The Domain Name Service, or DNS, was not conceived as an integral part of the fundamental routing mechanism of the Internet. It was designed to be a means to allow Internet hosts to be referred to using names that are meaningful to humans and can be remembered. The basic service provided by the DNS is to bind a domain name to a record consisting of a collection of records which have names and types determined by the DNS service. The most commonly used field is the A (address) field, whose value is an IP address. A single domain name can be bound to more than one record of each type.

The names that are bound by the DNS are lists of identifiers which are written using a period (".") as a delimiter. This format is used in the implementation of the DNS as a hierarchically organized collection of “domains” each of which defines a collection of “subdomains”. The DNS service is essentially a distributed, asynchronous database of records which are arranged into these hierarchically related domains.

The routing mechanisms of the Internet work in terms of IP addresses, not domain names. Thus a domain name can be thought of as an ease-of-use feature for human users. If every domain name corresponded to a single IP address and this never changed, then there would be no difference in functionality to using a domain name and using the IP address to which it was bound.

Of course, the generality of the domain name service meant that it could immediately be used to service other purposes. The fact that a domain can be bound to more than one address means that a single name can be used to denote all of the network interfaces that are attached to a single node. This means that a server can be connected to more than one network and traffic to it can be spread across those networks. Note, however, that the application layer is supposed to have no knowledge of network topology, and the DNS is itself implemented at the application layer. There is thus no way for a client to make an informed choice among the available addresses bound to a single domain name. The assumption is that no significance can be assigned to the order in which IP addresses are returned, and clients may use only the first, use them in order (if the first one fails), use them in random order or choose in any other way to balance the load of multiple requests globally.

Another form of generality is that records other than A records can be associated with domain names. For example another domain name can associated with a name through an MX record, which defines a mail server to which mail directed to the domain should be directed. This provides indirection, so that a domain name can be the target of mail even if it is not bound to a node that implements a mail server. Thus an email address mbeck@utk.edu is directed to the directed to the mail server returned by the MX (mail exchange) record bound to the domain utk.edu. This means that a domain that has no A records can still be the target of email
messages. The MX record was later generalized to an S (server) record that can bind multiple arbitrary services to a domain.

The DNS is implemented hierarchically, with each subdomain of a given domain potentially assigned to a different server. A name is resolved to an IP address by finding the DNS server associated with the highest level domain (the one written furthest to the right in standard order) and proceeding right to left, looking up an NS (name server) record for each name in the list until the last name, which associated with an A record. If a domain is being use for mail or some other service, then it is associated with the appropriate record type (MX or S) and the domain bound to that record is then resolved to an IP address (starting at the beginning).

The hierarchical nature of domain name resolution is mirrored on the client side by hierarchical caching servers that save the results of previous resolution requests in order to cut down on repetitive resolution requests. Each A record returned is stored by each caching server in the path to the client for a time associated with the record. This means that the administrator of a domain can determined how long a binding will linger in caches and thus how long it takes to change the binding of a name to an address.

All of this generality and control by the domain server binding an A address means that DNS resolution can be used in ways that may not be as simple as was originally contemplated. For instance if a particular server is specified by a domain name then it can be implemented in replica at different IP addresses. If a service bound to the domain name service.utk.edu has a prefered IP address ip1 and a backup address ip2 then rather than advertise both ip1 abnd ip2 in A records and letting the client choose which one to use, the administrator could decide to bind only ip1 as long as that server was available and to change the binding to ip2 in case ip1 failed. This would mean that for the period of time that ip1 was cached clients would not be able to reach the service at all, which would be unfortunate. To avoid this, the administrator could set the period of time that a cache could hold ip1 to be very short, or even set it to zero. Setting the cache time to zero would maximize the number of redundant translations of service.utk.edu in order to eliminate the period of unavailability. In exchange for fielding redundant requests, the domain administrator can exercise a great deal of control over the resolution of a particular domain to an IP address.

Later in the course we will discuss the problem of how high traffic servers (such as those that implement popular commercial Web sites) are replicated and requests to those domains are controlled. The DNS has become a key tool in making such resolution decisions in ways that are informed by information about network topology. As you might imagine, this requires breaking the rule that an application level DNS server cannot know the details of network topology. As we will see, in this case the simple DNS service has morphed into an integral part of the service replication and resolution framework that defines the modern Internet, essentially becoming a tool to manage IP routing in these cases.