The genesis of the DNS was the observation, circa 1982, that the HOSTS.TXT system for publishing the mapping between host names and addresses was encountering or headed for problems.

The base design assumptions for the DNS were that it must:

- Provide at least all of the same information as HOSTS.TXT.
- Allow the database to be maintained in a distributed manner.
- Have no obvious size limits for names, name components, data associated with a name, etc.
- Interoperate across the DARPA Internet and in as many other environments as possible.
- Provide tolerable performance.
- The cost of implementing the system could only be justified if it provided extensible services. In particular, the system should be independent of network topology, and capable of encapsulating other name spaces.
- In order to be universally acceptable, the system should avoid trying to force a single OS, architecture, or organizational style onto its users.

The active components of the DNS are of two major types: name servers and resolvers. Name servers are repositories of information, and answer queries using whatever information they possess. Resolvers interface to client programs, and embody the algorithms necessary to find a name server that has the information sought by the client.

The DNS internal name space is a variable-depth tree where each node in the tree has an associated label. The domain name of a node is the concatenation of all labels on the path from the node to the root of the tree.

Data for each name in the DNS is organized as a set of resource records (RRs); each RR carries a well-known type and class field, followed by applications data. Types are meant to represent abstract resources or functions, for example, host addresses and mailboxes. The class field is meant to divide the database orthogonally from type, and specifies the protocol family or instance.

The DNS provides two major mechanisms for transferring data from its ultimate source to ultimate destination: zones and caching. Zones are sections of the system-wide database which are controlled by a specific organization. The organization controlling a zone is responsible for distributing current copies of the zones to multiple servers which make the zones available to clients throughout the Internet. Caching is a mechanism whereby data

---

1 HOSTS.TXT is the name of a simple text file, which is centrally maintained on a host at the SRI Network Information Center (SRI-NIC) and distributed to all hosts in the Internet via direct and indirect file transfers.
acquired in response to a client’s request can be locally stored against future requests by the same or other client.

Operation of the DNS has revealed several issues that came as surprises to the developers.

- Refinement of semantics: the form and content of the information of DNS was well understood. That was a bad assumption.
- Performance: The performance of the underlying network was much worse than the original design expected.
- Negative caching: any naming system that relies on caching for performance may need caching for negative results as well.

There were also successes.

- Variable-depth hierarchy: is used a great deal and was the right choice
- Organizational structuring of names: the principle that names are independent of network, topology, etc. is quite popular
- Datagram access: The use of datagrams as the preferred method for accessing name servers was successful and probably was essential, given the unexpectedly bad performance of the DARPA Internet.
- Additional section processing: When a name server answers a query, in addition to whatever information it uses to answer the question, it is free to include in the response any other information it sees fit, as long as the data fits in a single datagram.
- Caching: The caching discipline of the DNS works well, and given the unexpectedly bad performance of the Internet, was essential to the success of the system.
- Mail address cooperation: Agreement between representatives of the major Internet communities led to an agreement to use organizationally structured domain names for mail addressing and routing.

Some of the shortcomings of the DNS include:

- Type and class growth: A methodology or guidelines to aid in the design of new types of information is needed.
- Easy upgrading of applications: Converting network applications to use the DNS is not a simple task. Part of the problem is transient failure. Another part of the problem is that access to the naming system needs to be integrated into the operating system to a much greater degree.
- Distribution of control vs. distribution of expertise or responsibility: Distributing authority for a database does not distribute a corresponding amount of expertise. Systems designers should anticipate this, and try to compensate by technical means.