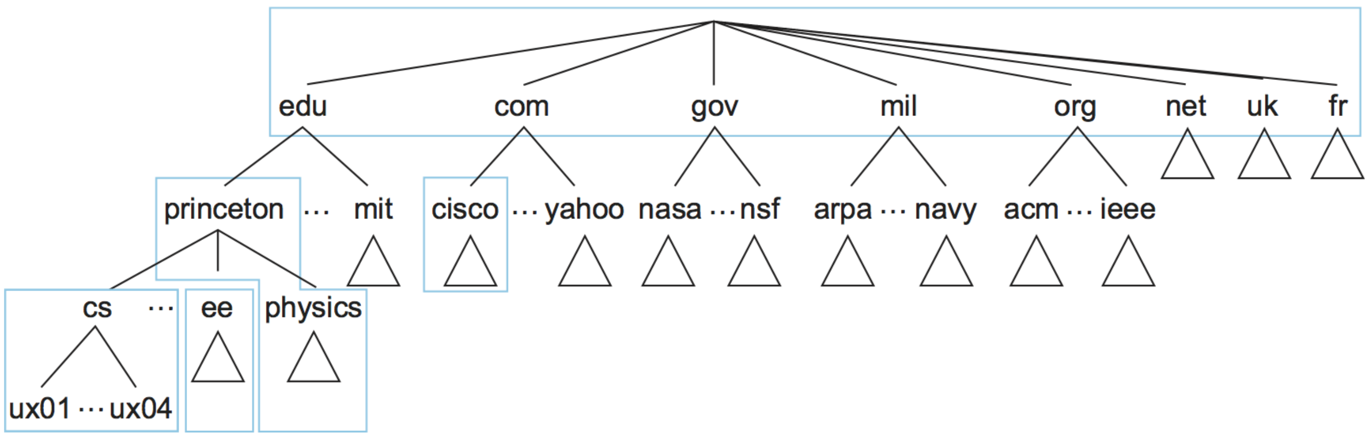
Application Layer

- Application model: (1) client-server; (2) peer-to-peer.
- Hyper Text Transfer Protocol (HTTP) relies on TCP.

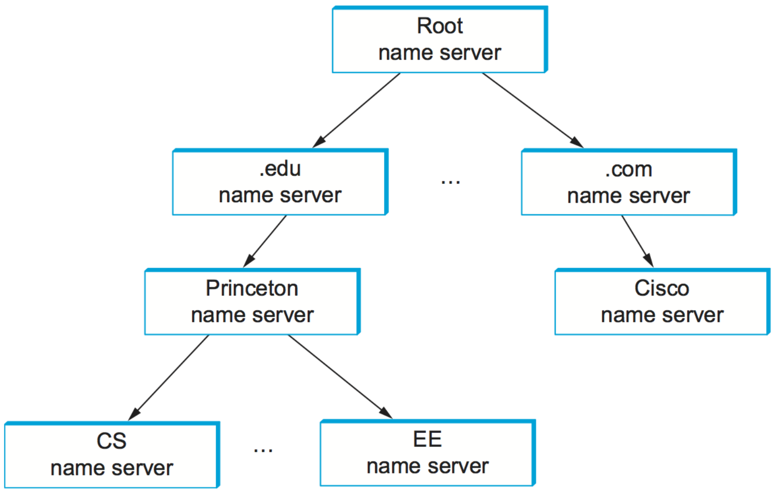
Domain Name Service (DNS)

A *name space* defines a set of possible names. A name space can be either *flat* or *hierarchical*. The naming system maintains mappings from names to values (usually addresses). A *resolution mechanism* is a procedure that takes a name and returns the corresponding value. A *name server* is a specific implementation of a resolution mechanism.

DNS names (alpha-numeric strings) are processed from right to left, using periods as separators. DNS hierarchy can be visualized as a tree, where each node in the tree is a domain, and the leaves in the tree are hosts being named.



The hierarchy is partitioned into subtrees called *zones*. Each zone corresponds to some administrative authority responsible for that portion of the hierarchy. The information in each zone is implemented in two or more name servers. Clients send queries to name servers, and name servers respond with the requested information. Sometimes the response contains the final answer that the client wants, and sometimes the response contains a pointer to another server that the client should query next. From an implementation perspective, DNS is a hierarchy of name servers.



Each name server implements the zone information as a collection of *resource records*, 5-tuples `(Name, Value, Type, Class, TTL)`.The `Type` field specifies how the `Value` should be interpreted. `NS` means the

`Value` field is the domain name for a name server that knows how to resolve names within the specified domain. `A` means IPv4 address.

Consider the process of resolving `penguins.cs.princeton.edu`. First, a root name server contains an `NS` record for each top-level domain (**TLD**) name server. This identifies a server that can resolve queries for this part of the DNS hierarchy. It also has `A` records that translates these names into the corresponding IP addresses. Taken together, these two records effectively implement a pointer from the root name server to one of the TLD servers.

```
(edu, a3.nstld.com, NS)
(a3.nstld.com, 192.5.6.32, A)
(com, a.gtld-servers.net, NS)
(a.gtld-servers.net, 192.5.6.30, A)
...
```

The TLD servers have:

```
(princeton.edu, dns.princeton.edu, NS, IN)
(dns.princeton.edu, 128.112.129.15, A, IN)
...
```

Down one level:

```
(penguins.cs.princeton.edu, dns1.cs.princeton.edu, NS, IN)
(dns1.cs.princeton.edu, 128.112.136.10, A, IN)
...
```

Finally, a third-level name server, such as the one managed by domain `cs.princeton.edu`, contains `A` records for all of its hosts.

```
(penguins.cs.princeton.edu, 128.112.155.166, A, IN)
(cs.princeton.edu, mail.cs.princeton.edu, MX, IN)
(mail.cs.princeton.edu, 128.112.136.72, A, IN)
...
```

Suppose the client wants to resolve the name `penguins.cs.princeton.edu`. The client could first send a query containing this name to one of the root servers. The root server, unable to match the entire name, returns the best match it has—the `NS` record for `edu` which points to the TLD server `a3.nstld.com`. The server also returns all records that are related to this record, in this case, the `A` record for `a3.nstld.com`. The client, having not received the answer it was after, next sends the same query to the name server at IP host `192.5.6.32`. This server also cannot match the whole name and so returns the `NS` and corresponding `A` records for the `princeton.edu` domain. Once again, the client sends the same query as before to the server at IP host `128.112.129.15`, and this time gets back the `NS` record and corresponding `A` record for the `cs.princeton.edu` domain. This time, the server that can fully resolve the query has been reached. A final query to the server at `128.112.136.10` yields the `A` record for `penguins.cs.princeton.edu`, and the client learns that the corresponding IP address is `128.112.155.166`.

Internet Caching

Main idea: move content closer to clients to reduce latency, improves robustness as well.

Content Delivery Network (CDN). The problem is that DNS would resolve the domain name to the IP address of the data center, instead of the local cache infrastructure.

Detail skipped. Not a popular topic for interview.