Network Working Group
Request for Comments: 830

A Distributed System for Internet Name Service

by
Zaw-Sing Su

This RFC proposes a distributed name service for DARPA Internet. Its purpose is to focus discussion on the subject. It is hoped that a general consensus will emerge leading eventually to the adoption of standards.

October 1982

SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

(415) 859-4576

A Distributed System for Internet Name Service

1 INTRODUCTION

For many years, the ARPANET Naming Convention "<user>@<host>" has served its user community for its mail system. The substring "<host>" has been used for other user applications such as file transfer (FTP) and terminal access (Telnet). With the advent of network interconnection, this naming convention needs to be generalized to accommodate internetworking. The Internet Naming Convention [1] describes a hierarchical naming structure for serving Internet user applications such as SMTP for electronic mail, FTP and Telnet for file transfer and terminal access. It is an integral part of the network facility generalization to accommodate internetworking.

Realization of Internet Naming Convention requires the establishment of both naming authority and name service. In this document, we propose an architecture for a distributed System for Internet Name Service (SINS). We assume the reader’s familiarity with [1], which describes the Internet Naming Convention.

Internet Name Service provides a network service for name resolution and resource negotiation for the establishment of direct communication between a pair of source and destination application processes. The source application process is assumed to be in possession of the destination name. In order to establish communication, the source application process requests for name service. The SINS resolves the destination name for its network address, and provides negotiation for network resources. Upon completion of successful name service, the source application process provides the destination address to the transport service for establishing direct communication with the destination application process.

2 OVERVIEW

2.1 System Organization

SINS is a distributed system for name service. It logically consists of two parts: the domain name service and the application interface (Figure 1). The domain name service is an application independent network service for the resolution of domain names. This resolution is provided through the cooperation among a set of domain

name servers (DNSs). With each domain is associated a DNS.* The reader is referred to [2] for the specification of a domain name server. As noted in [1], a domain is an administrative but not necessarily a topological entity. It is represented in the networks by its associated DNS. The resolution of a domain name results in the address of its associated DNS.

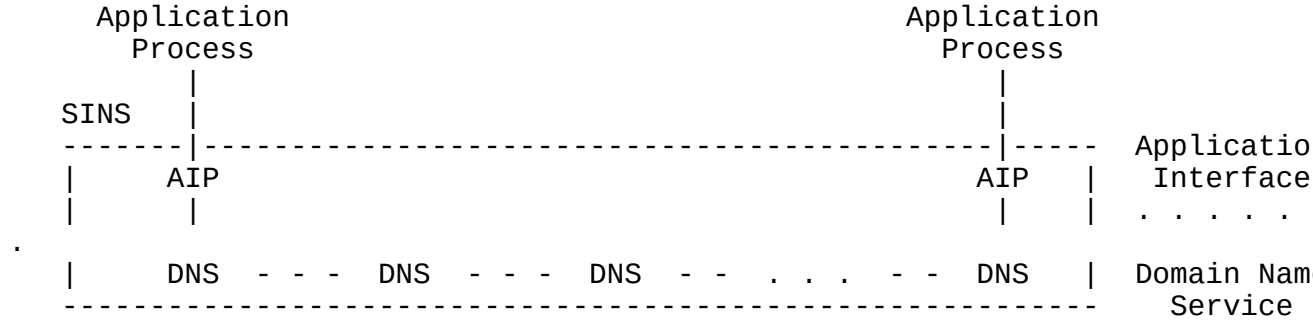


Figure 1 Separation of Application Interface

The application interface provides mechanisms for resolution beyond that of destination domain and negotiation to ensure resource availability and compatibility. Such negotiation is sometimes referred to as the "what-can-you-do-for-me" negotiation. The application interface isolates domain name service from application dependence. It thus allows sharing of domain name service among various user applications.

The application interface consists of a set of application interface processes (AIPs) one for each endpoint domain. For operation efficiency, the AIP is assumed to be combined with its associated DNS forming an endpoint DNS (Figure 2).

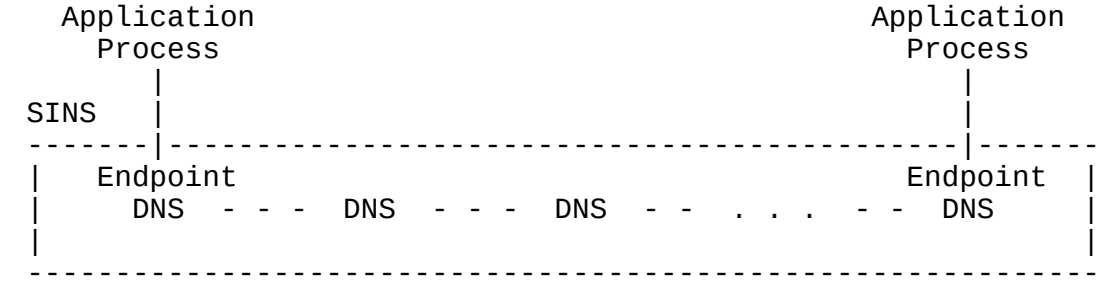


Figure 2 Distribution of Name Service Components Among Domains

---

* For reasons such as reliability, more than one DNS per domain may be required. They may be cooperating DNSs or identical for redundancy. In either case, without loss of generality we may logically view the association as one DNS per domain.

2.2 Domain Resolution

For name service, the source application process presents to its local AIP the destination name, and the application service it requests. For most applications, the application service the source application process requests would be the service it offers. The destination name is assumed to be fully qualified of the form:

<local name>@<domain>.<domain>. ... .<domain>

The domains named in the concatenation are hierarchically related [1]. The left-to-right string of simple names in the concatenation proceeds from the most specific domain to the most general. The concatenation of two domains,

... .<domain A>.<domain B>. ...

implies the one on the left, domain A, to be an immediate member (i.e., the first-generation descendent) of the one on the right, domain B. The right-most simple name designates a top-level domain, a first-generation descendent of the naming universe.

For domain resolution, the AIP consults the domain name service. It presents the co-located DNS with the fully qualified domain specification:

<domain>.<domain>. ... .<domain>

The DNSs participating in a resolution resolve the concatenation from the right. The source endpoint DNS resolves the right-most simple name and acts as a hub polling the other DNSs. It resolves the right-most simple name into an address for the DNS of the specified top-level domain, then polls that DNS with a request for further resolution. When polled, a DNS resolves the next right-most simple domain name. Upon successful resolution, an intermediate DNS may have a choice of either returning the resulting address or forwarding the request to the next DNS for continuing resolution.
When a intermediate DNS receives a reply from the next DNS, it must respond to the request it has received. To simplify the domain name service protocol, an intermediate DNS is not allowed to act as a hub for further polling.

2.3 Application Interface

Addressing for destination endpoint domain is in general not sufficient for the source application process to establish direct communication with the destination application process. In order to establish direct communication, further addressing may be necessary. Addressing beyond destination endpoin domain may be necessary when the addressing of application process cannot be derived from the address of the endpoint domain. To provide such derivation capability permanent binding and universal binding convention, such as TCP port number assignment, may be necessary.

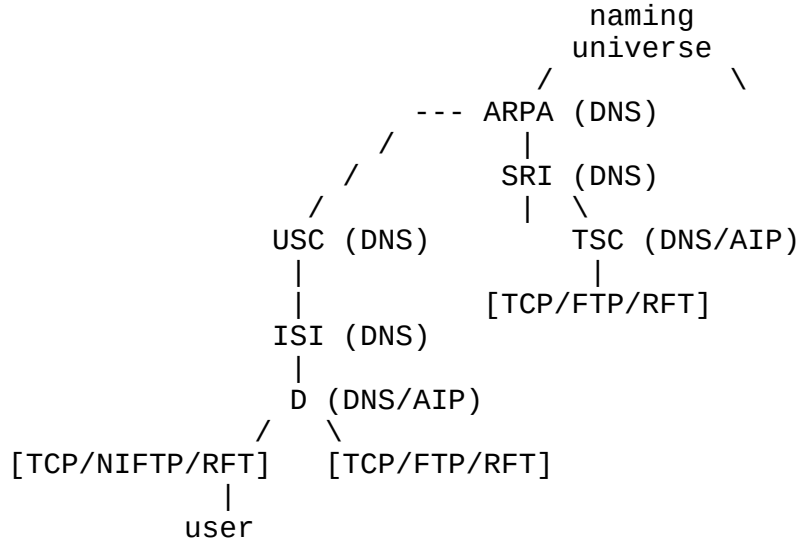
Beyond addressing, negotiation for resource availability and compatibility is often found necessary. The application interface provides a "what-can-you-do-for-me" negotiation capability between the source and destination endpoint domains. Such negotiation mechanisms provided in this design include those for the availability and compatibility of transport service, e.g., TCP or UDP, and application service, e.g., SMTP for mail transport. The availability of such negotiation service may allow dynamic binding and variations in system design.

The application interface offers an integrated service for various "what-can-you-do-for-me" negotiation capabilities.

2.4 Example

Let us assume that a request is made at ISID for remote file transfer using NIFTP to SRI-TSC. The domain name for ISID is D.ISI.USC.ARPA,* and TSC.SRI.ARPA for SRI-TSC. The hierarchical relationship between these two domains is as depicted in Figure 3 below. The NIFTP process (an application process) at ISID forwards the domain name TSC.SRI.ARPA" to the local AIP in domain D for name service. The AIP forwards the fully qualified domain name, "TSC.SRI.ARPA", to its co-located DNS for domain resolution.

ARPA, the right-most simple name, is assumed to designate a top-level domain. The DNS of D recognizes this simple name, resolves it into the address of the ARPA domain DNS, and forwards the request to that DNS with a pointer pointing to the next domain "SRI". The ARPA DNS recognizes "SRI" as one of its subdomains, resolves the address of the subdomain's DNS. It has a choice at this point whether to return this address to the source endpoint DNS or to forward the request to the DNS of SRI.



---

* Domain names used in the examples are for illustration purposes only. The assignment of domain names is beyond the scope of this writeup.

If it returns the address, the source endpoint DNS at D, would continue polling by forwarding the request to the SRI DNS. When the DNS of SRI detects TSC as the last domain in the concatenation, it resolves the address for the DNS at TSC, and returns it to the source DNS at domain D. Upon receiving a successful domain resolution, the source DNS returns the obtained address to its associated AIP.

Since the destination AIP is co-located at this address, the source AIP is able to forward a request with the service designation "TCP/NIFTP/RFT" for "what-can-you-do-for-me" negotiation. Realizing that within TSC there is no NIFTP but FTP provided for remote file transfer, the destination AIP would respond accordingly. Since ISID also offers FTP service, the "what-can-you-do-for-me" negotiation may conclude successfully. The user request for file transfer may thus be satisfied.

3 SYSTEM COMPONENTS

3.1 Component Processes

The two basic distributed components of SINS are the endpoint DNS and the intermediate DNS. An endpoint DNS is associated with each endpoint domain. An intermediate DNS is associated with a domain without any associated application process.

The intermediate DNS is rather simple. It has the resolution capability for translating simple names of first-generation subdomains to addresses of their associated DNS. It also communicates with other DNS for domain resolution.

An endpoint DNS consists of an AIP and a source DNS. The source DNS implements the polling mechanism which communicates with other DNSs as a hub for polling. It also has capability for the resolution of top-level domains. It responds to requests from the local AIP for domain resolution (Section 4.2.3).

The major function of an AIP implements the intellegence of "what-can-you-do-for-me" negotiations. A communication module realizes negotiation exchanges between the source and destination AIPs (Section 4.2.2). As an interface between the application processes and the local DNS, it must also implement communication capabilities for exchanges with the DNS and the application processes.

3.2 Databases for Name Resolution

There is a database associated with each resolution module. The database associated with an endpoint domain contains name-to-address

correspondences for the top-level domains, first-generation descendents of the naming universe. It facilitates the endpoint DNS resolving the right-most simple name of a fully-qualified domain specification.

The database associated with an intermediate domain contains name-to-address correspondences for the first-generation subdomains of this domain. Thus, the required database contents among the intermediate DNS databases are disjoint, and updates are local.

It is also noticed that with the implementation of the SINS, there is no need for database format standardization.

3.3 Caching

The component processes and resolution databases constitute the basic System for Internet Name Service. The distributed components are related according to the domain hierarchy. The databases associated with the endpoint domains are all identical. Containing only name-to-address correspondence for top-level domains, the endpoint database should be rather small in size. The disjoint nature of intermediate DNS databases allows easy local updates.

However, communications will be very inefficient if the Internet name service is called for the establishment of every transaction. A standard solution to aleviate such inefficiency is the use of caching.

Caching is a mechanism reusing previous resolution results. To expedite establishment of communication, the resolution results are stored for future reference. We do not incorporate caching as a standard feature of the SINS. However, we assume the use of caching for efficient operations at individual implementor’s discretion.

4 INTER-COMPONENT COMMUNICATIONS (THE INTERNET NAME SERVICE PROTOCOLS)

In this section, we present a format specification for correspondences between various component pairs. For co-located components, communication becomes interprocess, and the exact format less important. For inter-host communication, the format specification here defines a name service protocol.

The communicating component pairs of concern here are application process/AIP, AIP/DNS, and AIP/AIP. The communications employ request/response commands. A single command structure is adopted for all three pairs; while communications between a particular pair may employ a subset of the commands. Such uniformity allows minimum processing and maximum code sharing for implementation.

4.1 Command Structure

The basic command structure begins with two octets indicating command type and the number of items in the command. They are followed by the indicated number of items. The type of an item is indicated in its first octet, followed by a one-octet content length, and then the item content. Required presence or absence and order of the items for each component pair are specified in this section.

```
Command Type   Number of Items

Item Indicator   Content Length   Item Content
  .
  .
```

Command Type

This type coded in binary number indicates whether this command is a request, an affirmative response, or some other type of response (see Appendix A for the command types and their corresponding code). This type specification implies the presence or absence and order of the following items.

Number of Items

This number is expressed in binary number. It specifies the number of following items. Owing to the possibility of a multiple response, this number may vary for a particular command.

Item Indicator

This indicator defines the item type. The possible types include: service, name, address, and comment. The type of an item implies its content structure.

Content Length

This length specification, in binary, indicates the length of the following content in octets. The maximum can be specified is 255, thus the maximum length of the content. However, this maximum may also be constrained by the total length of the command (Section 4.3).

Item Content

The contents for different items are:

Service -- Transport protocol/service protocol/service type (ASCII). (See Appendix A for standard identifiers for service specifications.)

Name -- Whole or partial name string according to Internet Naming Convention [1] (ASCII).

Address -- The address is presented in binary form. In this writeup, double quotes, " ", are used around decimal values separated by a space to represent octets of the binary form.

Parsing of the address is implied by the specified transport protocol. In the case of TCP, the first four octets gives the 32-bit IP address, the 5th octet the IP-specific protocol number, and the 6th the TCP or UDP port number for the application service.

Comment -- The item is mostly optional. Its presence may allow an intermediate server passing comment to the end user. Error comments explaining resolution failure is an example of its use.

4.2 Command Specification

In this section, we define the name service commands for the various communication pairs.

4.2.1 Application Process/AIP Communication

From the name service point of view, there is no need for communication between the AIP and an application process at the destination. Thus, here we discuss communications at the originating domain.

An application process initiates a dialogue by making a request for name service to its local AIP. It provides the requested application service and a destination name for resolution.

REQUEST

Command Type Number of Items

Service Indicator Length Transport Protocol/Service/Service Type

Name Indicator Name Length Name String

Examples:

```
1  2
3 13 TCP/SMTP/mail
1 21 Postel@F.ISI.USC.ARPA

1  2
3 13 TCP/NIFTP/RFT
1 12 TSC.SRI.ARPA
```

The first example is a resolution request for the name "Postel@F.ISI.USC.ARPA". It is 21 octets in length. The requested application service is TCP/SMTP/mail. The second example is a resolution request for application service NIFTP at TSC.SRI.ARPA.

AFFIRMATIVE RESPONSE

Command Type Number of Items

Service Indicator Length Transport Protocol/Service/Service Type

Name Indicator Name Length Name String

Address Indicator Address Length Address

Examples:

```
2  3
3 13 TCP/SMTP/mail
1 21 Postel@F.ISI.USC.ARPA
2  6 "10  2  0 52  6 25"

2  4
3 13 TCP/NIFTP/RFT
1 12 TSC.SRI.ARPA
2  6 "10  3  0  2  6 47"
2  6 "39  0  0  5  6 47"
```

An affirmative response implies that the destination offers the requested service. The parsing of an address is implied by the indicated transport protocol. In the first example, the transport protocol is TCP. Thus, the address is composed of three fields: the internet address ("10 2 0 52"), the protocol number ("6" for TCP [3]), and the port number ("25" for SMTP [3]). A multiple-address response in the second example indicates that TSC is multi-homed via both ARPANET (net 10), and SRINET (net 39). A multiple-resolution response is preferred. It offers the source a choice.

NEGATIVE RESPONSE

Command Type Number of Items

Service Indicator Length Transport Protocol/Service/Service Type

Name Indicator Name Length Name String

Name Indicator Name Length Partial Name String

[Comment Indicator Comment Length Comment]

This indicates difficulty in resolution. Returned with this command is the left-most portion of the specified name including the difficulty encountered. An optional comment item may be included.

Examples:

```
3  4
3 13 TCP/SMTP/mail
1 16 Postel@F.ISI.USC
1 16 Postel@F.ISI.USC
9 18 Resolution Failure

3  4
3 13 TCP/NIFTP/RFT
1 13 TSC..SRI.ARPA
1  5 TSC..
9 17 Syntactic Anomaly
```

In the first example, the resolution failed because USC is not top-level domain. The syntactic error of adajacent dots in the second example is obvious.

INCOMPATIBLE SERVICE

This response indicates no compatible application and/or transport service is available at the destination. For example, the requested application service may be SMTP, while only FTP-mail is available at the destination. Return with this command is the available corresponding available service, if any, and its address. If no service is available for that service type, an empty string for service specification is returned.

```
Command Type   Number of Items

Service Indicator   Length   Transport Protocol/Service/Service Type

Name Indicator   Name Length   Name String

Service Indicator   Length   Transport Protocol/Service/Service Type

[Address Indicator   Address Length   Address]
```

Examples:

```
9  3
3 14 TCP/NIFTP/mail
1 21 Postel@F.ISI.USC.ARPA
3  0

9  5
3 13 TCP/NIFTP/RFT
1 12 TSC.SRI.ARPA
3 11 TCP/FTP/RFT
2  6 "10  3  0  2  6 21"
2  6 "39  0  0  5  6 21"
```

4.2.2 AIP/AIP Communication

Communication between the AIPs accomplishes the "what-can-you-do-for-me" negotiation. Examples in this section correspond to those of Section 4.2.1.

REQUEST

```
   Command Type   Number of Items

   Service Indicator   Length   Transport Protocol/Service/Service Type
```

Examples:

```
   1  1
   3 13 TCP/SMTP/mail

   1  1
   3 13 TCP/NIFTP/RFT
```

AFFIRMATIVE RESPONSE

```
   Command Type   Number of Items

   Service Indicator   Length   Transport Protocol/Service/Service Type

   Address Indicator   Address Length   Address
```

Examples:

```
   2  2
   3 13 TCP/SMTP/mail
   2  6 "10  2  0 52  6 25"

   2  3
   3 14 TCP/NIFTP/RFT
   2  6 "10  3  0  2  6 47"
   2  6 "39  0  0  5  6 47"
```

An affirmative response implies that the destination offers the same service as that of the originator. A multi-resolution response is possible. The parsing of an address is implied by the indicated transport protocol. In the second example, the transport protocol is TCP. Thus, the address is composed of three fields: the internet address (10 2 0 52), the protocol number (6 for TCP), and the port number (25 for SMTP). The returned address(es) is to be relayed to the originating application process.

INCOMPATIBLE SERVICE

Command Type Number of Items

Service Indicator Length Transport Protocol/Service/Service Type

Service Indicator Length Transport Protocol/Service/Service Type

Address Indicator Address Length Address

This response indicates no compatible application and/or transport service available serving the destination. For example, SMTP may be the requested application service, while only NIFTP-mail is available serving the destination. Return with this command is the available service of that type. If no service available for that service type, a empty text string is returned.

Examples:

```
9  2
3 14 TCP/NIFTP/mail
3  0

9  4
3 13 TCP/NIFTP/RFT
3 11 TCP/FTP/RFT
2  6 "10  3  0  2  6 21"
2  6 "39  0  0  5  6 21"
```

In the first example, the destination does not offer any kind of mail service. The second example indicates that there is no NIFTP, but FTP available for remote file transfer service at the destination.

4.2.3 AIP/DNS Communication

The source AIP presents its associated DNS with a fully qualified domain specification for resolution. The expected resolution result is the network address for the destination endpoint DNS. We assume no need for communication between the DNS and AIP at the destination.

REQUEST

Command Type Number of Items

Name Indicator Name Length Name String

Examples:

```
1  1
1 14 F.ISI.USC.ARPA

1  1
1 12 TSC.SRI.ARPA
```

AFFIRMATIVE RESPONSE

```
   Command Type   Number of Items

   Name Indicator   Name Length    Name String

   Service Indicator   Service Length   Transport Protocol

   Address Indicator   Address Length   Address
```

Examples:

```
   2  3
   1 14 F.ISI.USC.ARPA
   3  3 UDP
   2  6 "10  2  0 52 17 42"

   2  4
   1  7 TSC.SRI.ARPA
   3  3 UDP
   2  6 "10  3  0  2 17 42"
   2  6 "39  0  0  5 17 42"
```

An affirmative response returns an address of the destination endpoint DNS. This returned address is that of the destination DNS. The destination transport service needs to be indicated for guiding the parsing of the destination address.

NEGATIVE RESPONSE

```
   Command Type   Number of Items

   Name Indicator   Name Length   Name String

   Name Indicator   Name Length   Partial Name String

   [Comment Indicator  Comment Length   Comment]
```

This response indicates that the domain name service is unable to resolve the given destination domain name. It could be caused by an unknown simple name, which may result from, for example, misspelling. Returned with this command is the left-most portion of the specified name containing the cause of resolution failure.

Example:

```
     1  3
     1  9 F.ISI.USC
     1  9 F.ISI.USC
     9 18 Resolution Failure
```

4.2.4 DNS/DNS Communication

The domain name service is an application independent network service. It provides the resolution of domain names. For the specification of this service the reader is referred to [2].

4.3 Transport Protocol

For generality, this specification is intentionally transport protocol independent. Implications for the use of TCP and UDP are specifically considered.

Typically, for distributed name service a server A makes a request to a server B, server B may need to in turn contact other servers to complete a resolution. TCP is a connection-oriented protocol. It offers reliable transport, but also imposes certain amount of overhead for connection establishment and maintenance. For most cases, the use of TCP is not recommended.

UDP is a datagram service offering a transport capacity per datagram in excess of 500 octets. Such capacity should suffice most conceivable commands within this specification. However, it does impose a limit on the total length of a command. In order to enhance reliability, the request is incorporated as part of every response command.

5 NCP TO TCP TRANSITION

The Internet Naming Convention, "<user>@<domain>. ... . <domain>" [1], is a generalization of "<user>@<host>", the ARPANET Naming Convention. It is a generalization in the sense that the ARPANET Naming Convention can be considered as a partially qualified form of the subset "<user>@<host>.ARPANET". (We assume here ARPANET is a top-level domain name.)

For the transition from NCP to TCP, we may initially treat each host name entry in the current host table as a subdomain of the top-level domain ARPANET. Thus, initially there would be a very flat domain structure. This structure can be gradually changed after the transition toward a hierarchical structure when more and more domains and subdomains are defined and name servers installed. In the process of this change, the host table would be gradually converted into distributed domain tables (databases). For the newly created domain tables, no standard format would be required. Each individual domain table may have its own format suitable to the design of its associated domain name server.

REFERENCES

[1] Su, Z. and J. Postel, "The Domain Naming Convention for Internet User Applications," RFC 819, SRI International (August 1982).

[2] Postel, J., "Domains Name Server," RFC XXX, USC/Information Sciences Institute (to appear).

[3] Postel, J., "Assigned Numbers," RFC 790, USC/Information Sciences Institute (September 1981).

Appendix A

CONVENTION ASSIGNMENTS

Command Types

Request	1
Affirmative Response	2
Negative Response	3
Imcompatible Service	9

INDICATORS

Name Indicator	1
Address Indicator	2
Service Indicator	3
Comment Indicator	9

TRANSPORT PROTOCOLS: TCP, UDP, NCP

SERVICES

Service Protocols	Service Type
MTP	mail
SMTP	mail
FTP (FTP mail)	mail
NIFTP (NIFTP mail)	mail
MMDF	mail
FTP	RFT (remote file transfer)
Telnet	RTA (remote terminal access)