Motivation

- IP addresses are great for computers but a problem for humans.
- In the 1950s, it was discovered that most people can remember about 7 digits per number.
- IP addresses contain as many as 12 digits and are hard for most humans to remember.
- To deal with this problem, Internet designers developed the Domain Name System (DNS).

Structure of DNS Names

- Each DNS name consists of alphanumeric components separated by periods.
- Example: **bacchus.cs.gettysburg.edu**
- Names are hierarchical with the most significant component on the right.
- The left-most component is the name of an individual computer.
- The right-most components, called top-level domains are defined by a global authority.

Structure of DNS Names

- An organization applies for a name in a top-level domain. Examples:

  gettysburg.edu **ford.com**

- An organization determines its own internal structure. Examples:

  cs.gettysburg.edu **math.stanford.edu**

- Top-level domains are “US-centric.” Other countries top-level domains end in a country code. Examples:

  .ac.uk **.edu.au**

  www.canterbury.ac.nz

Domain Names in an Organization

- Once an organization has a top-level domain, it can create any internal DNS hierarchy it wants.
- The uniqueness of the top-level domain and an organization name guarantee uniqueness of any internal name.
- domain - all but the left-most component of a domain name.

  Examples:

Domain Names in an Organization

Name	Domain
www.gettysburg.edu	**gettysburg.edu**
bacchus.cs.gettysburg.edu	**cs.gettysburg.edu**

- Authority for creating new subdomains is delegated to each domain. For example, the administrator of **gettysburg.edu** has authority to create **cs.gettysburg.edu**.

DNS Names & Physical Location

- DNS names are logical concepts and need not correspond to the actual physical location of an organization.
- The DNS domain for an organization can span multiple networks.
  - **gettysburg.edu** covers all networks at Gettyburg College.
  - **cs.gettysburg.edu** is on the second floor of Glatfelter Hall.
  - **laptop.cs.gettysburg.edu** could be connected to a network in Oregon.

DNS Clients & Servers

- DNS names are managed by a hierarchy of DNS servers.
- When an application needs to translate a name to an IP address, the application becomes a client of the naming system and sends a request to a DNS server.
- DNS servers are arranged in a hierarchy that matches the naming hierarchy.
- No single DNS server knows all DNS names.

DNS Clients & Servers

- The root server at the top of the tree knows about the next level of servers, but it doesn't know all possible domain names.
- The next level of servers knows how to get to lower levels, and so on.
- Small organizations can use a single server.
  - Easy to administer.
  - Inexpensive.
- Large organizations often use multiple servers.
  - Reliability through redundancy.

DNS Clients & Servers

- Improved response time through load-sharing.
- Delegation of naming authority.

- The principle of locality of reference applies in choosing a DNS server architecture.
  - A user tends to look up names of local computers more often than names of remote computers.
  - A user tends to look up the same names repeatedly.

Resolving a Name

- name resolution - the translation of a domain name into an equivalent IP address. The software that does this is called name resolver or resolver software.
- Resolver software is available as library routines.
  - Java has an **`InetAddress`** class.
  - UNIX has a utility **`nslookup`**.
- authority for a given name - a DNS server which knows about the name.

How It Works

- A program needing to have a name resolved to an address calls on resolver software.
- The resolver software constructs a DNS protocol message called a DNS request containing the name to be resolved. It then becomes a client and sends the request to a local DNS server.
- The DNS server resolves the name, constructs a DNS protocol message called a DNS reply containing the IP address, and sends the reply to the client.

How It Works

- If the DNS request to a server is outside the set of names for which the server is an authority, the server becomes the client of another DNS server. When the second server sends an answer, the first server sends a copy back to the original client.
- How does a DNS server know which other server is the authority for a given name? It doesn't. However, it does know how to reach a root server, and what it does is forward a request it can't resolve locally to a root server.

How It Works

- The root server then informs the first server what server to use next.
- This back-and-forth exchange occurs until an authoritative server is located and the IP address returned.
- iterative query resolution - requesting a list of other DNS servers to contact for the query.
- recursive query resolution - requesting a complete resolution of the query.

DNS Replication & Caching

- DNS resolution can be very inefficient.
  - Every host referenced by name triggers a DNS request.
  - Every DNS request for the address of a host in a different organization goes through a root server.
- To reduce the demand on a particular root server, many copies of a root server exist around the world.

DNS Replication & Caching

- To reduce the number of DNS requests, servers and hosts use caching.
  - cache - a list of recently resolved names and IP addresses.
  - When a server looks up a new name, it places a copy of the IP address in its cache. When it receives a new request, it checks its cache first.

Types of DNS Entries

- DNS can hold several types of records.
- Each record includes:
  - Domain name.
  - Record type.
  - Data value.
- Type A records map from a domain name to an IP address.
  - Domain name: **bacchus.cs.gettysburg.edu**
  - Record type: **A**
  - Data value: **138.234.44.50**
- A name that works with one application may not work with another!

Abbreviations

- Sometimes it’s convenient to use abbreviations. For example, **bacchus** instead of **bacchus.cs.gettysburg.edu**.
- DNS servers do not know abbreviations. They respond only to full domain names.
- Local resolver software can be configured to recognize abbreviations. Resolvers are given a list of suffixes to try until a match is found.