5 NAMES AND ADDRESSES

John Schoch's well-known differentiation of names, addresses, and routes (Schoch 1978) unintentionally spawned widespread misunderstanding of the role of naming in network architectures. It asserts that names and addresses are fundamentally different and that routes and addresses are unrelated. In fact, anything that serves semantically to identify is, by definition, a name. An address is simply a name with special properties: an address is also a name (of a point within a specified coordinate system), but not all names are addresses. A route is not an identifier at all, but a specification of a path from one point to another in a graph that represents the topology of a particular network at a particular point in time.

To understand the role that they play in open systems networking, it is not important to recognize a high-level distinction between names in general and the names of network locations (addresses). In this chapter, the discussion of "names" applies as well to the special class of names that are addresses. Routes, since they are not identifiers, are covered elsewhere (in Chapter 14). We also discuss the role of registration authorities in the administration of open system naming schemes and describe the most important classes of names and addresses (some of which are covered, in much greater detail, in the chapters that describe the network components that actually use them).

Names

The importance of naming schemes should be intuitively obvious to anyone who has ever dealt with a large distributed system of any kind (such as the postal mail system, a motor-vehicle registry, or a computer network). A naming scheme specifies the structure and significance of names and the way in which a name is allocated (selected from the set—the "name space"—of all possible names) and assigned (associated with, or "bound to," the particular object for which it is the identifier).

A naming scheme may be characterized according to a number of basic criteria, including:

- *Scope:* whether a naming scheme is intended to apply globally to an entire class (or to entire classes) of objects or is intended to be understood only within and with respect to a particular local context
- *User-friendliness:* whether a name must be intelligible to and usable by a human user or is intended only for communication among "nonhuman" network elements
- *Scale:* the ability of a naming scheme to accommodate an increase in both the number and (potentially) the internal complexity of names as the size of the corresponding system increases
- *Permanence:* whether the binding of a name to an object is transient (and whether or not the name may be reassigned to another object after its association with one object has ended) or persistent (and if the latter, whether or not the name or some part of it must be registered with a formal registration authority in order to ensure that it can be used unambiguously throughout an open system)

The way in which a particular type of name will be used determines the importance of each of these criteria to the design of an appropriate naming scheme. For example, a system of names for houses along a street ("street numbers") will generally be designed for limited scope (the context is provided by the street; different houses on different streets may have the same street number without ambiguity); a high degree of user-friendliness (since human "users" will be writing these street numbers on envelopes; they should probably not, for example, start at 5,354,201 or some other large number); the ability to scale to at least the number of houses that might reasonably be built along the street, coupled with an assignment rule that leaves enough space between adjacent house numbers to accommodate the future construction of new houses on side lots; and of course, permanence (for which the guarantor is often the town's tax assessor).

Hierarchy

The most straightforward way to keep track of the names that have been allocated from a particular name space is simply to enumerate them in a list, in which each name appears once. If the number of names that have been allocated is "small enough,"[1] this is also a perfectly practical way to keep track of them and to advertise them if necessary (so as to make the names, and their object bindings, known to potential users); but if the number grows too large, the cost of maintaining and distributing the simple enumerated list of names eventually becomes excessive. However, for a "flat" name space, from which names with no discernible structure are allocated, there is no alternative: every reference by name to a set of objects must name every member of the set individually.

If the names associated with a particular name space do have a discernible structure, however, they can be grouped accordingly. The structure may be syntactic or semantic, or some combination. In the case of street numbers, for example, it is common to allocate only odd numbers to houses along one side of the street and only even numbers to houses along the other side, which creates a discrimination that is based on a semantic attribute of the house number. In order to make a statement about the entire collection of houses on one side of the street, it is not necessary to enumerate every house number; it is sufficient to refer to "the houses with odd [or "even," as the case may be] house numbers." In large condominium complexes consisting of more than one building, it is also common to name individual condominium units by means of a two-part identifier, the first part of which names the building and the second part of which names the unit within the building (such as "B-42" for unit 42 in building B). The syntax of the name creates a discrimination that allows the local fire department, for example, to determine which building to train its hoses on without individually identifying all the units contained within that building (all of which, in the case of a building fire, share a common fate).

The second example illustrates the usefulness of *hierarchy* as a way of structuring an address space. In a hierarchically structured name space, individual names (and the objects to which they are bound) can be effectively grouped into larger and larger aggregations based on some property of the objects that is reflected in a corresponding hierarchical property of the name space. Such a structure, which is commonly represented by an inverted tree diagram, makes it possible to refer to a collec-

1. "Small enough" is, of course, a subjective measure; what is "small enough" for a computer's database-management system may be "too large" by far for a human administrator with a clipboard.

tion of objects by giving just the name of the collection (the subset, or "subtree," of the structure that contains all the objects that share a common property of interest) rather than exhaustively naming the individual elements of the subset.

Although hierarchies are generally established according to criteria that are related to some property of the named objects—their location within a network, for example—they may also be established purely for administrative convenience, without reference to any essential property of the named objects. It might be convenient, for example, to define a hierarchy within a hitherto flat name space in order to distribute the job of allocating and assigning names among several managers, even though a hierarchically structured name space does not confer any benefit with respect to dealing with the named objects; in such a case, there is no practical significance to the fact that the name of an object appears in one part of the hierarchy or another.

Occasionally, the desire to establish a hierarchical structure based on the corresponding real-world organization of the named objects collides with an equally strong desire to establish a hierarchical structure based on the administrative convenience of the registration authority or authorities responsible for the task of managing the assignment of names. More often, however, there is a natural relationship between the way in which objects are organized (the practical hierarchy) and the way in which their names are most readily managed (the administrative, or naming, hierarchy).

The difficulty of managing a flat name space increases linearly with its size. The difficulty of managing a hierarchical name space can be made to increase at a much slower rate by judicious selection of the hierarchy. As we will see in the chapters on the network layer (Chapter 13) and routing (Chapter 14), the selection of an appropriate hierarchy determines whether or not a network architecture *scales* well—that is, whether it supports a network that can continue to grow in size and extent or creates practical upper bounds on size and extent that inhibit the network's growth.

OSI Naming Architecture

The OSI naming architecture is described in part 3 of the OSI reference model (ISO/IEC 7498-3: 1989), *Naming and Addressing*. In the layered architecture of OSI, *entities* represent the active agents that operate within a layer to carry out its assigned functions (see Chapter 3); entities are named by *entity titles*, or simply *titles*. Service access points (SAPs) represent the logical interfaces between adjacent layers, at which the lower-layer service is conceptually presented to and received by the upper layer. Service access points are named by *service access point addresses*.

In principle, each of the seven layers of the OSI reference model contains named ("titled") entities, and each of the six layer service boundaries (there is no application-layer service boundary) is speckled with addressable service access points. In practice, however, the only important titles are those associated with entities in the application and network layers: application entity titles (AE-titles), because they are the basic unit of identification for OSI applications (the "end users" of an OSI network), and network entity titles (NETs), because routing information is exchanged among network entities independent of any particular data flow between network service access points. None of the other layers contains entities that need to refer to each other independent of the service access points to which they provide access. The truly important addresses are the network, or network service access point (NSAP), address (which identifies individual hosts connected to the network), and the subnetwork address (which identifies a point at which any system—host or router—is logically or physically attached to a real network).

Service access point addresses above the network layer are constructed by simply concatenating a *selector* to the next-lower-layer service access point address. The selectors are used to accomplish a "fan-out" at the layer service boundary to (potentially) multiple entities in the layer above. The presentation address is the culmination of this process of address composition—it identifies an application entity, which is as far up the stack as you can go in the OSI architecture. The presentation address is particularly significant as the point at which the names associated with applications are coupled to the addresses that are used by OSI-defined services and protocols (see Chapters 7 and 11). Below the network layer, the abstract concepts of "service access point" and "service access point address" bear only a forced and awkward relevance to the technology-specific data link and physical components of real networks and are best left unexplored.[2]

Application Entity Titles

Since the ultimate sources and sinks of information in an OSI network are application entities, every end-user interaction in an OSI environ-

2. This is an excellent example of the benefit of knowing what is and is not important! The ISO and CCITT committees concerned with data link–layer standards have spent endless hours debating the existence and properties of data link addresses without coming to any satisfying conclusions that apply to the data link service in general (rather than to one or more data link technologies in particular). This unproductive exercise has been pursued because the groups involved failed to recognize a very basic point: that the concept of "service access point address" is not useful in the context of the data link service (notwithstanding the reference model mandate that all service boundaries have SAPs and all SAPs have addresses).

ment depends, at some point, on the identification of the participating application entities by their application entity titles, and by implication, the presentation address associated with that title.

Prior to the development of the X.500-series standards for the OSI Directory, the syntax of application entity titles was poorly defined. However, this caused very few problems because, in the absence of a directory, the early end users of OSI were unlikely to make use of any of the application-layer features that would have required them to know what an application entity title was. The ASN.1-encoded, generic form of application entity title, contained in early drafts of the OSI standard for registration authorities (ISO/IEC 9834-1: 1991), was both very simple and not very useful:

```
AE-Title ::= SEQUENCE {
             AP-Title,
             AE-Qualifier }
```

The registration authority standard also defined an all-numeric form of application entity title, in which the application process title (`AP-Title` in the ASN.1 definition) part is syntactically an object identifier, and the application entity qualifier (`AE-Qualifier`) part is an integer:

```
AP-Title ::= IMPLICIT objectIdentifier
AE-Qualifier ::= INTEGER
```

The agreements reached by ISO/IEC and CCITT on the 1988 (and subsequent) directory standards (see Chapter 7) included the recognition of two well-defined forms for the application entity title, one of which is the object identifier form just shown. The other, much more useful, form is one in which the application entity title is syntactically equivalent to a directory distinguished name:

```
AP-Title ::= RDNSequence
AE-Qualifier ::= RelativeDistinguishedName
```

A relative distinguished name sequence (`RDNSequence`) is just a sequence of relative distinguished names of entries in the directory information tree; in this second form of application entity title, the application process title contains the relative distinguished names of the directory entries from the root down to (but not including) the entry that represents the application entity, and the application entity qualifier adds the relative distinguished name of the application entity entry itself. This makes it easy to list an application entity in the directory and to refer to it by a distinguished name that may be used as a valid title for the application entity.

System Titles

Recognizing that applications—even OSI applications—necessarily reside in systems, and that people have become accustomed over the years to thinking about applications as "running on" a particular computer system, the OSI naming scheme defines the concept of *system title*, the effect of which is to bring the decomposition of the application entity title, begun in the previous subsection, even closer to the real world.

The concept of system title is equivalent to the familiar concept of *host name* in TCP/IP; it is a permanent identifier for a particular OSI end system (host), and the identification of applications running on that end system can be based on it by further decomposing the application process title in either its object identifier form or its relative distinguished name form:

Generic form:

```
AP-Title ::= SEQUENCE {
                System-Title,
                AP-Qualifier }
```

Form 1:

```
System-Title ::= RDNSequence
AP-Qualifier ::= RelativeDistinguishedName
```

Form 2:

```
System-Title ::= objectIdentifier
AP-Qualifier ::= INTEGER
```

The entire process of application entity title formation can now be traced back to the registration of system titles, or host names. The distinguished name form of an application entity title begins with a high-order sequence of relative distinguished names registered (implicitly, as a simple consequence of being entered in the directory information base, or explicitly) by a system-title registration authority, to which another relative distinguished name is concatenated to form an application process title, to which another relative distinguished name is concatenated to form an application entity title—*et voilà!* A similar process of concatenation of subidentifiers to a base system-title object identifier creates the object identifier form of an application entity title.

By convention, registration authorities for system titles and application process titles require the simultaneous allocation of both forms (object identifier and relative distinguished name sequence), which ensures that the values of both forms are (1) interchangeably available for use (they appear, logically, in the same place in the register) and (2) protected from allocation as the value of any other title.

TCP/IP Naming Architecture

The naming architecture of TCP/IP, being less deliberately abstract than that of OSI, is correspondingly simpler. It is entirely captured by the Domain Name System, which is discussed in Chapter 7.

Addresses

In the context of open systems networking, an address identifies a specific point in a graph that represents the topology of a particular logical network. That's pretty general, but in the OSI architecture, the concept of "address" is even broader, including the identification of points on the interfaces between the layers of the OSI reference model.

It turns out that the only OSI addresses worth talking about are the ones that are concerned with network and subnetwork topologies. In the TCP/IP architecture, which has never been encumbered by addressable points on its layer interfaces, the interesting addresses are the TCP/IP equivalents of OSI's network (NSAP) and subnetwork addresses: the IP (internet) address and the addresses of network interfaces. Both OSI network addresses and IP addresses are best understood in the context of the network layer and the routing protocols that use them and are thus discussed in more detail in Chapters 13 and 14. Subnetwork and interface addresses are inherently technology-specific: no matter how wonderful and/or pervasive a particular network is, there is always, somewhere, another (different) network to which it eventually must be connected, wherefore no subnetwork-specific address is a substitute for an internetwork address.

Registration Authorities

In order for names to be useful, there must be a way to ensure that the allocation and assignment of names is unambiguous and that the relationship between a name and the "thing" it names is maintained in such a way that users can find out what it is (by "looking it up"). The latter function is performed by a directory service and is covered in Chapter 7; the former is performed by a *registration authority*.[3]

A registration authority plays two important roles with respect to a

3. It should be pointed out here that the term *registration* can imply either simple "listing," in which a name-to-thing relationship is recorded in a list (but about which the listing

particular class of names: it administers the name space, managing the actual allocation of values from the name space and the assignment of those values to things that are entitled to be identified by a name of that class; and it maintains a record of the name-to-thing bindings that it has created. This record of bindings may then be incorporated into a directory (see Chapter 7). It is important to recognize, however, that registration and directories are separate systems (although in some cases, as we shall see in our discussion of the Domain Name System in Chapter 7, they are inextricably linked). Registration may proceed whether or not there is a directory in which the registered names are listed. A directory, however, implies at least an informal or implicit registration authority responsible for the names that appear in the directory.

Registration Authority Procedures

In the OSI world, the way in which registration authorities operate is defined by an international standard, ISO/IEC 9834-1: 1991, which is (intentionally) identical to CCITT Recommendation X.660. This standard specifies a tree-structured name-registration hierarchy, within which individual registration authorities operate, exercising hegemony over that part of the global name space that is represented by the subtree for which the authority is the root. The topmost levels of the registration hierarchy are defined by ISO/IEC 8824 (CCITT X.208) and ISO/IEC 9834-1 (CCITT X.660), which together specify the registration hierarchy shown in Figure 5.1.

In the world of open systems, there are a great many identifier types that must be registered in order to ensure that they can be used unambiguously. In the OSI world, these include:

Document types
Standardized object identifiers
Virtual Terminal profiles and control objects profiles
AP titles and AE titles
Abstract syntaxes

agency makes no assertions of any kind), or "formal registration," in which the registering agency not only records and lists a name-to-thing relationship but also confers "legitimacy" on it—guaranteeing, for instance, that the name is not bound to any other object or that the person or organization registering the name has some legal right or entitlement that is expressed by the fact that the name is registered with the agency. At this time, it is not clear which model of "registration" applies to the registration authorities that have been and are being formed in support of open systems networking. It is likely, however, that registration authorities established by international standards or, within countries, by national governments or national standards organizations will operate in accordance with the "formal registration" model but that the directory systems in which name-to-thing bindings are recorded for others to "look up" will operate according to the "listing" model.

Transfer syntaxes
Application context identifiers
System titles
Organization identifiers
Administration (public) messaging domain/private messaging domain (of X.400 MHS) names for message handling systems

Many of these identifiers are registered statically by the specification of one or more ISO/IEC and/or CCITT standards or by national standards. Others are registered dynamically, as necessary, by registration authorities constituted specifically for that purpose.

The U.S. Registration Authority

The American National Standards Institute (ANSI) operates a registration authority that assigns organization names in two forms: (1) an alphanumeric form, which may consist of from 1 to 100 characters chosen from a specified character set,[4] and (2) a numeric form, which consists of a posi-

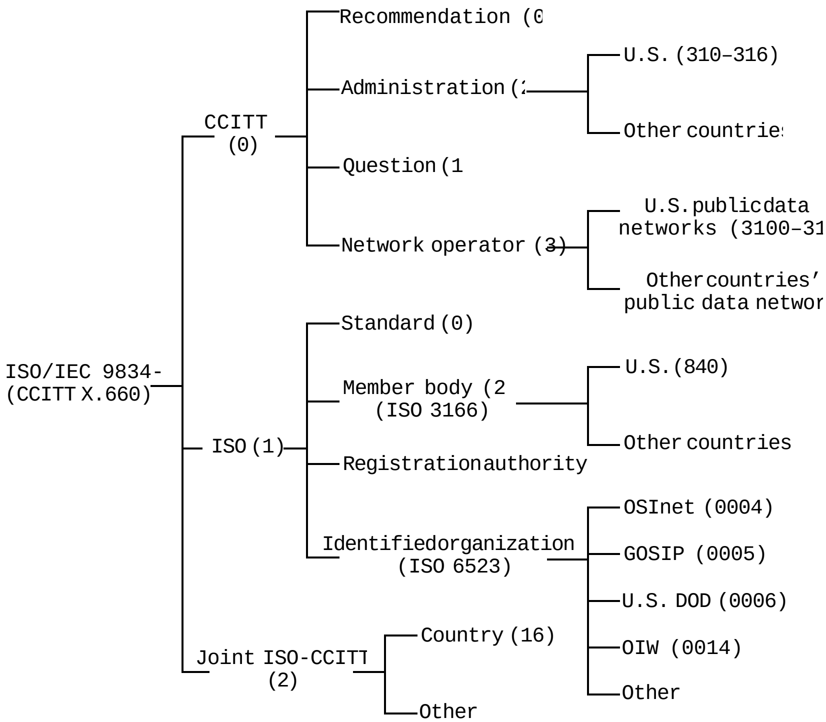

FIGURE 5.1 The OSI Name-Registration Hierarchy

4. The character set is defined as registration number 102, the Teletex Set of Primary

tive decimal integer greater than or equal to 113,527 and less than 16 million. The numeric form may be obtained with or without a corresponding alphanumeric form, but the alphanumeric form is always accompanied by (and in the register, associated with) a numeric form.

AHA *Many people have wondered why ANSI decided to begin the assignment of numeric organization names at the nonintuitive value "113,527," which is not, among other things, a power of 2 (or any other interesting number). Realizing that they had to specify some value as the starting point (and hoping to avoid conferring any special cachet on the recipients of the first few numbers by starting at, say, "1"), the members of the U.S. registration authority committee were about to pick a "logical" number ("1,000," perhaps, or "2,048") when Jack Veenstra, the chairman of the committee, shouted "113,527!"—which was promptly dubbed the "Veenstra constant" and written into the registration authority procedures. Later, the members of the committee arranged for AT&T (Veenstra's employer) to receive the numeric organization name "113,527" in Jack's honor.*

An organization name may be used in several different ways to form unambiguous identifiers. The alphanumeric form is most often used as the organization name in an electronic-mail address (as, for example, the value of the "organization" or "organizational unit" elements of an X.400 originator/recipient address containing "/C = US"; see Chapter 8). The numeric form may be used as the value of the "organization" field in an OSI NSAP address constructed according to ANSI Standard X3.216-1992 (see Chapter 13) or as part of an object identifier prefix for constructing unambiguous object identifiers for organization-specific objects (such as, for example, organization-specific management information base variables for use with a network management system). The registration authority does not specify or constrain the way in which an organization name may be used, nor does it guarantee the legal right of an organization to actually use the name. Only one guarantee comes with an organization name obtained from ANSI's registration authority: that ANSI has not previously assigned the same name to anyone else and will not do so in the future.

Until recently, the U.S. name-registration authority conducted its business under the `{ iso (1) member-body (2) us (840) }` arc of the registration hierarchy (see Figure 5.1), registering names for ANSI

Graphic Characters, of the ISO International Register of Character Sets to Be Used with Escape Sequences, plus the space character.

standards, private organizations with U.S. national standing, and the names of U.S. states and "state equivalents." In 1991, however, changes in the registration authority procedures standard—adopted as a result of joint efforts by ISO/IEC and CCITT to align their registration procedures, leading to a single standard (ISO/IEC 9834 | CCITT X.660)—invalidated this procedure; they required that private organization names with national standing be registered under the `{ joint-iso-ccitt (2) country (16) us (840) }` arc of the registration hierarchy. This had two immediate, significant consequences: organization names already registered under the `{ 1 2 840 }` arc were suddenly "homeless," and policy control over the assignment of organization names—previously vested solely in ANSI, which owns the `{ 1 2 840 }` arc as the U.S. member body in ISO, under the old rules—became the joint responsibility of ANSI and the U.S. Department of State (which is the official U.S. representative in the CCITT arena).

The solution to this problem was, fortunately, straightforward; Figure 5.2 illustrates the way in which the new `{ 2 16 840 }` subtree will be jointly administered.

The Federal Information Processing Standard 5 (FIPS-5) subtree will be managed by the existing U.S. government FIPS-5 commission (U.S. Department of Commerce 1987), which defines the names of "states and state equivalents" within the United States. Under the old rules, these names had been "imported" from FIPS-5 directly into the first 100 numeric name slots of the `{ 1 2 840 }` arc; however, since no name assignments were ever made under these values, they have simply been abandoned. Under the new rules, the FIPS-5 state names are expected to be available for use as the values of the `stateOrProvince` attribute in relative distinguished names in the X.500 Directory (see Chapter 7).

The existing register of private organization names will move, intact, from the `{ 1 2 840 }` arc to the `{ 2 16 840 2 }` arc, which will also be administered by ANSI. This "copy" operation, however, will not invalidate existing names (such as organization identifiers and applica-

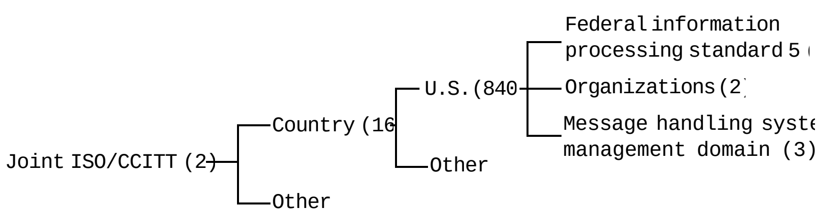

FIGURE 5.2 The U.S. Name-Registration Subtree

tion entity titles) that may already have been constructed using the `{ 1 2 840 }` prefix; in effect, two equivalent prefixes will exist (in perpetuity) for currently registered organization names. New registrations will be made only under the new `{ 2 16 840 2 }` arc, and organizations possessed of an "old" registration will be encouraged (but not required) to construct no new identifiers under the `{ 1 2 840 }` arc but to use the `{ 2 16 840 2 }` arc instead.

The numeric organization names under `{ 2 16 840 2 }` will be used as values of the "organization" field in the construction of NSAP addresses with authority and format identifier values of 38 or 39 and an initial domain identifier value of 840, according to American National Standard X3.210-1992, just as they were under the old rules (see Chapter 13); this usage is not affected by the switch to a new subtree, since only the numeric organization name, without the qualifying `{ 1 2 840 }` or `{ 2 16 840 2 }` prefix, is used in the construction of NSAP addresses.

Since the rule change covers only "private organization names with national standing," the existing mechanism for registering the names of American national standards under the `{ 1 2 840 }` arc is unaffected and will remain in place.

The creation of a new arc for message handling system management domain names recognizes the apparently unreconcilable difference between the general registration of organization names and the registration of names that will be used as administration messaging domain and private messaging domain names for X.400 (see Chapter 8). This is unfortunate, since it means that organization names registered under the organizations arc—including those names that have already been registered under the "old rules"—cannot be used as administration messaging domain or private messaging domain names in an X.400 originator/recipient address unless they are also registered under the new message handling system management domain arc. The very different requirements associated with the registration of organization names that are intended for use in an X.400 context make this situation unavoidable.

AHA *The designers of the original registration authority for the United States recognized that their general-purpose registry, which permitted great flexibility in the formation of an acceptable organization name (it could consist of up to 100 characters chosen from a very large character set), could not enforce the constraints that might be applicable to the use for which a registered name was intended. Message handling system management domain names, for example, are constrained by CCITT Recommendation X.411 to be no longer than 16 bytes and must be constructed from the "Printable-*

String" character set defined by ISO/IEC 8824. They hoped, however, that these additional constraints could be observed in practice without creating application-specific registries for different types of organization name. The difficulties of doing so, at least in the case of organization names that will be used as MHS administration or private domain names, have proved to be too great.

The U.K. Registration Authority

The United States is not, of course, the only country with a need to register names with national standing under the various arcs of the ISO/IEC 9834 registration hierarchy. ANSI's counterpart in the United Kingdom, the British Standards Institute, operates a similar U.K. Registration Authority (UKRA),[5] which is governed by British Standard (BS) 7306 (1990). The U.K. registration authority is more specific than ANSI's; whereas ANSI hands out organization names that can be used for a wide variety of purposes, BSI has elected to assign organization names that can only be used in the construction of OSI network-layer addresses (see Chapter 13) and in fact the UKRA (see Chapter 13) includes the following specific disclaimer: "Users of an X.400 service therefore obtain their O/R addresses through mechanisms provided for that purpose, independently of any allocations that may be made under this standard."

The GOSIP Registration Authority

In the United States, ANSI operates the U.S. registration authority (in conjunction with the Department of State with respect to some of the arcs below `{ 2 16 840 }`; see the subsection entitled "The U.S. Registration Authority," earlier in the chapter), but it is not the only organization registering "national standing" identifiers. The U.S. General Services Administration, acting for the National Institute of Standards and Technology (NIST), administers a GOSIP registration authority that assigns identifiers for use in two specific contexts:

1. GSA will provide U.S. government agencies with strings (all of which begin with the characters "GOV+") that can be used in X.400 mail addresses as part of constructed private management domain or organization names, under a delegation from ANSI to NIST that is registered as the arc `{ iso (1) member-body (2) us (840) gov (101) }`.
2. GSA will provide, to anyone who asks, a numeric value that can be used in the administrative authority identifier field of U.S. GOSIP version 2 NSAP addresses (see Chapter 13), under a delegation of authority from the ISO 6523 part of ISO's subtree that is registered as the arc `{ iso (1) identified-organization (3) NIST (5) }`.

5. In fact, BSI has delegated the responsibility for actually operating the U.K. Registration Authority to the Electronics and Business Equipment Association.

Fortunately, these identifiers are in every case readily distinguishable, in context, from identifiers registered with ANSI under other arcs of the registration hierarchy. ANSI has agreed not to allocate message handling system management domain names beginning with "GOV+" under its `{ 2 16 840 3 }` arc, and the administrative authority identifier values allocated by GSA, although they are semantically equivalent to the organization field values allocated by ANSI, can appear only in NSAP addresses that follow the GOSIP version 2 format (with AFI = 47 and IDI = 0005), whereas the org values can appear only in NSAP addresses that follow ANSI Standard X3.216-1992.

The Internet Assigned Numbers Authority

In the Internet world, the list of names that are formally registered is (remarkably) even longer than it is for OSI. The Internet Assigned Numbers Authority (IANA) is responsible for assigning and registering names (which, in Internet jargon, are simply called "numbers," since that's what they are) of the following types:

Version numbers
Protocol numbers
Port numbers
Internet multicast addresses
The Internet Ethernet address block
IP type-of-service parameter values
IP time-to-live parameter values
Domain Name System parameter values
BOOTP parameter values
Network management parameter values
ARPANET and MILNET logical addresses
ARPANET and MILNET link numbers and X.25 address mappings
IEEE 802 numbers of interest
XNS protocol types
PRONET 80 type numbers
Ethernet numbers of interest
Ethernet vendor address components
UNIX port numbers
Address resolution protocol parameters
Reverse address resolution protocol parameters
Dynamic reverse address resolution protocol parameters
X.25 type numbers
Public data network numbers
TELNET options

Mail encryption types
Machine names
System names
Protocol and service names
Terminal type names

These number registrations are published periodically in an Internet RFC entitled *Assigned Numbers*.

The preceding list does not include the most visible Internet registration activity: the Internet registry, which assigns Internet network numbers (the high-order part of a 32-bit Internet address, from which individual IP host numbers are generated). This activity is formally the responsibility of the Internet Assigned Numbers Authority, but the actual assignment task has traditionally been carried out by the Internet's Network Information Center (NIC). The Network Information Center was operated for many years by SRI International in Menlo Park, California, but was recently reassigned to Network Solutions, Inc., operating under contract to Government Systems, Inc. (GSI), operating under contract to the Defense Information Systems Agency (DISA, which was formerly the Defense Communications Agency, or DCA).[6] In most cases, a public or commercial network operator (service provider) will apply to the Internet registry for network numbers on behalf of its clients; in some cases, the registry preallocates blocks of numbers to a service provider, which is then free to further assign them to its clients as needed.

Object Identifiers

Of the data types predefined by the Abstract Syntax Notation One (ASN.1) standard (see Chapter 4), one—the OBJECT IDENTIFIER data type—is particularly important to naming in OSI. The value of an object identifier names an information object; it is an ordered set of non-negative integer values whose root is defined in ISO/IEC 9834-1 and whose branches or arcs are derived from the registration authorities described earlier and depicted in Figure 5.2.

What are information objects? Applications and systems are objects. The documents that define OSI protocols and services are objects. OSI Directory entries and attributes of those entries (see Chapter 7); X.400

6. If you followed all that, you might be qualified for a career in Internet name assignment; see Appendix D for the address and phone number of the Internet Network Information Center.

Message Handling System mail addresses (see Chapter 8); and the sets of counters, gauges, and status indicators used in the management of network resources by OSI's Common Management Information Service and TCP/IP's Simple Network Management Protocol (see Chapter 9) are among the many things OSI considers objects. Essentially, anything in a network can be modeled as an object; among these, objects that require a name for unique and unambiguous identification are registered by a naming authority.

Conclusion

This chapter has described the roles names play in OSI and the formal composition of OSI addresses. (The roles of names in TCP/IP have intentionally been deferred to Chapter 7, where the Domain Name System is discussed, and the details of network addresses have been deferred to Chapter 13, where OSI network service access point addresses and IP addresses are discussed.) The names most relevant to understanding how OSI applications identify each other for the purpose of conducting information exchange have been specified as well. The authors have also explained that the means by which identifiers in general are guaranteed uniqueness is through the establishment of registration authorities recognized by ISO/CCITT and that a similar infrastructure exists for TCP/IP—i.e., the Internet Assigned Numbers Authority. The chapter concluded with a discussion of the role played by object identifiers in the "name game" and the relationship of this ASN.1 data type to name registration.