Network Working Group
Request for Comments: 819
Zaw-Sing Su (SRI)
Jon Postel (ISI)
August 1982

The Domain Naming Convention for Internet User Applications

1. Introduction

For many years, the naming convention "<user>@<host>" has served the ARPANET user community for its mail system, and the substring "<host>" has been used for other applications such as file transfer (FTP) and terminal access (Telnet). With the advent of network interconnection, this naming convention needs to be generalized to accommodate internetworking. A decision has recently been reached to replace the simple name field, "<host>", by a composite name field, "<domain>" [2]. This note is an attempt to clarify this generalized naming convention, the Internet Naming Convention, and to explore the implications of its adoption for Internet name service and user applications.

The following example illustrates the changes in naming convention:

```
ARPANET Convention:   Fred@ISIF
Internet Convention:  Fred@F.ISI.ARPA
```

The intent is that the Internet names be used to form a tree-structured administrative dependent, rather than a strictly topology dependent, hierarchy. The left-to-right string of name components proceeds from the most specific to the most general, that is, the root of the tree, the administrative universe, is on the right.

The name service for realizing the Internet naming convention is assumed to be application independent. It is not a part of any particular application, but rather an independent name service serves different user applications.

2. The Structural Model

The Internet naming convention is based on the domain concept. The name of a domain consists of a concatenation of one or more <simple names>. A domain can be considered as a region of jurisdiction for name assignment and of responsibility for name-to-address translation. The set of domains forms a hierarchy.

Using a graph theory representation, this hierarchy may be modeled as a directed graph. A directed graph consists of a set of nodes and a collection of arcs, where arcs are identified by ordered pairs of distinct nodes [1]. Each node of the graph represents a domain. An ordered pair (B, A), an arc from B to A, indicates that B is a subdomain of domain A, and B is a simple name unique within A. We will refer to B as a child of A, and A a parent of B. The directed graph that best describes the naming hierarchy is called an "in-tree", which is a rooted tree with all arcs directed towards the root (Figure 1). The root of the tree represents the naming universe, ancestor of all domains. Endpoints (or leaves) of the tree are the lowest-level domains.

```
                         U
                       / | \
                     /   |   \          U -- Naming Universe
                    ^    ^    ^         I -- Intermediate Domain
                    |    |    |         E -- Endpoint Domain
                    I    E    I
                  /   \       |
                 ^     ^      ^
                 |     |      |
                 E     E      I
                            / | \
                           ^  ^  ^
                           |  |  |
                           E  E  E
```

Figure 1
The In-Tree Model for Domain Hierarchy

The simple name of a child in this model is necessarily unique within its parent domain. Since the simple name of the child's parent is unique within the child's grandparent domain, the child can be uniquely named in its grandparent domain by the concatenation of its simple name followed by its parent's simple name.

For example, if the simple name of a child is "C1" then no other child of the same parent may be named "C1". Further, if the parent of this child is named "P1", then "P1" is a unique simple name in the child's grandparent domain. Thus, the concatenation C1.P1 is unique in C1's grandparent domain.

Similarly, each element of the hierarchy is uniquely named in the universe by its complete name, the concatenation of its simple name and those for the domains along the trail leading to the naming universe.

The hierarchical structure of the Internet naming convention supports decentralization of naming authority and distribution of name service capability. We assume a naming authority and a name server associated with each domain. In Sections 5 and 6 respectively the name service and the naming authority are discussed.

Within an endpoint domain, unique names are assigned to <user> representing user mailboxes. User mailboxes may be viewed as children of their respective domains.

In reality, anomalies may exist violating the in-tree model of naming hierarchy. Overlapping domains imply multiple parentage, i.e., an entity of the naming hierarchy being a child of more than one domain. It is conceivable that ISI can be a member of the ARPA domain as well as a member of the USC domain (Figure 2). Such a relation constitutes an anomaly to the rule of one-connectivity between any two points of a tree. The common child and the sub-tree below it become descendants of both parent domains.

```
                    U
                  / | \
                /   .   \
              .     .   ARPA
            .       .     | \
                   USC    |   \
                      \   |     .
                        \ |       .
                         ISI
```

Figure 2
Anomaly in the In-Tree Model

Some issues resulting from multiple parentage are addressed in Appendix B. The general implications of multiple parentage are a subject for further investigation.

3. Advantage of Absolute Naming

Absolute naming implies that the (complete) names are assigned with respect to a universal reference point. The advantage of absolute naming is that a name thus assigned can be universally interpreted with respect to the universal reference point. The Internet naming convention provides absolute naming with the naming universe as its universal reference point.

For relative naming, an entity is named depending upon the position of the naming entity relative to that of the named entity. A set of hosts running the "unix" operating system exchange mail using a method called "uucp". The naming convention employed by uucp is an example of relative naming. The mail recipient is typically named by a source route identifying a chain of locally known hosts linking the sender's host to the recipient's. A destination name can be, for example,

"alpha!beta!gamma!john",

where "alpha" is presumably known to the originating host, "beta" is known to "alpha", and so on.

The uucp mail system has demonstrated many of the problems inherent to relative naming. When the host names are only locally interpretable, routing optimization becomes impossible. A reply message may have to traverse the reverse route to the original sender in order to be forwarded to other parties.

Furthermore, if a message is forwarded by one of the original recipients or passed on as the text of another message, the frame of reference of the relative source route can be completely lost. Such relative naming schemes have severe problems for many of the uses that we depend upon in the ARPA Internet community.

4. Interoperability

To allow interoperation with a different naming convention, the names assigned by a foreign naming convention need to be accommodated. Given the autonomous nature of domains, a foreign naming environment may be incorporated as a domain anywhere in the hierarchy. Within the naming universe, the name service for a domain is provided within that domain. Thus, a foreign naming convention can be independent of the Internet naming convention. What is implied here is that no standard convention for naming needs to be imposed to allow interoperations among heterogeneous naming environments.

For example:

There might be a naming convention, say, in the FOO world, something like "<user>%<host>%<area>". Communications with an entity in that environment can be achieved from the Internet community by simply appending ".FOO" on the end of the name in that foreign convention.

John%ISI-Tops20-7%California.FOO

Another example:

One way of accommodating the "uucp world" described in the last section is to declare it as a foreign system. Thus, a uucp name

"alpha!beta!gamma!john"

might be known in the Internet community as

"alpha!beta!gamma!john.UUCP".

Communicating with a complex subdomain is another case which can be treated as interoperation. A complex subdomain is a domain with complex internal naming structure presumably unknown to the outside world (or the outside world does not care to be concerned with its complexity).

For the mail system application, the names embedded in the message text are often used by the destination for such purpose as to reply to the original message. Thus, the embedded names may need to be converted for the benefit of the name server in the destination environment.

Conversion of names on the boundary between heterogeneous naming environments is a complex subject. The following example illustrates some of the involved issues.

For example:

A message is sent from the Internet community to the FOO environment. It may bear the "From" and "To" fields as:

```
From: Fred@F.ISI.ARPA
To:   John%ISI-Tops20-7%California.FOO
```

where "FOO" is a domain independent of the Internet naming environment. The interface on the boundary of the two environments may be represented by a software module. We may assume this interface to be an entity of the Internet community as well as an entity of the FOO community. For the benefit of the FOO environment, the "From" and "To" fields need to be modified upon the message's arrival at the boundary. One may view naming as a separate layer of protocol, and treat conversion as a protocol translation. The matter is complicated when the message is sent to more than one destination within different naming environments; or the message is destined within an environment not sharing boundary with the originating naming environment.

While the general subject concerning conversion is beyond the scope of this note, a few questions are raised in Appendix D.

5. Name Service

Name service is a network service providing name-to-address translation. Such service may be achieved in a number of ways. For a simple networking environment, it can be accomplished with a single central database containing name-to-address correspondence for all the pertinent network entities, such as hosts.

In the case of the old ARPANET host names, a central database is duplicated in each individual host. The originating module of an application process would query the local name service (e.g., make a system call) to obtain network address for the destination host. With the proliferation of networks and an accelerating increase in the number of hosts participating in networking, the ever growing size, update frequency, and the dissemination of the central database makes this approach unmanageable.

The hierarchical structure of the Internet naming convention supports decentralization of naming authority and distribution of name service capability. It readily accommodates growth of the naming universe. It allows an arbitrary number of hierarchical layers. The addition of a new domain adds little complexity to an existing Internet system.

The name service at each domain is assumed to be provided by one or more name servers. There are two models for how a name server completes its work, these might be called "iterative" and "recursive".

For an iterative name server there may be two kinds of responses. The first kind of response is a destination address. The second kind of response is the address of another name server. If the response is a destination address, then the query is satisfied. If the response is the address of another name server, then the query must be repeated using that name server, and so on until a destination address is obtained.

For a recursive name server there is only one kind of response -- a destination address. This puts an obligation on the name server to actually make the call on another name server if it can’t answer the query itself.

It is noted that looping can be avoided since the names presented for translation can only be of finite concatenation. However, care should be taken in employing mechanisms such as a pointer to the next simple name for resolution.

We believe that some name servers will be recursive, but we don’t believe that all will be. This means that the caller must be prepared for either type of server. Further discussion and examples of name service is given in Appendix C.

The basic name service at each domain is the translation of simple names to addresses for all of its children. However, if only this basic name service is provided, the use of complete (or fully qualified) names would be required. Such requirement can be unreasonable in practice. Thus, we propose the use of partial names in the context in which their uniqueness is preserved. By construction, naming uniqueness is preserved within the domain of a common ancestry. Thus, a partially qualified name is constructed by omitting from the complete name ancestors common to the communicating parties. When a partially qualified name leaves its context of uniqueness it must be additionally qualified.

The use of partially qualified names places a requirement on the Internet name service. To satisfy this requirement, the name service at each domain must be capable of, in addition to the basic service, resolving simple names for all of its ancestors (including itself) and their children. In Appendix B, the required distinction among simple names for such resolution is addressed.

6. Naming Authority

Associated with each domain there must be a naming authority to assign simple names and ensure proper distinction among simple names.

Note that if the use of partially qualified names is allowed in a sub-domain, the uniqueness of simple names inside that sub-domain is insufficient to avoid ambiguity with names outside the subdomain. Appendix B discusses simple name assignment in a sub-domain that would allow the use of partially qualified names without ambiguity.

Administratively, associated with each domain there is a single person (or office) called the registrar. The registrar of the naming universe specifies the top-level set of domains and designates a registrar for each of these domains. The registrar for any given domain maintains the naming authority for that domain.

7. Network-Oriented Applications

For user applications such as file transfer and terminal access, the remote host needs to be named. To be compatible with ARPANET naming convention, a host can be treated as an endpoint domain.

Many operating systems or programming language run-time environments provide functions or calls (JSYSs, SVCs, UUOs, SYSs, etc.) for standard services (e.g., time-of-day, account-of-logged-in-user, convert-number-to-string). It is likely to be very helpful if such a function or call is developed for translating a host name to an address. Indeed, several systems on the ARPANET already have such facilities for translating an ARPANET host name into an ARPANET address based on internal tables.

We recommend that this provision of a standard function or call for translating names to addresses be extended to accept names of Internet convention. This will promote a consistent interface to the users of programs involving internetwork activities. The standard facility for translating Internet names to Internet addresses should include all the mechanisms available on the host, such as checking a local table or cache of recently checked names, or consulting a name server via the Internet.

8. Mail Relaying

Relaying is a feature adopted by more and more mail systems. Relaying facilitates, among other things, interoperations between heterogeneous mail systems. The term "relay" is used to describe the situation where a message is routed via one or more intermediate points between the sender and the recipient. The mail relays are normally specified explicitly as relay points in the instructions for message delivery. Usually, each of the intermediate relays assume responsibility for the relayed message [3].

> A point should be made on the basic difference between mail relaying and the uucp naming system. The difference is that although mail relaying with absolute naming can also be considered as a form of source routing, the names of each intermediate points and that of the destination are universally interpretable, while the host names along a source route of the uucp convention is relative and thus only locally interpretable.

The Internet naming convention explicitly allows interoperations among heterogeneous systems. This implies that the originator of a communication may name a destination which resides in a foreign system. The probability is that the destination network address may not be comprehensible to the transport system of the originator. Thus, an implicit relaying mechanism is called for at the boundary between the domains. The function of this implicit relay is the same as the explicit relay.

9. Implementation

The Actual Domains

The initial set of top-level names include:

ARPA

This represents the set of organizations involved in the Internet system through the authority of the U.S. Defense Advanced Research Projects Agency. This includes all the research and development hosts on the ARPANET and hosts on many other nets as well. But note very carefully that the top-level domain "ARPA" does not map one-to-one with the ARPANET -- domains are administrative, not topological.

Transition

In the transition from the ARPANET naming convention to the Internet naming convention, a host name may be used as a simple name for an endpoint domain. Thus, if "USC-ISIF" is an ARPANET host name, then "USC-ISIF.ARPA" is the name of an Internet domain.

10. Summary

A hierarchical naming convention based on the domain concept has been adopted by the Internet community. It is an absolute naming convention defined along administrative rather than topological boundaries. This naming convention is adaptive for interoperations with other naming conventions. Thus, no standard convention needs to be imposed for interoperations among heterogeneous naming environments.

This Internet naming convention allows distributed name service and naming authority functions at each domain. We have specified these functions required at each domain. Also discussed are implications on network-oriented applications, mail systems, and administrative aspects of this convention.

APPENDIX A

The BNF Specification

We present here a rather detailed "BNF" definition of the allowed form for a computer mail "mailbox" composed of a "local-part" and a "domain" (separated by an at sign). Clearly, the domain can be used separately in other network-oriented applications.

<mailbox> ::= <local-part> "@" <domain>

<local-part> ::= <string> | <quoted-string>

<string> ::= <char> | <char> <string>

<quoted-string> ::= """ <qtext> """

<qtext> ::= "\" <x> | "\" <x> <qtext> | <q> | <q> <qtext>

<char> ::= <c> | "\" <x>

<domain> ::= <naming-domain> | <naming-domain> "." <domain>

<naming-domain> ::= <simple-name> | <address>

<simple-name> ::= <a> <ldh-str> <let-dig>

<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>

<let-dig> ::= <a> | <d>

<let-dig-hyp> ::= <a> | <d> | "-"

<address> :: = "#" <number> | "[" <dotnum> "]"

<number> ::= <d> | <d> <number>

<dotnum> ::= <snum> "." <snum> "." <snum> "." <snum>

<snum> ::= one, two, or three digits representing a decimal integer value in the range 0 through 255

<a> ::= any one of the 52 alphabetic characters A through Z in upper case and a through z in lower case

<c> ::= any one of the 128 ASCII characters except <s> or <SP>

<d> ::= any one of the ten digits 0 through 9

<q> ::= any one of the 128 ASCII characters except CR, LF, quote ("), or backslash (\)

<x> ::= any one of the 128 ASCII characters (no exceptions)

<s> ::= "<", ">", "(", ")", "[", "]", "\", ".", ",", ";", ":", "@", """, and the control characters (ASCII codes 0 through 31 inclusive and 127)

Note that the backslash, "\", is a quote character, which is used to indicate that the next character is to be used literally (instead of its normal interpretation). For example, "Joe\,Smith" could be used to indicate a single nine character user field with comma being the fourth character of the field.

The simple names that make up a domain may contain both upper and lower case letters (as well as digits and hyphen), but these names are not case sensitive.

Hosts are generally known by names. Sometimes a host is not known to the translation function and communication is blocked. To bypass this barrier two forms of addresses are also allowed for host "names". One form is a decimal integer prefixed by a pound sign, "#". Another form, called "dotted decimal", is four small decimal integers separated by dots and enclosed by brackets, e.g., "[123.255.37.2]", which indicates a 32-bit ARPA Internet Address in four 8-bit fields. (Of course, these numeric address forms are specific to the Internet, other forms may have to be provided if this problem arises in other transport systems.)

APPENDIX B

An Aside on the Assignment of Simple Names

In the following example, there are two naming hierarchies joining at the naming universe 'U'. One consists of domains (S, R, N, J, P, Q, B, A); and the other (L, E, F, G, H, D, C, K, B, A). Domain B is assumed to have multiple parentage as shown.

```
                                U
                              /   \
                            /       \
                          J           L
                        /               \
                      N                   E
                    /   \               /   \
                  R       P           D       F
                /           \         | \      \
              S               Q       C  (X)     G
                                \   /   \          \
                                  B       K          H
                                /
                              A
```

Figure 3
Illustration of Requirements for the Distinction of Simple Names

Suppose someone at A tries to initiate communication with destination H. The fully qualified destination name would be

    H.G.F.E.L.U

Omitting common ancestors, the partially qualified name for the destination would be

    H.G.F

To permit the case of partially qualified names, name server at A needs to resolve the simple name F, i.e., F needs to be distinct from all the other simple names in its database.

To enable the name server of a domain to resolve simple names, a simple name for a child needs to be assigned distinct from those of all of its ancestors and their immediate children. However, such distinction would not be sufficient to allow simple name resolution at lower-level domains because lower-level domains could have multiple parentage below the level of this domain.

In the example above, let us assume that a name is to be assigned to a new domain X by D. To allow name server at D to resolve simple names, the name for X must be distinct from L, E, D, C, F, and J. However, allowing A to resolve simple names, X needs to be also distinct from A, B, K, as well as from Q, P, N, and R.

The following observations can be made.

Simple names along parallel trails (distinct trails leading from one domain to the naming universe) must be distinct, e.g., N must be distinct from E for B or A to properly resolve simple names.

No universal uniqueness of simple names is called for, e.g., the simple name S does not have to be distinct from that of E, F, G, H, D, C, K, Q, B, or A.

The lower the level at which a domain occurs, the more immune it is to the requirement of naming uniqueness.

To satisfy the required distinction of simple names for proper resolution at all levels, a naming authority needs to ensure the simple name to be assigned distinct from those in the name server databases at the endpoint naming domains within its domain. As an example, for D to assign a simple name for X, it would need to consult databases at A and K. It is, however, acceptable to have simple names under domain A identical with those under K. Failure of such distinct assignment of simple names by naming authority of one domain would jeopardize the capability of simple name resolution for entities within the subtree under that domain.

APPENDIX C

Further Discussion of Name Service and Name Servers

The name service on a system should appear to the programmer of an application program simply as a system call or library subroutine. Within that call or subroutine there may be several types of methods for resolving the name string into an address.

First, a local table may be consulted. This table may be a complete table and may be updated frequently, or it may simply be a cache of the few latest name to address mappings recently determined.

Second, a call may be made to a name server to resolve the string into a destination address.

Third, a call may be made to a name server to resolve the string into a relay address.

Whenever a name server is called it may be a recursive server or an interactive server.

If the server is recursive, the caller won't be able to tell if the server itself had the information to resolve the query or called another server recursively (except perhaps for the time it takes).

If the server is iterative, the caller must be prepared for either the answer to its query, or a response indicating that it should call on a different server.

It should be noted that the main name service discussed in this memo is simply a name string to address service. For some applications there may be other services needed.

For example, even within the Internet there are several procedures or protocols for actually transferring mail. One need is to determine which mail procedures a destination host can use. Another need is to determine the name of a relay host if the source and destination hosts do not have a common mail procedure. These more specialized services must be specific to each application since the answers may be application dependent, but the basic name to address translation is application independent.

APPENDIX D

Further Discussion of Interoperability and Protocol Translations

The translation of protocols from one system to another is often quite difficult. Following are some questions that stem from considering the translations of addresses between mail systems:

What is the impact of different addressing environments (i.e., environments of different address formats)?

It is noted that the boundary of naming environment may or may not coincide with the boundary of different mail systems. Should the conversion of naming be independent of the application system?

The boundary between different addressing environments may or may not coincide with that of different naming environments or application systems. Some generic approach appears to be necessary.

If the conversion of naming is to be independent of the application system, some form of interaction appears necessary between the interface module of naming conversion with some application level functions, such as the parsing and modification of message text.

To accommodate encryption, conversion may not be desirable at all. What then can be an alternative to conversion?

GLOSSARY

address

An address is a numerical identifier for the topological location of the named entity.

name

A name is an alphanumeric identifier associated with the named entity. For unique identification, a name needs to be unique in the context in which the name is used. A name can be mapped to an address.

complete (fully qualified) name

A complete name is a concatenation of simple names representing the hierarchical relation of the named with respect to the naming universe, that is it is the concatenation of the simple names of the domain structure tree nodes starting with its own name and ending with the top level node name. It is a unique name in the naming universe.

partially qualified name

A partially qualified name is an abbreviation of the complete name omitting simple names of the common ancestors of the communicating parties.

simple name

A simple name is an alphanumeric identifier unique only within its parent domain.

domain

A domain defines a region of jurisdiction for name assignment and of responsibility for name-to-address translation.

naming universe

Naming universe is the ancestor of all network entities.

naming environment

A networking environment employing a specific naming convention.

name service

Name service is a network service for name-to-address mapping.

name server

A name server is a network mechanism (e.g., a process) realizing the function of name service.

naming authority

Naming authority is an administrative entity having the authority for assigning simple names and responsibility for resolving naming conflict.

parallel relations

A network entity may have one or more hierarchical relations with respect to the naming universe. Such multiple relations are parallel relations to each other.

multiple parentage

A network entity has multiple parentage when it is assigned a simple name by more than one naming domain.

REFERENCES

[1] F. Harary, "Graph Theory", Addison-Wesley, Reading, Massachusetts, 1969.

[2] J. Postel, "Computer Mail Meeting Notes", RFC-805, USC/Information Sciences Institute, 8 February 1982.

[3] J. Postel, "Simple Mail Transfer Protocol", RFC-821, USC/Information Sciences Institute, August 1982.

[4] D. Crocker, "Standard for the Format of ARPA Internet Text Messages", RFC-822, Department of Electrical Engineering, University of Delaware, August 1982.