On Names, Addresses and Routings

DESTINATIONs and SOURCEs of communication have names. These destinations and sources are processes which exist outside the communication subnet (e.g., in HOSTs) or inside it (e.g., in GATEWAYs, NET-NODEs, etc).

Names are not necessarily unique to destinations. Certain processes may have more than a single name. There are two kinds of names, names which specify a unique process and names which specify a set of processes (e.g., by capability).

A name tells WHO (or WHAT) the process is.

It is conceivable the names are arbitrary alphanumeric string suited for human readability.

In addition to names, processes have addresses which indicate WHERE they are. Again, it is possible that several different names share the same address, and also that the same name has several addresses.

In order to move messages from sources to destinations, not only the address of the destination (i.e., where it is) has to be known, but also HOW-TO-GET-THERE. This knowledge is the ROUTING.

It is, obviously, possible that the same address can be reached by more than one route.

These beautiful definitions of NAMEs (what), ADDRESSes (where) and ROUTING (how-to-get-there) are borrowed from John Shoch.

To summarize: destinations and sources have names. Names are "converted" into addresses, and addresses into routings. However, none of the above mappings is a one-to-one correspondence.

Names have to be unambiguous within a certain environment.

From outside of this environment, the environment, too, has to be specified in order to complete the name specification i.e., to disambiguate it completely. This is the essence of hierarchical naming. A very similar argument holds for addresses. Within a certain environment addresses are unique, outside of it, the environment has to

be specified ("addressed") too. Obviously, environments have names/addresses which are unique within some bigger super-environment, out of which, the super-environment has to be specified, too. And so on.

Note that in general address (and names) are hierarchical. The non-hierarchical case (or the "flat" situation) is a special case of hierarchy, with depth equal to one, and width equal to the total number of addresses.

The mapping of names to addresses cannot be computed, and can be resolved only by reference to directories. Hence names from which addresses can be computed are not names in the true sense, but addresses in disguise.

These directories may be implemented in various ways, ranging from off-line distributed directories (like paper based telephone directories) to online centralized services (like 411) even by or utilizing some broadcasting techniques.

The sole purpose of addressing is to support routing. The address which tells where the destination is, is only important for determining the routing (i.e., how to get there).

The key question is who performs the mapping of addresses into routing.

Several avenues can be pursued. (1) Sources supplying the entire routing from source to destination (source-routing), or (2) source supplying only the name (or address) of the destination, and the communication subsystem performing the routing (communication-routing), and (3) a hybrid of the above, by the source supplying some intermediate destinations along the route, such that the communication subsystem can perform the routing between these given destinations.

Note that (3) is a generalization of both (1) and (2). If done right, it should have the advantages of both, else it may have their combined disadvantages.

There are too many (obvious) problems associated with a complete source-routing, (1), which are amplified in dynamically changed networks. Basically, the user is faced with the need to specify completely the routing whenever a message is sent. Since most of the information needed for finding the routing is originated in the communication subsystem, this is an undue burden which should be minimized.

On the other hand the other extreme, the communication-routing, (2), is not problem-free either.

In the case of flat address-spaces too much knowledge has to be maintained, with all the problems associated with this requirement. Therefore, it is preferred to implement hierarchical addressing and hierarchical routing, which matches it in a unique way. As a matter of

fact, it seems that the most "natural" avenue is to follow the well-debugged paradigm of the telephone system, and have both hierarchical names, which are uniquely mapped into hierarchical addresses.

One of the bigger drawbacks of this scheme is that it will not support an application of INTERnet fast facilities in order to expedite INTRAnet traffic, like using satellite links (which belong to SATnet) in order to expedite internal coast-to-coast ARPAnet traffic.

Since the flat address space is a special case of the hierarchical one, we would like to treat all address spaces as hierarchical ones, and consider the argument of flat vs. hierarchical as optimizing the depth/width combination.

The disadvantage of a big flat name/address subspace (as opposed to a hierarchical one) is that it requires that large table of addresses be maintained, that names/addresses be cleared before being put into use and that everyone knows about everyone, a requirement which is not feasible to fulfill when the dimension of this space keeps increasing.

An example for a very flat name space is the social security number, as opposed to telephone numbers and mailing addresses. Note that this space is a subspace of a bigger space which includes also SSNs (or other unique IDs) of people in other countries.

Note that both telephone numbers and mailing addresses are a kind of routing of the 3rd flavor. It is supplied by the source, and may have as much (or as little) information to get to the environment where the next piece of routing is nonambiguous.

An example of source-routing, (1), is calling from a centrex system to a centrex system in another area code.

An example of communication-routing, (2), is the interHOST traffic within the ARPAnet.

An example of destination-routing, (3), is an operator-assisted person to person call without knowing the number. For example, one has to tell the communication subsystem (here, a sequence of operators) that the other person is (a) in Massachusetts, (b) in Cambridge, (c) at MIT, and finally (d) the final destinations name. In this case the source supplied the sub-destination "Masachusetts" (or 617) but the communication system was free to choose the optimal route there.

Another example is a call which is placed from a military base to another. Normally the AUTOVON system is used, but many times at the discretion of the individuals, they use the AUTOVON system to get outside to the commercial network, and back to the destination base, where the AUTOVON system is used again for the terminal contact. This is a example of an INTRAnet (AUTOVON) call which for quality, convenience and similar reasons, uses INTERnet facilities.

Another interesting observation about the telephone systems. Telephone
addresses are of variable length. This does not mean only that
different addresses may have different lenghts, but that the same
address may have several lenghts, depending from where it is refered to.
For example inside most organizations extensions may be reached by
dialing about 4 digits only. from outside the organization, but still
in the neighborhood 7 digits are usually needed, from a larger
neighborhood it is 8, then 10, and so on. This is to be expected since
telephone numbers are the basis for the hybrid routing used. Telephone
numbers are parsed sequentially, and can traverse hierarchies up and
down because of the prefixes used to indicate the "up" direction (like
the 9 for outside, and the 1 for area code).

It seems to me that we might like to implement a very similar structure
for addressing.

One way to implement such a user assisted routing is to implement a
special FORWARDER process (on a well-known PORT), whose function is to
receive messages, retrieve the "next-address" from its data and resubmit
it for transportation to the next sub-destination, while doing the right
things to acknowledgments, duplicates and the like. These processes do
not have to be implemented on all systems, but only on a certain subset,
where this capability is needed. This would eliminate the need for
making all processes able to handle variable length multilevel
addressing.

SUMMARY

The address space should be hierarchical, because it is more general
than the flat space, and mainly because it may be not practical to have
everyone knowing about all the other participants in this space.

The format of the ADDRESS should be made more general (extensible).

Routing which follows the hierarchy of the address space has
difficulties in utilizing internet resources for intranet traffic.

The distribution of all the knowledge needed to support routing which is
entirely performed by the communication subsystem may also turn out to
be impractical. Relying always on the source to supply the entire
routing has its obvbious problems.

Therefore we highly recommend that the hybrid approach will be used.

ACKNOWLEDGMENT

Recent discussions with John Shoch and David Boggs were a very
educational experience for me, and helped me to understand these issues
better. This note and John Shoch's note on the same subject (both are
prepared for the same January 1978 TCP meeting) cover similar issues,
and are basically in agreement about most issues. However, minor
differences between the them seem to justify their existance separately.