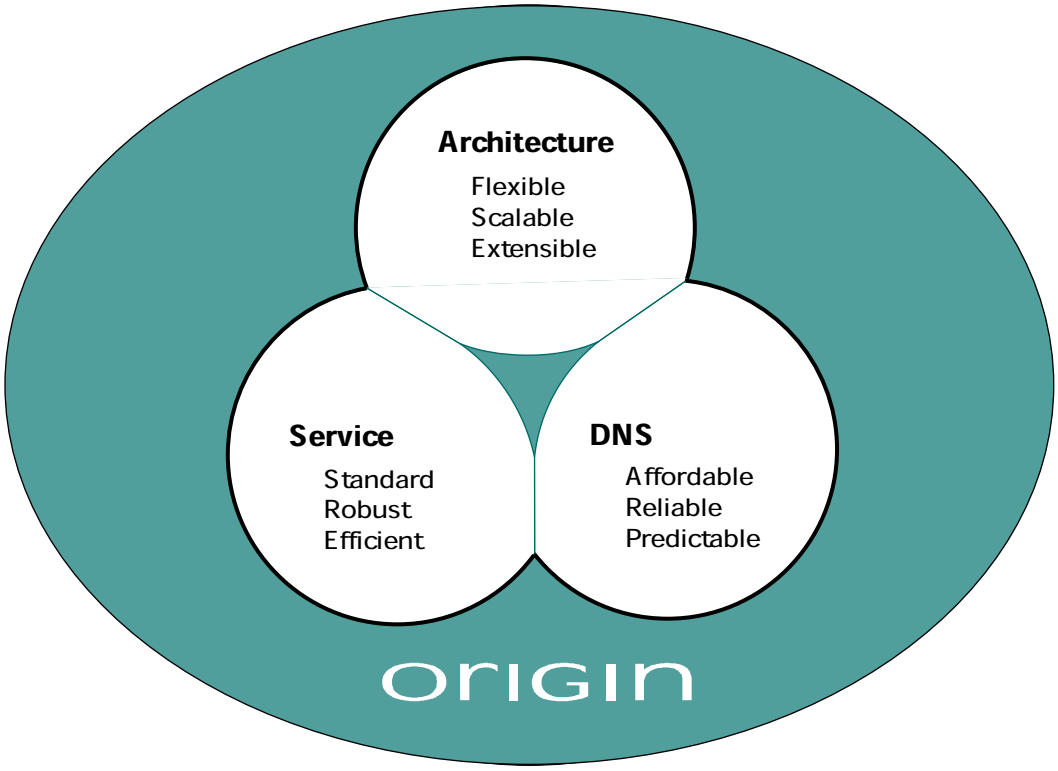
Domain Name System:
The Origin Solution



ORIGIN

Introduction 3
DNS: Some Background Information 3
IP Addresses 3
Descriptive Addresses 3
Mapping between IP Addresses and Domain Names 4
DNS usage 4
DNS: What Is Needed To Build DNS? 4
Contents 4
Servers 5
Resolvers 5
DNS: How Does DNS work? 5
Redundancy: Master & Slave Servers 6
Efficiency: Caching 6
From Technology To The DNS Service 6
Architecture: The Origin Technical DNS Architecture 7
The Server Architecture 8
The Resolver Architecture 9
The Contents Architecture and Management 10
Architecture: The Total Architecture 11
Connecting the Internal Network to the Internet 12
Dynamic Host Configuration Protocol (DHCP) 13
Service: Organizational Aspects 14
Customer Roles 14
Origin Roles 14
Procedures 14
Further Reading 15
Books 15
World-Wide Web sites 15
RFCs 15

SUMMARY

The Domain Name System (DNS) is one of the core enabling technologies in Internets and Intranets. It performs the key role of mapping between network addresses and descriptive names.

Because DNS is generally considered to be difficult for non-specialists to implement and maintain, Origin has developed a flexible, integrated, managed DNS Service, whereby customers can retain differing levels of responsibility for the DNS backbone and the DNS content, and delegate the remainder to Origin.

DNS is not an end in itself – just a requirement.

Jerry Scharf: Vixie Enterprises

Introduction

Of all the Internet services, the Domain Name System (DNS) is one of the least well understood, while at the same time being one of the most important. The DNS affects every other Internet service, from e-mail, to the Web, to audio conferencing. A good understanding of DNS is crucial to understanding the implications of the Origin solution.

This whitepaper describes: the general principles of DNS, the Origin solution to implementing a corporate DNS, and the organizational aspects of the Origin solution.

The intended audience is very wide and includes both Origin's employees and its customers.

DNS: Some Background Information

Before we can say what DNS does, we must know what problem it was designed to solve, and before we can understand that, we must know a little about addressing on the Internet.

IP Addresses

Each resource (computer, printer, router, and so on) on the Internet or an Intranet has a globally unique numeric address called an *IP address* (called 'IP' after the Internet Protocol). It is these numbers that are used by the underlying network for communication between resources. The IP addresses are conventionally written as aaa.bbb.ccc.ddd, for example, the Origin external web site has address 192.68.44.134.

It is generally easier for people to refer to descriptive names rather than IP addresses. Descriptive names are usually easier to remember, and usually convey more information, such as the type of the resource, for example, web server, mail machine, printer; the owner of the resource; and the location of the resource. For example, 192.68.44.134 tells us very little, whereas www.origin-it.com tells us that it is most probably a web server, that Origin IT is an important factor, and that it is run by a commercial enterprise.

Descriptive Names

So how are these descriptive names constructed? All descriptive names exist in a notional *domain space*, which can best be likened to a tree (it is traditional to visualize such trees with their root at the top). The root of the domain space is considered to be to the right of com in www.origin-it.com: com is a *subdomain* of the root, along with other world-wide subdomains, such as, edu, org and net, and the national subdomains, such as, uk and se.

To the left of com we see origin-it, and this is a subdomain within the com subdomain (dots are used as separators). At this level company names and organizational names often appear, such as, mit.edu, philips.com, and ietf.org.

Working leftwards through the descriptive name, we come across subdomains of the subdomain to the right – every domain is actually a subdomain, with the exception of the root domain. All subdomains within a subdomain must be unique – for example, there can only be one origin-it subdomain within the com subdomain. Henceforth we will refer to *descriptive names* as *domain names*.

Mapping between IP Addresses and Domain Names

In the early days, the mapping between IP addresses and domain names was kept in a file which was regularly replicated on all systems that wanted to use it. As the number of systems grew, the frequency with which the file needed to be updated grew, the size of the file grew, and the number of systems it had to be sent to grew. It was clear that a better, scaleable, solution was needed. This is the problem DNS was created to solve.

The key thinking behind DNS is: 1) if the mapping information is too large to be sent to the systems, then let the systems look up what they need from where it is held; 2) don't keep all the information in one massive database in one place, but hold the information in a distributed form, and; 3) develop a scaleable model which will be able to accommodate the projected Internet/Intranet growth.

DNS usage

The main use of the DNS system is to translate domain names, such as, www.origin-it.com into IP addresses. These are called *forward lookups*. DNS is also used the other way around. For example, web masters can log the IP addresses of the computers from which their site has been visited, and using DNS map the IP addresses back to domain names. This is done by *reverse lookups*, that is, passing an IP address to a DNS server, and asking for the associated domain name.

The DNS is also used for routing e-mail between machines on the net. When you send mail to, for example, tis-intranet@nl.origin-it.com, the mail actually gets delivered to a machine called mail.nl.origin-it.com, and if mail.nl.origin-it.com is down, then a backup machine called mail.de.origin-it.com will receive the mail. It is the DNS which contains such mail routing information.

One other important function of DNS is that it allows changes to be made to the infrastructure without interrupting the services running on top of the infrastructure. For example, if a web server is to be upgraded, a new machine can be put in place, the DNS changed, and the old web server removed. Visitors continue to access the server by its domain name, but reach a different machine, and there is no service disruption.

DNS: What Is Needed To Build DNS?

Three components are needed to build a Domain Name System:

- **contents**: the information stored in the system
- **servers**: to store the information
- **resolvers**: to request information from servers.

To make a DNS service from a DNS system a fourth component, namely **management**, is required.

Contents

We have seen that domain names have a tree-like, or hierarchical, structure. This hierarchical structure of domain names serves several purposes. First, it gives humans a rough guide to the type, and maybe location, of the organization a name is associated

with. Second, it gives the ability for automated navigation through the entire DNS information. Third, it makes it simple to delegate responsibility for a branch of the tree to a particular authority.

The content consists of a number of types of *resource record*. Resource records include: *MX* records defining where mail should be delivered; *A* records mapping resource names to IP addresses; *CNAME* records mapping aliases to 'real' resource names; and more.

The DNS content in its entirety constitutes the DNS database.

Servers

A DNS server contains information about some part of the DNS database. For instance, the Origin DNS server contains all IP addresses and mail routing information for all machines which have origin-it.com as the last part of their name.

Resolvers

All machines, PCs, Macs, UNIX, whatever, that make use of the Internet include a resolver. The resolver's job is to send queries to a DNS server whenever an application, for example, a web browser, wants information from the DNS. A PC on a network must be configured with the IP address of the DNS server it is to use.

DNS: How Does DNS work?

Each DNS server maintains information about a part of the DNS database; a number of DNS servers, called *root servers*, maintain information about the top of the DNS hierarchy. The root servers' information about origin-it.com, for example, only stretches as far as knowing which servers maintain information about the com domain, and these servers in turn maintain information about servers that have information about IP addresses in origin-it.com domain.

The root servers each have an entry in their tables that essentially says, "for information about origin-it.com, ask a server that knows about the com domain". The servers that know about the com domain have an entry that essentially says "for information about origin-it.com, ask the machine with IP address 193.79.128.34".

Assume a user somewhere outside Origin wants to access web pages on the web server www.origin-it.com. How does his PC connect to the correct IP address?

- First of all the resolver in the PC sends a query to the local DNS server (assuming there is one) asking for the IP address of www.origin-it.com.
- The local DNS server doesn't know the answer, (of course it could know the answer, in which case that is the end of the story) so it sends a query to one of the root DNS servers asking for the IP address for www.origin-it.com.
- The root DNS server sends back a reply saying that it doesn't know the answer, but it knows a server that knows about the com domain, and it returns its IP address.
- The local DNS server then repeats the previous query, but this time sends it to the address of the com server it has just been given by the root server.

- That server sends back a reply saying that it doesn't know the answer, but it knows a DNS server that does, namely 193.79.128.34 (the Origin DNS server).
- The local DNS server then sends the identical query to the server with address 193.79.128.34.
- The machine at 193.79.128.34 (the Origin DNS server) knows the information, so it sends back a reply
- The user's local DNS server finally sends back a reply to the PC, and the PC contacts www.origin-it.com using the IP address returned.

It can be seen that two kinds of "hard coded" linkage in the DNS system are needed. Individual PCs need to have the IP address of the DNS servers that will resolve any type of query for them, often a local DNS server; and when the DNS servers are started they need to have the IP addresses of the DNS root servers.

Redundancy: Master & Slave Servers

Since DNS servers are so important, it is common sense to make sure that information is always available. The DNS does this by replicating information contained in one server on other servers.

There is always one, and only one, computer which contains the master set of DNS information for a particular domain name. This computer is called the *master name server* for the domain. A *slave* is a DNS server which periodically downloads a copy of the master's domain table. Slaves do this by querying the master regularly to see if the master's information has changed. If it has, the slave simply downloads the entire table again from the master. This downloading is called a *zone transfer*.

Efficiency: Caching

All DNS servers remember recently issued queries and their replies. If a server is asked again for the same domain name, it can return the reply immediately from its cache, without having to send out a query. Along with the caching mechanism, there is a series of parameters which tell the servers how long they are permitted to cache particular domain names.

From Technology To The DNS Service

The DNS technology is now well established on the Internet. Deploying DNS on an Intranet brings with it its own particular problems:

- **DNS is difficult to configure and maintain**. The distributed capability makes DNS difficult to configure. Resource records have arcane formats for non-specialists. For example, omitting a dot at the end of a name can result in the complete disappearance of a domain from the DNS. Furthermore, the distributed capability and the caching facility can make it difficult to track down DNS problems.
- **Intranets tend to grow rapidly.** The facilities offered by an Intranet are being exploited all over the organization. This leads to a rapid growth of the TCP/IP network and, as a consequence, of the DNS. Unfortunately, training and education of support personnel does not keep pace with the deployment of technology. Support

personnel, therefore, rarely have any knowledge about DNS. This can lead to persistent connectivity problems because of DNS misconfigurations.

- **DNS is not the only mechanism to map names to numbers**. For example, hosts files, NIS, and WINS are also used. As a consequence, redundant administrations need to be maintained. Another consequence is that most operating systems try to support some, or all, possible technologies for mapping names to numbers. This results in complex system configurations, which can give rise to unreliable resolvers and servers on many operating systems, and can require the installation of patches which, in turn, may affect other services.

- **DNS technology is continually being extended**. The Internet Engineering Task Force has several active working groups which produce additions to the DNS protocol. For example, work is being done on the efficiency of the protocol (notifies and incremental zone transfers), and on Secure DNS, which allows for strong authentication of DNS information. Initiatives for linking DHCP to DNS will result in extensions to the protocol in the near future.

  The software almost universally used for DNS is `BIND` from the Internet Software Consortium. This product is regularly updated to take advantage of the latest Internet and Intranet developments.

Architecture: The Origin Technical DNS Architecture

The Origin DNS architecture is built from three main parts:

- **The server architecture:** where the servers are placed in the network, and their function.

- **The resolver architecture:** how the resolvers (DNS client programs) connect to the servers and to which servers they must connect.

- **The contents architecture:** the way in which the information is stored and updated in the DNS database.

The goals of Origin's DNS architecture are:

Flexibility The entire system must be adaptable to customer requirements, including the need of some customers to have complete control over their local DNS servers, and of others to have as much of the process as possible controlled by the central network administration.

Scalability As the network grows, perhaps by orders of magnitude, it should not be necessary to change the architecture. It should be possible to add servers and client machines with minimal changes.

Extensibility The architecture must be able to accommodate new protocols and mechanisms introduced as the result of technical developments, without fundamental design changes or large-scale replacement of existing hardware or software.

Simplicity The entire system must require a minimum of skilled staff to maintain and expand it, and should not require special skills at the local level. DNS servers must be simple to install and to update. Simplifying the process of configuring and maintaining DNS is a crucial goal.

Reliability Failure of any single component must not stop the system as a whole from working, although some degradation in performance may be tolerable in the event of hardware or network failures.

Efficiency The system must enable domain name lookups to take place with minimum delay, and minimize the use of network bandwidth.

Elegance Details of the inner working of the system should be hidden from the LAN administrator.

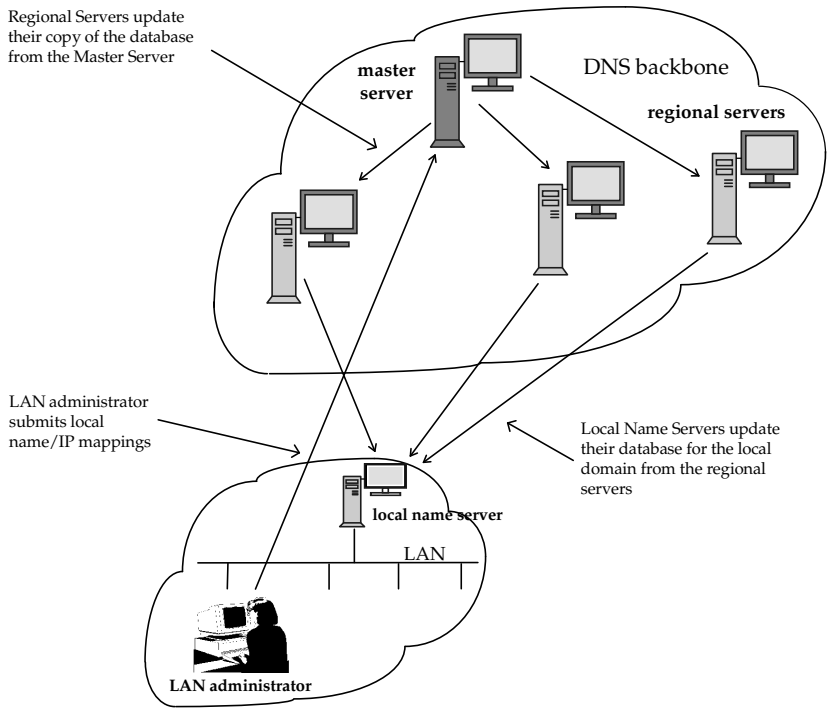
Predictability Changes to the local DNS database must be reflected throughout the system in a guaranteed time. LAN administrators must be informed whether or not their changes have been successfully incorporated.

Cost reduction The cost of hardware and of the creation and installation of the software must be moderate. Domains of arbitrary size must be created and maintained by a small number of experts in a central support team. This team must manage the underlying technology while the information base is to be managed by LAN administrators.

The Server Architecture

Origin's DNS service is based on a DNS backbone with name servers distributed globally. One of these servers is designated as a master server. The servers have almost identical configurations and hold DNS information for the entire organization, including all local domains.

The backbone servers allow any DNS server on the Intranet to take copies of DNS information, provided the server is appropriately connected in the server architecture.



In addition to the regional name servers, there are local (LAN) name servers that provide DNS services for local domains. The way in which the databases on these local name servers is updated depends on the type of domain that each serves. There are three types, each of which is described in the section *The Contents Architecture and Management*.

The architecture does not require a local name server, but one is highly recommended. A local name server brings with it the following advantages:

- Resolving of local names can continue even when there is a problem with the WAN, although addresses that are not known to the local name servers cannot be retrieved if the WAN is not available.
- Popular names need not be fetched from the remote server every time they are needed, thus ensuring the fastest possible name resolution, and minimizing the bandwidth used on the WAN. This is a result of DNS requests being cached on the local name servers: names fetched by the local name server, whether local or not, are temporarily cached and used for answering subsequent requests.

The Resolver Architecture

When an application requires an IP address, it passes the domain name of the host it needs to contact to a resolver. The resolver is a client for a DNS server, and is usually built in the application, for example, in a web browser. The resolver sends a query to a name server in order to obtain an answer. The resolver must be configured with the IP address of at least one (and preferably more than one) DNS server to contact.

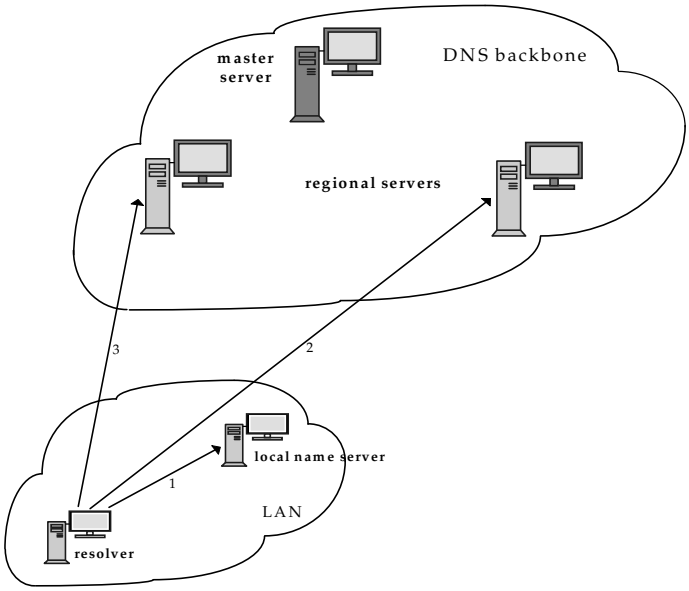
Resolvers can also obtain domain names for IP addresses, and certain other information stored in the DNS database, including machines for routing e-mail using SMTP.

If the resolver is configured to use more than one DNS server, it tries each server in turn. If the first server listed does not respond, the resolver contacts the next server on the list, and so on until a response is obtained.

Resolvers should be configured to use DNS servers in the following order:

1. The local name servers (ensuring that requests are handled locally when possible).
2. The "closest" DNS server on the DNS backbone.
3. Other DNS servers on the DNS backbone.

Using a local name server in this way increases overall efficiency because of caching, and adds robustness – it can still answer some queries even if the connection to the WAN is interrupted.



The Contents Architecture and Management

A domain name is part of the hierarchical naming scheme used for hosts on Intranets and on the Internet. A domain comprises a specific level of the hierarchy referred to by a domain name, and all the subdomains below that point. In practice, domains are assigned to meet the needs of customers, and can represent sites, offices, or departments within an organization.

The following three types of domain, each with different management approaches, can be supported by the Origin DNS backbone.

Centrally managed domains

In a centrally managed domain, both the backbone infrastructure and the LAN domain content is managed on behalf of the customer by Origin. The customer specifies the content: Origin performs the implementation, under strict change control. The domain origin-it.com is an example of a centrally managed domain.

The master set of DNS information for all centrally managed domains is on the master server. The other backbone servers take their DNS information from the master server.

Where, for whatever reason, it is inappropriate for a local LAN administrator to take responsibility for a LAN domain's content, Origin can assume that responsibility, and manage it centrally.

Partially delegated domains

In a partially delegated domain, the backbone infrastructure is managed by Origin, and the LAN domain content is managed by a local LAN administrator. (Partially delegated LAN domains are usually bound to unique IP ranges.) Most LAN domains on an Intranet are expected to be partially delegated domains.

The advantages of partially delegated domains are that administrative costs are kept to a minimum and that LAN administrator tasks are kept simple. In order to minimize the effort and knowledge required by the LAN administrators, whose main responsibility is not DNS, the process of updating the DNS database has been automated. This currently involves central processing software and a client program called `dnssend`. The client can be called from the command line in UNIX and Windows NT.

Partially delegated domains also offer flexibility – if all computers on a network are to be transferred to a new domain (that is, the domain is to be migrated), the new domain can be created alongside the old. The old domain names can then be aliased to the new in one step. This is carried out smoothly by the central processing software.

Each LAN administrator is required to maintain a LAN resources file containing names and IP addresses of all resources in the domain. When changes are made at the LAN level, the administrator updates this file, and uses `dnssend` to send a copy of the file to the central processing software.

The central processing software checks the newly arrived file for errors. Errors that can be detected include: IP numbers outside the IP range for the LAN; invalid names (RFC 1912 defines valid names); and duplicate names and IP addresses.

If errors are found the entire file is rejected, and the LAN administrator is notified by e-mail. After the errors have been rectified, the administrator can resubmit the file.

However, errors that assign valid names to the wrong LAN IP addresses cannot be detected by the central processing software. The LAN administrator is responsible for making sure that such errors do not occur. The consequence of this type of error could be that computers on the LAN concerned can no longer access some or all network services.

When the data has been checked and found to be correct, the central processing software sends the changes to all the backbone DNS servers (both master and slaves) and to the appropriate local name servers.

The master set of DNS information for all partially delegated domains is normally on the master server. The other backbone servers take their DNS information for partially delegated domains from the master server. The Origin solution includes tools which guarantee that the data required for forward lookups and reverse lookups is kept synchronized.

Delegated domains

In a delegated domain, the local LAN administrators assume full responsibility for all DNS information. (Delegated domains are bound to unique IP ranges.) Origin organizes the proper delegation of the domain, and ensures that there is a copy of the domain data on the DNS backbone, but does not take any responsibility for the content of the domain.

Updates to the DNS database are carried out locally, following locally determined change management procedures. Each LAN administrator has total responsibility for ensuring that the DNS configuration is correct for his domain.

The master set of DNS information for a delegated domain is on a local name server (the master name server for that domain): not on the DNS backbone. The DNS backbone servers pull a copy of the data from this local server. This is done with zone transfers. All DNS backbone servers act as slave servers for the delegated domains. Origin delegates the forward and reverse domains to the site master server.

Architecture: The Total Architecture

Having seen the three parts of the DNS service architecture, we can now see how the goals set out in the section *Architecture: The Origin Technical DNS* Architecture have been met.

Flexibility Customers are offered several levels of service; using a centrally managed domain, a partially delegated domain, or a fully delegated domain. This choice may be based on the customer's expertise in this field, on cost, or on availability of hardware, software or personnel.

Scalability Scaling is inherent in the delocalized, hierarchical design of DNS. In this specific service, new backbone and LAN name servers can be added at any time to meet the requirements of expansion. The backbone servers are identical and can be easily cloned.

Extensibility Extensibility is achieved by using the Internet standard DNS software, `BIND`, for implementation of the service. This is updated regularly to take advantage of new developments in Intranet and Internet services, and all existing DNS protocols and services are supported.

Simplicity Simplicity is achieved because the management of all the backbone servers is identical. Also, the update of both backbone and LAN name servers is automated as much as possible. This minimizes the effort and skill required from both LAN and central administrators.

Reliability Reliability comes from the distribution of accurate information across servers: if one server is down, another can usually serve the client. Local database updates for partially delegated domains are automatically checked for logical errors. Resolving can continue even when there is a problem with the WAN, although DNS information that is not on the LAN name server cannot be fetched.

Efficiency The system of LAN name servers and regional backbone servers, with caching of frequently requested data, ensures that DNS queries are answered locally whenever possible, and minimizes use of the WAN.

Elegance All LAN administrator interaction with the central processing software is carried out through the `dnssend` software, with a simple interface.

Predictability LAN administrators are informed within a specified time by e-mail whether their changes have been successful or whether they have failed.

Cost reduction For the partially delegated domains the information is managed entirely by LAN administrators, with automated checking by the central processing software. The simplicity of the system as a whole means that central administration does not need to increase to match the expansion of the network, and LAN administrators do not need special training.

Connecting the Internal Network to the Internet

A *split DNS* is used for connection between the Intranet and the Internet. In a split DNS, no external names can be resolved in the internal DNS service, and no internal names can be resolved in the external DNS. External names are obtained by *proxy servers* on the boundary between the internal and external networks.

The proxy servers act as clients for DNS servers on the Internet. They obtain external DNS data in the normal way, and then provide the data to the internal client that has requested it. The internal client may in turn be a proxy server for some other Internet service, for example, the World-Wide Web (HTTP).

Split DNS is needed for a number of reasons.

Availability: Use of external DNS means dependence on Internet root server access. If no Internet root servers are available (for example, because the Internet link is down) an internal computer might not be able to connect to some or all internal services.

Confidentiality: The internal name space is not visible to external entities on the Internet nor on attached Intranets. This ensures company confidentiality. The name space presented on external networks can be controlled to reflect corporate security and access policies. For instance, Internet users see only those resources that the company network allows them to see.

Service levels: Separation of the internal and external name space means that Origin can take responsibility for the operation of the internal DNS, and guarantee service levels.

Security: DNS has some security risks. A malicious intruder might be able to compromise a DNS server from an external network and change DNS data to point a well-known service name at another IP address. That IP address could then be used,

for example, to collect e-mail. Similarly, a split DNS protects against recurring errors and attacks on DNS servers used elsewhere on external networks.

Monitoring: Access between the internal network and external networks, including the Internet, can generally be controlled and monitored.

A similar model can be used for connecting the Intranet and third-party networks.

Dynamic Host Configuration Protocol (DHCP)

DHCP is a protocol that allows a host machine to get its operating parameters, which can include IP address, subnet mask, host name, proxy server addresses and resolver configuration, from a DHCP server. The use of DHCP avoids the need for each client machine to be configured by hand. Because IP addresses and host names are involved, DHCP has an impact on DNS.

There is no existing standard for integrating DHCP with DNS, although a draft exists (RFC 2136). There are security risks associated with the draft standard unless it is combined with additional measures.

Origin recommends not following any of the insecure or proprietary routes to DNS/DHCP integration, but rather following current best practice until standards are defined and products are available.

The current options for using DHCP with DNS are outlined below. The method chosen depends on how the DHCP clients are managed:

- dynamic allocation: that is, clients are assigned any address from the DHCP pool. For example, if three machines were to be given addresses from the range 172.16.8.9 to 172.16.8.11 then the DNS entries would be:

  ```
  172.16.8.9    dhcppc-9.domain-name
  172.16.8.10   dhcppc-10.domain-name
  172.16.8.11   dhcppc-11.domain-name
  ```

  where *domain-name* is the domain of the hosts.

  This ensures that there is an entry in the DNS for each client, although it is not possible to tell from the DNS which machine has a particular address: this is under the control of DHCP. The domain name can be used as a form of authentication, but the host names are of no use for that purpose.

- automatic allocation: that is, each client is always assigned a fixed IP address by the DHCP server. For example, if there are three clients, tom, dick and harry, that always have IP addresses 172.16.5.6, 172.16.5.7, 172.16.5.8 respectively, then the DNS entries would be:

  ```
  172.16.5.6    tom.domain-name
  172.16.5.7    dick.domain-name
  172.16.5.8    harry.domain-name
  ```

  The DNS has an entry for the IP address and the permanent name of each client. This DHCP mechanism is, in fact, the same as the BOOTP protocol from the naming and addressing point of view. Other DHCP functionality like handing out subnet masks and default routes is still possible. Both the domain names and the host names can be used as a form of authentication.

Once the DHCP entries have been added to the LAN administrator's resource file, the file can be submitted for processing and inclusion in the database, in the usual way.

Both Origin and the customer have roles and responsibilities defined within the service.

Customer Roles

The customer must ensure that appropriate persons are assigned for the following:

- **Ownership of the naming standard:** the authority to define and control the naming standard.
- **Ownership of the centrally managed domains:** the authority to request changes to corporate domains.
- **Connecting domains:** the authority to connect and disconnect domains. This authority can also change the DNS data for those domains.
- **Responsibility for domain contents:** the responsibility for initial installation of the Origin tool and the on-going management of the DNS contents of the domain. This role can be performed by the LAN administrator.

Origin Roles

- **First-line support:** including first contact in case of problems, and simple problem solving. Primary operational contact for subscription and removal of DNS domains.
- **Second-line support:** including change implementation and problem resolution.
- **Third-line support:** including overall service architecture, delivery and consultancy.

Procedures

The customer and Origin work together according to predefined procedures. These must be formalized for:

- Connecting a domain.
- Disconnecting a domain.
- Changing centrally managed domains.
- Changing partially delegated domains.
- Changing the naming standard.

Further Reading

Books

TCP/IP Network Administration Craig Hunt (1994)	O'Reilly & Associates ISBN: 0-937175-82-X
TCP/IP Illustrated, Volume 1 W. Richard Stevens (1994)	Addison-Wesley Publishing Company ISBN: 0-201-63346-9
DNS and BIND, Second Edition Paul Albitz & Cricket Liu (1997)	O'Reilly & Associates ISBN: 1-56592-236-0
Domain Name System (DNS): Guides for local site administrators Origin (1997)	Origin BID-NBS-1003 and 1004

World-Wide Web sites

Internet Engineering Task Force	http://www.ietf.org
DNS Resources Directory	http://www.dns.net/dnsrd/
Internet Software Consortium (software and documentation for Berkeley Internet Name Daemon (BIND) and DHCP)	http://www.isc.org/bind.html
Internet Society	http://www.isoc.org
Origin TIS Intranet Services	http://pww.origin.philips.com/tis-ins/services/dns1.htm http://channels.origin-it.com/tis-ins/services/dns1.htm

RFCs

RFC920	Domain Requirements	Oct 1984
RFC1032	Domain Administrators Guide	Nov 1987
RFC1033	Domain Administrators Operations Guide	Nov 1987
RFC1034	Domain Names - Concepts and Facilities	Nov 1987
RFC1035	Domain names - Implementation and Specification	Nov 1987
RFC1101	DNS Encoding of Network Names and Other Types	Apr 1989
RFC1591	Domain Name System Structure and Delegation	Mar 1994
RFC2131	Dynamic Host Configuration Protocol	Mar 1997
RFC2136	Dynamic Updates in the Domain Name System (DNS UPDATE)	Apr 1997

Origin BV
Origin TIS Intranet Services
Eindhoven
The Netherlands

tis-intranet@nl.origin-it.com

April 1998 Printed in The Netherlands © Origin BV 1998 Document Number: BID-NBS-0054

origin