Charting Networks in the Directory

Glenn Mansfield Thomas Johannsen Mark Knopper

September 1993

OSI-DS-37-01

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

To learn the current status of any Internet-Draft, please check the 1id-abstracts.txt listing contained in the Internet-Drafts Shadow Directories on ds.internic.net, nic.nordu.net, ftp.nisc.sri.com, or munnari.oz.au.

Abstract

There is a need for a framework wherein the infrastructural and service related information about communication networks can be made accessible from all places and at all times in a reasonably efficient manner and with reasonable accuracy. This document presents a model in which a communication network with all its related details and descriptions can be represented in the X.500 Directory. Schemas of objects and their attributes which may be used for this purpose are presented. The model envisages physical objects and several logical abstractions of the physical objects.

Contents

1 Introduction

The rapid and widespread use of computer networking has highlighted the importance of holding and servicing information about the networking infrastructure itself. The growing and active interest in network management, which has concentrated mainly in the areas of fault and performance management on a local scale, is severely constrained by the lack of any organized pool of information about the network infrastructure itself. Some attempts have been made, on a piecemeal basis, to provide a larger view of some particular aspect of the network (WHOIS, DNS, .. in the case of the Internet; [WHOIS], [DNS]). But to date, little or no effort has been made in setting up the infrastructural framework, for such an information pool. In this work we explore the possibility of setting up a framework to hold and serve the infrastructural information of a network.

2 Infrastructural information requirements

Network operation and management requires information about the structure of the network, the nodes, links and their properties. Further, with current networks extending literally beyond bounds, the scope of the information covers networks beyond the span of local domain of authority or administration. When the Network was relatively small and simple the map was already known to the knowledgable network administrator. Based on this knowledge the course of the packets to different destinations would be charted. But presently the size of the Network is already beyond such usages. The current growth of the Network is near explosive. This is giving rise to the urgent necessity of having infrastructural and service related information made accessible from all places and at all times in a reasonably efficient manner and with reasonable accuracy. In the rest of this work a *network* is the media for transmitting information. *Network elements* are equipment with one or more *network interfaces* whereby it is possible to exchange information with the network. Network elements with multiple interfaces e.g. gateways/routers/bridges/repeaters... may be used to connect networks. Network related information, referred to as 'network map' in the rest of this paper, should

1. Show the interconnection between the various network elements. This will basically represent the Network as a graph where vertices represent objects like gateways/workstations/ subnetworks and edges indicate the connections.
2. Show properties and functions of the various network elements and the interconnections. Attributes of vertices will represent various properties of the objects e.g. speed, charge, protocol, OS, etc. Functions include services offered by a network element.
3. Contain various name and address information of the networks and network elements

4. Contain information about various administrative and management details related to the networks and network elements.
5. Contain the policy related information, part of which may be private while the other part may be made public.

Using this map the following services may be provided

1. Configuration management:
   - Display the physical configuration of a network, i.e. nodes and their physical interconnections
   - Display the logical configuration of a network, i.e. nodes and their logical interconnections.
2. Route management:
   - Find alternate routes by referring to the physical and logical configurations.
   - Generate routing tables considering local policy and policy of transit domains
   - Check routing tables for routing loops, non-optimality, incorrect paths, etc.
3. Fault management: In case of network failures alternatives may be found and used to bypass the problem node or link.
4. Service management: Locate various services and servers in the Network.
5. Optimization: The information available can be used to carry out various optimizations, for example cost, traffic, response-time, etc.
6. Provide mappings between the various names and addresses of elements
7. Depict administrative/autonomous domains.
8. Network Administration and Management: References to people responsible for administering and technically maintaining a network will be useful.

Examples of such usages are described in [IP], [SPP].

3 The Nature of the Network Map - The X.500 solution

Implementing and maintaining a detailed map of the network poses a serious problem. The scope of the map is global and the network itself is expanding. Some of the problems that are peculiar to the network map are listed below-

- The Network configuration is quasi-static. Nodes, links and networks are being added,updated and deleted someplace or the other.
- The Network is huge and geographically distributed.
- The network spans several political and administrative areas. The related information is also controlled and maintained in a distributed fashion.

In short, global network configuration information is unwieldy and growing continuously. It is impossible to service such information in a centralized fashion. There is need for a distributed framework which allows users and applications to access information about users, services, networks, ... easily and globally. The OSI X.500 Directory services [X.500-88] provides a rich framework to support a globally distributed information service system. The X.500 Directory is intended to be a very large and highly distributed database. It is structured hierarchically with entries arranged in the form of a tree in which each object corresponds to a node or an entry. Information is stored about an object as a set of attributes.

4 The hierarchical model of a network

For representing networks in the Directory we use the following hierarchical model.

A network is the media for transmitting information with zero or more network elements each having at least one network interface on the media. The media may be any kind of a line (physical circuit/virtual circuit), or a collection of interconnected networks.

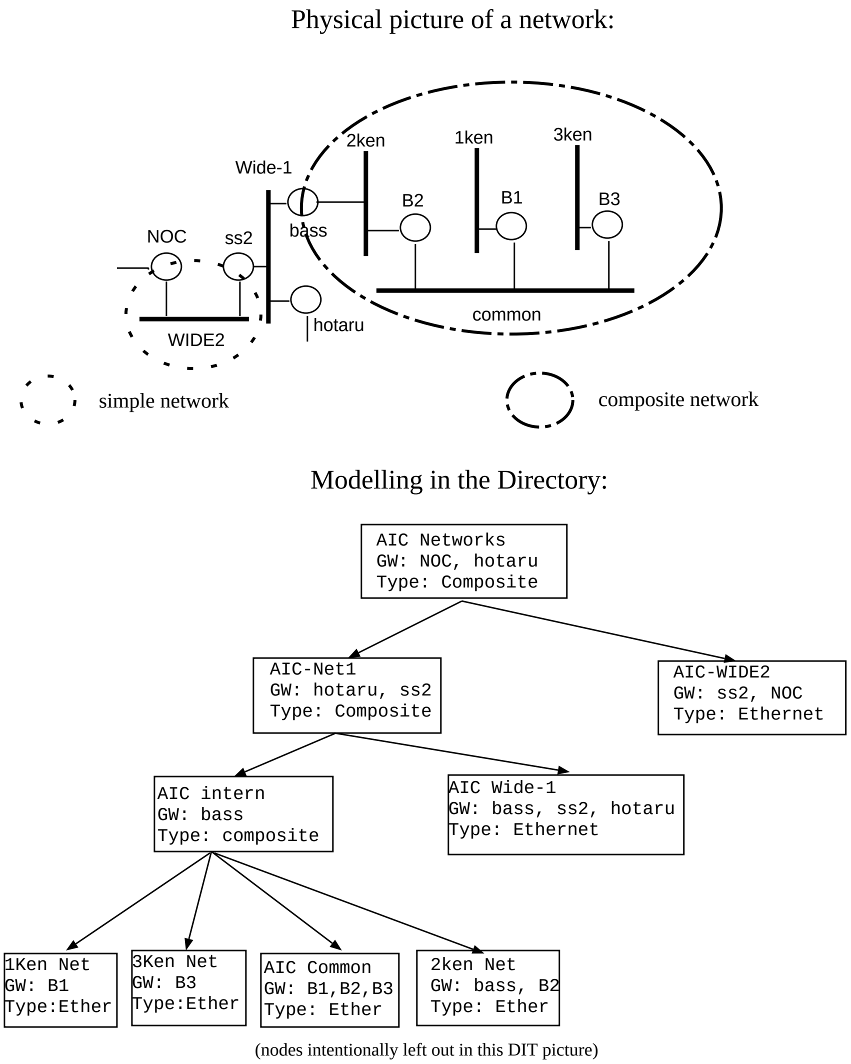


Figure 1: Simple and composite networks and their mapping to the DIT.

The model allows hierarchy of subnetworks. Network elements with multiple interfaces may act as *external gateways* to the attached network and to networks higher up in the hierarchy. Thus, a gateway may be the external gateway of several networks which are either interconnected or have a hierarchical relationship.

A network may be *simple* consisting of zero or more network elements or *composite* consisting of several sub-networks. Examples of simple networks are ethernets, optical fiber/copper cables, free space, ...

4.1 Network Maps

Using the above model it is straight forward to draw the topological graph of the network where the vertices represent the components of the network and edges indicate the connections. For visual representation the graph may be translated to a more "physical" illustration (figure 1) .

Just as there are several maps of the same geographical domain (political, natural...) one can envisage several views of the same network and its components. A view (called "image" in the remainder) could pertain to a particular protocol suite (IP/OSI/...), an administrative domain or purpose. Using images, several abstractions of the same object are possible.

4.2 Representation in the X.500 Directory

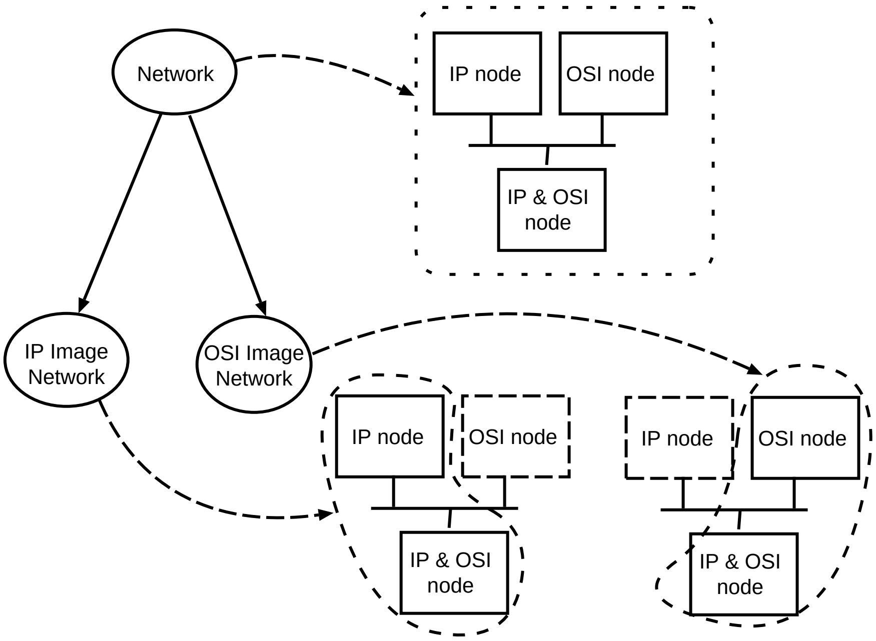
To represent the various images of networks and its components along with the real-image relationship among the various objects we introduce the following classes of objects:

- Communication Object Class (CO): All objects defined furtheron in this document belong to this class. Common attributes for all communication objects are defined in section 6.
- Physical Communication Object Class (PCO): A subclass of CO-class, this class defines common properties for all objects representing physical communication objects.
- Image Communication Object Class (ICO): A subclass of CO-class, this class defines common properties for all objects representing images of communication objects.

The above classes sort communication objects into physical or image object. As is implied in the nomenclature a physical object will have several attributes describing it physical properties - location, weight, size, ... etc. An image object will have an *Image-of* attribute. The *Image-of* attribute will point to a physical object or to another image object.

Using this schema it is possible to represent the case of several logical network systems (running different protocol stacks - IP, XNS, SNA, OSI, ...) which coexist on the same physical network. Information related to different types of networks, no matter what the underlying communication protocol is, will reside in the Directory in harmony. Also, their interrelation will be represented and accessed in a fashion independent of the source/destination network, namely, using the OSI X.500 protocol.

Schemes for physical networks and logical images of physical networks are defined in section 6.



FUNCTIONAL IMAGES OF A NETWORK

Figure 2: Several logical views of the same physical network

All objects are defined in section 6.

5 Position in the Directory Information Tree (DIT)

Information about networks usually will be contained in the DIT as subordinate of the organization maintaining the network. The network model gets mapped into a tree structure for network elements. There is one *network* object giving general descriptions of the network. Subordinates of this network object are *node* objects for each node element present in the network. Node objects hold *networkInterface* objects as subordinates. A network can be physically or logically subdivided into several (sub)networks. In this case, a network entry will have network objects as subordinates which again build the same structure. These entries may be kept as subordinates of organizationalUnit entries as well, with pointers from the "root" network.

This structure holds for physical and logical elements. Physical elements are named network, node and networkInterface, and logical elements are named networkImage, nodeImage and networkInterfaceImage.

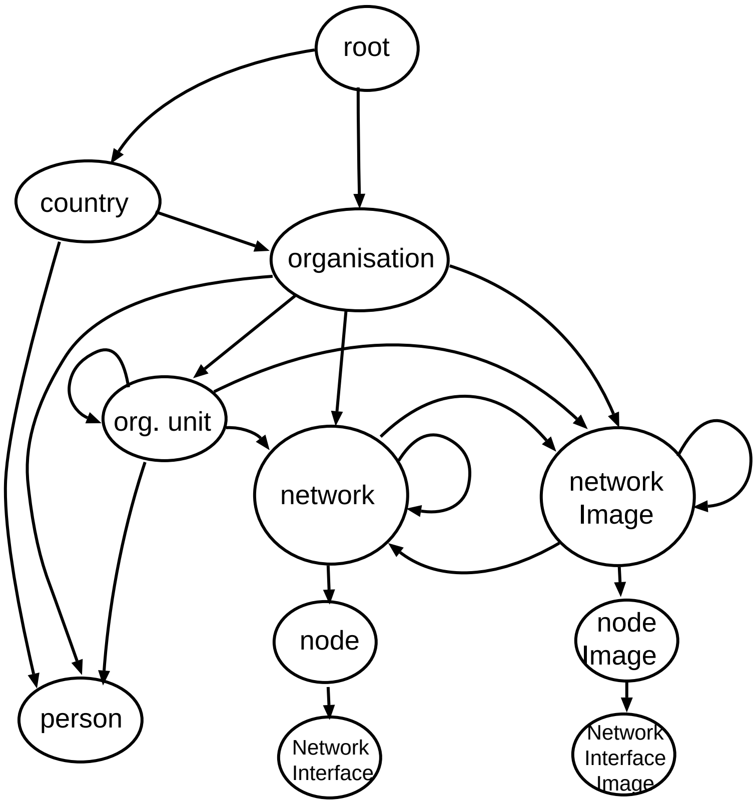


Figure 3: Part of the Directory Information Tree, showing relations of White Pages and network objects

6 Proposed Schemes

A physical network comprises of wires and machines. The physical map of the network will show the interconnections of these nodes by these circuits.

For each physical network element, one or more images may exist. Similarly, an image may be attached to one or more physical objects. The types of images can grow along with the requirements. Relationship between elements (physical or logical) are expressed by attributes or the position in the Diretory tree.

Problems that are addressed in the schema:

1. Avoiding data duplication
2. Preserving administrative boundaries/controls.
3. Simple representation (minimal number of pointers)
4. Security: Though no special emphasis has been placed in this work we believe the X.500 access control policies will provide reasonable privacy.

Problems that are not addressed:

1. Caching policies, etc.: to be decided locally

6.1 Communication Object Classes

The object classes introduced in section 4 are defined as follows:

```
CommunicationObject OBJECT CLASS
    SUBCLASS of top
    MAY CONTAIN {
          description ::  CaseIgnoreStringSyntax,
                   /* can contain any information about the object,
                   however, whereever an appropiate attribute
                   exists, this should be used first to hold
                   information */
          adminContact ::  distinguishedNameSyntax,
                   /* points to the person which is responsible for
                   the administration of the instance this object
                   describes;
                   This refers to the instance only in the context
                   of the concrete object class */
          technContact ::  distinguishedNameSyntax,
                   /* points to the person which is responsible for
```

```
                   the technical maintanence of the instance this
                   object describes;
                   This refers to the instance only in the context
                   of the concrete object class;
                   Availability (e.g.  hours of service) is not
                   covered by this attribute.  */
          }

PhysicalCommunicationObject OBJECT CLASS
    SUBCLASS of CommunicationObject
    MAY CONTAIN {
          owner ::  distinguishedNameSyntax,
                   /* refers to organization or person owning the
                   physical element;
                   Note that more detailed information (like lease,
                   rental, etc.)  can be covered in a specific image
                   (ownerImage) of this element */
          localityName ::  CaseIgnoreStringSyntax
                   /* where the object is located;
                   can be used freely to "spot" a network element,
                   e.g.  state/city/street/building/floor/room/
                   desk/...  */
          ICO ::  distinguishedNameSyntax
                   /* points to image object the physical object
                   is related to;
                   might have several values if physical object
                   is use for several applications at the same time */
          }

ImageCommunicationObject OBJECT CLASS
    SUBCLASS of CommunicationObject
    MAY CONTAIN {
          type ::  caseIgnoreStringSyntax,
                   /* expresses the view this object referes to, e.g.
                   view of provider/user/IP/OSI/...;
                   Note that this information can be covered by
                   the object class in some cases
                   (e.g.  ipNetworkImage gives the IP view) */
          imageOf ::  distinguishedNameSyntax,
                   /* points to physical/image object the image
                   is related to;
                   might have several values if view applies to
                   several physical objects at the same time */
          }
```

6.2 Physical elements

The following objects describe network elements without saying anything about their usage. All objects inherit properties of the PhysicalCommunicationObject class.

6.2.1 Network

The network object supplies general descriptions which are common for a set of nodes and circuits comprising one network. This includes information about the type of circuits (medium, broadcast or point-to-point, etc.) and properties (speed etc.).

```
network OBJECT CLASS
    SUBCLASS of PhysicalCommunicationObject
    MUST CONTAIN {
          networkName :: caseIgnoreStringSyntax }
    MAY CONTAIN {
          externalGateway :: distinguishedNameSyntax,
                   /* points to one or more nodes that connect
                   this network to neighbor networks;
                   whether a node actually is used as gateway
                   for one or the other protocol, is defined
                   in a related networkImageObject */
          nwType :: caseIgnoreStringSyntax,
                   /* type of this network;
                   either "composite" (if consisting of subnetworks)
                   or type of a line:
                   bus, ring, star, mesh, point-to-point */
          media :: caseIgnoreStringSyntax,
                   /* if network is not composite,
                   describes physical media:
                   copper, fiber optic, etc. */
          speed :: numericStringSyntax,
                   /* nominal bandwidth, e.g. 64 kbps */
          traffic :: numericStringSyntax
                   /* (average) use in percent of nominal bandwidth
                   [ this needs more specification later ] */
          configurationDate :: uTCTimeSyntax,
                   /* date when network was configured in current
                   shape */
          configurationHistory :: caseIgnoreStringSyntax
                   /* list of configuration changes, like:
                   added/removed nodes, lines */
          }
```

6.2.2 Node

The node object describes any kind of device that is part of the network, such as simple nodes, printer, bridges.

```
node OBJECT CLASS
    SUBCLASS of PhysicalCommunicationObject
    MUST CONTAIN {
          nodeName ::  caseIgnoreStringSyntax }
    MAY CONTAIN {
          machineType ::  caseIgnoreStringSyntax,
                   /* e.g.  main frame, work station, PC, printer;
                   might include manufacturer */
          OS ::  caseIgnoreStringSyntax,
                   /* e.g.  VM, UNIX, DOS; might include release */
          }
```

6.2.3 NetworkInterface

Each node object will have one or more *networkInterface* objects as subordinates. NetworkInterface objects provide information about interfaces of the node and connectivity.

```
networkInterface OBJECT CLASS
    SUBCLASS of PhysicalCommunicationObject
    MUST CONTAIN {
          networkInterfaceName ::  caseIgnoreStringSyntax }
                   /* It is suggested that the networkInterface name
                   is derived from the name of the logical
                   device this networkInterface represents for the
                   operating system, e.g.  le0, COM1 */
    MAY CONTAIN {
          networkInterfaceAddress ::  caseIgnoreStringSyntax,
                   /* this should contain a protocol-independent
                   interface address, e.g.  Ethernet board number */
          connectedNetwork ::  distinguishedNameSyntax,
                   /* pointer to object of network which this networkInterface is
                   connected to */
          }
```

6.3 Logical Elements

An abstract view of a physical element is called image in this document. The word image gets appended to the object type, leading to the new objects networkImage, nodeImage and networkInterfaceImage. Images will either build Directory trees of themselves or be stored as part of the physical network tree (see section 5.).

Image objects inherit properties of the ImageCommunicationObject class.

Each image object has specific attributes which vary depending on the point of view (IP, OSI, ...). Also, the user and provider of an image will view an object differently; further a user of an object may also be providing the services of the same object to another user.

Therefore, in the following a complete and general list of attributes cannot be given. We recommend to define subclasses of Image classes for each logical view. These subclasses inherit all attributes defined with the object classes below and add more specific attributes. Examples for an IP-view are given in [IP].

6.3.1 Network

There may be several network images for one and the same physical network: one for each protocol, application, etc.

```
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 images referes to */
          speed ::  numericStringSyntax,
                   /* nominal bandwidth for the channel dedicated
                   to this protocol or application ,
                   e.g.  64 kbps */
          traffic ::  numericStringSyntax,
                   /* (average) use in percent of nominal bandwidth
                   [ this needs more specification later ] */
          charge ::  numericStringSyntax
                   /* amount of money that has to be paid to
                   service provider for usage;
                   [ it is felt that this needs further definition:
                   e.g.  monetary unit / time unit, monetary unit /
                   data unit ] */
          }
```

6.3.2 Node

Name and functionality within the network might vary for a node from protocol to protocol considered. In particular, a node might act as gateway for one protocol but not for the other. Routing policy is stored in the case of policy gateways.

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

6.3.3 NetworkInterface

As with physical nodes, nodeImages have networkInterfaces (networkInterfaceImages) which describe connectivity to other network elements. NetworkInterfaceImages are only given if the protocol is establishing connections via this networkInterface.

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

7 Security Considerations

Security Considerations are not discussed in this draft. However, data access control will be taken care of by means of the Directory (namely X.509).

8 Authors' Addresses

Thomas Johannsen
Dresden University of Technology
Institute of
Communication Technology
D-01062 Dresden
Germany

EMail:
Thomas.Johannsen@ifn.et.tu-dresden.de

Glenn Mansfield
AIC Systems Laboratories
6-6-3 Minami Yoshinari
Aoba-ku, Sendai 989-32
Japan
Phone: +81-22-279-3310

glenn@aic.co.jp

Mark Knopper
Merit Network, Inc.
1071 Beal Avenue
Ann Arbor, MI 48109

EMail: mak@merit.edu

References

[X.500-88] CCITT:
The Directory - Overview of Concepts, Models and Services;
Recommendations X.500 - X.521

[HK 91] Hardcastle-Kille, S.:
X.500 and Domains;
RFC 1279, November 1991.

[IP] Johannsen, Th., G. Mansfield, M. Kosters, S. Sataluri:
Representing IP information in the X.500 Directory;
Draft proposal OSI-DS-38, August 1993.

[SPP] Johannsen, Th., G. Mansfield:
The Soft Pages Project;
Draft proposal OSI-DS-39, February 1993.

[OSI-DS-14] Weider, Ch., M. Knopper:
Interim Schema for Network Infrastructure Information in X.500;
Internet Draft osi-ds-14, March 1991.

[OSI-DS-16] Weider, Ch., M. Knopper:
Schema for Network Information Center (NIC) Profiles in X.500;
Internet Draft osi-ds-16-01, March 1992.

[OSI-DS-19] Weider, Ch., M. Knopper, R. Lang:
Interim Directory Structure for Schema for Network Infrastructure Information;
Internet Draft osi-ds-19, April 1991.

[Knopper] Knopper, M:
Representing Networking Infrastructure Information in X.500;
Internet Draft, July 1992.

[WHOIS] Harrenstein, K., M.K. Stahl, E.J. Feinler:
NICNAME/WHOIS;
RFC 954, October 1985.

[DNS] Mockapetris, P.V.:
Domain names - implementation and specification;
RFC 1035, November 1987.