= <host-name> , <host-address> <end-of-line>
<attribute-item> ::= <attribute-name> = <attribute-value>
<end-of-line>
This leaves the following terms undefined:
<end-of-line>: I don't know what end-of-line indication is in
favor in the network community these days. I personally favor
carriage-return followed by line-feed. TENEX tends to use the
single character octal 37, which is totally non-standard and
inappropriate for this application.

<host-name>: an official host name as specified in the recent
RFC 597 (NIC 20826) by NJN and JAKE. It is my understanding
that these names are restricted to letters, digits, hyphens,
and parentheses (including the network name).

<host-address>: a decimal host address, relative to its own
network (I would assume). There has been no general discussion
of multi-network addressing -- although there is apparently an
unpublicized Internetworking Protocol experiment in progress --
and some other convention may be more desirable.
<attribute-name>: an arbitrary name containing only letters,
digits, and hyphens. We will have to agree on some names like
BEST-FTP-BYTE-SIZE (?), but I am willing to let the NIC pick
them.

<attribute-value>: an arbitrary string not containing
<end--of-line>, whose interpretation depends in general on the
attribute. For example, there might be an attribute SERVERS
whose value was a list of the servers customarily run by the
site.

The following are some specific attributes that I think would be
worthwhile:

NICKNAMES -- value is a list of acceptable nicknames for the
host. Any system that provides name-to-address translation is
encouraged (although of course not required) to accept these
names as alternatives to the official host name.

FTP-BYTE-SIZES -- value is a list of the byte sizes supported
by the FTP server. The first byte size is the one which leads
to the least computational overhead (e.g. 36 for PDP-1O's, 32
for 36O's).

ECHOING -- value is L or R depending on whether the host
expects the terminal to echo (Remote) or expects to do its own
echoing (Local).

Note that no attribute is actually required and that the values
under a given attribute need not be complete. In other words,
this list is meant not to replace option negotiation,
word-of-mouth, or any other means bo which one host discovers
the properties of another, but merely to provide an alternate
source of information which can be accessed in a simple and
uniform way.

I realize that there is a time-honored pitfall associated with
suggestions such as the present one: it represents a specific
solution to a specific problem, and as such may not be compatible
with or form a reasonable basis for more general solutions to more
general problems. However, (1) this particular problem has been
irking me and others I have spoken to for well over a year, and it
is really absurd that it should have gone unsolved this Long; (2)
no one seems particularly interested in solving any more general
problem.

Except the Datacomputer: PLEASE, if there is an easy way to
accomplish the same function through the Datacomputer, someone
write un RFC specifying it.