Chapter 17

Domain Name System: DNS

Objectives

Upon completion you will be able to:

- ***Understand how the DNS is organized***
- ***Know the domains in the DNS***
- ***Know how a name or address is resolved***
- ***Be familiar with the query and response formats***
- ***Understand the need for DDNS***

17.1 NAME SPACE

The names assigned to machines 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.

The topics discussed in this section include:

Flat Name Space
Hierarchical Name Space

17.2 DOMAIN NAME SPACE

The domain name space is hierarchical in design. The names are defined in an inverted-tree structure with the root at the top. The tree can have 128 levels: level 0 (root) to level 127.

The topics discussed in this section include:

Label
Domain Name
Domain

Figure 17.1 ***Domain name space***

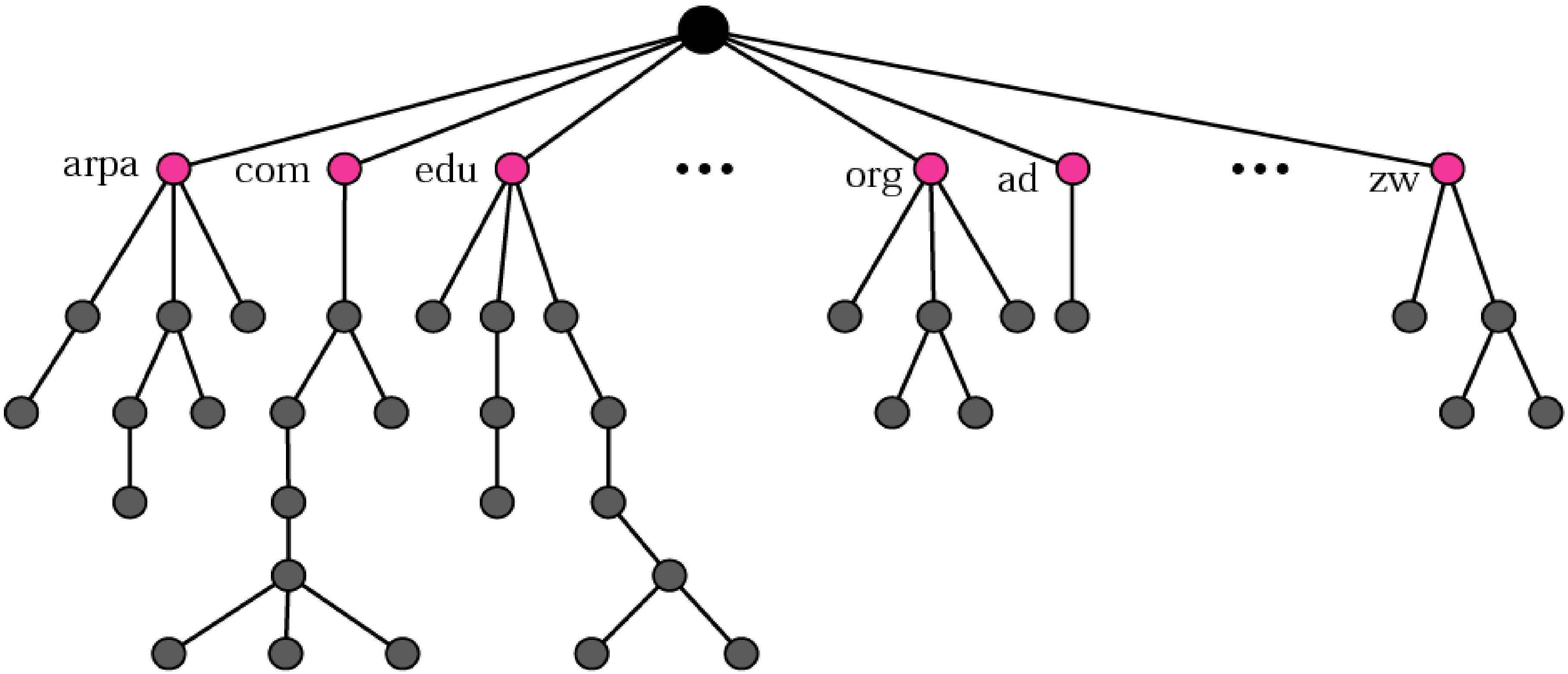


Figure 17.2 *Domain names and labels*

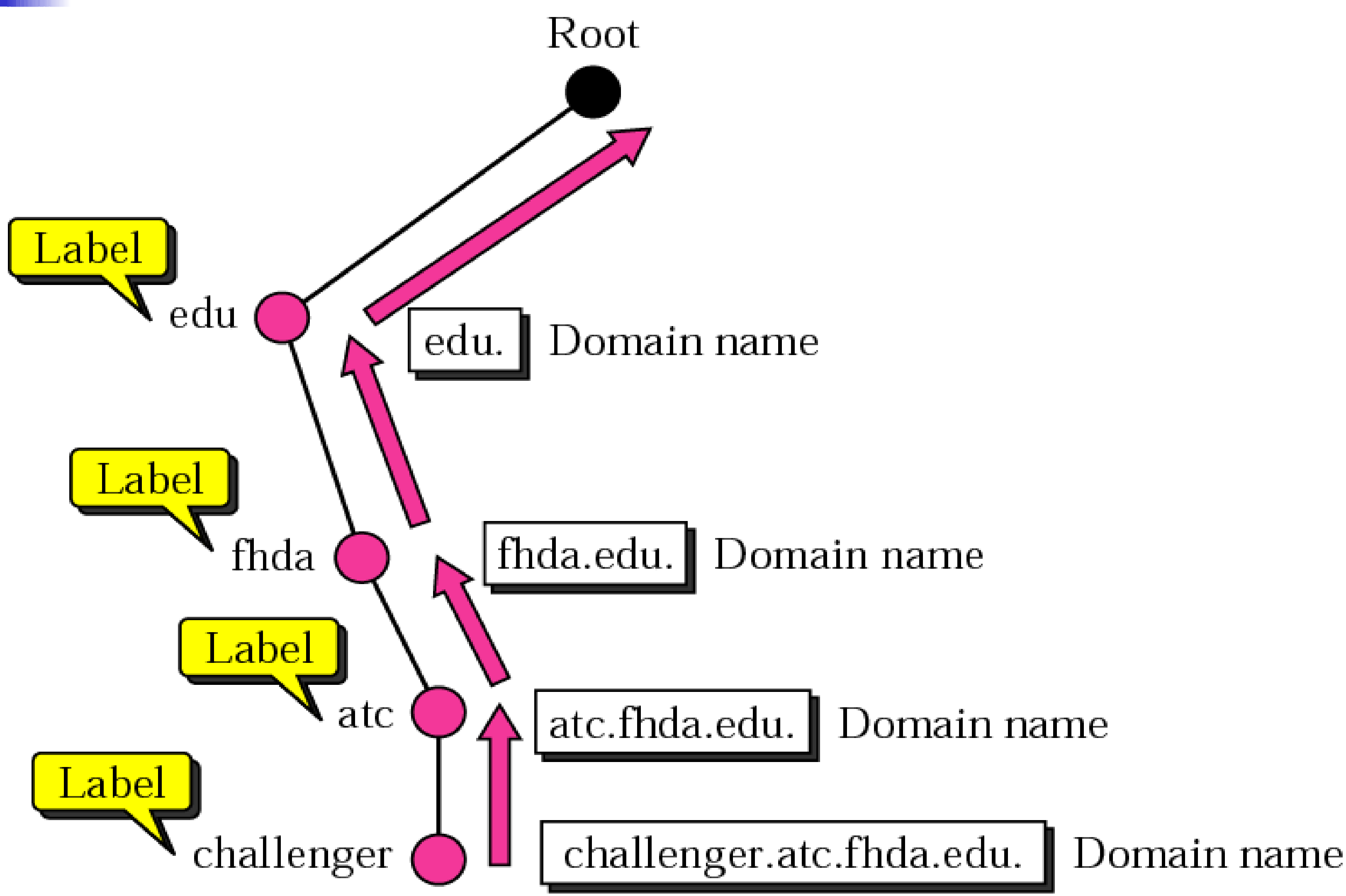
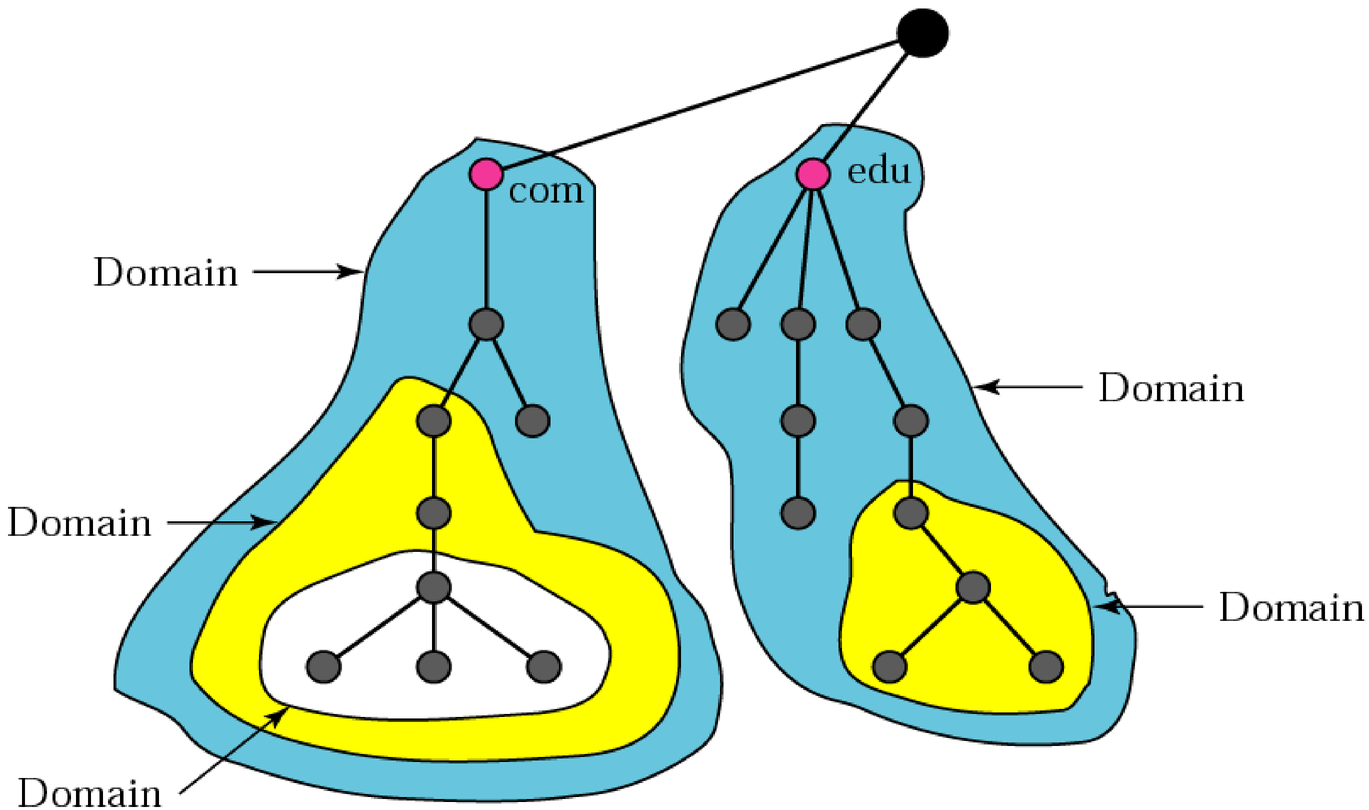


Figure 17.3 *FQDN and PQDN*

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

Figure 17.4 ***Domains***



17.3 DISTRIBUTION OF NAME SPACE

The information contained in the domain name space is distributed among many computers called DNS servers.

The topics discussed in this section include:

Hierarchy of Name Servers
Zone
Root Server
Primary and Secondary Servers

Figure 17.5 ***Hierarchy of name servers***

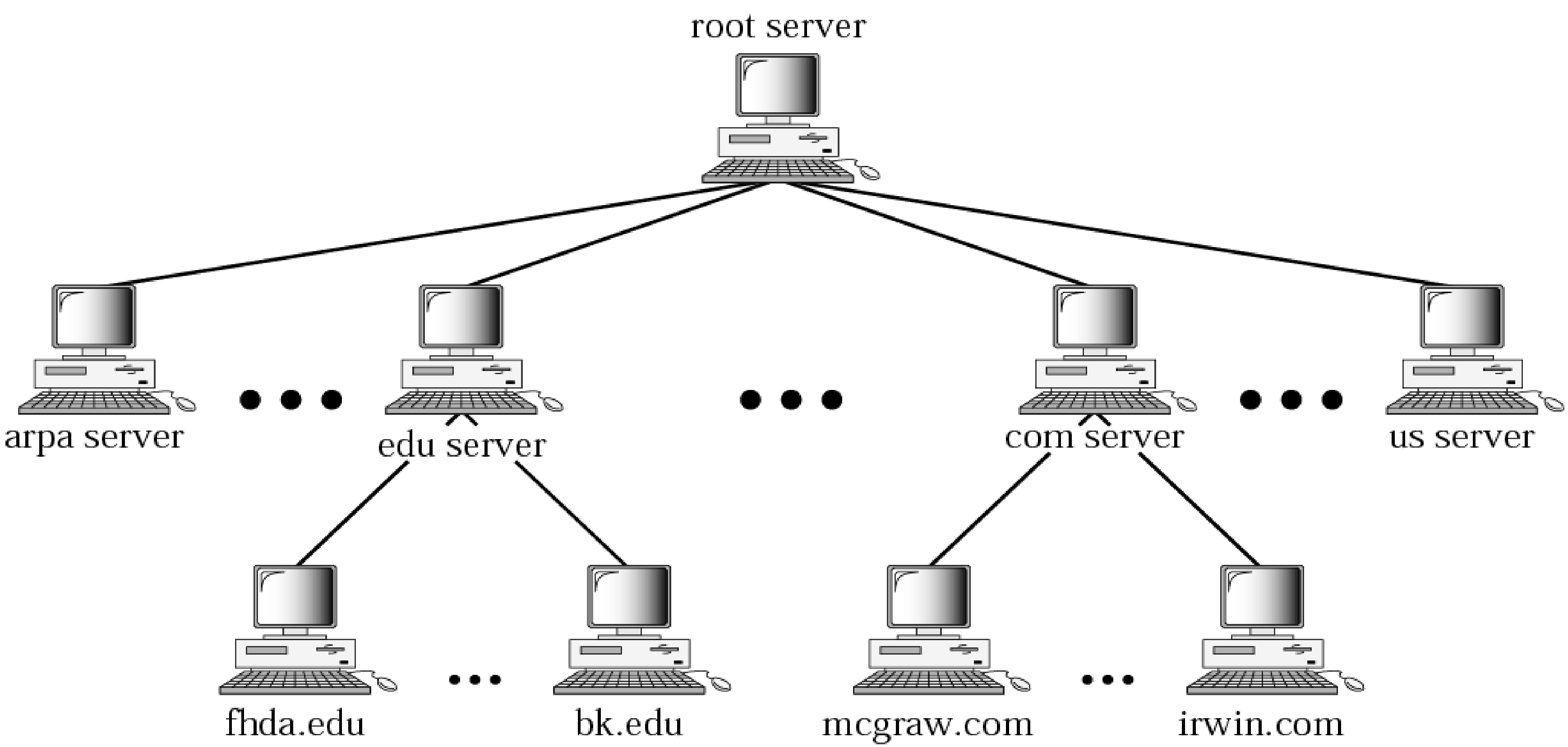
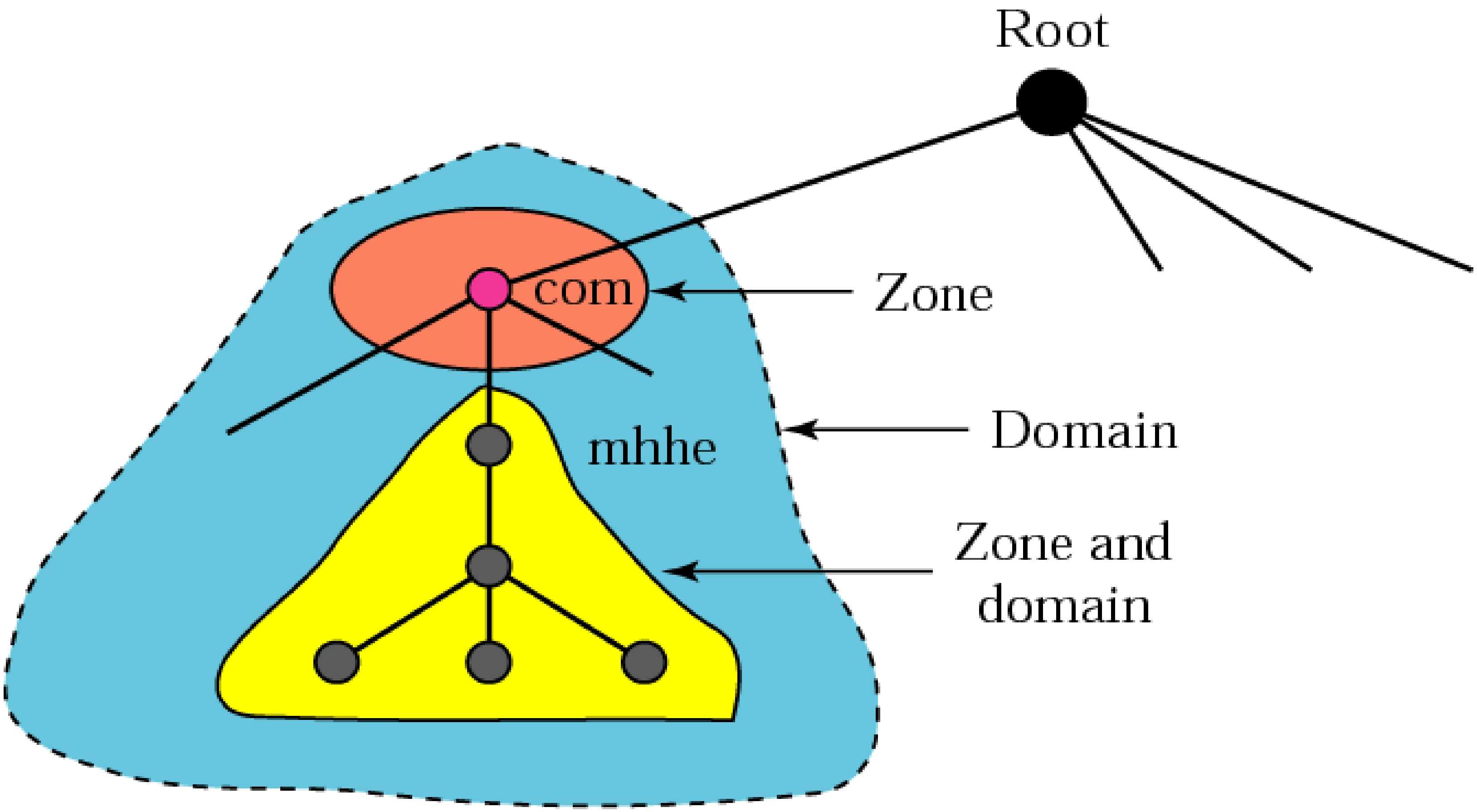
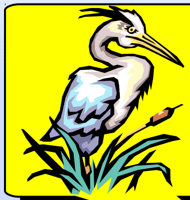


Figure 17.6 ***Zones and domains***





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.

17.4 DNS IN THE INTERNET

The domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.

The topics discussed in this section include:

Generic Domains
Country Domains
Inverse Domain
Registrar

Figure 17.7 *DNS used in the Internet*

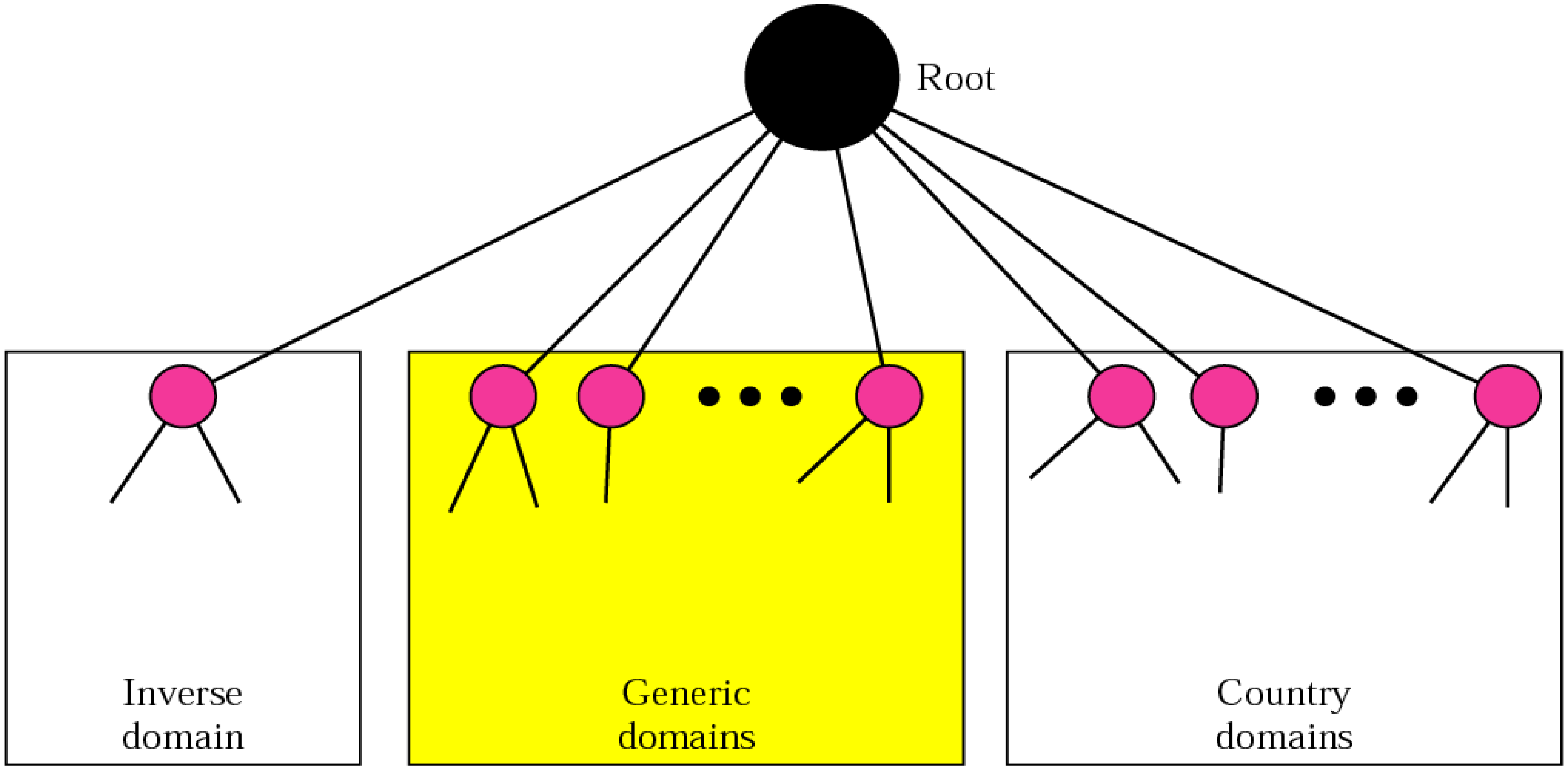


Figure 17.8 *Generic domains*

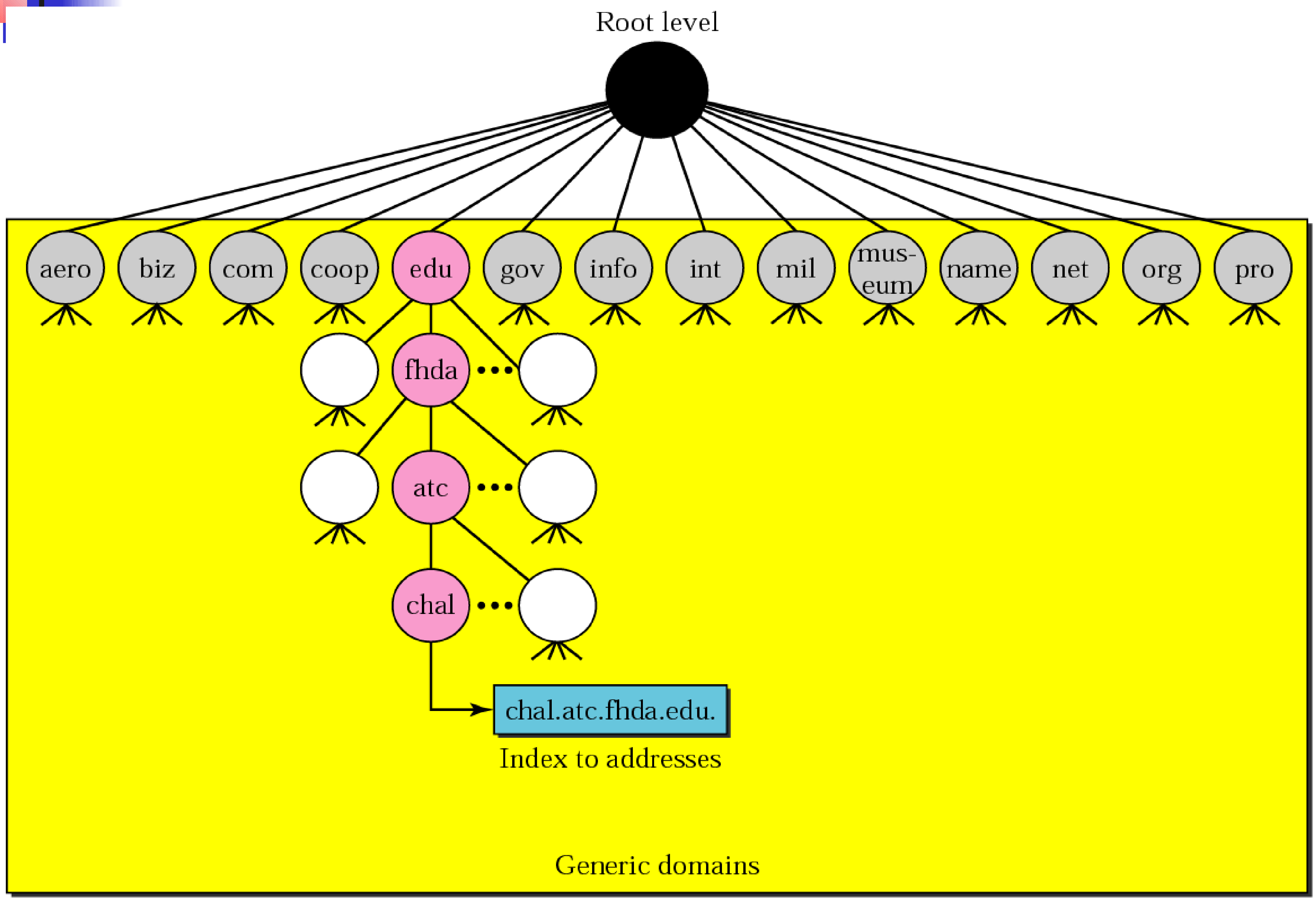


Table 17.1 ***Generic domain labels***

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

Table 17.1 Generic domain labels (Continued)

Label	*Description*
int	International organizations
mil	Military groups
museum	Museums and other non-profit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Figure 17.9 ***Country domains***

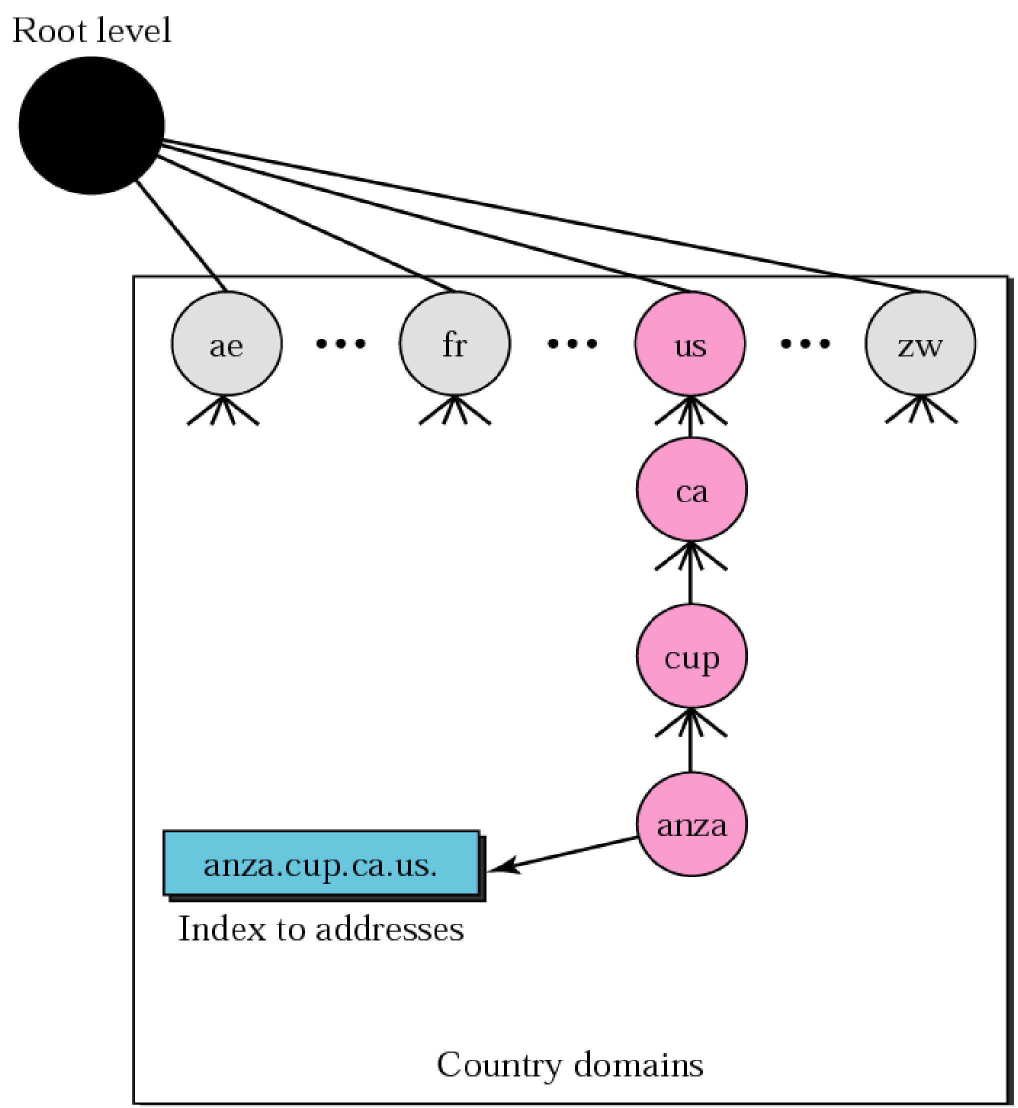
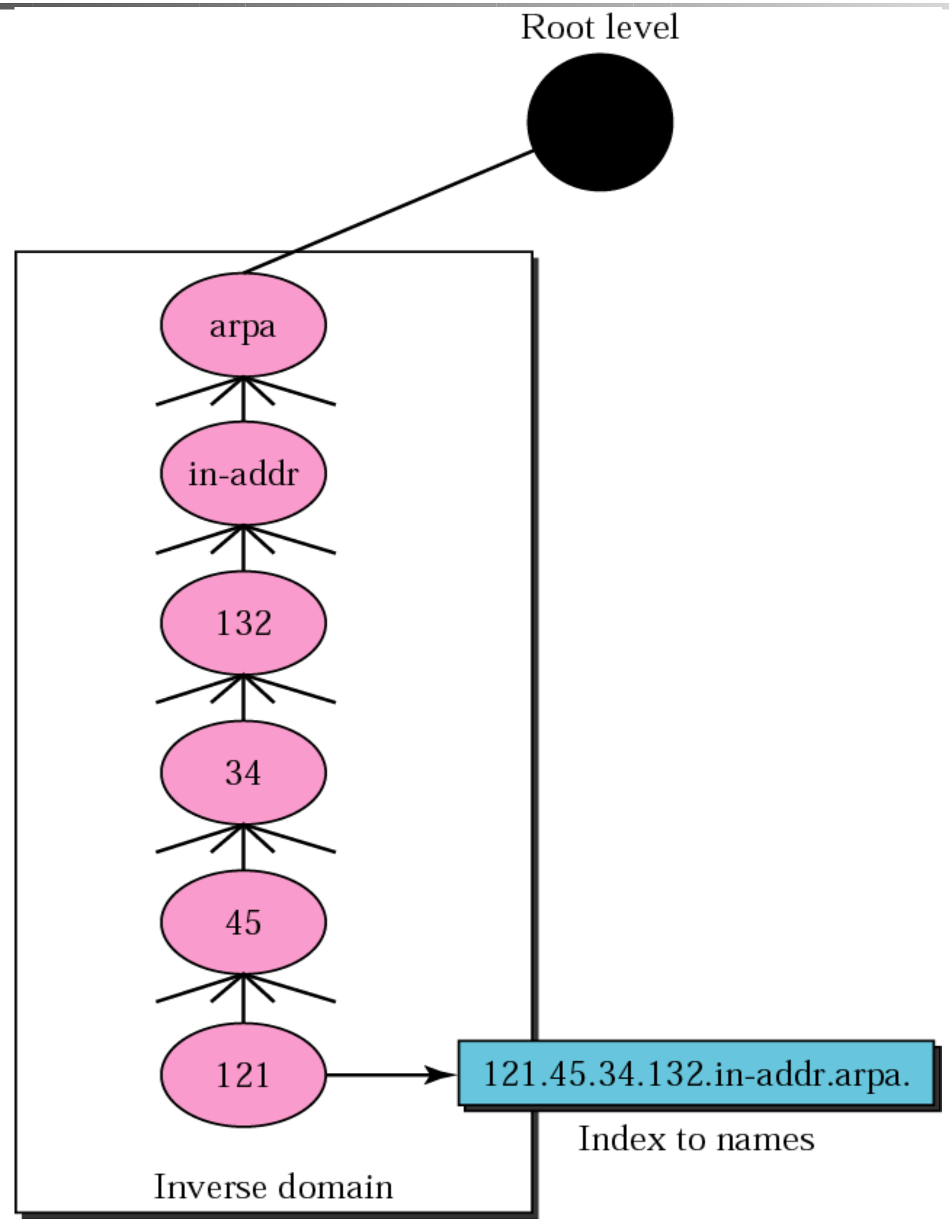


Figure 17.10 *Inverse domain*



17.5 RESOLUTION

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

The topics discussed in this section include:

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

Figure 17.11 *Recursive resolution*

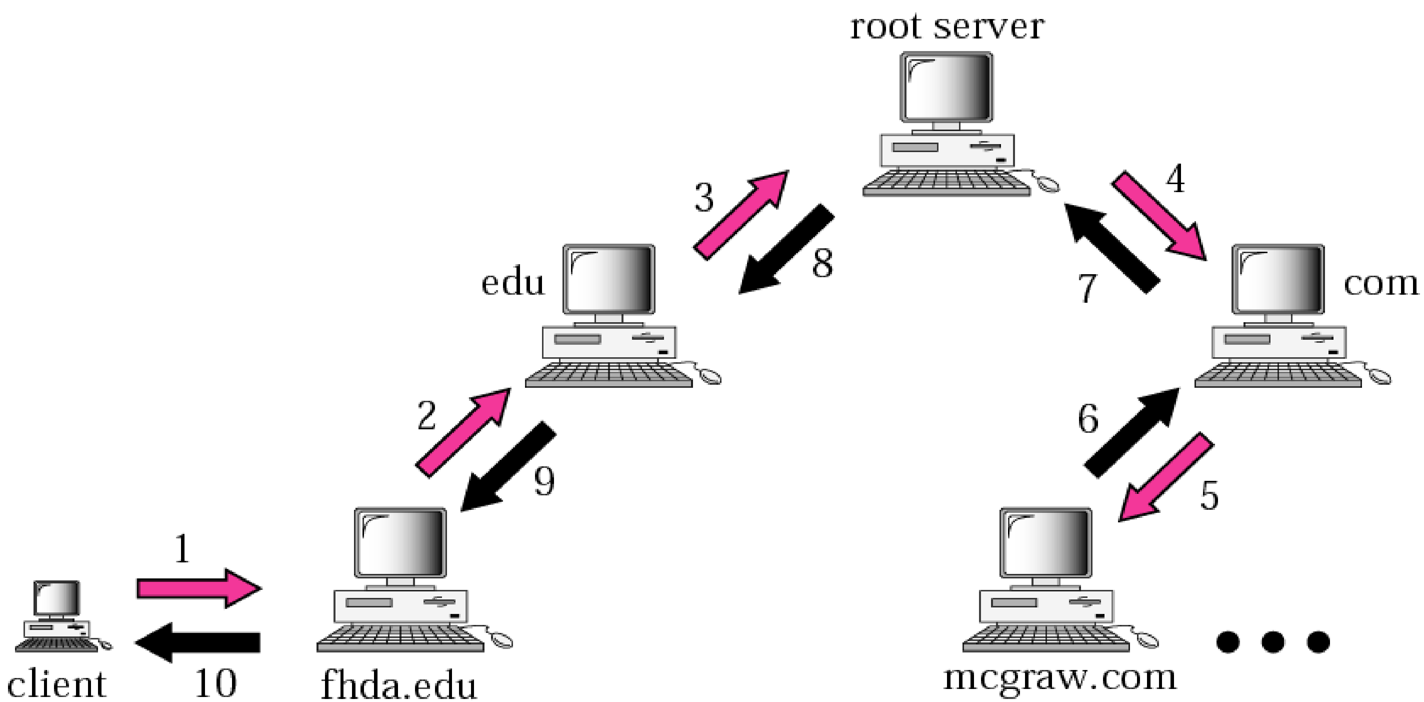
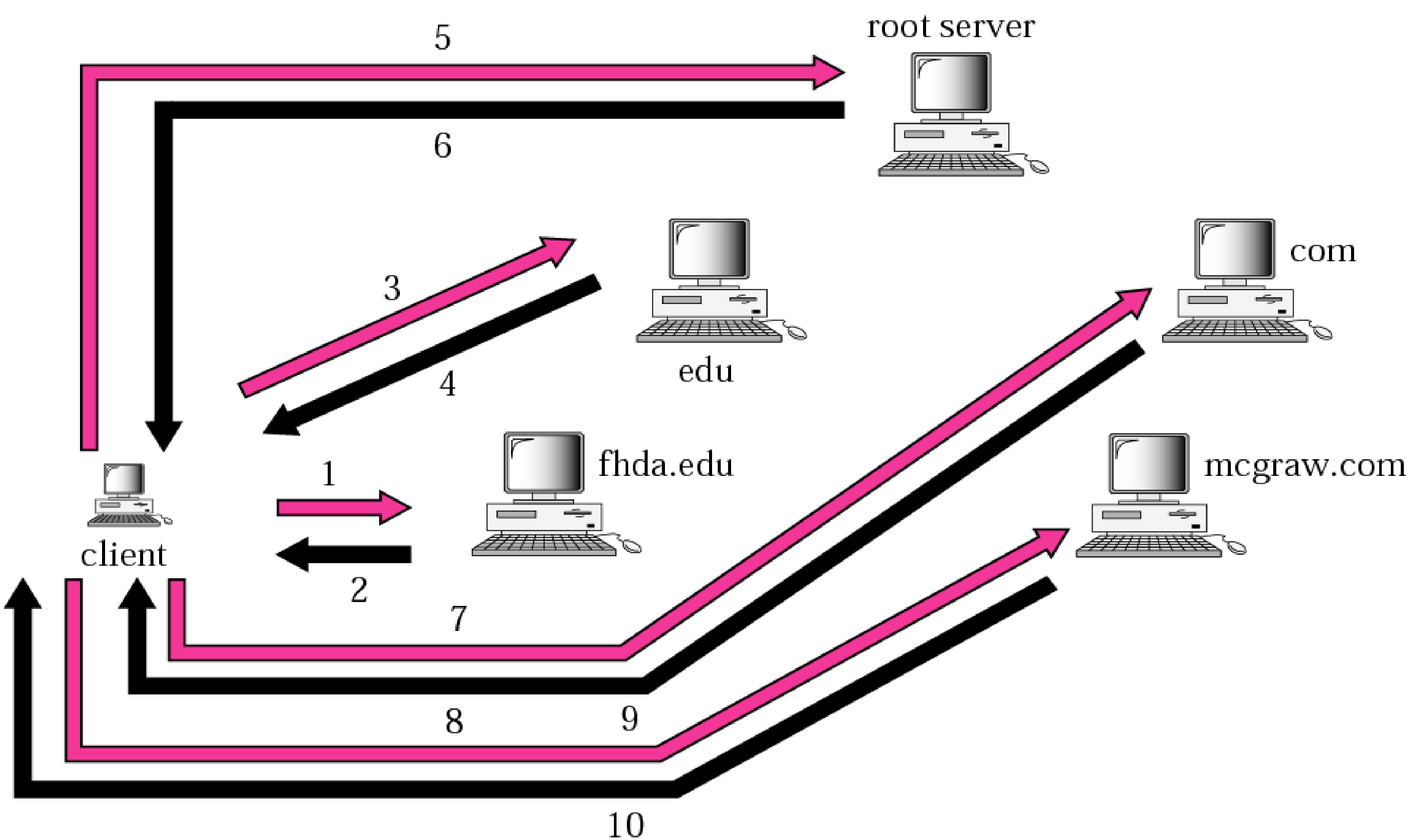


Figure 17.12 ***Iterative resolution***



17.6 DNS MESSAGES

The DNS query message consists of a header and question records; the DNS response message consists of a header, question records, answer records, authoritative records, and additional records.

The topics discussed in this section include:

Header

Figure 17.13 ***DNS messages***

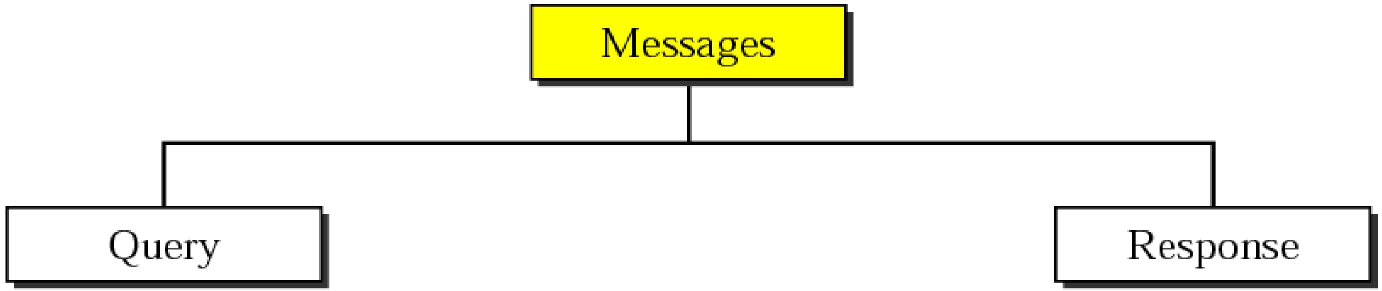


Figure 17.14 *Query and response messages*

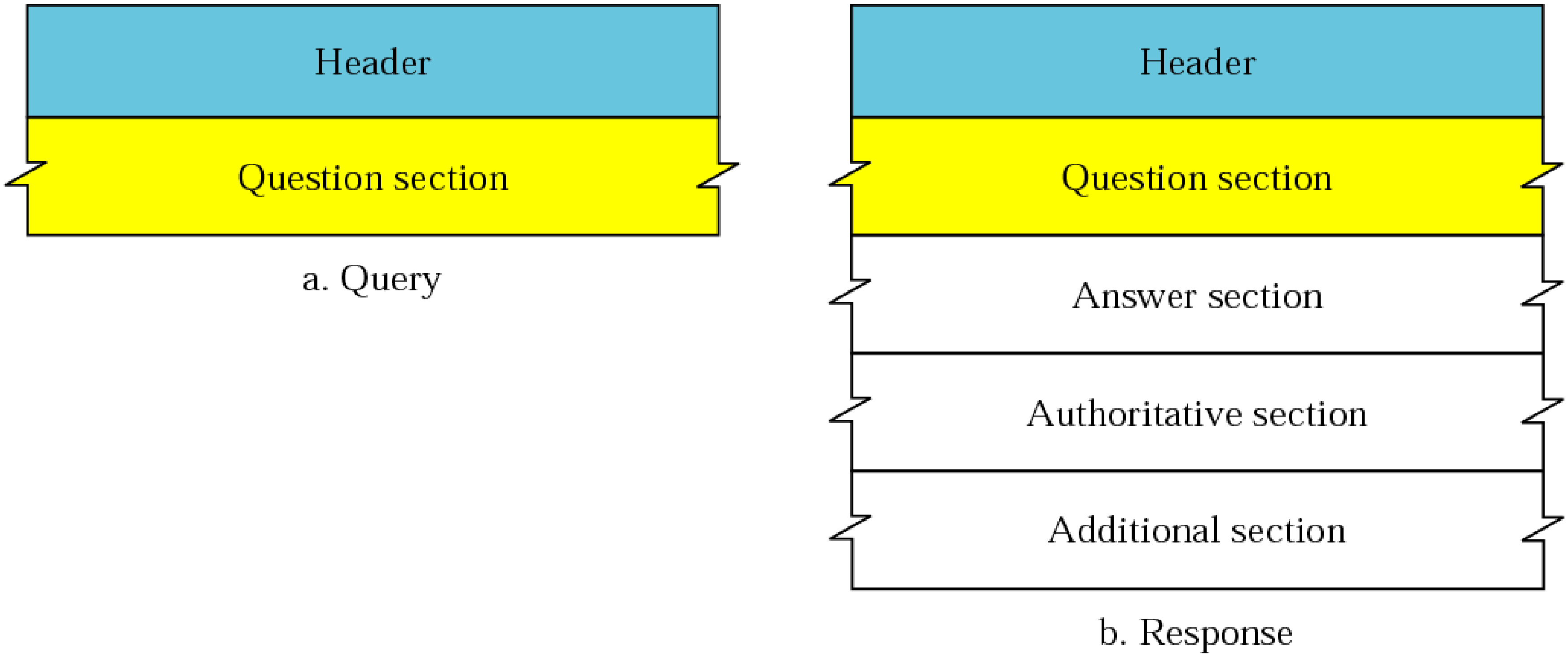


Figure 17.15 *Header format*

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)

Figure 17.16 ***Flags field***

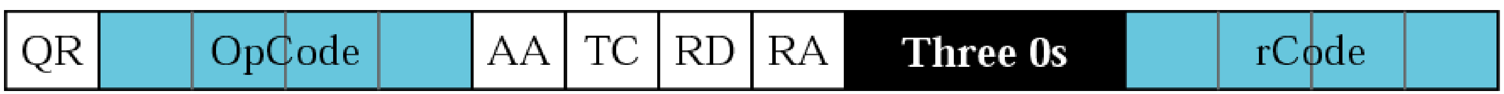


Table 17.2 Values of rCode

Value	*Meaning*
0	No error
1	Format error
2	Problem at name server
3	Domain reference problem
4	Query type not supported
5	Administratively prohibited
6–15	Reserved

17.7 TYPES OF RECORDS

Two types of records are used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative, and additional information sections of the response message.

The topics discussed in this section include:

Question Record
Resource Record

Figure 17.17 *Question record format*

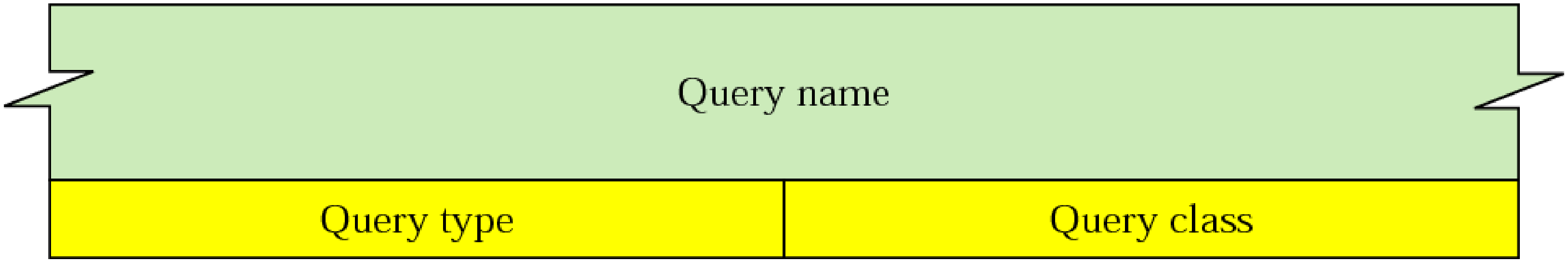


Figure 17.18 ***Query name format***

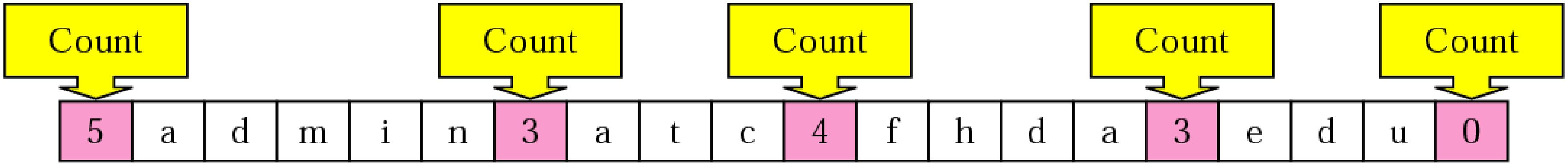


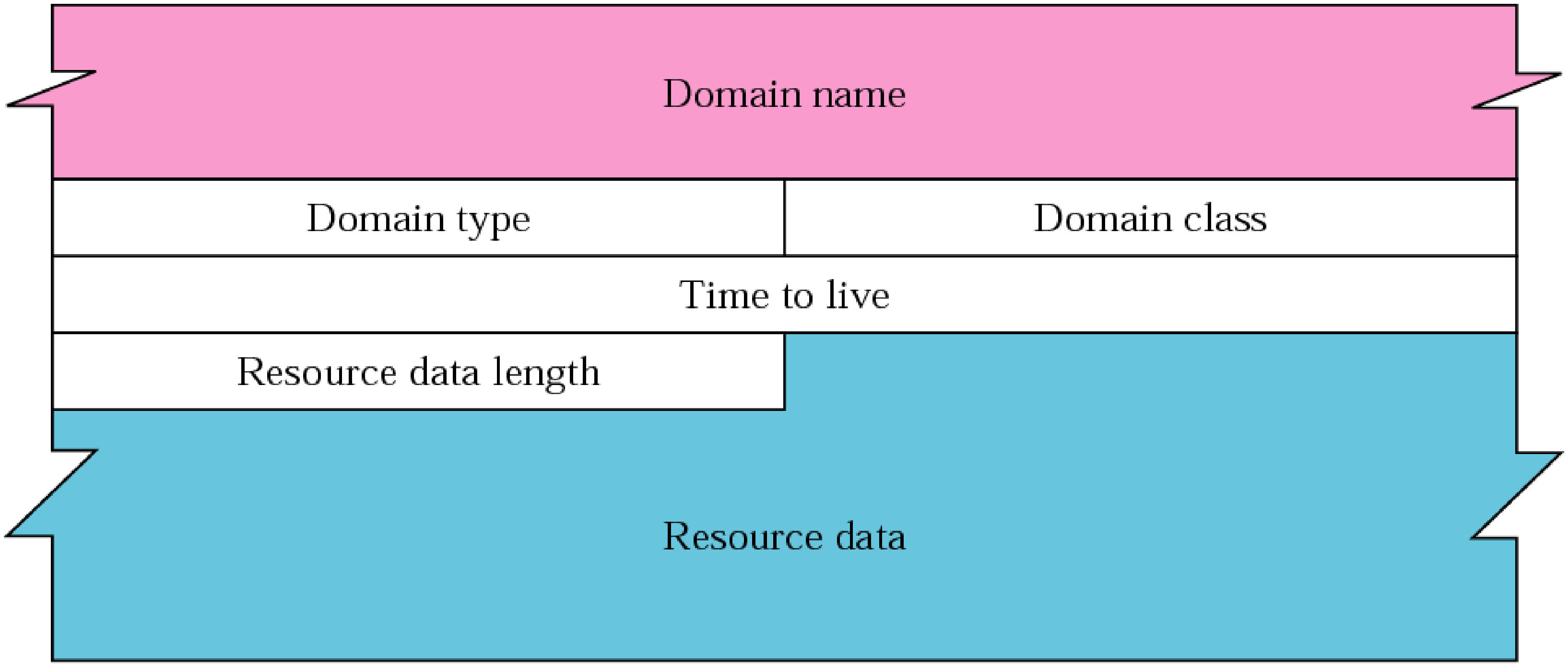
Table 17.3 Types

Type	*Mnemonic*	*Description*
1	A	Address. A 32-bit IPv4 address. It is used to convert a domain name to an IPv4 address.
2	NS	Name server. It identifies the authoritative servers for a zone.
5	CNAME	Canonical name. It defines an alias for the official name of a host.
6	SOA	Start of authority. It marks the beginning of a zone. It is usually the first record in a zone file.
11	WKS	Well-known services. It defines the network services that a host provides.
12	PTR	Pointer. It is used to convert an IP address to a domain name.
13	HINFO	Host information. It gives the description of the hardware and the operating system used by a host.
15	MX	Mail exchange. It redirects mail to a mail server.
28	AAAA	Address. An IPv6 address (see Chapter 27).
252	AXFR	A request for the transfer of the entire zone.
255	ANY	A request for all records.

Table 17.4 Classes

Class	*Mnemonic*	*Description*
1	IN	Internet
2	CSNET	CSNET network (obsolete)
3	CS	The COAS network
4	HS	The Hesiod server developed by MIT

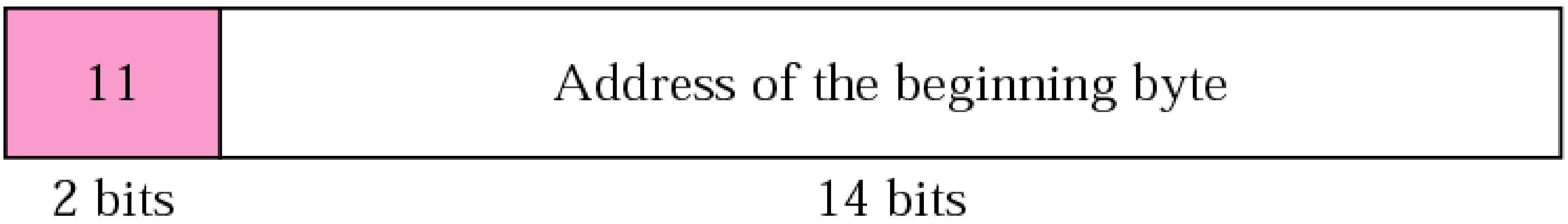
Figure 17.19 *Resource record format*



17.8 COMPRESSION

DNS requires that a domain name be replaced by an offset pointer if it is repeated. DNS defines a 2-byte offset pointer that points to a previous occurrence of the domain name or part of it.

Figure 17.20 *Format of an offset pointer*



Example 1

A resolver sends a query message to a local server to find the IP address for the host "chal.fhda.edu.". We discuss the query and response messages separately.

Figure 17.21 shows the query message sent by the resolver. The first 2 bytes show the identifier (1333). It is used as a sequence number and relates a response to a query. Because a resolver may even send many queries to the same server, the identifier helps to sort responses that arrive out of order. The next bytes contain the flags with the value of 0x0100 in hexadecimal. In binary it is 0000000100000000, but it is more meaningful to divide it into the fields as shown below:

QR	OpCode	AA	TC	RD	RA	Reserved	rCode
0	0000	0	0	1	0	000	0000

Figure 17.21 ***Example 1: Query message***

0x1333		0x0100	
1		0	
0		0	
4	‘c’	‘h’	‘a’
‘l’	**4**	‘f’	‘h’
‘d’	‘a’	**3**	‘e’
‘d’	‘u’	**0**	Continued on next line
1	1		

Example 1 *(Continued)*

The QR bit defines the message as a query. The OpCode is 0000, which defines a standard query. The recursion desired (RD) bit is set. (Refer back to Figure 17.16 for the flags field descriptions.) The message contains only one question record. The domain name is 4chal4fhda3edu0. The next 2 bytes define the query type as an IP address; the last 2 bytes define the class as the Internet.

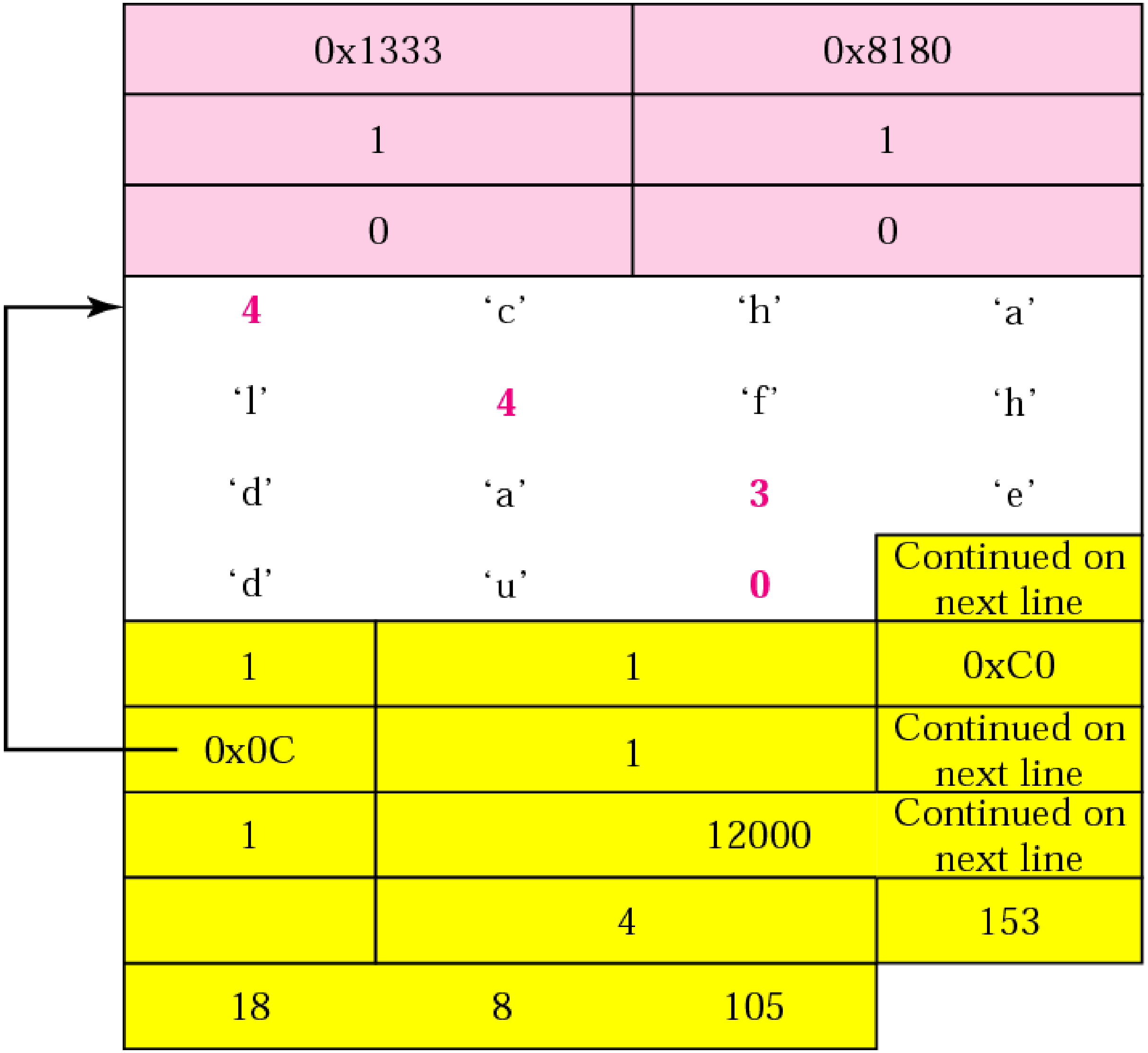
Figure 17.22 shows the response of the server. The response is similar to the query except that the flags are different and the number of answer records is one. The flags value is 0x8180 in hexadecimal. In binary it is 1000000110000000, but again we divide it into fields as shown below:

QR	OpCode	AA	TC	RD	RA	Reserved	rCode
1	0000	0	0	1	1	000	0000

Example 1 (Continued)

The QR bit defines the message as a response. The OpCode is 0000, which defines a standard response. The recursion available (RA) and RD bits are set. The message contains one question record and one answer record. The question record is repeated from the query message. The answer record has a value of 0xC00C (split in two lines), which points to the question record instead of repeating the domain name. The next field defines the domain type (address). The field after that defines the class (Internet). The field with the value 12,000 is the TTL (12,000 s). The next field is the length of the resource data, which is an IP address (153.18.8.105).

Figure 17.22 *Example 1: Response message*



Example 2

An FTP server has received a packet from an FTP client with IP address 153.2.7.9. The FTP server wants to verify that the FTP client is an authorized client. The FTP server can consult a file containing the list of authorized clients. However, the file consists only of domain names. The FTP server has only the IP address of the requesting client, which was the source IP address in the received IP datagram. The FTP server asks the resolver (DNS client) to send an inverse query to a DNS server to ask for the name of the FTP client. We discuss the query and response messages separately.

Example 2 (Continued)

Figure 17.23 shows the query message sent from the resolver to the server. The first 2 bytes show the identifier (0x1200). The flags value is 0x0900 in hexadecimal. In binary it is 0000100100000000, and we divide it into fields as shown below:

QR	OpCode	AA	TC	RD	RA	Reserved	rCode
0	0001	0	0	1	0	000	0000

The OpCode is 0001, which defines an inverse query. The message contains only one question record. The domain name is 19171231537in-addr4arpa. The next 2 bytes define the query type as PTR, and the last 2 bytes define the class as the Internet.

Figure 17.23 *Example 2: Inverse query message*

0x1200		0x0900	
1		0	
0		0	
1	'9'	**1**	'7'
1	'2'	**3**	'1'
'5'	'3'	**7**	'i'
'n'	'-'	'a'	'd'
'd'	'r'	**4**	'a'
'r'	'p'	'a'	**0**
12		1	

Example 2 ***(Continued)***

Figure 17.24 shows the response. The flags value is 0x8D80 in hexadecimal. In binary it is 1000110110000000, and we divide it into fields as shown below:

QR	***OpCode***	***AA***	***TC***	***RD***	***RA***	***Reserved***	***rCode***
1	***0001***	***1***	***0***	***1***	***1***	***000***	***0000***

Figure 17.24 *Example 2: Inverse response message*

0x1200		0x8D80	
1		1	
0		0	
1	‘9’	**1**	‘7’
1	‘2’	**3**	‘l’
‘5’	‘3’	**7**	‘i’
‘n’	‘-’	‘a’	‘d’
‘d’	‘r’	**4**	‘a’
‘r’	‘p’	‘a’	**0**
12		1	
0xC00C		12	
1		Continued on next line	
24000		10	
4	‘m’	‘h’	‘h’
‘e’	**3**	‘c’	‘o’
‘m’	**0**		

Example 3

In UNIX and Windows, the nslookup utility can be used to retrieve address/name mapping. The following shows how we can retrieve an address when the domain name is given.

```
$ nslookup fhda.edu
Name: fhda.edu
Address: 153.18.8.1
```

The nslookup utility can also be used to retrieve the domain name when the address is given as shown below:

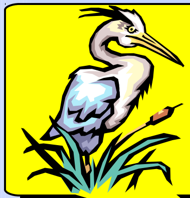
```
$ nslookup 153.18.8.1
1.8.18.153.in-addr.arpa name = tiptoe.fhda.edu.
```

17.9 DDNS

The Dynamic Domain Name System (DDNS) updates the DNS master file dynamically.

17.10 ENCAPSULATION

DNS uses UDP as the transport protocol when the size of the response message is less than 512 bytes. If the size of the response message is more than 512 bytes, a TCP connection is used.



Note:

DNS can use the services of UDP or TCP using the well-known port 53.