Chapter 5

Application Layer

- STANDARD CLIENT SERVER APPLICATIONS
  - HTTP
  - FTP
  - SMTP
  - Telnet
  - SSH
  - DNS

Domain Name System (DNS)

- Name Space
  - Domain Name Space
  - Domain
  - Distribution of Name Space
  - Zone
  - Root Server
- DNS in the Internet
  - Generic Domains
  - Country Domains

- Resolution
  - Recursive Resolution
  - Iterative Resolution
  - Caching
- Resource Records
- DNS Messages
- Encapsulation
- Registrars
- DDNS
- Security of DNS

Domain Name System (DNS)

To identify an entity, TCP/IP 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, the Internet needs to have a directory system that can map a name to an address. This is analogous to the telephone network. A telephone network is designed to use telephone numbers, not names. People can either keep a private file to map a name to the corresponding telephone number or can call the telephone directory to do so.

Since the Internet is so huge today, a central directory system cannot hold all the mapping. In addition, if the central computer fails, the whole communication network will collapse.

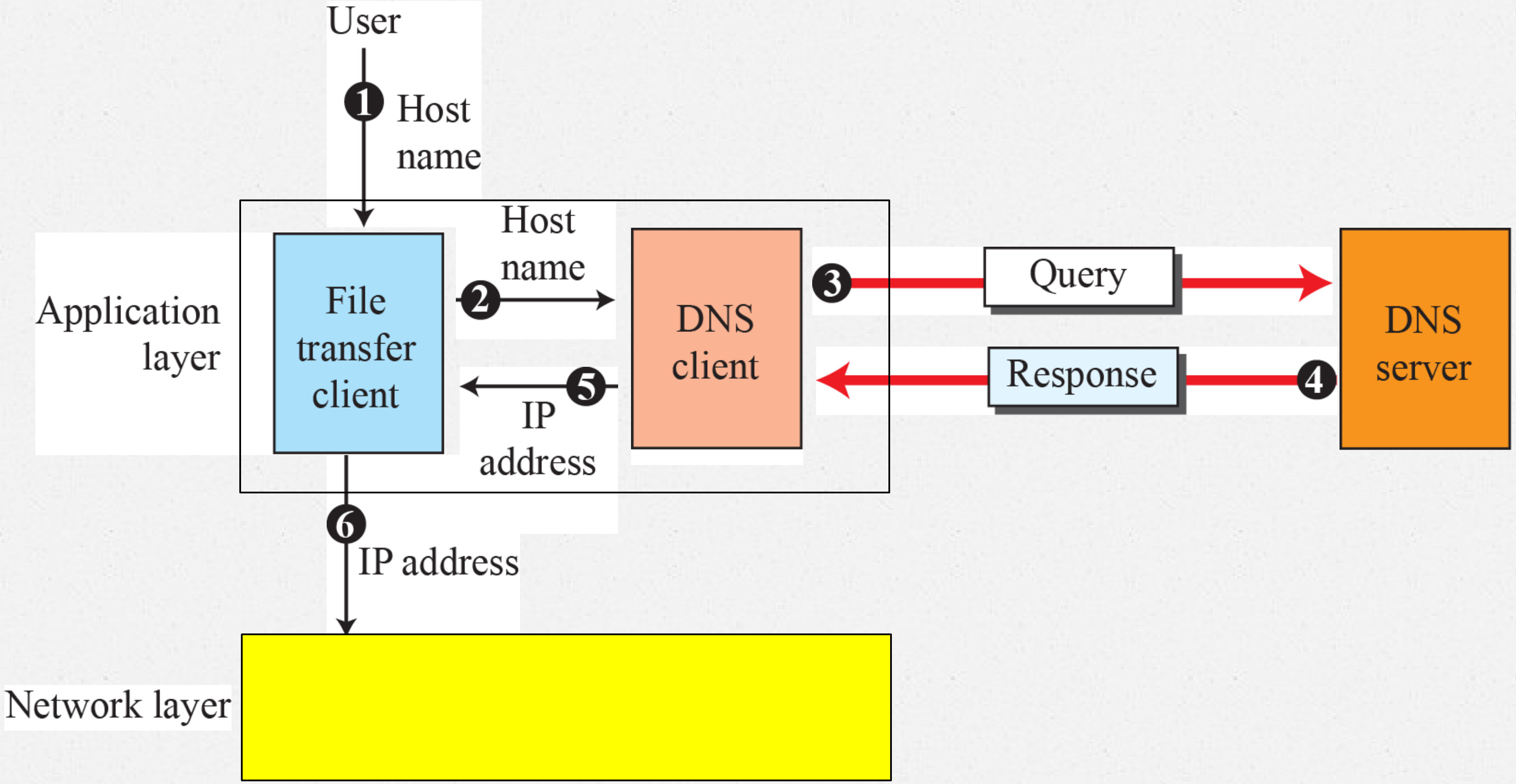
A better solution is to distribute the information among many computers in the world. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the **Domain Name System (DNS).**

TCP/IP uses a DNS client and a DNS server to map a name to an address

A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as afilesource.com. However, the TCP/IP suite needs the IP address of the file transfer server to make the connection. The following six steps map the host name to an IP address:

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The file transfer client now uses the received IP address to access the file transfer server.

Purpose of DNS



Name Space

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. The names must be unique because the addresses are unique.

A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

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 main disadvantage of a flat 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.

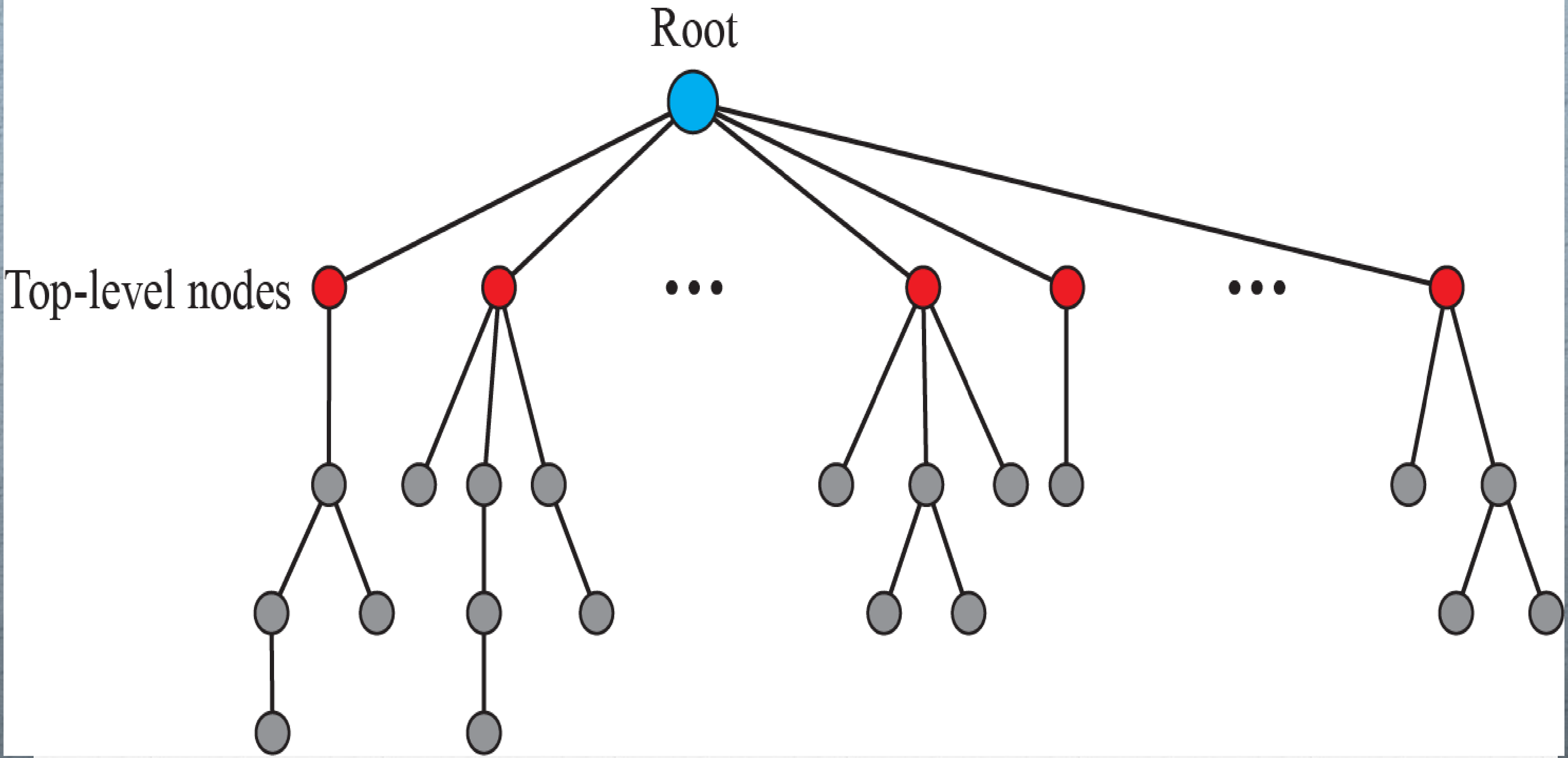
Hierarchical Name Space

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. In this case, the authority to assign and control the name spaces can be decentralized.

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.

Domain name space



Label

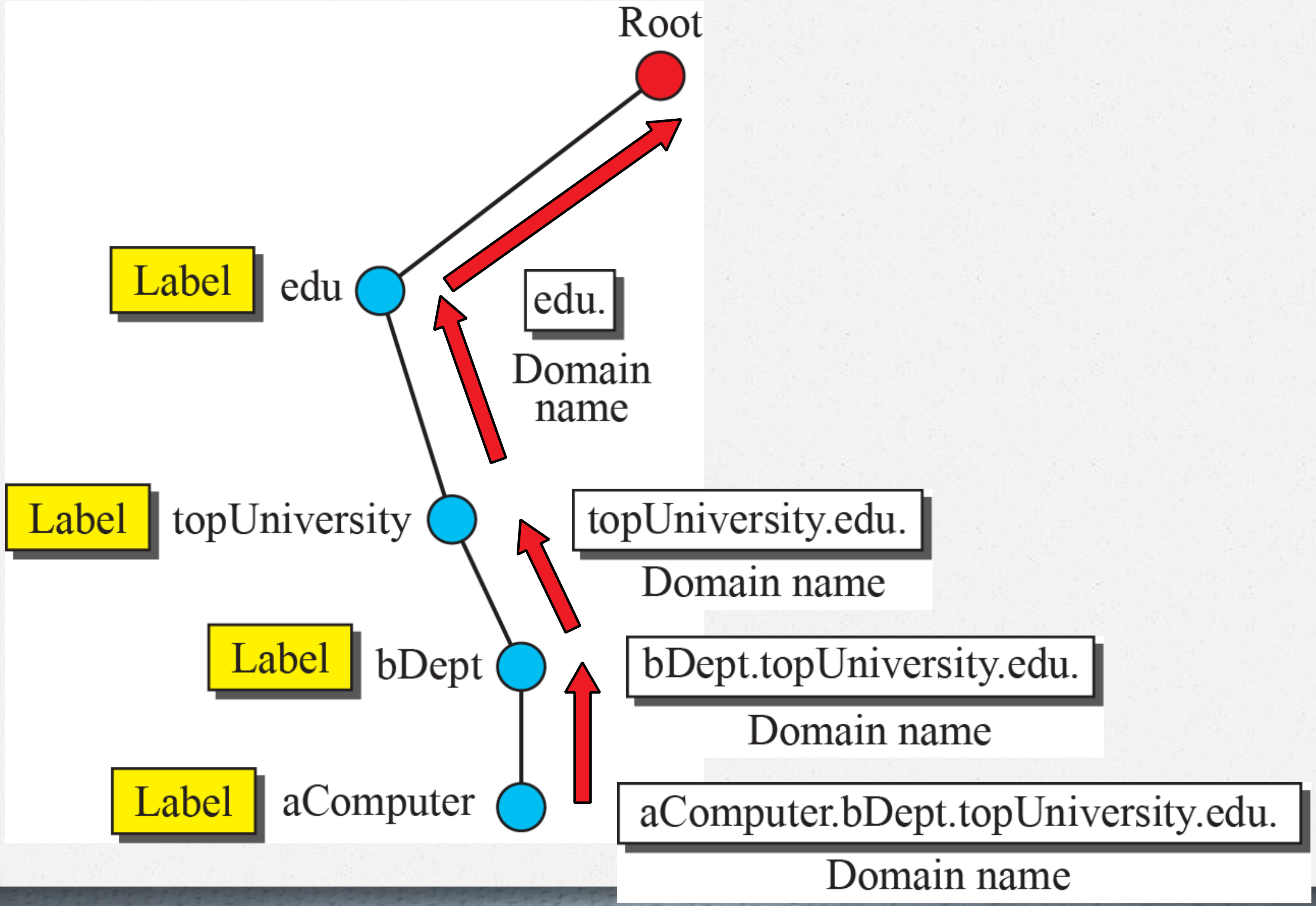
Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

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.

If a label is terminated by a null string, it is called a **fully qualified domain name (FQDN)**. The name must end with a null label, but because null means nothing, the label ends with a dot. 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.

Domain names and labels



Domain

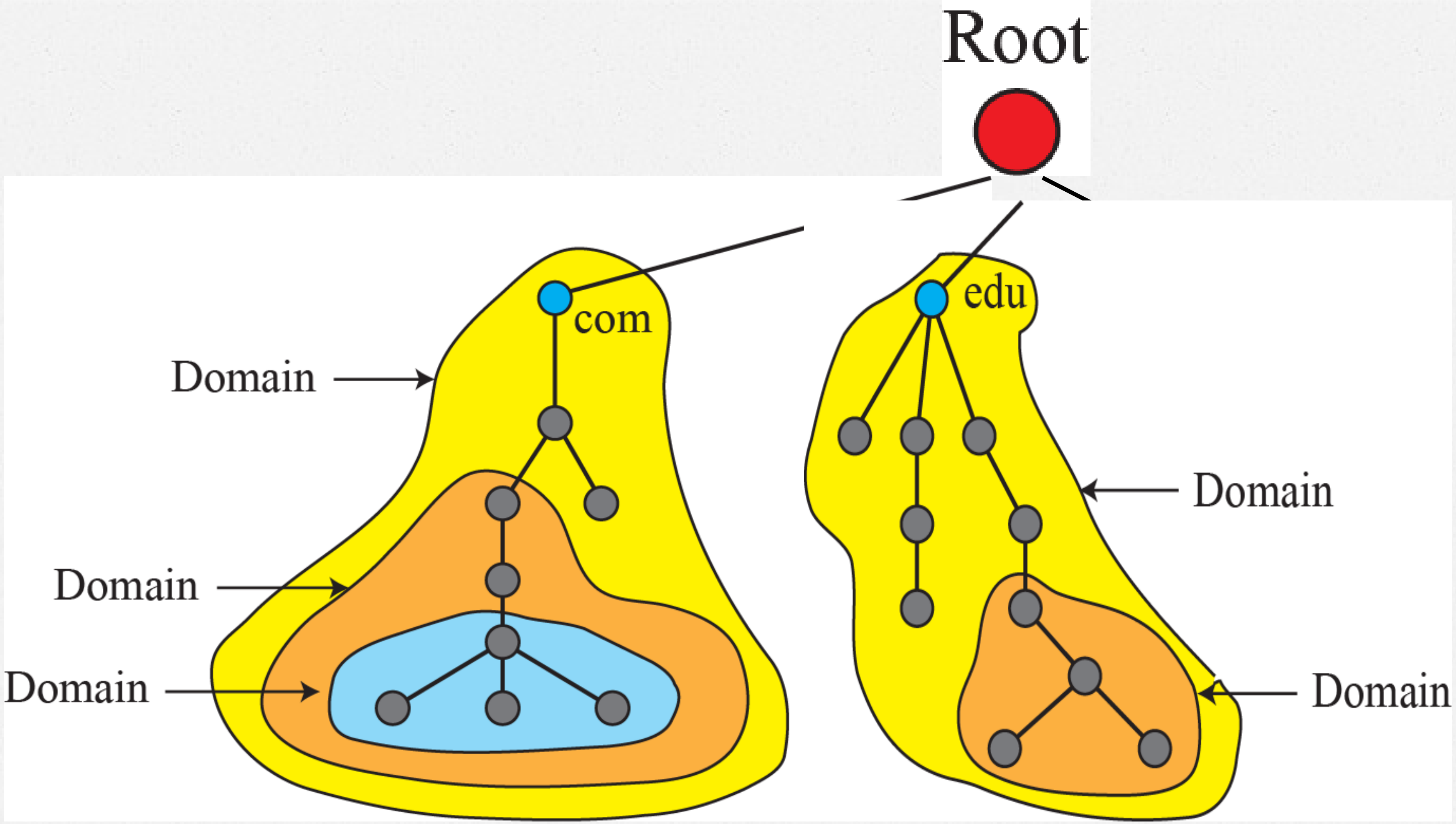
A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree.

A domain may itself be divided into domains.

Distribution of Name Space

The information contained in the domain name space must be stored. However, it is very inefficient and also not reliable 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 reliable because any failure makes the data inaccessible.

Domains

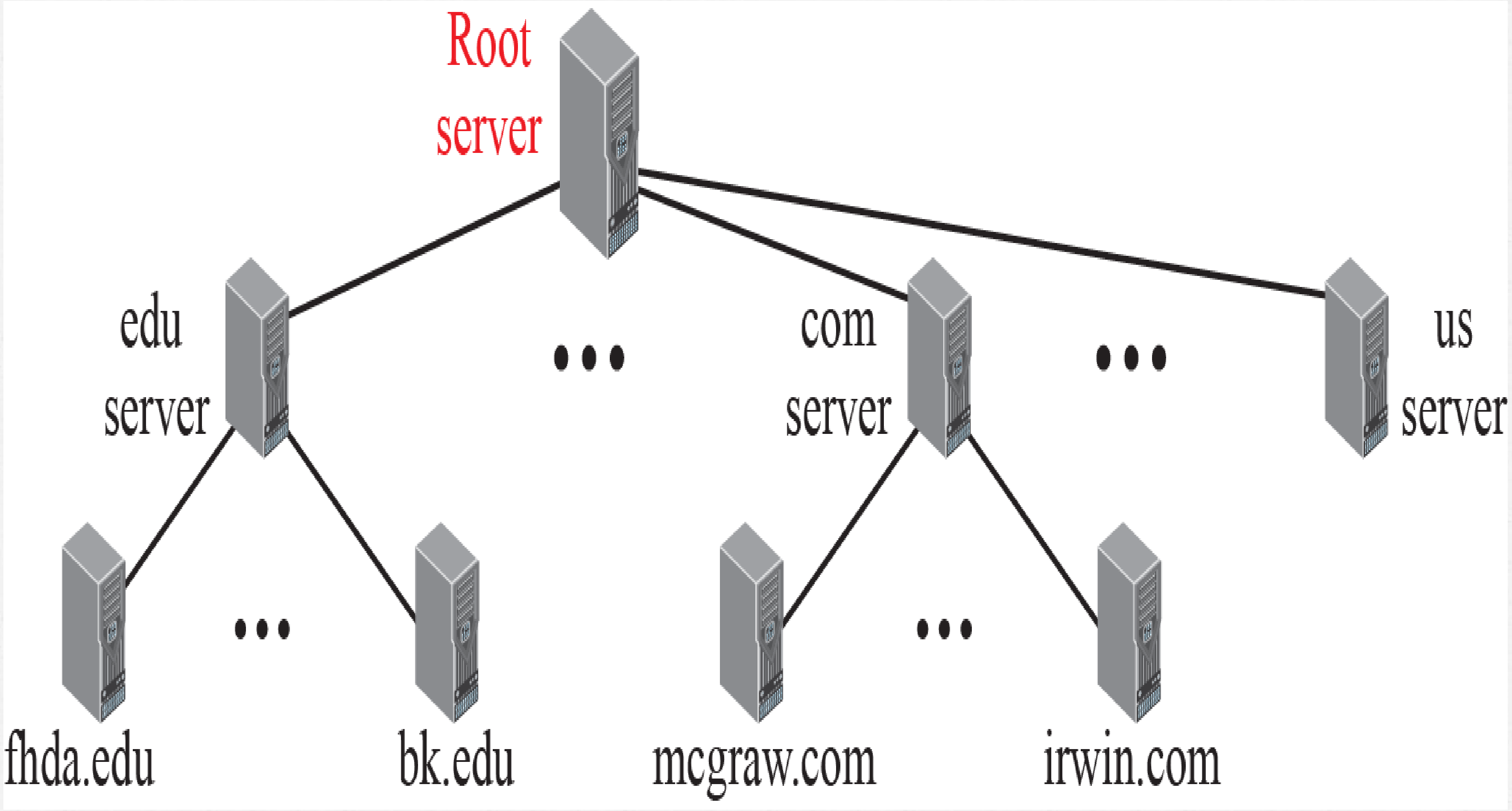


Hierarchy of Name Servers

One way to do this is to divide the whole space into many domains 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 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 small domain.

Hierarchy of name servers



Zone

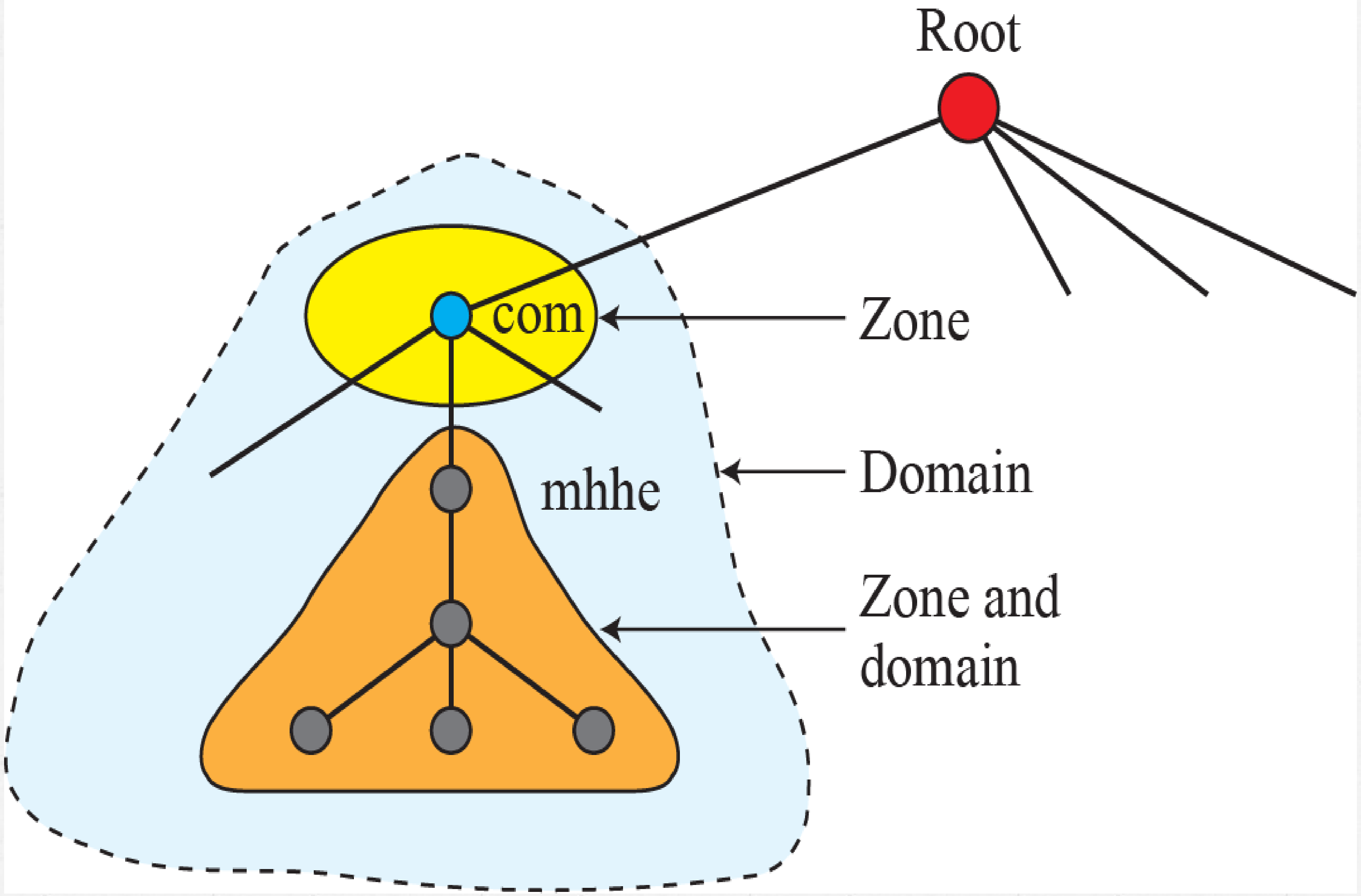
Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree.

The server makes a database called a zone file and keeps all the information for every node under that domain.

Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space.

Zone



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.

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

DNS in the Internet

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

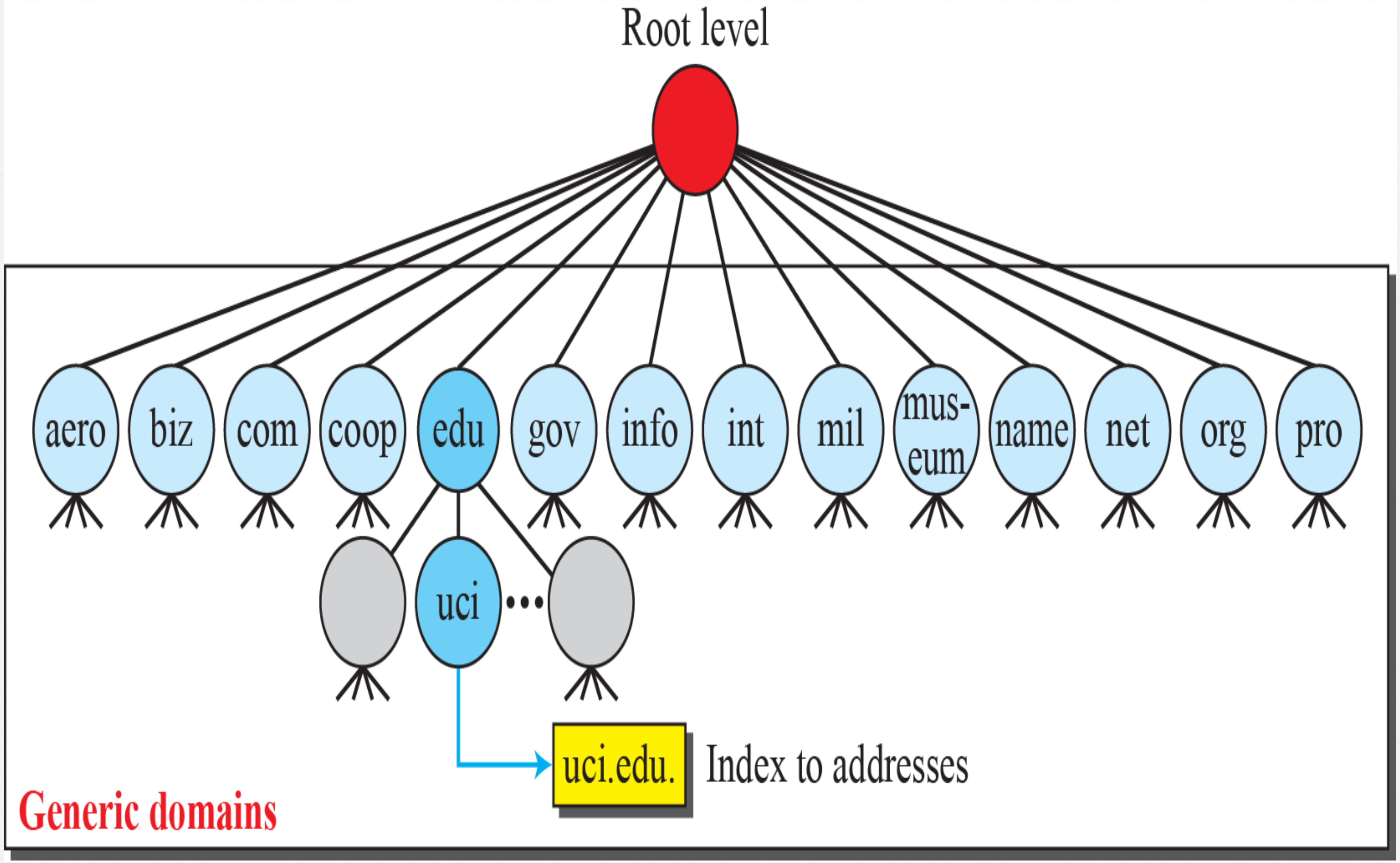
Generic Domains

The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

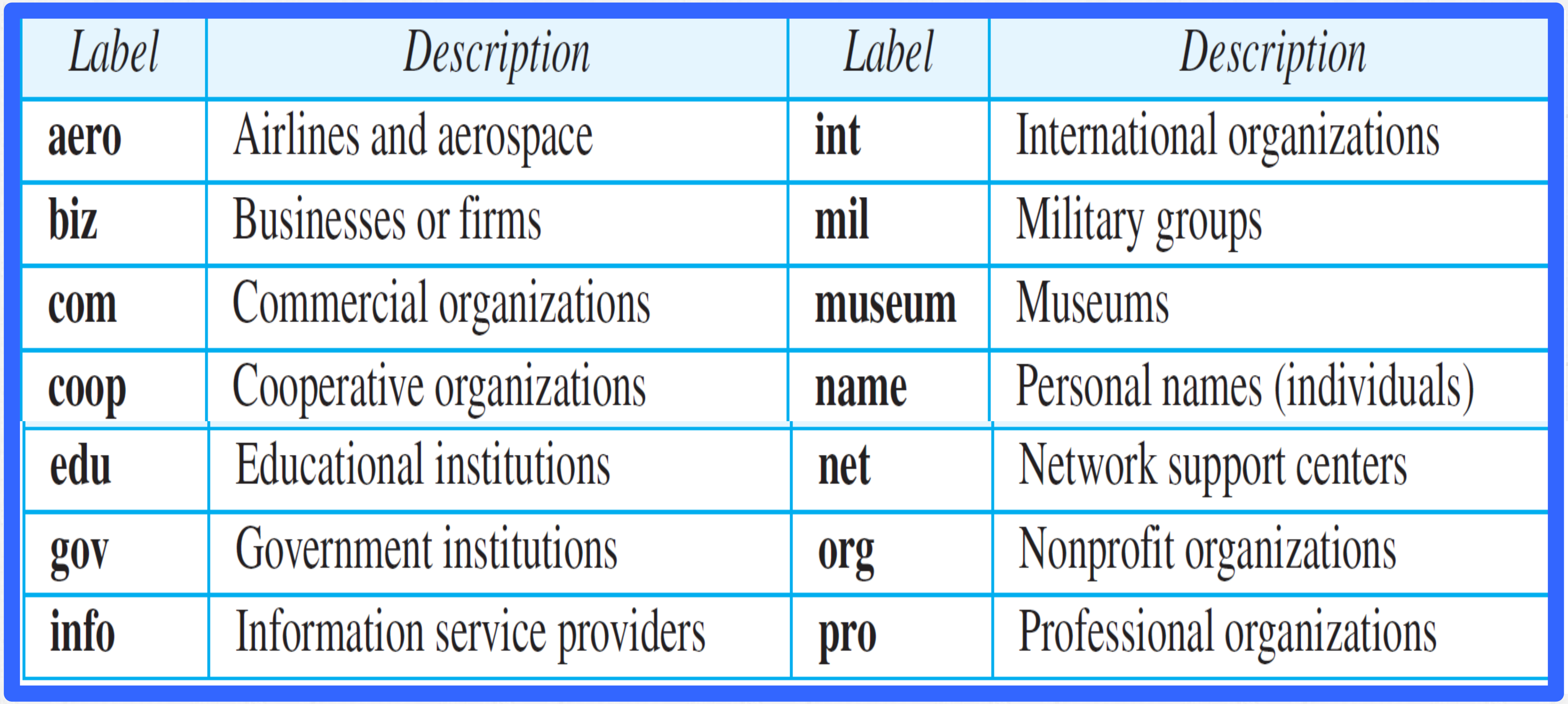
Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific, national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).

Generic domains

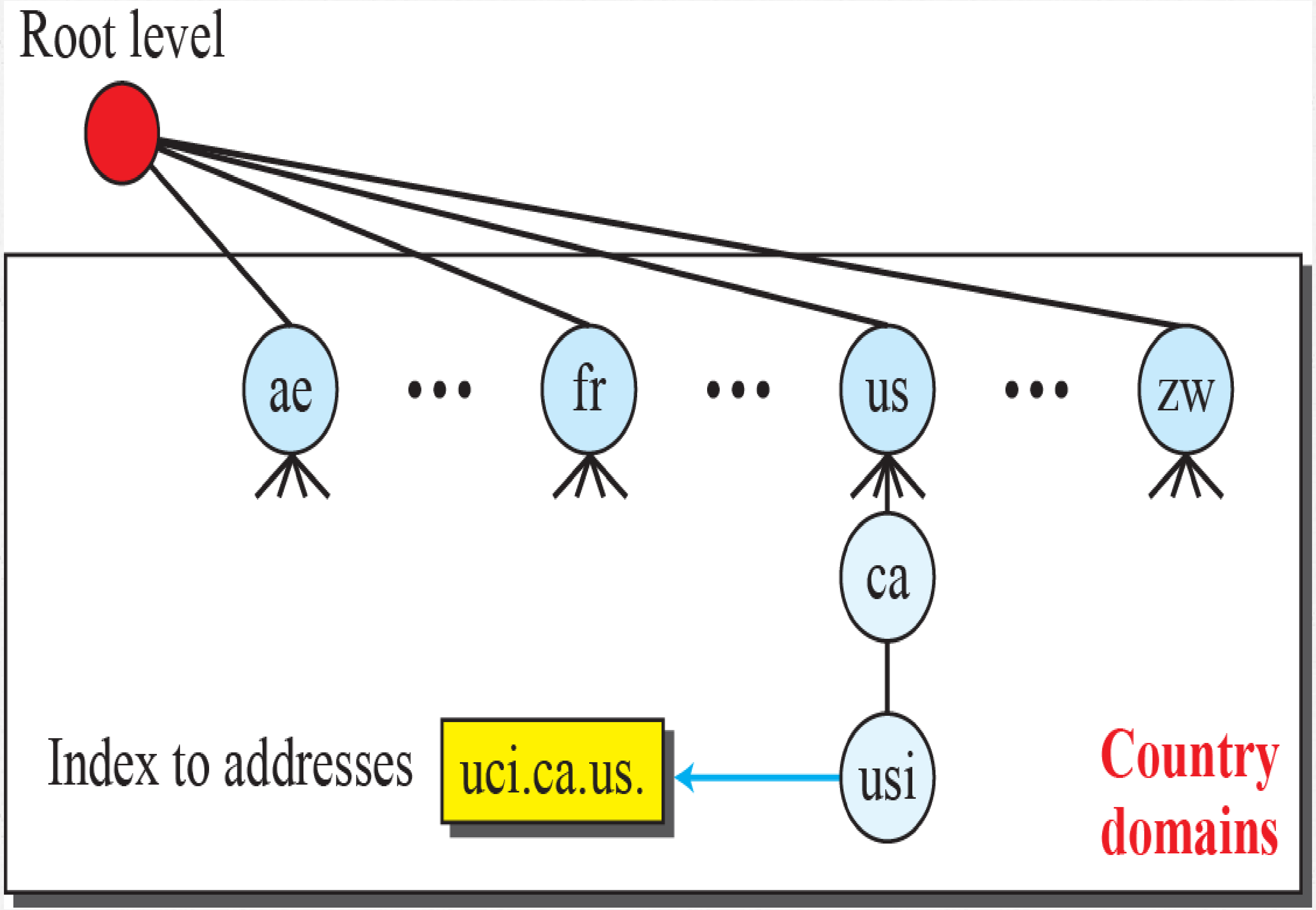


Generic domain labels



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

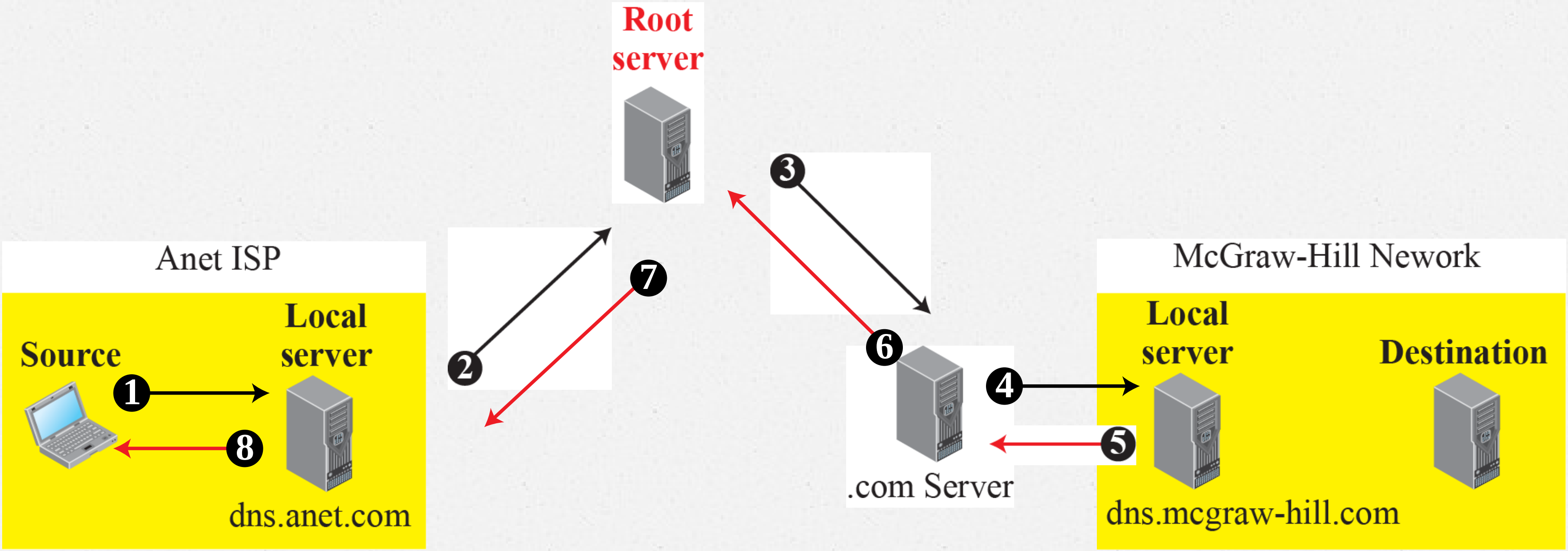
Country domains



Resolution

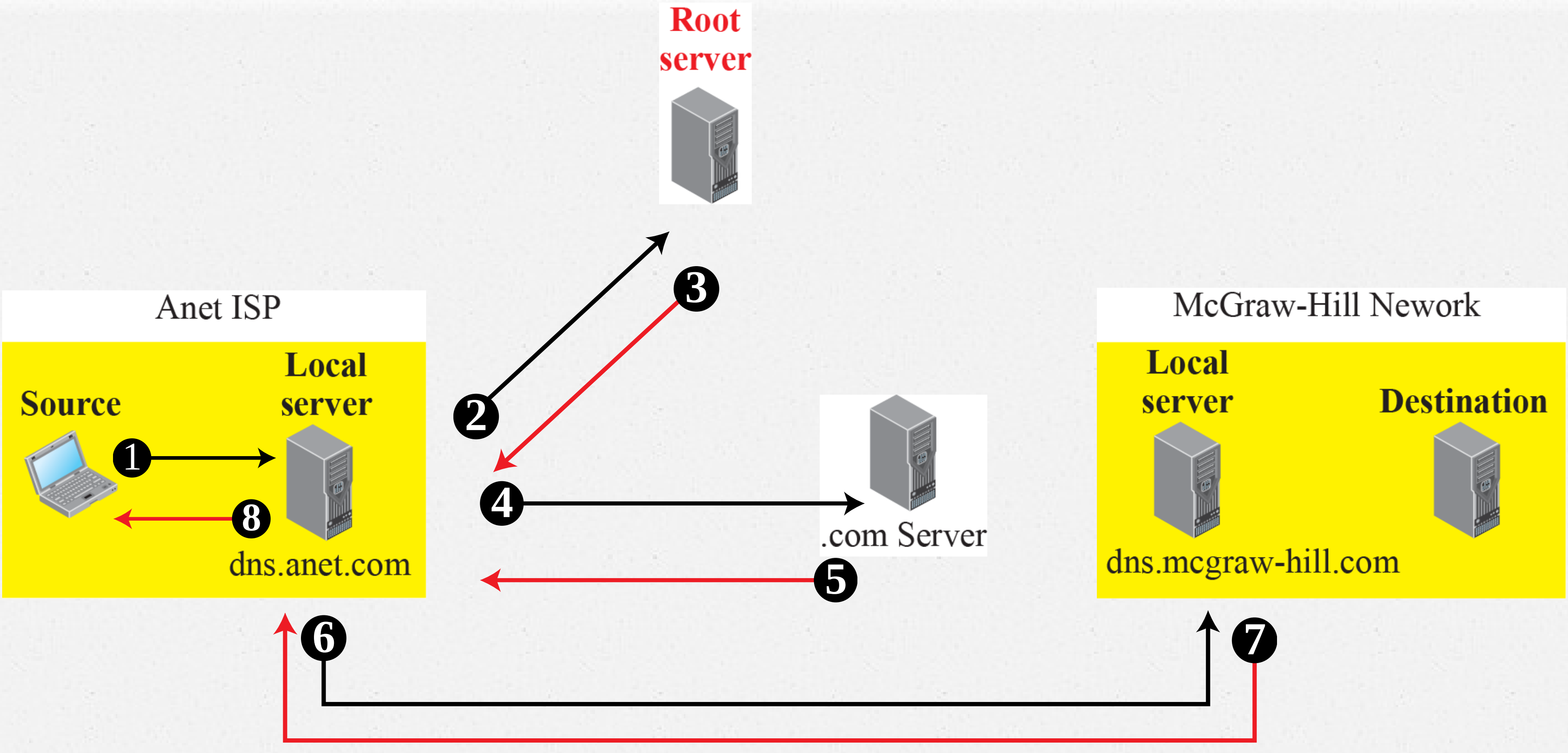
- Mapping a name to an address is called **name-address resolution**. DNS is designed as a client-server application.
- A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request.
- If the server has the information, it satisfies the resolver;
- otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
- After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.
- A resolution can be either **recursive** or **iterative**.

Recursive resolution



Source: some.anet.com
Destination: engineering.mcgraw-hill.com

Iterative resolution



Source: some.anet.com
Destination: engineering.mcgraw-hill.com

Caching

Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with a mechanism called **caching**.

When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.

Caching speeds up resolution, but it can also be problematic. If a server caches a mapping for a long time, it may send an outdated mapping to the client To counter this, two techniques are used.

- **Time To Live (TTL).** It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server.
- DNS requires that each server keep a **TTL counter** for each mapping it caches. The cache memory must be searched periodically and those mappings with an expired TTL must be purged

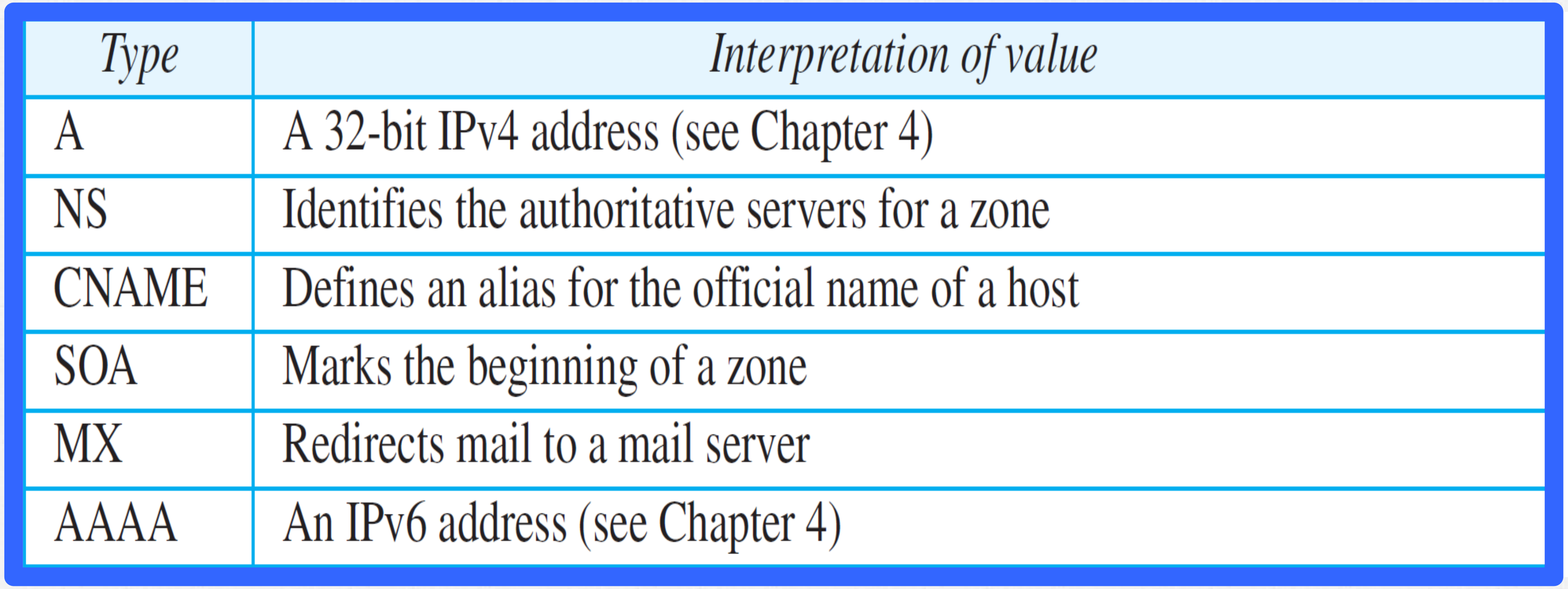
Resource Records

The zone information associated with a server is implemented as a set of resource records. In other words, a name server stores a database of resource records. A resource record is a 5-tuple structure, as shown below:

(Domain Name, Type, Class, TTL, Value)

- The domain name field is what identifies the resource record.
- The value defines the information kept about the domain name.
- The TTL defines the number of seconds for which the information is valid.
- The class defines the type of network class address
- The type defines how the value should be interpreted.

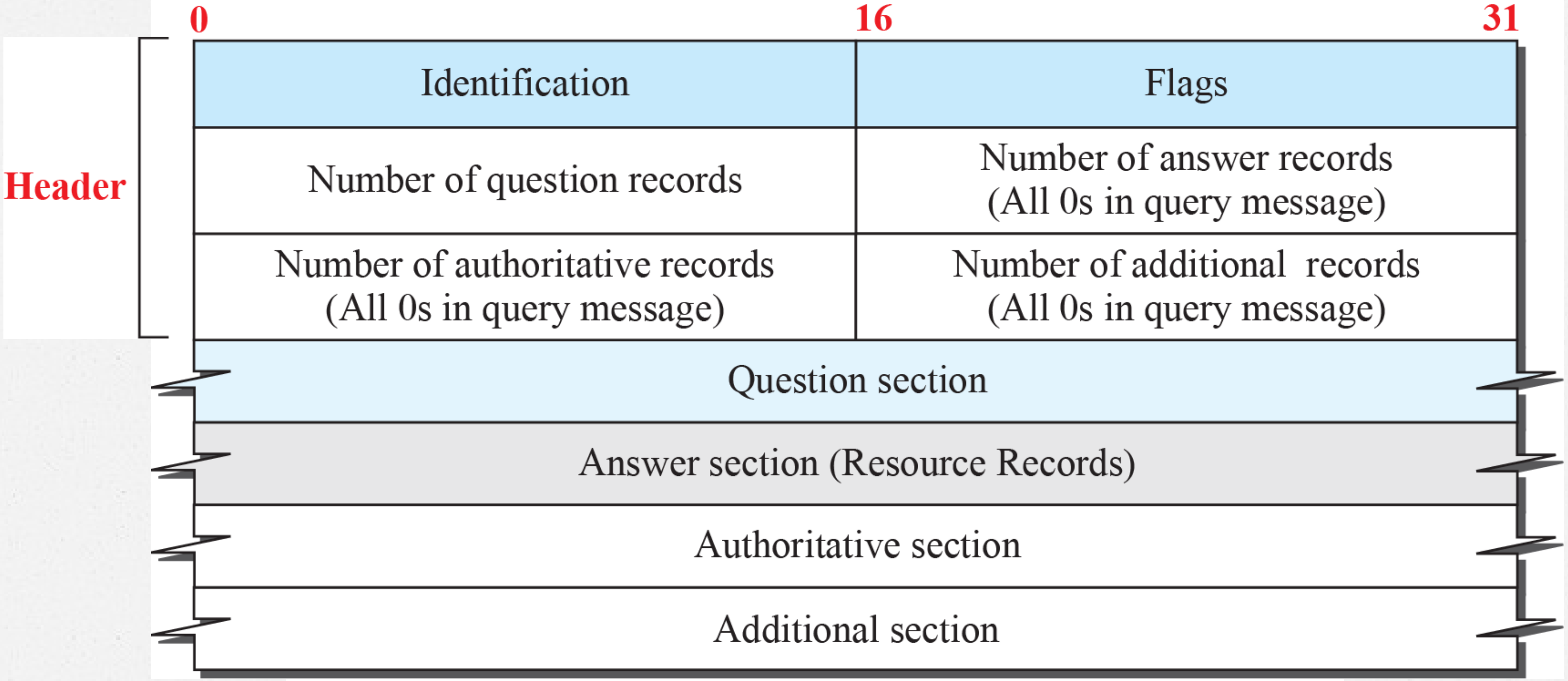
DNS types



Type	*Interpretation of value*
A	A 32-bit IPv4 address (see Chapter 4)
NS	Identifies the authoritative servers for a zone
CNAME	Defines an alias for the official name of a host
SOA	Marks the beginning of a zone
MX	Redirects mail to a mail server
AAAA	An IPv6 address (see Chapter 4)

DNS message

To retrieve information about hosts, DNS uses two types of messages: query and response. Both types have the same format



Identification	Flags
Number of question records	Number of answer records (All 0s in query message)
Number of authoritative records (All 0s in query message)	Number of additional records (All 0s in query message)
Question section	
Answer section (Resource Records)	
Authoritative section	
Additional section	

Note:

The query message contains only the question section. The response message includes the question section, the answer section, and possibly two other sections.

DNS message

- The **identification field** is used by the client to match the response with the query.
- The **flag field** defines whether the message is a query or response. It also includes status of error.
- The next four fields in the header define the number of each record type in the message.
- The **question section**, which is included in the query and repeated in the response message, consists of one or more question records. It is present in both query and response messages.
- The **answer section** consist of one or more resource records. It is present only in response messages.
- The **authoritative section** gives information (domain name) about one or more authoritative servers for the query.
- The **additional information section** provides additional information that may help the resolver

Encapsulation

- DNS can use either UDP or TCP. In both cases the well-known port used by the server is **port 53**.
- UDP is used when the size of the response message is less than 512 bytes because most UDP packages have a 512-byte packet size limit.
- If the size of the response message is more than 512 bytes, a TCP connection is used. In that case, one of two scenarios can occur:
  - If the resolver has prior knowledge that the size of the response message is more than 512 bytes, it uses the TCP connection. For example, if a secondary name server (acting as a client) needs a zone transfer from a primary server.
  - If the resolver does not know the size of the response message, it can use the UDP port. However, if the size of the response message is more than 512 bytes, the server truncates the message and turns on the TC bit. The resolver now opens a TCP connection and repeats the request to get a full response from the server.

Registrars

- New domains added to DNS through **a registrar**, a commercial entity accredited by ICANN.
- A registrar first verifies that the requested domain name is unique and then enters it into the DNS database. A fee is charged.
- To register, the organization needs to give the name of its server and the IP address of the server.

Dynamic Domain Name System (DDNS)

- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file. These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation. The DNS master file must be updated dynamically. The **Dynamic Domain Name System (DDNS)** therefore was devised to respond to this need.
- The primary server updates the zone. The secondary servers are notified either actively or passively.
    - In **active notification**, the primary server sends a message to the secondary servers about the change in the zone, whereas
    - in **passive notification**, the secondary servers periodically check for any changes.

In either case, after being notified about the change, the secondary server requests information about the entire zone (called the **zone transfer**).

Security of DNS

- DNS is one of the most important systems in the Internet infrastructure; it provides crucial services to Internet users.
- DNS can be attacked in several ways: flooding, message interception,
- To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides message origin authentication and message integrity using a security service called digital signature.
- DNSSEC, however, does not provide confidentiality for the DNS messages.
- There is no specific protection against the denial-of service attack in the specification of DNSSEC. However, the caching system protects the upper-level servers against this attack to some extent.

Chapter 2: Summary

- ***Applications in the Internet are designed using either a client-server paradigm or a peer-to-peer paradigm. In a client-server paradigm, an application program, called a server, provides services and another application program, called a client, receives services. A server program is an infinite program; a client program is finite. In a peer-to- peer paradigm, a peer can be both a client and a server.***

- ***The World Wide Web (WWW) is a repository of information linked together from points all over the world. Hypertext and hypermedia documents are linked to one another through pointers. The HyperText Transfer Protocol (HTTP) is the main protocol used to access data on the World Wide Web (WWW).***

Chapter 2: Summary (continued)

- ***File Transfer Protocol (FTP) is a TCP/IP client-server application for copying files from one host to another. FTP requires two connections for data transfer: a control connection and a data connection. FTP employs NVT ASCII for communication between dissimilar systems.***

- ***Electronic mail is one of the most common applications on the Internet. The e-mail architecture consists of several components such as user agent (UA), main transfer agent (MTA), and main access agent (MAA). The protocol that implements MTA is called Simple Main Transfer Protocol (SMTP). Two protocols are used to implement MAA: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).***

Chapter 2: Summary (continued)

- ***File Transfer Protocol (FTP) is a TCP/IP client-server application for copying files from one host to another. FTP requires two connections for data transfer: a control connection and a data connection. FTP employs NVT ASCII for communication between dissimilar systems.***

- ***TELNET is a client-server application that allows a user to log into a remote machine, giving the user access to the remote system. When a user accesses a remote system via the TELNET process, this is comparable to a time-sharing environment.***

Chapter 2: Summary (continued)

- ***The Domain Name System (DNS) is a client-server application that identifies each host on the Internet with a unique name. DNS organizes the name space in a hierarchical structure to decentralize the responsibilities involved in naming. TELNET is a client-server application that allows a user to log into a remote machine, giving the user access to the remote system.***