Chapter 25

Domain Name System

- The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.
- To identify an entity, TCPIIP protocols use the IP address, which uniquely identifies the connection of a host to the Internet.
- However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.

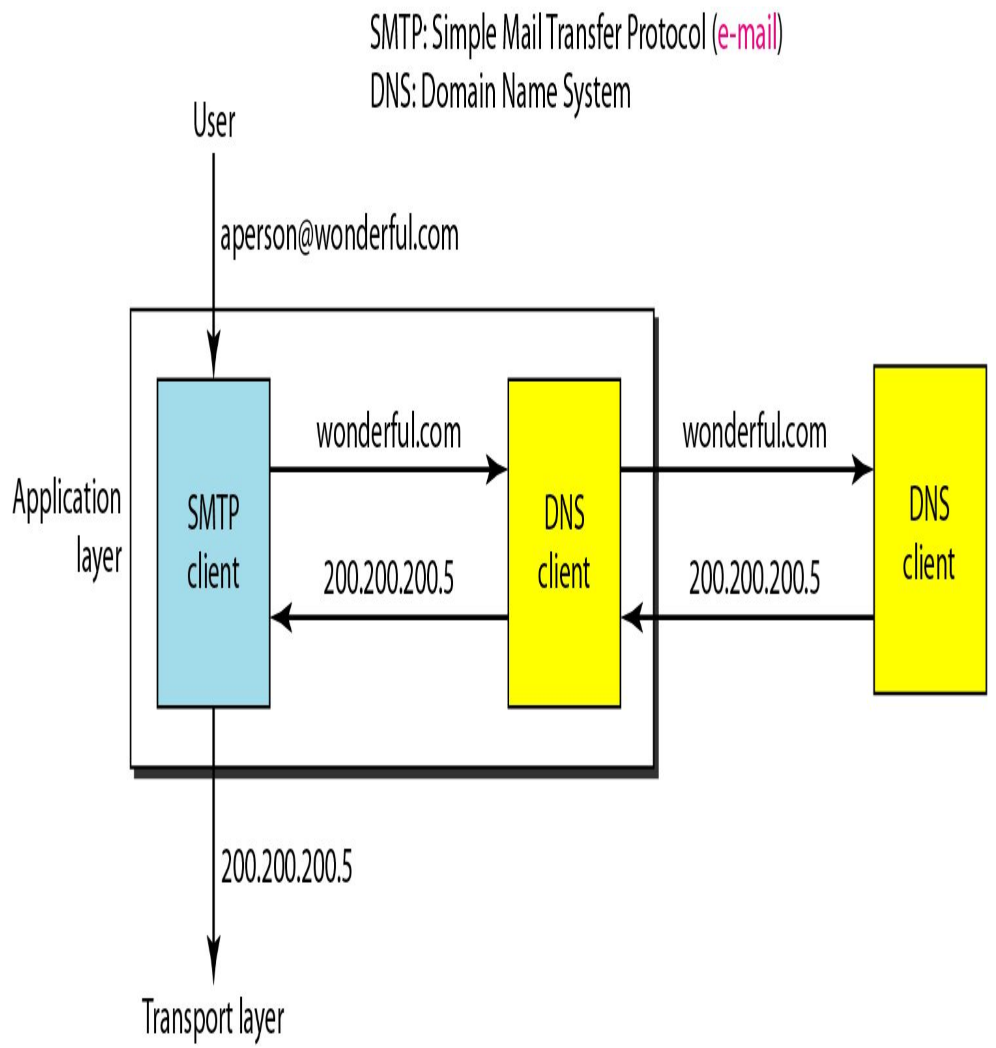


Figure 25.1 *Example of using the DNS service*

- When the Internet was small, mapping was done by using a host file.
- The host file had only two columns: name and address. Every host could store the host file on its disk and update it periodically from a master host file.
- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.
- Today, however, it is impossible to have one single host file to relate every address with a name and vice versa.
- The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there was a change.
- One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping.
- But this would create a huge amount of traffic on the Internet.
- Another solution, the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer

25-1 NAME SPACE

To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.

Topics discussed in this section:

Flat Name Space
Hierarchical Name Space

Flat Name Space

- In a flat name space, a name is assigned to an address.

- A name in this space is a sequence of characters without structure.

- The names mayor may not have a common section; if they do, it has no meaning.

- The main disadvantage of a fiat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Narne Space

- Each name is made of several parts.
- The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on.
- The authority to assign and control the name spaces can be decentralized.
- A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.
- The responsibility of the rest of the name can be given to the organization itself.
- The organization can add suffixes (or prefixes) to the name to define its host or resources.
- The management need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different.

25-2 DOMAIN NAME SPACE

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.

Topics discussed in this section:

Label
Domain Name
Domain

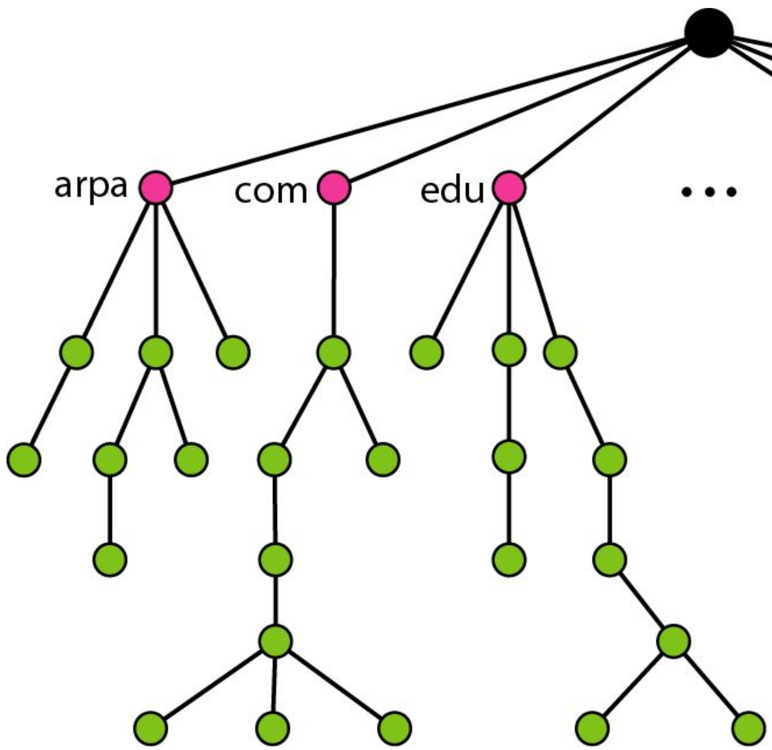


Figure 25.2 ***Domain name space***

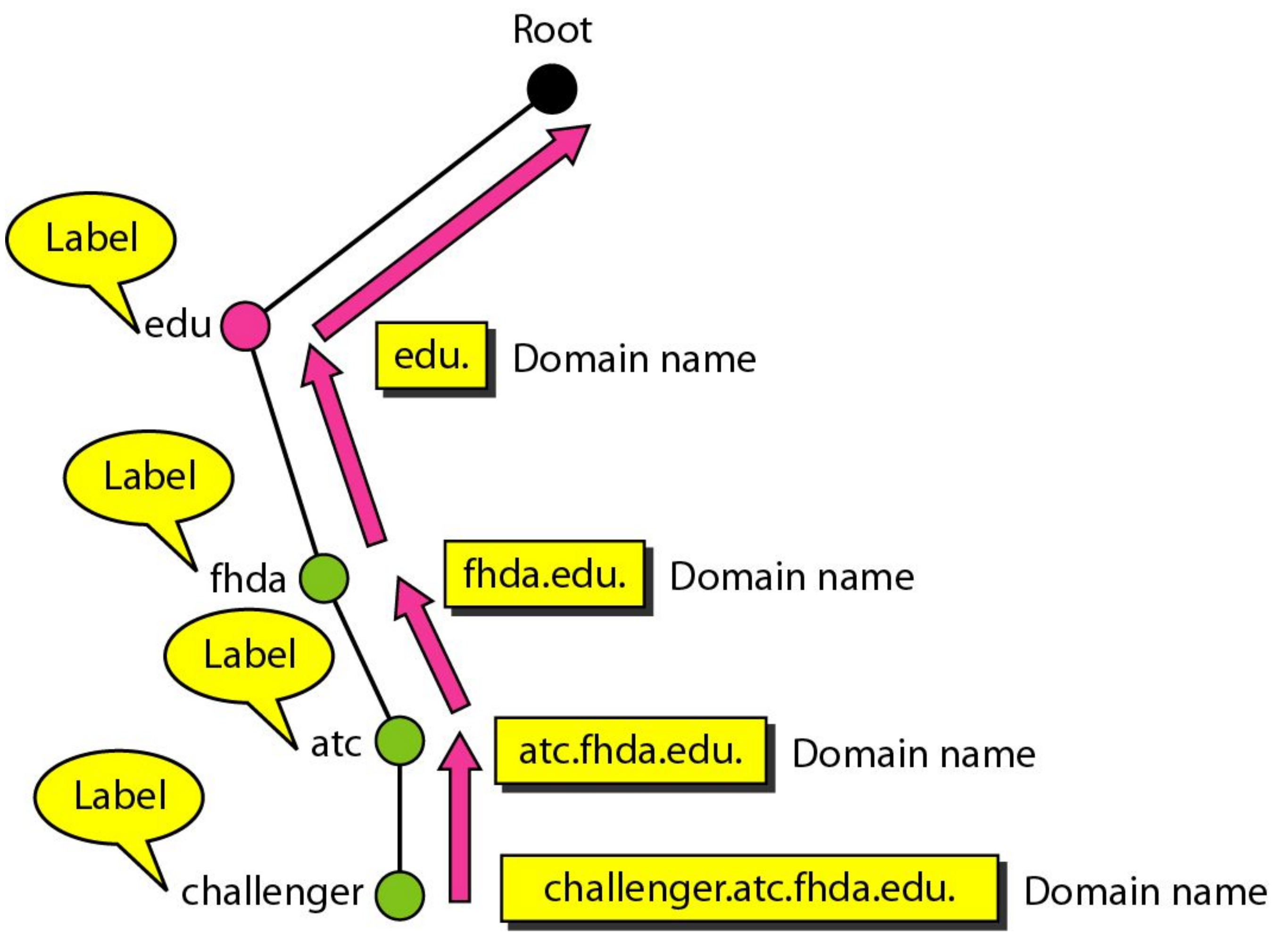
1) Label

- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- The root label is a null string (empty string). DNS requires that children of a node have different labels, which guarantees the uniqueness of the domain names.

2) Domain Name

- Each node in the tree has a domain name.
- A full domain name is a sequence of labels separated by dots (.).
- The domain names are always read from the node up to the root.
- The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

Figure 25.3 *Domain names and labels*



Fully Qualified Domain Name

- If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).

- is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host.

- For example, the domain name challenger.ate.tbda.edu. is the FQDN of a computer named *challenger installed at the Advanced Technology* Center (ATC) at De Anza College.

- A DNS server can only match an FQDN to an address. Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

challenger.ate.tbda.edu.

Partially Qualified Domain Name

- If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root.
- It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the suffix, to create an FQDN.
- For example, if a user at the jhda.edu. site wants to get the IP address of the challenger computer, he or she can define the partial name.

challenger

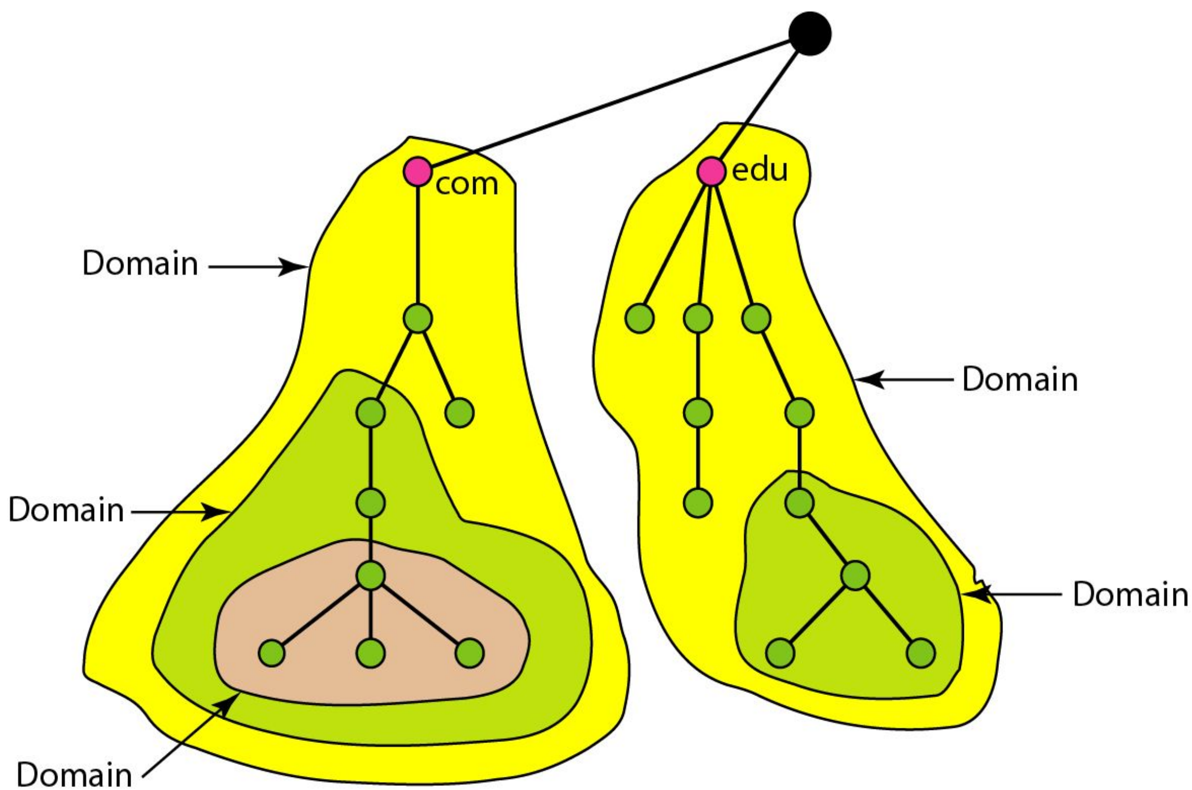
- The DNS client adds the suffix atc.jhda.edu. before passing the address to the DNS server.

Figure 25.4 *FQDN and PQDN*

FQDN	PQDN
challenger.atc.fhda.edu.	challenger.atc.fhda.edu
cs.hmme.com.	cs.hmme
www.funny.int.	www

3) Domain

- A domain is a subtree of the domain name space.
- The name of the domain is the domain name of the node at the top of the subtree. Below Figure shows some domains.
- Note that a domain may itself be divided into domains (or subdomains as they are sometimes called).



25-3 DISTRIBUTION OF NAME SPACE

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information. In this section, we discuss the distribution of the domain name space.

Topics discussed in this section:

Hierarchy of Name Servers
Zone
Root Server
Primary and Secondary Servers

- The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information.

- It is inefficient because responding to requests from all over the world places a heavy load on the system. It is not unreliable because any failure makes the data inaccessible.

- The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level.

- In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes. Because a domain created in this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or a small domain.

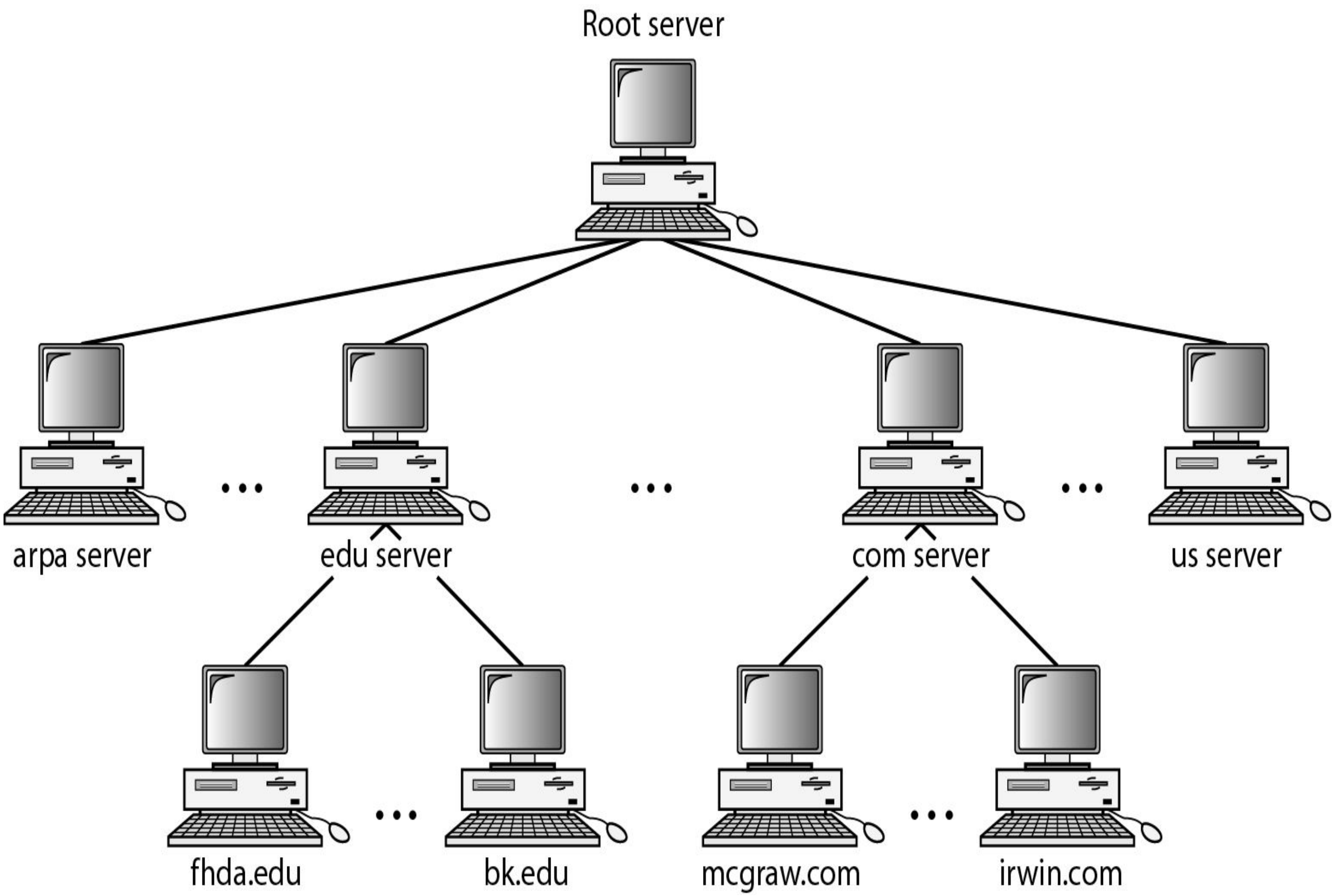
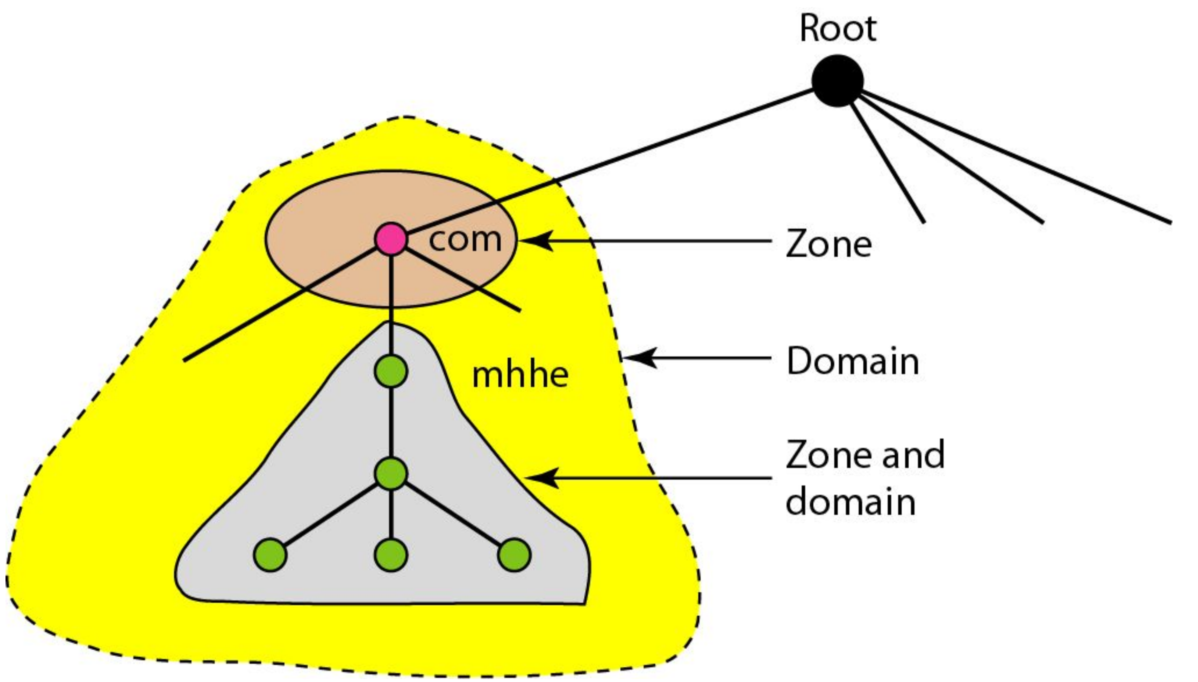


Figure 25.6 ***Hierarchy of name servers***

Figure 25.7 *Zones and domains*



Primary and Secondary Servers

- DNS defines two types of servers: primary and secondary.

- A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

- A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

The idea is not to put the secondary server at a lower level of authority but to create redundancy for the data so that if one server fails, the other can continue serving clients.

Note

A primary server loads all information from the disk file; the secondary server loads all information from the primary server.

When the secondary downloads information from the primary, it is called zone transfer.

25-4 DNS IN THE INTERNET

DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.

Topics discussed in this section:

Generic Domains
Country Domains
Inverse Domain

Figure 25.8 *DNS IN THE INTERNET*

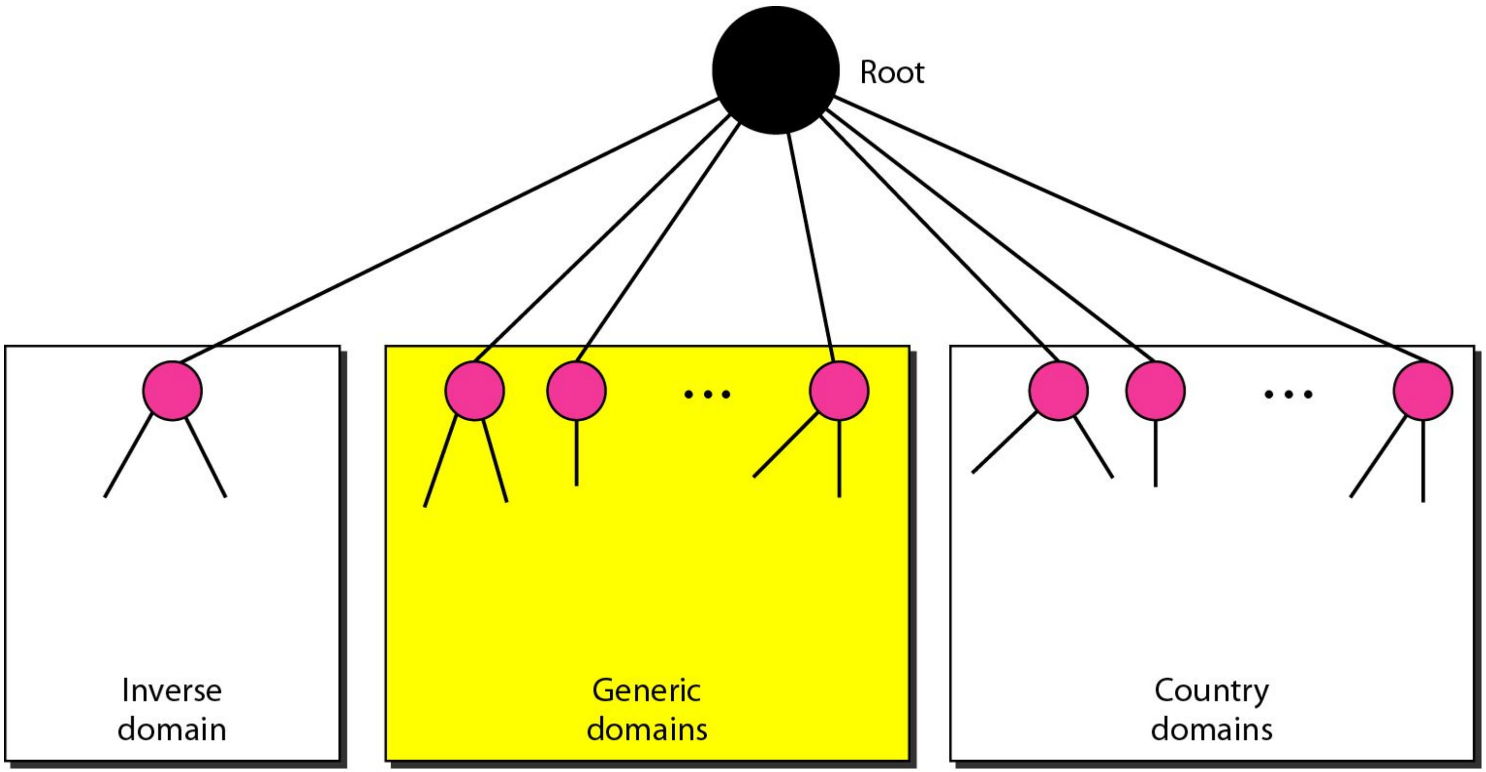


Figure 25.9 ***Generic domains***

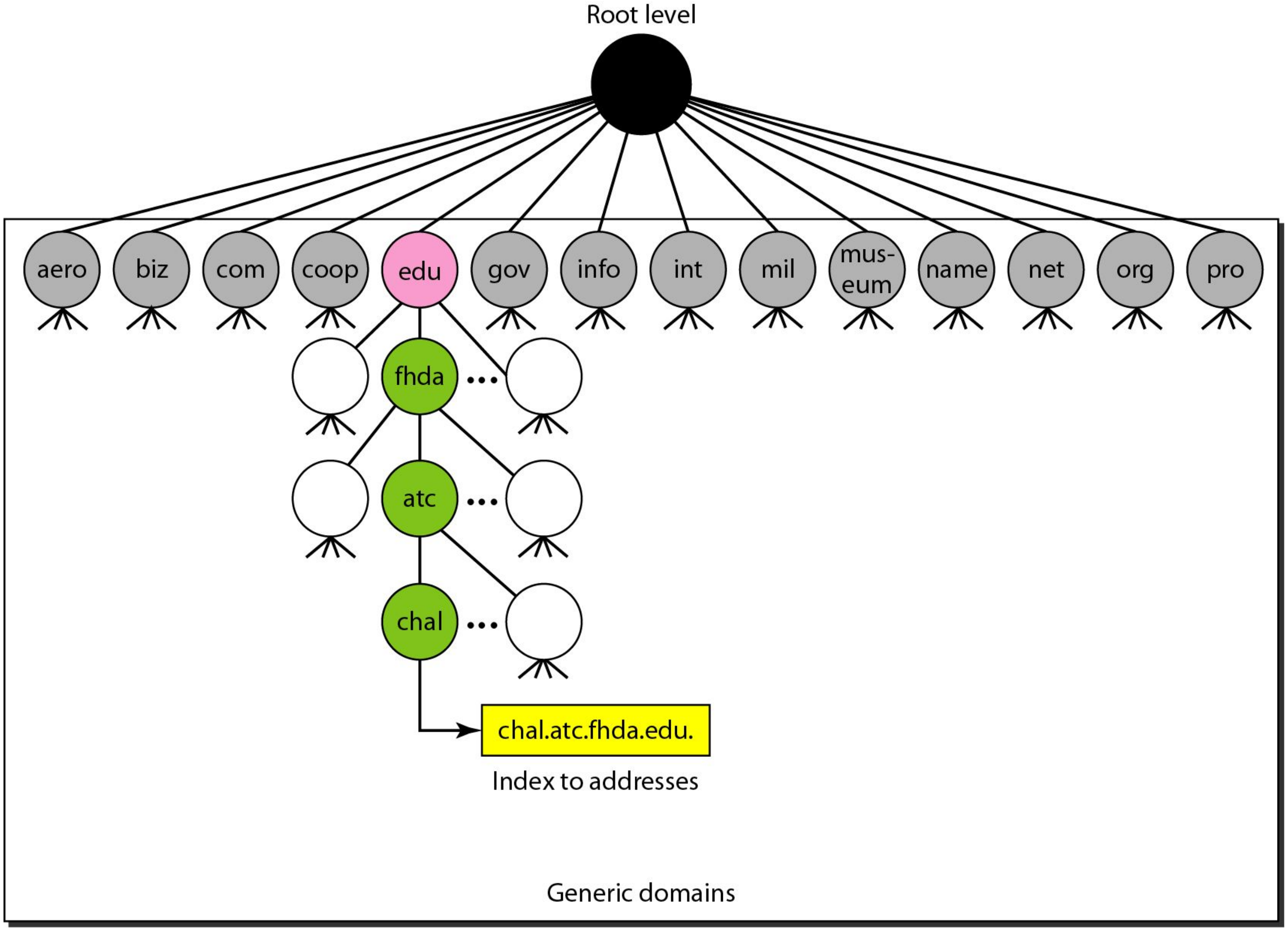


Table 25.1 ***Generic domain labels***

Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to “com”)
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Figure 25.10 *Country domains*

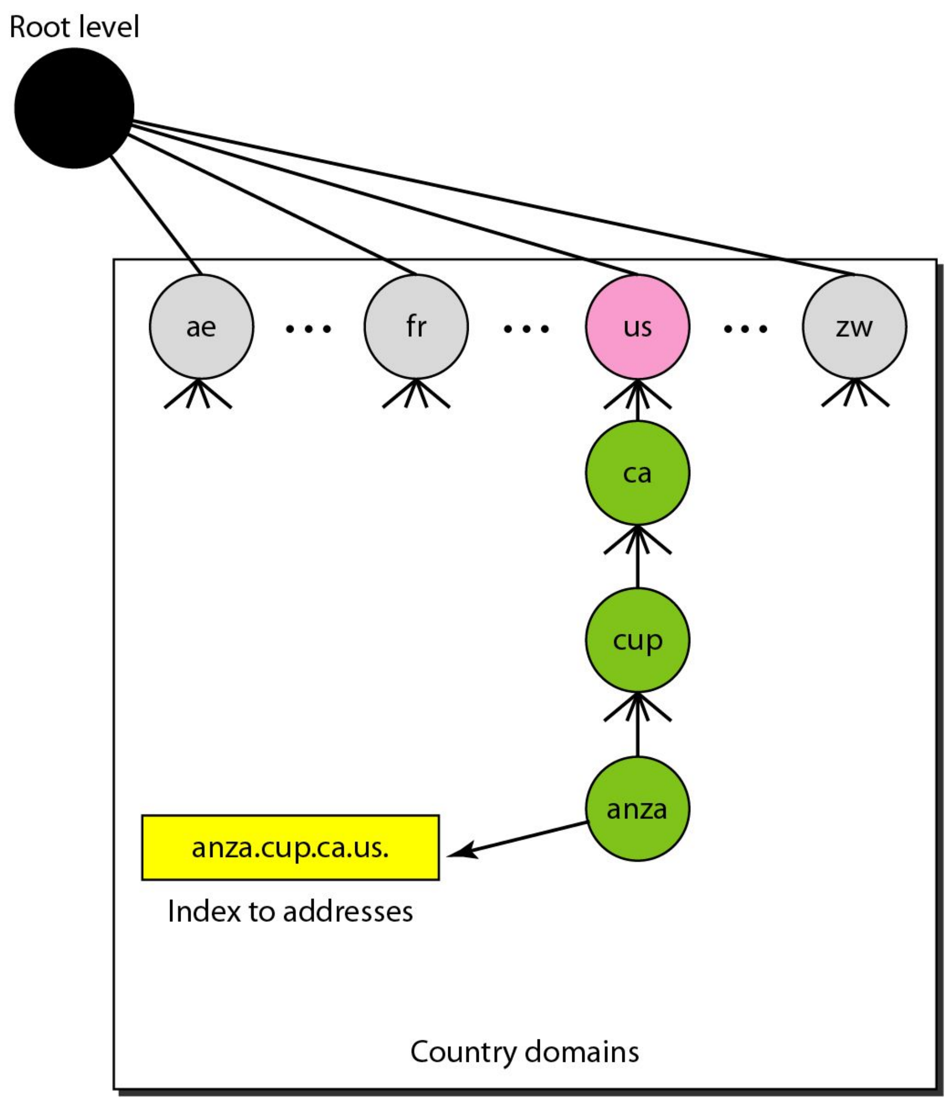
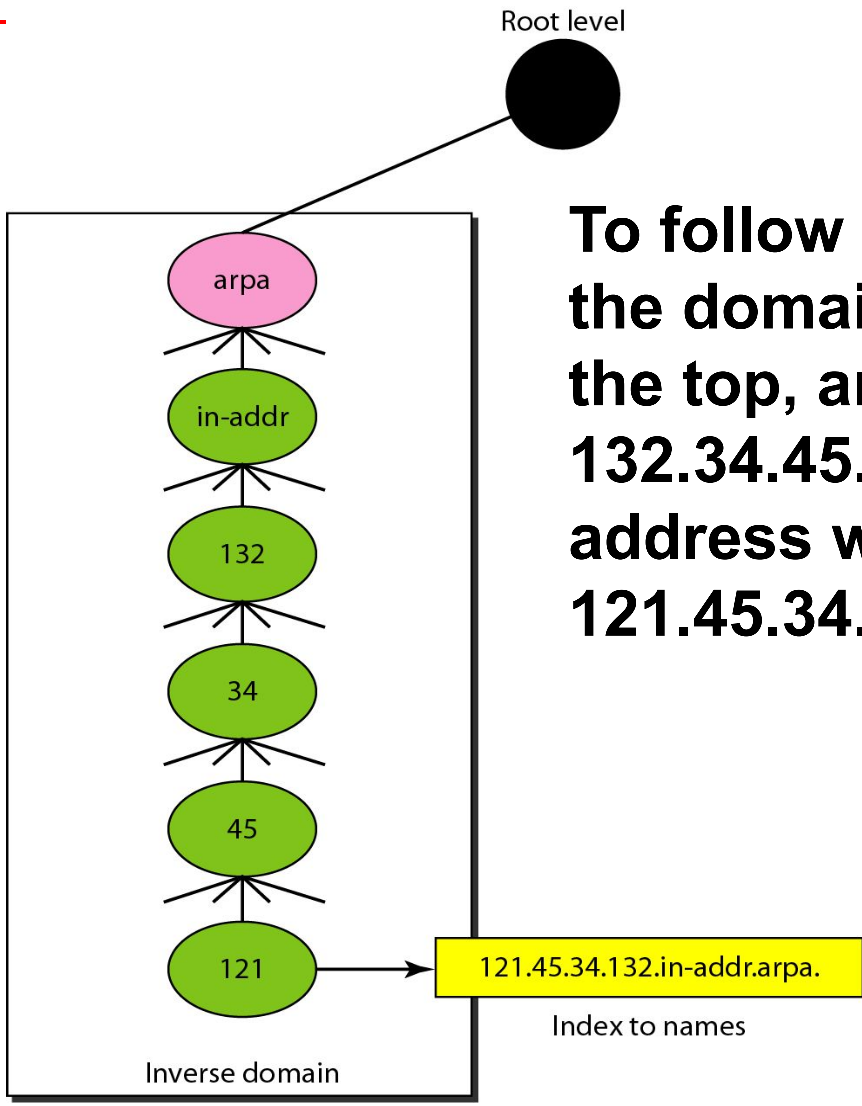


Figure 25.11 *Inverse domain*



To follow the convention of reading the domain labels from the bottom to the top, an IF address such as 132.34.45.121 (a class B address with netid 132.34) is read as 121.45.34.132.in-addr. arpa.

- The inverse domain is used to map an address to a name. for example,when a server has received a request from a client to do a task.
- Although the server has a file that contains a list of authorized clients, only the IP address of the client is listed.
- The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.
- This type of query is called an inverse or pointer (PTR) query.
- To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called *arpa.*
- *The second level is also one single node named in-addr (for inverse address).*
- *The rest of the domain defines IP addresses.*

25-5 RESOLUTION

Mapping a name to an address or an address to a name is called name-address resolution.

Topics discussed in this section:

Resolver
Mapping Names to Addresses
Mapping Addresses to Names
Recursive Resolution
Caching

Figure 25.12 *Recursive resolution*

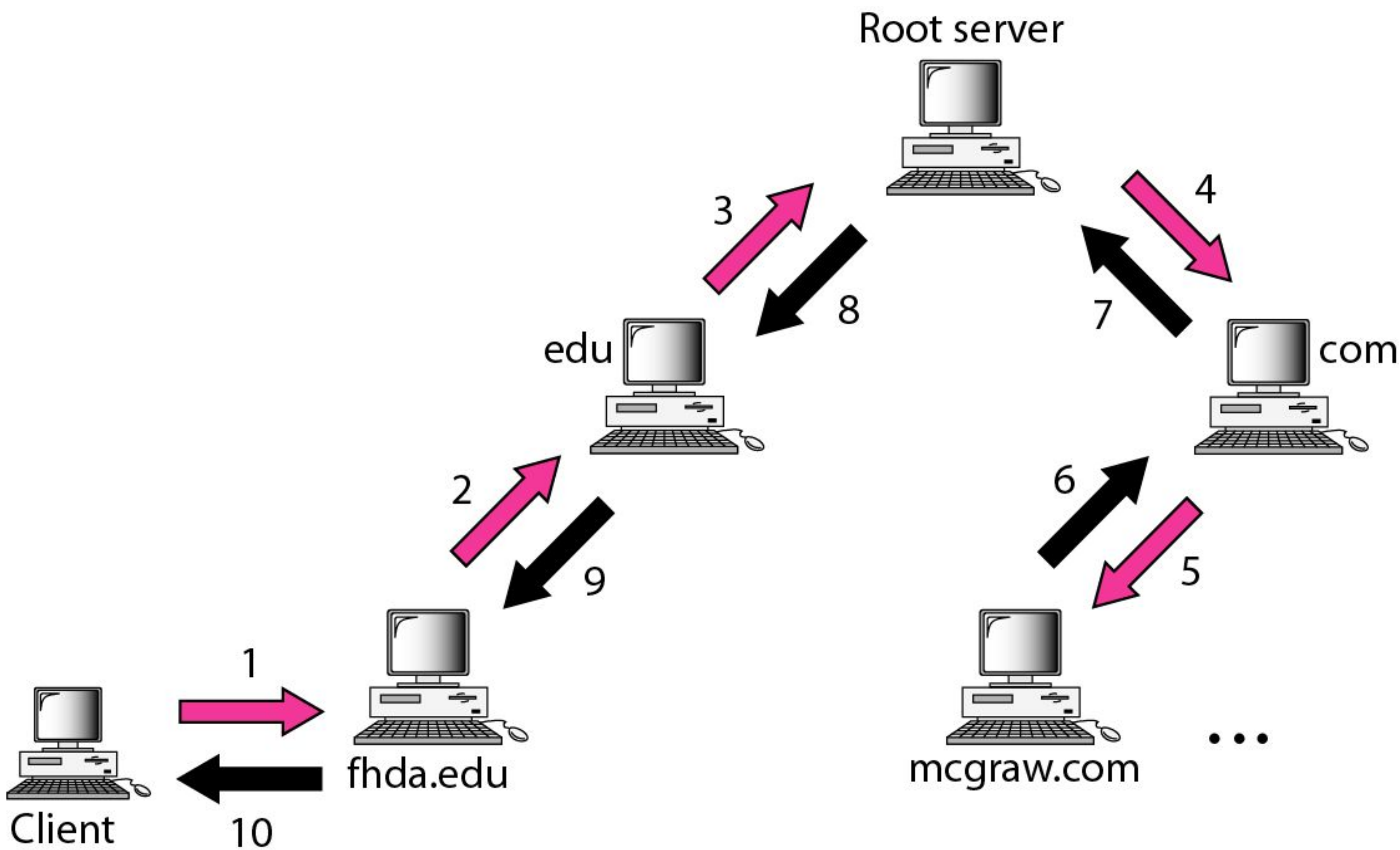


Figure 25.13 *Iterative resolution*