                 Representing IP Information in the X.500 Directory

Status of this Memo

   This memo defines an Experimental Protocol for the Internet
   community. This memo does not specify an Internet standard of any
   kind. Discussion and suggestions for improvement are requested.
   Distribution of this memo is unlimited.

Abstract

   This document describes the objects necessary to include information
   about IP networks and IP numbers in the X.500 Directory. It extends
   the work "Charting networks in the X.500 Directory" [1] where a
   general framework is presented for representing networks in the
   Directory by applying it to IP networks. This application of the
   Directory is intended to support the work of IP network assigning
   authorities, NICs, as well as other applications looking for a
   mapping of IP numbers to data of related networks. Furthermore,
   Autonomous Systems and related routing policy information can be
   represented in the Directory along with their relationship to
   networks and organizations.

Table of Contents

1. Introduction

   Information related to the Internet Network Infrastructure is created
   and stored by a number of different organizations, such as the
   Internet Assigned Numbers Authority (IANA), Internet Registry (IR),
   Network Information Centers (NICs), and the NSFNET Network Operations
   Center (NOC). This information is generally "mastered" (stored and
   maintained) by these organizations on a centralized basis, i.e.,
   there is a single place to look for a definitive list of entries for
   these categories. This has worked well in the past but given the
   tremendous growth of the Internet and its number of users and
   networks, it is essential that a distributed schema be used.

   The X.500 Directory offers the appropriate technology for
   implementing this distributed method of managing network
   infrastructure information.

   The following goals are addressed in this document:

    o Provision of IP specific images of network elements
    o Mapping from Network Number to network, owner, provider etc.
    o Support of delegation of IP address blocks
    o Storage of high-level routing policies and AS information
    o Support of "classless" network address formats

       o Provision of several organizational images of a network

    It may be noted that the document basically consists of two parts.
    In the first part, an IP specific extension of the general framework
    "Charting networks in the Directory" [1] is given. Objects defined
    by the application of this framework are related to IP numbers. An IP
    namespace is defined in the second part of this paper, referring to
    IP network elements defined at the beginning.

2. IP images of networks

    As defined in [1], networks are modeled as a set of subnetworks and
    nodes, called network elements in the remainder. To obtain a
    particular view of a network element, images were introduced. Below,
    images of network elements for an IP specific view are defined.
    Please note that images contain references to underlying physical
    information about elements.

    In the remainder, ipStringSyntax is used as attribute type for all
    attributes that are to hold an IP number. Currently, there are two
    definitions for a syntax like this:

     o IpAddress as of [5]
     o ip as of [6]

    It is suggested to use CaseIgnoreStringSyntax for implementations for
    the time being with the convention to use the ordinary IP syntax.
    Nevertheless, an OID has been reserved for ipStringSyntax (see
    Appendix).

2.1 IP network image

    IP network image is one application of network images and therefore
    inherits the networkImageClass. This class is given below for
    reference only, its definition can be found in [1].

    networkImage OBJECT CLASS
     SUBCLASS of ImageCommunicationObject
     MAY CONTAIN
      externalGateway :: distinguishedNameSyntax,
       /* points to one or more nodes that act
          as gateway for the protocol application
          this image refers to */

    An IP network combines all network elements that have a common IP
    network number. Presently, IP network numbers fall into one of the
    classes A, B, or C. However, sub- and supernetworking is already done
    broadly, and classless networks numbers are expected to be assigned

soon. [2] proposes to assign bitwise contiguous blocks of class-C-
addresses to all networks with more than 255 nodes. The definition of
IP network, therefore, is always related to common network number and
network mask.

IP networks have a very close relationship to the Domain Name System.
Several attributes are introduced to take care of these relations.
Though we do not intend to abolish the existing DNS service, the
schema defined here is able to provide the mapping IP number to
domain name and vice versa.

An IP network image object as defined below is intended to provide
technical and administrative data for one IP network. Attributes hold
information that a NIC-WHOIS server would reveal for the network, and
more.

```
ipNetworkImage OBJECT CLASS
 SUBCLASS of NetworkImage
 MUST CONTAIN
  ipNetworkImageName :: CaseIgnoreString,
   /* common name */
  ipNwNumber :: IPStringSyntax,
   /* the IP network number for
      this (sub)network */
  ipNwMask :: IPStringSyntax
   /* mask that applies to ipNwNumber
      in order to define classless
      network number; also used for routing */

 MAY CONTAIN
 /* DNS related info ; e.g. - */
  associatedDomain :: CaseIgnoreStringSyntax,
   /* the domain name associated to this IP network;
      As there is not always a 1:1 mapping between traditional
      IP numbers and domain names, the name given here
      should contain the "closest match".
      1) (sub)network does not have a domain name
         of its own, but is part of a bigger domain:
         take name of that domain
      2) network is divided into several domains,
         therefore having more than one domain name:
         list all of them.
      Note: a reverse mapping of domain names to
      networks/nodes can be achieved by a modified
      implementation of RFC1279 */
  inAddrServer :: DistinguishedNameSyntax,
   /* refers to the ipNodeImageObject of the
      inaddr Server that holds information about
```

```
      this network */
  /* routing policy; e.g. - */
   asNumber :: integerStringSyntax,
    /* number of Autonomous System this network belongs to */
   acceptedUsagePolicy :: caseIgnoreStringSyntax,
    /* semantics to be defined */
  /* Any other - */
   provider   :: DistinguishedNameSyntax,
    /* points to network provider */
   onlineDate :: uTCTimeSyntax
    /* date when network got connected to the Internet */
```

2.2 IP node image

   If a node in the network is running the IP protocol, an
   ipNodeImageObject should be created for this node. This image is a
   subclass of nodeImageClass and holds IP specific information. The
   nodeImageClass is given below for reference only, its definition can
   be found in [1].

```
   nodeImage OBJECT CLASS
    SUBCLASS of ImageCommunicationObject
     /* no attributes common for all nodeImages have been
        defined yet */
```

   An ipNodeImage is used to obtain technical and administrative
   information on a host. The object is defined as follows:

```
   ipNodeImage OBJECT CLASS
    SUBCLASS of NodeImage
    MUST CONTAIN
     ipNodeName :: CaseIgnoreString
      /* common name, it is advised to use
         the hostname for this purpose */
    MAY CONTAIN
     protocol :: CaseIgnoreString,
      /* name and version of IP protocol running */
     domainName :: CaseIgnoreString,
      /* the complete domain name of this node;
         CNAMEs can be stored additionally to the
         DNS A record name;
         further relationships, like MX record entries,
         should be taken care of by the domain name tree
         according to RFC 1279 */
```

2.3 IP network interface image

   The most important IP related information of a node (its IP
   addresses) is registered with ipNetworkInterfaceImageObjects. This
   picture is accurate as a node can have several IP addresses, but at
   most one per interface. Furthermore, it shows clearly the
   relationship between interface and neighboring IP network.

   IpNetworkInterfaceImage is a subclass of networkInterfaceImageClass.
   The networkInterfaceImageClass is given below for reference only, its
   definition can be found in [1]. Note that for
   ipNetworkInterfaceImage all references are drawn in the context of
   IP, i.e., networkInterfaceAddress becomes an IP address and
   connectedNetwork points to an ipNetworkImageObject.

   networkInterfaceImage OBJECT CLASS
    SUBCLASS of ImageCommunicationObject
    MAY CONTAIN
     networkInterfaceAddress   :: caseIgnoreStringSyntax,
       /* this interface's address in the context the
          image refers to, e.g. IP number, NSAP */
     connectedNetwork   :: distinguishedNameSyntax
       /* pointer to networkImageObject that describes
          the network this interface is attached to in terms
          of the protocol or application the images
          indicates */

   Additionally, ipNetworkInterfaceImageObject has the following
   properties:

   ipNetworkInterfaceImage OBJECT CLASS
    SUBCLASS of NetworkInterfaceImage
    MUST CONTAIN
     ipNetworkInterfaceName :: CaseIgnoreStringSyntax
       /* It is suggested that the interface name
          is derived from the name of the logical
          device this interface represents for the
          operating system, e.g. le0, COM1 */
    MAY CONTAIN
     ipNwMask :: IPStringSyntax
       /* mask that applies to networkInterfaceAddress for
          routing of packets to nodes on the connected
          (broadcast) network;
          Note: This is only a fraction
          of the routing table information
          for this interface, namely for one hop. */

2.4 Autonomous Systems

   Autonomous Systems (AS) are defined in asObject which is a subclass of
   imageCommunicationObject. It provides technical and administrative
   information of an AS. Furthermore, routing policies can be stored with
   the AS object. The definition of the AS object is aligned with the
   corresponding database object defined in [3].

   as OBJECT CLASS
    SUBCLASS of ImageCommunicationObject
    MUST CONTAIN
     asNumber :: IntegerStringSyntax

    MAY CONTAIN
     asName :: CaseIgnoreStringSyntax,
      /* if there is a name different from AS */
     asIn :: CaseIgnoreStringSyntax,
      /* accepted routes from other AS */
      /* for syntax and semantics refer [3] */
     asOut :: CaseIgnoreStringSyntax,
      /* generated routes announced to others */
      /* for syntax and semantics refer [3] */
     asDefault ::CaseIgnoreStringSyntax,
      /* how default routing is handled */
      /* for syntax and semantics refer [3] */
     asGuardian :: DistinguishedNameSyntax, */
      /* DN of guardian of this AS */
     lastModifiedDate :: UTCtimeSyntax */
      /* important as routes change frequently */

   Note that routing policies for an Autonomous System are represented
   by the {asIn, asOut} attributes of asObject. Due to performance
   constraints of present X.500 technology, it will probably not be used
   directly by routers for dynamic routing. However, it certainly can
   be used in network management systems to determine the allowed paths
   [i.e., in accordance with published policies] between two networks.
   This will be useful in finding alternate paths, and evaluating the
   connectivity of networks.

3. Number assignment information

   In the following, Directory objects have been defined to represent IP
   and AS (Autonomous System) namespace in the Directory. Their purpose
   is to provide

     o mapping from IP number to IP network element (network or node)
     o mapping from IP number to AS number and vice versa
     o assignment and delegation information

   The need for a global, distributed database supporting the objectives
   arises mainly from distributed IP- and AS-number assignment.

   Describing all IP numbers with one of the new objects delegatedBlock,
   ipGroup and ipReference leads to the desired information. AS number
   information is stored with the objects asBlock and asReference.
   Furthermore, all assigned numbers have some properties in common.
   Therefore, an objectclass assignedNumberClass is introduced. This
   class exports attributes to delegatedBlock, ipGroup, ipReference,
   asBlock, and asReference.

   AssignedNumberClass is defined as follows ("number" always refers to
   IP number of delegatedBlock, network, host, and AS number, resp.):

   assignedNumberClass OBJECT CLASS
    SUBCLASS of top
    MAY CONTAIN
     assBy :: DistinguishedNameSyntax,
      /* refers to an organization or organizationalRole
         that assigned the number to assTo (see below) */
     assTo :: DistinguishedNameSyntax,
      /* refers to organization or organizationalRole
         that the number was assigned to. This does not
         imply that assTo "owns" this number now. */
     assDate :: uTCTimeSyntax,
      /* date of assignment for this number */
     nicHandle :: CaseIgnoreStringSyntax,
      /* gives the unique ID for a description
         related to this number.
         format: "handle : nic-domain-name"
         example: MAK21 : rs.internic.net */
     relNwElement :: DistinguishedNameSyntax,
      /* the network element related to this number
         (network or node) */

3.1 Delegated Block object

   This object provides information on a block of IP addresses delegated
   to some local-authority or service provider. Only contiguous blocks
   can be represented with the following schema. If an organization
   (say, a NIC) has been assigned several IP network numbers which do
   not form a contiguous block, it might want to use a different form of
   representing that fact (e.g., using imageNetworks). The
   delegatedBlock object holds lower and upper bounds of the block.

   Note that the above does not make any assumption about the network
   masks being constrained by byte boundaries. We can thus represent
   subnetting within a "network (number)" that often happens within an

organization in the same framework.

This schema does lead to some granularity in the otherwise flat IP-
number space. Further, the granularity is significant as it may be
used to identify the administrator of the block - a service provider
or a domain manager. E.g., it fits well into the schema of
aggregating networks for routing purposes as has been proposed in
[4].

The object delegatedBlock is of the form:

```
delegatedBlock OBJECT CLASS
 SUBCLASS of AssignedNumberClass
 MUST CONTAIN
  delegatedBlockName :: caseIgnoreStringSyntax,
  lowerBound :: IPStringSyntax,
   /* smallest IP address belonging to the
      block, e.g. 195.100.0.0 */
  upperBound :: IPStringSyntax
   /* highest  IP address belonging to the
      block, e.g. 195.103.255.255 */
```

The attribute relNwElement (inherited from AssignedNumberClass) can
point to a networkImage covering all networks within the block if
this makes sense.

3.2 IP Group object

This object provides information for an IP network number. Its
purpose is basically only to

    o show that the number has been assigned, and
    o provide a reference to the descriptive ipNetworkObject for
      this network.

Regardless of the actual value of x, IP group objects may exist for
IP numbers x.0.0.0, x.y.0.0 and x.y.z.0. This approach includes
"classical" class-A, -B and -C network addresses as well as any kind
of super- and subnetworking.

The IP group object is a subclass of assignedNumberClass. The
attribute relNwElement points to an ipNetworkImage as defined above.

```
ipGroup OBJECT CLASS
 SUBCLASS of  AssignedNumberClass
 MUST CONTAIN
  ipGroupName :: IPStringSyntax,
   /* IP number; x.0.0.0 or x.y.0.0 or x.y.z.0
```

```
     where x, y, z in 1..255 */
   ipNwMask   ::    IPStringSyntax
    /* mask that applies to all numbers
       within the group; used to define
       classless networking;  */
```

3.3 IP Reference object

There is one IP reference object for each IP host address. The
purpose of this object is to

  o tell that this IP number is already assigned to a node
  o give a pointer to the related ipNodeImageObject

The IP reference object is a subclass of assignedNumberClass. The
attribute relNwElement points to an ipNodeImage.

```
ipReference OBJECT CLASS
 SUBCLASS of  AssignedNumberClass
 MUST CONTAIN
  ipReferenceName :: IPString
   /* value is always IP address */
```

3.4 AS block object

The AS block object is used to show delegation of blocks of AS
numbers to regional registries. This is similar to delegatedBlock of
ipNetwork numbers.

```
asBlock OBJECT CLASS
 SUBCLASS of  AssignedNumberClass
 MUST CONTAIN
  asBlockName :: caseIgnoreStringSyntax,
  asLowerBound :: integerStringSyntax,
  asUpperBound :: integerStringSyntax
```

An AS block will comprise several consecutive AS numbers. Objects to
describe these numbers may be stored in asObjects.

3.5 AS reference object

An AS reference object is used to show that an Autonomous System
number has been assigned (and thus can not be given to somebody
else). Similar to ipGroup, asReference does not contain technical
details about an autonomous system itself but rather points (with
relNwElement) to a descriptive asObject.

```
   asReference OBJECT CLASS
    SUBCLASS of  AssignedNumberClass
    MUST CONTAIN
     asNumber :: integerStringSyntax
```

4. Directory tree

```
                           root
                            |
                +------------+--------------+
                |                           |
               c=                        o=Internet
                |                           |
           +-----+------+             +------+-------+
           |            |             |              |
         ipNw=        as=           dbl=           asB=
           |                         |              |
         ipNd=                      ipG=           asRef=
           |                         |
         ipNwIf=                    ipRef=
```
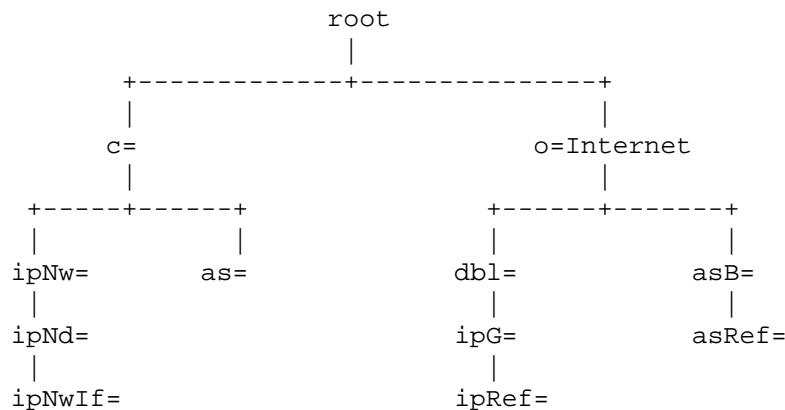
                 Figure 1: Overall relationship of objects.

4.1 IP image objects

   According to [1], IP image entries will be stored underneath the
   organization / organizationalUnit entry of the entity responsible for
   that network. The case that such an entry does not yet exist in the
   white-pages pilot is discussed in 4.4 below.

4.2 AS objects

   The technical and administrative description of an AS is basically
   maintained by NICs, network providers, or other special
   organizations. It is suggested that these organizations build a
   subtree for information on AS which they are responsible for.

4.3 Namespace objects

   The new IP namespace objects build a single tree in the Directory. It
   is suggested that this tree will have a root of type
   organizationalUnit within @o=Internet@ou=Network Information.

```
     objectClass= organizationalUnit
     organizationalUnitName= IP networks
     description= root of IP number tree
```

The tree is built under an administrative and an implementational
view. Nowadays, network numbers usually are assigned to
organizations by (national) Network Information Centers (NIC) which
themselves have got a block of IP network numbers assigned from
another authority (e.g., IR at top level). This concept of delegated
blocks falling apart in smaller delegated blocks and IP network
numbers is used to model the Directory tree. Thus, an ipGroup object
is always subordinate of a delegated block object (namely the
delegated block including this IP number). Network numbers that were
directly assigned by a top-level authority, i.e., have not been
object of a delegation to a local assigning authority, will all be at
one level in the Directory. Already today, however, we find many
delegations within the traditional class A-, B- and C-addresses.
Such a delegation is represented by a delegated block object, having
the assigned IP network numbers as subordinates. Also, part of the
block can be further delegated to another authority, leading to
another delegated block object within the parent delegated block's
tree. Usually, subordinates of ipGroup objects are ipReferences,
i.e., single IP addresses as assigned to nodes. To support
subnetworking, it is also allowed to divide ipGroups into several
subnetwork ipGroups, each representing an IP subnetwork. In such
cases, subnetwork numbers are given as subordinates to the assigned
IP network number. Network masks clarify what the subnetwork
addresses are.

```
                        ou=IP networks
                              |
          +-----------------+---------------------+
         /                   |                     \
             dbl=150.0.0.0-150.100.0.0
                              |
          +-----------------+---------------------+
         /                   |                     \
                     ipG=150.80.0.0
                              |
          +-----------------+---------------------+
         /                   |                     \
                     ipG=150.80.240.0
                              |
          +-----------------+---------------------+
         /                   |                     \
    ipRef=150.80.254.1    ipRef=150.80.254.2     ipRef=150.80.254.3
```
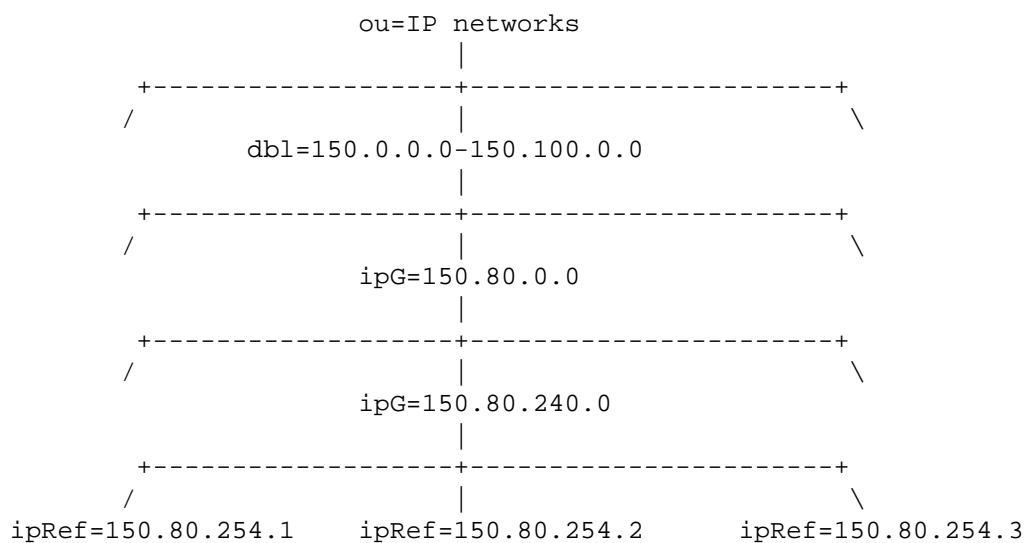
        Figure 2: Example population of IP namespace tree according
                 to delegation and subnetworking.

For some applications, the separation of ipImage (description of the
network) and ipGroup (description of the namespace element) will bear

disadvantages in the look-up procedure. In that case one might think
of combining both object classes with the aim to provide one object
describing administrative and technical data for an IP network.

As Autonomous Systems are an additional namespace to the existing IP
number space, they should go into a separate subtree. It is suggested
that this is an organizationalUnit within @o=Internet@ou=Network
Information.

```
  objectClass= organizationalUnit
  organizationalUnitName= AS numbers
  description= root of Autonomous System number space
```

Similar to blocks of IP network numbers, blocks of AS numbers are
sometimes delegated to another registry. This is expressed by asBlock
objects. These objects come below the root of the AS number space.
All AS numbers falling into such a block are stored as subordinates
of the block. An AS block may have smaller AS blocks underneath if
delegation is extended.

4.4 Relationship to organizational entries

Organizational information (i.e., white-pages-like information about
an organization, its departments and employees) occurs at several
places in the network DIT - [org of IP-Number, org of AS-Number, org
of Admin- contact, However, it will be basically mastered
[administered, maintained] by the organization itself in the
Directory Management Domain (DMD) over which the organization has the
authority. This gives rise to some tricky problems - a typical
example is that of a NIC which holds the AS, DNS, IP, ... subtrees
of the DIT.

A good strategy would avoid explicit duplication of information. By
explicit duplication of information we understand information being
duplicated outside the directory framework, e.g., by having several
master entries for one and the same piece of information. The only
way to avoid duplication would be to have relevant entries point to
the pertinent organizational entry for organizational information.
But since

  o most organizations do not, as yet, have an entry in the DIT and
  o the reliability of the access to an organizations DIT when
    stored in a remote DSA cannot be taken for granted,

the following framework is adopted to accommodate the conflicting
requirements /conditions.

   o A copy of all the necessary organization-info is retained
     at the NICs DSA. Since only the necessary info will be kept
     the NIC will not be burdened to act as the repository of the
     organizations DIT. These objects may be kept in a separate
     subtree of affiliated-organizations [organizations
     affiliated to the NIC]. Though the affiliated organizations node
     does not really represent a locality, it is suggested to define
     the node as objectClass locality. This does not break the
     Directory schema when entries of organizations shall become
     subordinate to the NICs organization's entry.

   o The problem of information duplication/consistency will arise when
     organizational DITs/DSAs do come into existence. At that stage a
     shadowing mechanism which will attempt to maintain the data
     consistency may be resorted to. The X.500/ISO 9594(1993)
     implementations are expected to provide appropriate shadowing
     mechanisms along X.525.

   It may be noted that what is suggested is not a duplication of an
   entire white-pages-like structure at the NIC. It suggests an
   "affiliated organizations node". The entries under this node will be
   organization objects with a limited number of attributes, i.e., the
   attributes to hold the organization info necessary for the NIC:
   nothing more, nothing less. Operationally, and content wise the NIC
   DSA will hold exactly the amount of info that is desired by the NIC.
   When an organization sets up its DSA and when the organization
   informs the NIC about it, the NIC will set up the shadowing
   arrangement to obtain info on changes of interest [and forget about
   it].

   It may be emphasized that the entries under affiliated organizations
   are physical entities [replicated and refined from the Master
   entries, if and when they exist...] rather than alias entries. If a
   NIC dislikes the idea of users poring over the entries in the
   affiliated organizations - appropriate access control can be applied.
   Though duplication is unavoidable, the proposal attempts to make it
   transparent, by delegating the responsibility of maintaining the
   integrity to the Directory.

   This issue is discussed in greater detail in a separate document [7].

5. Security Considerations

   Security issues are not discussed in this memo.

6. Authors' Addresses

Thomas Johannsen
Dresden University of Technology
Institute of Communication Technology
D-01062 Dresden, GERMANY

Phone: +49 351 463-4621
EMail: Thomas.Johannsen@ifn.et.tu-dresden.de

Glenn Mansfield
AIC Systems Laboratory
6-6-3 Minami Yoshinari, Aoba-ku
Sendai 989-32, JAPAN

Phone: +81 22 279-3310
EMail: glenn@aic.co.jp

Mark Kosters
Network Solutions, Inc.
505 Huntmar Park Dr.
Herndon, VA 22070

Phone: +1 703 742-4795
EMail: markk@internic.net

Srinivas R. Sataluri
AT&T Bell Laboratories
Room 1C-429, 101 Crawfords Corner Road
Holmdel, NJ 07733-3030

Phone: +1 908 949-7782
EMail: sri@qsun.att.com

References

[1]   Mansfield, G., Johannsen, T., and M. Knopper, "Charting Networks
        in the X.500 Directory", RFC 1609, AIC Systems Laboratory,
        Dresden University, Merit Networks,Inc., March 1994.

[2]   Gerich, E., "Guidelines for Management of IP Address Space", RFC
        1466, Merit, May 1993.

[3]   Bates, T., Jouanigot, J.-M., Karrenberg, D., Lothberg, P., and M.
        Terpstra, "Representation of IP Routing Policies in the RIPE
        Database", Document ripe-81, RIPE, February 1993.

[4]   Fuller, V., Li, T., Yu, J., and K. Varadhan, "Supernetting: An
        Address Assignment and Aggregation Strategy", RFC 1338, BARRNet,
        cisco, MERIT, OARnet, June 1992.

[5]   Rose, M., and K. McCloghrie, "Structure and Identification of
        Management Information for TCP/IP-based internets", STD 16, RFC
        1155, Performance Systems International, Hughes LAN Systems, May
        1990.

[6]   Barker, P., and S. Kille, "The COSINE and Internet X.500 Schema",
        RFC 1274, University College London, November 1991.

[7]   Mansfield, G., Johannsen, T., and J. Murai, J., "Deployment
        Strategy for the Directory in the Internet", AIC Systems
        Laboratory, Dresden University, Keio University, Work in
        Progress, July 1993.

Appendix: OID tables

   This appendix lists object identifiers for object classes and
   attributes type defined in [1] and this document.

   OIDs are given in quipu-oidtable format to make it easy for people to
   include them into their pilots.

   IMPORT from oidtable.gen:

   iso: 1
   identifiedOrganization: iso.3
   dod: identifiedOrganization.6
   internet: dod.1
   experimental: internet.3
   network-objects: experimental.53

   -- localoidtable.gen

   id-nw-oc: network-objects.1
   id-nw-at: network-objects.2
   id-nw-as: network-objects.3
   ipStringSyntax: ip-nw-as.1

   -- localoidtable.oc

   # general class definitions
   # Format is -
   #   Object: OID: SubClassOf: MustHave: MayHave

   CommunicationObject: id-nw-oc.1 : top : \
            : \
           adminContact, technContact, description

   PhysicalCommunicationObject: id-nw-oc.2 : CommunicationObject : \
            : \
           owner, localityName, ICO

   ImageCommunicationObject: id-nw-oc.3 : CommunicationObject : \
            : \
           imageType, imageOf

   # physical communication elements

   network: id-nw-oc.4 : PhysicalCommunicationObject : \
           networkName : \

```
        externalGateway, nwType, media, speed, traffic, \
        configurationDate, configurationHistory

node: id-nw-oc.5 : PhysicalCommunicationObject : \
        nodeName : \
        typeOfMachine, OS

networkInterface: id-nw-oc.6 : PhysicalCommunicationObject : \
        networkInterfaceName : \
        networkInterfaceAddress, connectedNetwork

# image communication elements

networkImage: id-nw-oc.7 : ImageCommunicationObject : \
        : \
        externalGateway, speed, traffic, charge

nodeImage: id-nw-oc.8 : ImageCommunicationObject : \
        :

networkInterfaceImage: id-nw-oc.9 : ImageCommunicationObject : \
        : \
        networkInterfaceAddress, connectedNetwork

# IP image elements

ipNetworkImage: id-nw-oc.10 : networkImage : \
        ipNetworkImageName, ipNwNumber, ipNwMask : \
        associatedDomain, inAddrServer, asNumber, \
        provider, onlineDate

ipNodeImage: id-nw-oc.11 : nodeImage : \
        ipNodeName : \
        protocol, domainName

ipNetworkInterfaceImage: id-nw-oc.12 : networkInterfaceImage : \
        ipNetworkInterfaceName : \
        ipNwMask

as: id-nw-oc.13 : ImageCommunicationObject : \
        asNumber : \
        asName, asIn, asOut, asDefault, asGuardian, \
        lastModifiedDate

# number assignement objects

assignedNumberClass: id-nw-oc.14 : top : \
        : \
```

```
           assBy, assTo, assDate, nicHandle, relNwElement, \
           description

   delegatedBlock: id-nw-oc.15 : AssignedNumberClass : \
           delegatedBlockName, lowerBound, upperBound :

   ipGroup: id-nw-oc.16 : AssignedNumberClass : \
           ipGroupName, ipNwMask :

   ipReference: id-nw-oc.17 : AssignedNumberClass : \
           ipReferenceName :

   asBlock: id-nw-oc.18 : AssignedNumberClass : \
           asBlockName, asLowerBound, asUpperBound :

   asReference: id-nw-oc.19 : AssignedNumberClass : \
           asNumber :



   -- localoidtable.at

   adminContact:                 id-nw-at.1     :DN
   technContact:                 id-nw-at.2     :DN
   ICO:                          id-nw-at.3     :DN
   imageType:                    id-nw-at.4     :caseIgnoreString
   imageOf:                      id-nw-at.5     :DN
   networkName,nw:               id-nw-at.6     :caseIgnoreString
   externalGateway:              id-nw-at.7     :DN
   nwType:                       id-nw-at.8     :caseIgnoreString
   media:                        id-nw-at.9     :caseIgnoreString
   speed:                        id-nw-at.10    :numericString
   traffic:                      id-nw-at.11    :numericString
   configurationDate:            id-nw-at.12    :utcTime
   configurationHistory:         id-nw-at.13    :caseIgnoreString
   nodeName,nd:                  id-nw-at.14    :caseIgnoreString
   typeOfMachine:                id-nw-at.15    :caseIgnoreString
   OS:                           id-nw-at.16    :caseIgnoreString
   networkInterfaceName,ni:      id-nw-at.17    :caseIgnoreString
   networkInterfaceAddress:      id-nw-at.18    :caseIgnoreString
   connectedNetwork:             id-nw-at.19    :DN
   charge:                       id-nw-at.20    :numericString
   ipNetworkImageName,IPnw:      id-nw-at.21    :caseIgnoreString
   ipNwNumber:                   id-nw-at.22    :caseIgnoreString
   ipNwMask:                     id-nw-at.23    :caseIgnoreString
   inAddrServer:                 id-nw-at.24    :DN
   asNumber,asN:                 id-nw-at.25    :integerString
   provider:                     id-nw-at.26    :DN
```

```
onlineDate:                   id-nw-at.27    :utcTime
ipNodeName,IPnd:              id-nw-at.28    :caseIgnoreString
protocol:                     id-nw-at.29    :caseIgnoreString
domainName:                   id-nw-at.30    :caseIgnoreString
ipNetworkInterfaceName,IPni:  id-nw-at.31    :caseIgnoreString
asName:                       id-nw-at.32    :caseIgnoreString
asIn:                         id-nw-at.33    :caseIgnoreString
asOut:                        id-nw-at.34    :caseIgnoreString
asDefault:                    id-nw-at.35    :caseIgnoreString
asGuardian:                   id-nw-at.36    :DN
assBy:                        id-nw-at.37    :DN
assTo:                        id-nw-at.38    :DN
assDate:                      id-nw-at.39    :utcTime
nicHandle:                    id-nw-at.40    :caseIgnoreString
relNwElement:                 id-nw-at.41    :DN
delegatedBlockName,dbl:       id-nw-at.42    :caseIgnoreString
lowerBound:                   id-nw-at.43    :caseIgnoreString
upperBound:                   id-nw-at.44    :caseIgnoreString
ipGroupName,IPgr:             id-nw-at.45    :caseIgnoreString
ipReferenceName,IPref:        id-nw-at.46    :caseIgnoreString
asBlockName,asBl:             id-nw-at.47    :caseIgnoreString
asLowerBound:                 id-nw-at.48    :integerString
asUpperBound:                 id-nw-at.49    :integerString
```