X.500 and Domains

Abstract

    This RFCconsiders X.500 in relation to Internet and UK Domains.
    A basic model of X.500 providing a higher level and more
    descriptive naming structure is emphasised. In addition, a
    mapping of domains onto X.500 is proposed, which gives a range of
    new management and user facilities over and above those currently
    available. This specification proposes an experimental new
    mechanism to access and manage domain information on the Internet
    and in the UK Academic Community. There is no current intention
    to provide an operational replacement for DNS.

1 The Domain Name System

The Domain (Nameserver) System (DNS) provides a hierarchical resource
labelling system [Moc87a] [Moc87b] [Lar83]. Example domains are:

MIT.EDU
VENERA.ISI.EDU
CS.UCL.AC.UK

Entries usually have a single name, although pointers to entries (not
subtrees) may be provided by CNAME records. Information (resource
records) is associated with each entry. Name components are typically
chosen to be shortish (e.g., ``CS'').
RFC 822 mailbox names are closely related [Cro82]. For example:

     <S.Kille@CS.UCL.AC.UK>

The local-part of the RFC 822 mailbox can be considered as one level
lower in the domain hierarchy.

2 X.500

The OSI Directory, usually known as X.500, provides a very general
naming framework [CCI88]. A basic usage of X.500 is to provide
Organisationally Structured Names. A Schema for this is defined
within the standard. Name components will typically have longish
values. This is an example directory name represented in Tabular
form:

               Country   GB
               Organisation   University College London
               Organisational Unit Computer Science
               Common Name   Stephen E. Hardcastle-Kille

This can also be written in the ``User Friendly Name'' notation
defined in [HK91]. This syntax is used for names in the rest of this
document:

     Stephen E. Hardcastle-Kille, Computer Science,

      University College London, GB

This type of structure is termed ``organisational X.500''. This is a
subset of the general capabilities.

3 The basic model

      X.500 has as much relation to the DNS as DNS has to ARP. Paul
      Mockapetris

This is, essentially, the position adopted here. The basic model is
that organisational X.500 is providing a layer of naming at the level
above domain names. These structured names can be considered to form
a naming layer above domain names. There are the following key
differences:

 o Organisational X.500 tends to use longer and more descriptive
    values

 o The organisational X.500 DIT is slightly shallower than the DNS
    tree

 o X.500 has a richer information framework than DNS

These differences suggest that the following should NOT be done:

 o Represent X.500 information in the DNS

 o Have an algorithmic mapping between the two hierarchies

This note proposes to represent DNS information in the DIT, and to
provide for a loose coupling between the two trees. This note does
not propose an equivalencing of X.500 and Domains.

The proposed model is illustrated in Figure 1. Both an organisational
and domain structure is represented in the DIT, by use of appropriate
object classes and attribute types. A weak linkage is provided
between the two parts of the tree by use of special attributes. Here,
the linkage is 1:1, but it may be more complex for some parts of the
organisational DIT or domain namespace. The linkage is achieved by
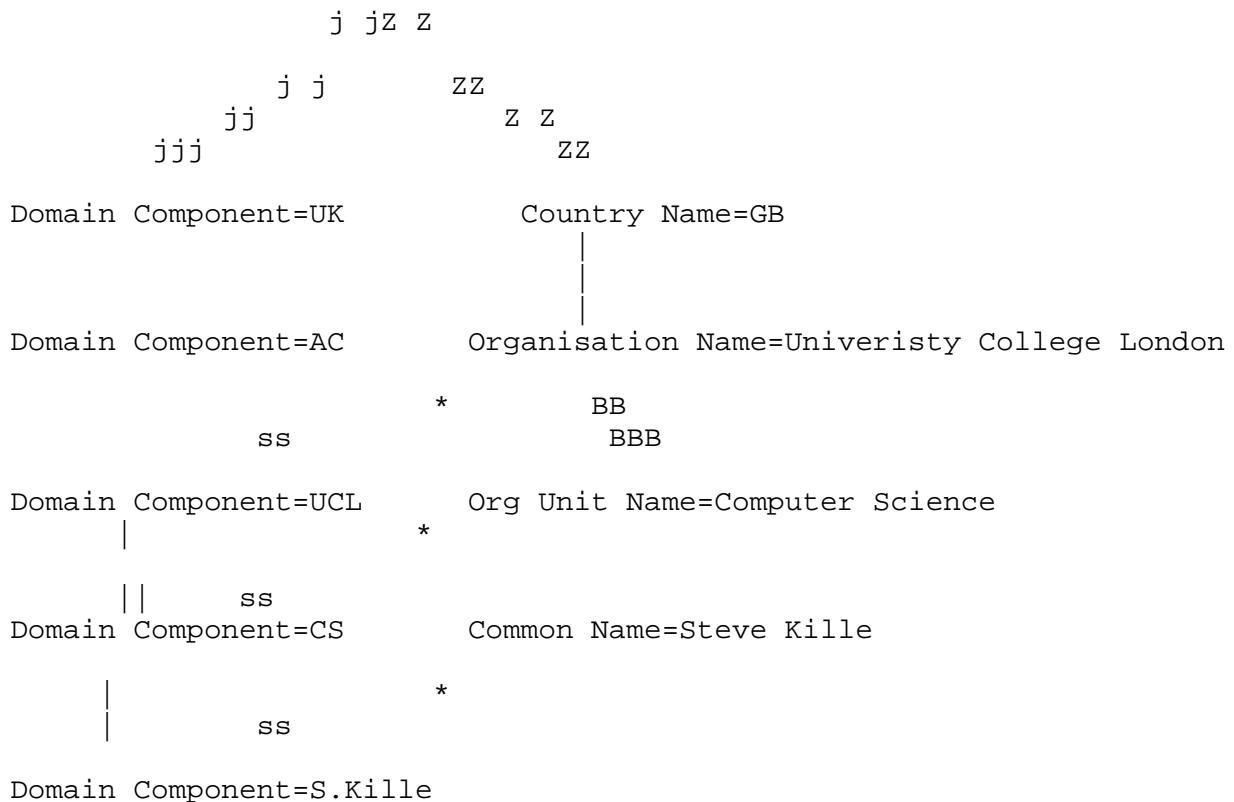use of special attributes, as described in Section 11.

```
                   j jZ Z

              j j        ZZ
            jj              Z Z
          jjj                ZZ

Domain Component=UK          Country Name=GB
                                 |
                                 |
                                 |
Domain Component=AC       Organisation Name=Univeristy College London

                     *          BB
               ss                BBB

Domain Component=UCL     Org Unit Name=Computer Science
     |                       *

     ||      ss
Domain Component=CS       Common Name=Steve Kille

     |                   *
     |          ss

Domain Component=S.Kille
```

                        Figure 1: Example X.500 tree

4 Representing Domains in X.500

Domains are at the level below X.500 names of the form illustrated in
the previous section. However, it is also possible to use X.500 in
other ways. In particular, there are benefits from representing
Domains in X.500. Note that this is very different to equivalencing,
as no attempt is made to represent X.500 information within the domain
scheme. There are the following potential advantages:

o Domain Services (DNS and NRS) could be replaced with an OSI
   service (some may not view this as an advantage). This is
   particularly attractive for OSI services, where use of a non-OSI
   directory may be inappropriate.

o For Internet sites, access to domain information (beyond MX
   records) could be provided for systems registered remotely. For
   UK Academic Community sites, access to domain information for
   domains not registered in the NRS could be given. For sites
   neither on the Internet nor in the UK Academic Community there
   will usually be even more of an advantage, as they usually have
   very limited information on domains.

o Assuming that information is downloaded from an X.500 database
   into a DNS or NRS system, the remote management facilities of
   X.500 could be used. This is possible because of the extra
   security features of X.500.

   Note: For initial work, the converse situation of information
       being mastered in Domain Databases and uploaded into the X.500
       DIT is more likely.

o User access to the domain data, and in particular searching, could
   be provided. This would allow users to browse the domain
   namespace, and to determine information associated with the
   domains.

o The X.500 framework would allow for additional management
   information to be stored, and to relate the domain names into a
   more complex structure of information. For example, this might
   allow for the managers of a system to be identified, and
   information on how to contact the manager.

   o A facility to map RFC 822 mailbox into a Directory Name (and thus
      access other user information on the basis of this key) could be
      provided. This may be useful for the user to determine
      information about a message originator.

   o This technique may be useful to facilitate introduction of
      security, as it will enable certificates to be associated with
      domains and mailboxes. This may be very useful for the privacy
      enchanced mail work [Lin89].

5 Representing Domain Names

A new attribute syntax is defined:

CaseIgnoreIA5StringSyntax ATTRIBUTE-SYNTAX
    IA5String
    MATCHES FOR EQUALITY SUBSTRINGS ORDERING

A new attribute and two new object classes are defined:

DomainComponent ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX caseIgnoreIA5StringSyntax
    SINGLE VALUE

Domain OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN -DomainComponent"
    MAY CONTAIN -AssociatedName,
        organizationName,
        organizationalAttributeSet,
        manager"

RFC822Mailbox OBJECT-CLASS
    SUBCLASS OF Domain
    MAY CONTAIN -commonName,
        surname,
        description,
        telephoneNumber,
        postalAttributeSet,
        telecommunicationAttributeSet "

Note that the attribute AssociatedName is defined in Section 11. The
manager attribute is defined in the COSINE and Internet naming
architecture [BHK91]. It allows a manager to be associated with the
domain, which is useful where the manager of the domain is different
to the manager of the object defined by the AssociatedName. This will
allow any domain to be represented in an X.500 hierarchy. The local
part of an RFC 822 mailbox is treated as a special sort of domain
component, and so these can be represented in the tree as a natural
extension of the hierarchy.
For example, consider the mailbox S.Kille@cs.ucl.ac.uk. This will
lead to the following structure in the DIT:

```
 _____
|_Object_Class__|RDN_Type_____|RDN_Value_|
| Domain        |DomainComponent  |UK        |
| Domain        |DomainComponent  |AC        |
| Domain        |DomainComponent  |UCL       |
| Domain        |DomainComponent  |CS        |
|_RFC822Mailbox_|DomainComponent_|S.Kille__ |
```

This can be represented in User Friendly Name format as:

DomainComponent=S.Kille, DomainComponent=CS, DomainComponent=UCL,
DomainComponent=AC, DomainComponent=UK

Note that the RFC822Mailbox Object Class is a subclass of Domain.
Some attributes are allowed to be associated with these objects.
There may be other additional management attributes which it is useful
to define (e.g., Machine Type, Owner, Location etc.). This allows
some information which truly belongs to the domain to be represented
there. It also allows for further information to be associated with
the domain/mailbox when there is not a relevant part of the
organisationally structure DIT to be pointed at. When there is an
associated part of the DIT, information from that part of the DIT
should not be duplicated in the domain entry.

6 Wildcards

Wildcards are supported by having "*" as a special domain component
name. If there is a need to emulate wildcard matching using the
directory, the following algorithm must be employed. For example, the

wildcard entry for *.*.PODUNK.COM would be represented in the DIT as:

    DomainComponent=*, DomainComponent=*,
    DomainComponent=MIT, DomainComponent=COM

If A.B.PODUNK.COM is looked up in the directory, the query will fail
and indicate that two components are matched. A substitution should
be made, and *.*.PODUNK.COM looked up explicitly to identify the
associated information.

7 DNS Information

DNS information can be associated with an entry in the DIT. It is
important that this is done in a manner which is exactly equivalent to
the information stored in the DNS. This will allow the DIT to have
information loaded from the DNS or vice versa. All (authoritative)
records associated with a domain will be stored in the DIT. There is
no attempt made by the OSI Directory to emulate DNS caching or TTL
handling. It is assumed that the master entries are maintained by use
of DNS Zone Transfer (or equivalent), and that they can be treated as
authoritative. There is a need to define an attribute syntax which
represents a DNS record. This then allows DNS records to be stored in
the DIT. There are three possible encodings of this record:

ASN.1 Encoded This is the most natural approach in terms of X.500.
    However, it would require all users of this service to handle the
    new syntax, which would be awkward. There is a problem with
    handling the resource format in a general manner.

DNS Binary Encoded Use the formally defined record syntax. This
    would be convenient for access to the data by DNS related
    software, but would be an awkward encoding for independent X.500
    DUAs.

Text encoded Use of a text encoding derived from the DNS
    specifications. This is straightforward to map onto DNS protocol,
    and easy to support in a naive X.500 DUA. This approach is chosen.

The syntax is defined in IA5 characters. The BNF of the record uses
the definitions of section 5.1 of RFC 1035. It is

    <rr> [";" <comment>]

Three examples of this (for domain C.ISI.EDU) might be:


```
500 A   10.1.0.52                           ; Basic address record
IN 600 MX  10 VENERA.ISI.EDU.                  ; MX record
600 IN MX  10 VENERA.ISI.EDU.                  ; MX record - other order
```

Note that:

 o The class and TTL may be in either order (following RFC 1035)

 o The class defaults to IN

 o Domains must always be fully specified (i.e., master file
    abbreviate rules are not used).

 o The TTL for a record must always be present (this saves looking at
    the parent entry to find the SOA record).

 o Records (e.g., SOA) may be multiline. Lines should be separated
    with the two IA5 characters <CR><LF>.

CNAME records are mapped symmetrically onto Directory Aliases.

This is now defined in terms of attribute and object class
definitions. A single record type is defined, as opposed to one
attribute type per record type. This allows the definition to not
require extension when new DNS Record types are define. However,
there is some loss of efficiency if only a single record type is
needed, as filtering must be done by the DUA.
Similarly, no distinction is made on the basis of DNS class. This
means that if there are two class hierarchies, that they must be
represented in a single DIT, and that information for different
classes must be separated by DUA filtering.


```
DNSDomain OBJECT-CLASS
    SUBCLASS OF Domain
    MAY CONTAIN -
        DNSRecord "
```

DNSRecord ATTRIBUTE
    ATTRIBUTE-SYNTAX IA5String
    MATCHES FOR EQUALITY

Lookup of a domain is achieved by translating it algorithmically to a
Distinguished Name (DN), and reading back attributes desired. This
information can be managed and searched in a straightforward fashion.

The information may also be downloaded into a DNS database. This
should be done by use of zone transfer. A tool to perform zone
transfer (in both directions) between a DNS Server and a DSA would
seem to be both straightforward and useful. This would be a key tool
in a transition to X.500 based management of the DNS. It would also
allow a large part of the DNS namespace to be rapidly made available
in an X.500 pilot.
Inverse information can be derived by the usual IN-ADDR domain, which
will be represented in the same manner in the DIT.

8 NRS Information

Information associated with the UK NRS (Name Registration Scheme) can
be handled in a similar manner [Lar83]. This is being developed in a
separate document by Alan Turland.

9 Application Entity Titles

In many cases, Application entities will be closely related to
domains. In some cases, it may be appropriate to give Application
Entities names which are related to the DNS part of the DIT. In this
case, the Domain Name will be used to identify the application, and
the entry for the domain will also be of object class Application
Process. The children of this entry will identify Application
Entities, with names such as ''FTAM Service''.

10 Networks

It is clearly useful to represent networks within the DIT. A short
note on how to do this is given here. It is likely that this
specification will later be evolved in a separate document. This

defines an Object Class for a general network, and shows how it can be
subclassed to define technology specific networks.

```
Network OBJECT-CLASS
     SUBCLASS OF TOP
     MAY CONTAIN -
       Manager,
       Locality,
       Description "

IPNetwork OBJECT-CLASS
     SUBCLASS OF Network
     MUST CONTAIN -AssociatedDomain"
```

The Network Object Class allows networks to be defined, and for useful
attributes to be associated with the entry. A network will often
appear in more than one organisational structure, and this linkage
should be achieved by use of aliases. This grouping can facilitate
management of networks.
The subclass IPNetwork mandates linkage into the DNS part of the DIT.
This will be represented in the DIT using the structures of RFC 1101
[Moc89]. Both of the domains which identify the network should be
represented in the Object Class. For example, a network might have
the (user friendly) name:

 UCL-Ethernet, University College London, GB

This would have associated domains 0.0.40.128.IN-ADDR.ARPA and
UCL-ETHERNET.UCL.AC.UK. These would both have the analogous DIT
representations. For example:

DomainComponent=0, DomainComponent=0, DomainComponent=40,
DomainComponent=128, DomainComponent=IN-ADDR, DomainComponent=ARPA

11 Linkage

There is a need to associate the organisational X.500 DIT and the DNS
tree. The objects represented are different (Domain 6= Organisation;
Person 6= RFC 822 Mailbox). Therefore aliasing is not an appropriate
linkage. However, in many cases, there is a linkage which is rather

stronger than that implied by the seeAlso attribute. Therefore, we
define new attributes, which represent this stronger cross-linkage.
The same mechanism can be used to link a domains with an Application
Entity or an Application Process.
Links from the organisational X.500 DIT to the DNS tree are provided
by a new attribute, which could be present in Organisation or
Organisational Unit entries.

ObjectWithAssociatedDomain OBJECT-CLASS
  SUBCLASS OF top
  MUST CONTAIN -AssociatedDomain"

AssociatedDomain ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ia5StringSyntax

For example, the organisational entry:

    University College London, GB

would have an attribute:

    AssociatedDomain = UCL.AC.UK

Similarly, an RFC 822 mailbox attribute is used to link entries of
Person Object Class to their associated DNS entry. This attribute is
defined in the Cosine and Internet Naming Architecture [BHK91].
Conversely, there are pointers from the DNS represented tree into the
organisational X.500 DIT:

AssociatedName ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax

This attribute is associated with the Domain object class.

This entry is used to provide linkage from the DNS X.500 Hierarchy
into the organisational X.500 hierarchy. Where such entries do not
exist, attributes in the DNS entry (such as phone number) may be used.
It is recommended that information is not duplicated. The preferred
setup is for the DNS attributes to be rather skeletal, with pointers
into the organisational X.500 DIT.

For example, the domain UCL.AC.UK would be represented in the DIT as:

    DomainComponent=UCL, DomainComponent=AC,
    DomainComponent=UK

This entry would have in it an AssociatedName attribute with value:

    University College London, GB

This example shows a simple case with 1:1 linkage. There are cases
where a domain might be associated with multiple organisations, or an
organisation with multiple domains.

12 Conclusions and proposals for evaluation

Experiments should be undertaken to determine the practicality and
utility of this scheme, in a pilot environment. A possible approach
to this experimentation is described in Appendix A.
Object Identifiers have been assigned for this purpose in the Cosine
and Internet Naming Architecture [BHK91].

References

[BHK91] P. Barker and S.E. Hardcastle-Kille. The COSINE and Internet
         X.500 schema. Request for Comments RFC 1274, Department of
         Computer Science, University College London, November 1991.

[CCI88] The Directory --- overview of concepts, models and services,
         December 1988. CCITT X.500 Series Recommendations.

[Cro82] D.H. Crocker. Standard of the format of ARPA internet text
         messages. Request for Comments 822, University of Delaware,
         August 1982.

[HK91]   S.E. Hardcastle-Kille. Using the OSI directory to achieve
         user friendly naming. Request for Comments in preparation,
         Department of Computer Science, University College London,
         November 1991.

[Lar83] J. Larmouth. JNT name registration technical guide, April
         1983.

[Lin89] J. Linn. Privacy Enhancement for Internet Electronic Mail:
         Part 1 --- Message Encipherment and Authentication
         Procedures. Request for Comments 1113, Bolt, Beranek, and
         Newman, August 1989.

[Moc87a] P. Mockapetris. Domain names - concepts and facilities.
         Request for Comments RFC 1034, USC/Information Sciences
         Institute, November 1987.

[Moc87b] P. Mockapetris. Domain names - implementation and
         specification. Request for Comments RFC 1035,
         USC/Information Sciences Institute, November 1987.

[Moc89] P. Mockapetris. DNS encoding of network names and other
         types. Request for Comments RFC 1101, USC/Information
         Sciences Institute, April 1989.

13 Security Considerations

This memo does not directly address security issues. However, due to
the facilities of X.500, this proposal could lead to a more secure way
to access and manage domain information.

14 Author's Address

    Steve Hardcastle-Kille
    Department of Computer Science
    University College London
    Gower Street
    WC1E 6BT
    England

    Phone: +44-71-380-7294

    EMail: S.Kille@CS.UCL.AC.UK

A Possible Deployment Approach

This appendix notes a possible approach to deploying an experiment to
evaluate this mechanism. The following components of a possible
experiment are noted.

1. User tool. This will take a domain or mailbox as input. This
    will be looked up in the DIT. This tool should be capable of:

        o Attempting to correct user input

        o Helping browsing

        o Looking up information associated with the domain (or mailbox)
           and associated name, in particular the manager (of both domain
           and associated name) and information on the manager (e.g.,
           phone number and mailbox).

        o Supply DNS records

        o Handle IN-ADDR.ARPA inverse lookups if supplied with an IP
           Address

        o Look up networks

2. A procedural library to allow user interfaces to make easy use of
    these facilities.

3. Zone transfer tool. This will use the zone transfer protocol to
    transfer information between a DSA and Domain Nameserver. When
    writing to the DSA, attributes in an entry which are not DNS
    records should remain untouched.

4. Linkage patching tool. When the organisational DIT is
    established, associated domain pointers are usually inserted. A
    tool can be written to search the DIT and insert the reverse
    pointers.

5. DNS Manager Tool. This will allow user addition of additional
    information into the DNS part of the DIT. A standard DUA can
    probably be used for this.

6. Mailbox download tool. This will allow download of local
    mailboxes, with pointers to the user entries.

7. Emulation DNS Server, using the Directory as a database. The
    server should maintain a permanent connection to its local DSA. As
    there is no OSI bind, the response of this server can be at least
    as fast as a normal DNS server. There can be two variants of this
    server.

    (a) Using a local DSA as a local database but using DNS
         distributed operations.

    (b) Do all lookups in the directory (using Directory Distributed
         Operations).

An initial experiment is straightforward. The Zone Transfer Tool (3)
can be used to download a large part of the DNS space into a single
DSA (there will be some restrictions, as parts of the DNS hierarchy do
not permit zone transfer). This can be used repeatedly to maintain
the information. The linkage patching tool (4) can be used to put in
pointers to parts of the DIT. The user tool can then be used (by all
sites participation the the directory pilot) to look up domain
information. This will allow the utility of the approach to be
evaluated. The manager tool (5) will allow extra information to be
added to parts of the DNS tree.

The next stage will be to distribute the DNS part of the DIT over
multiple DSAs using Directory distribution techniques.
The emulation DNS Server (7) will be useful to ensure that equivalent
functionality is being offered by the Directory. It can also be used
to examine performance differences.
A final step is to master some parts of the DNS hierarchy in the DIT.
Because of the zone transfer technique, this will be entirely
transparent to the DNS user. Management benefits can then be
examined.