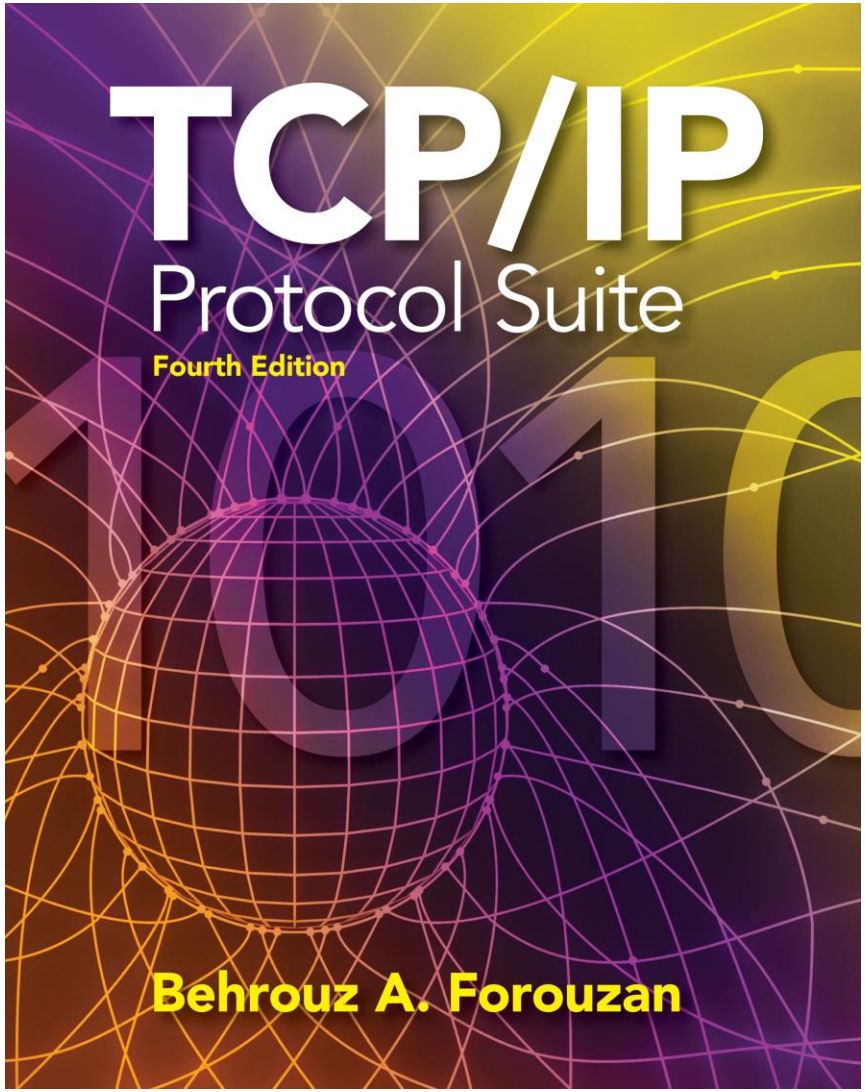
Chapter 19

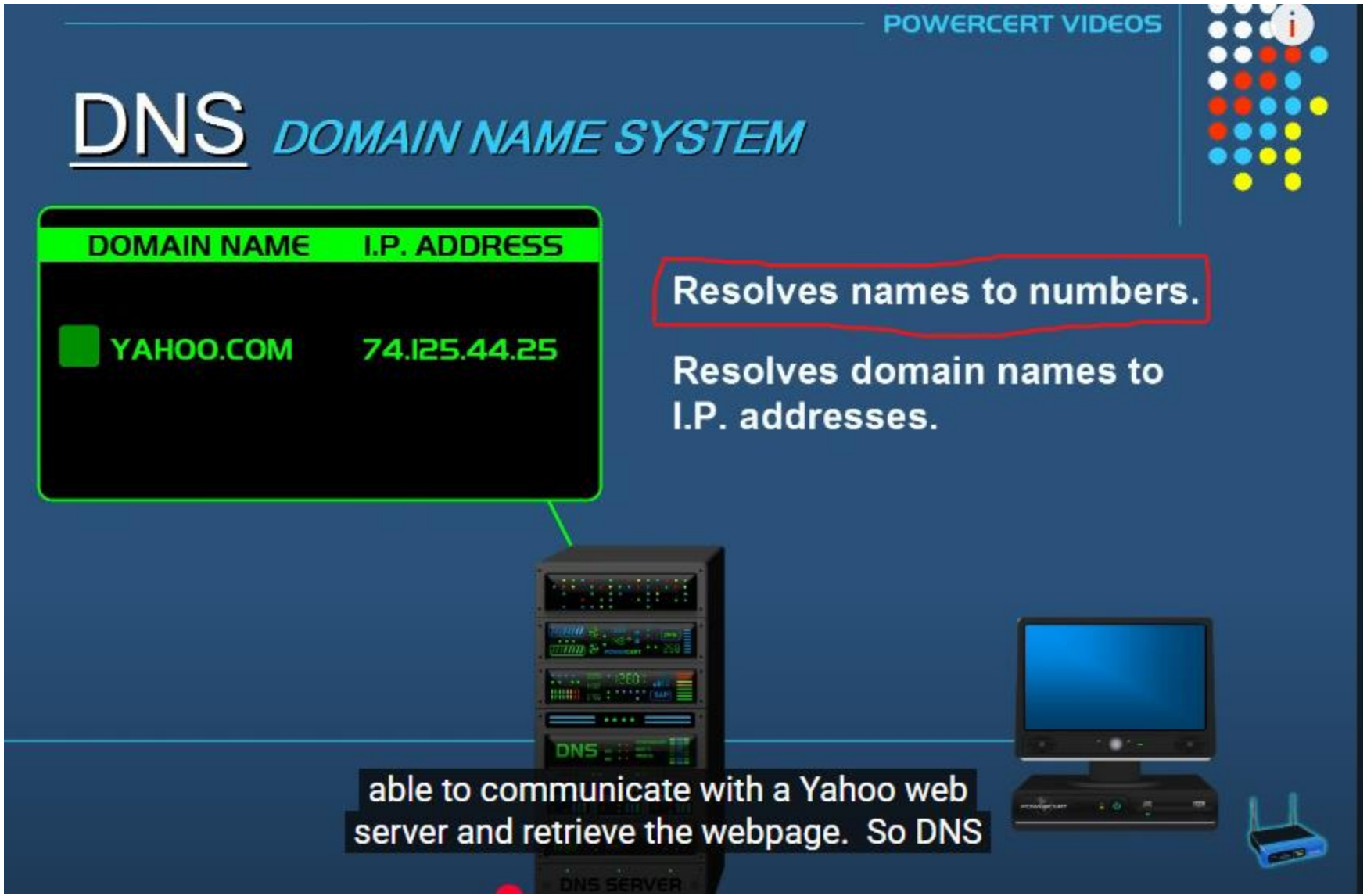
Domain Name System (DNS)



OBJECTIVES:

- To describe the purpose of DNS.
- To define the concept of domains and domain name space.
- To describe the distribution of name spaces and define zones.
- To discuss the use of DNS in the Internet and describe three categories of domains: generic, country, and reverse.
- To discuss name-address resolution and show the two resolution methods: recursive and iterative.
- To show the format of DNS message and how they can be compressed.
- To discuss DDNS and DNSSEC..

Chapter Outline


POWERCERT VIDEOS
DNS
DOMAIN NAME SYSTEM
DOMAIN NAME
I.P. ADDRESS
YAHOO.COM
74.125.44.25
Resolves names to numbers.
Resolves domain names to I.P. addresses.
DNS
DNS SERVER
able to communicate with a Yahoo web server and retrieve the webpage. So DNS

🔹 উদাহরণ দিয়ে ব্যাখ্যা:

ধরি, আপনি ব্রাউজারে লিখলেন:

```
www.pstu.ac.bd
```

তাহলে ধাপে ধাপে যা ঘটে:

1. আপনার কম্পিউটার প্রথমে **Root DNS Server**-কে জিজ্ঞেস করে `.bd` কোথায় আছে।
2. তারপর **TLD Server** বলে দেয় `.bd` এর জন্য দায়ী সার্ভার।
3. এরপর, আপনার কম্পিউটার **Authoritative Name Server**-এর কাছে যায়, যেটা **pstu.ac.bd** এর জন্য **সঠিক IP address** জানে।

এই **Authoritative Name Server**-ই নিশ্চিত করে বলে দেয়:

> "`www.pstu.ac.bd` -এর IP address হলো: `123.45.67.89`"

🔸 একটি সহজ উদাহরণ চিত্র (Text Version):

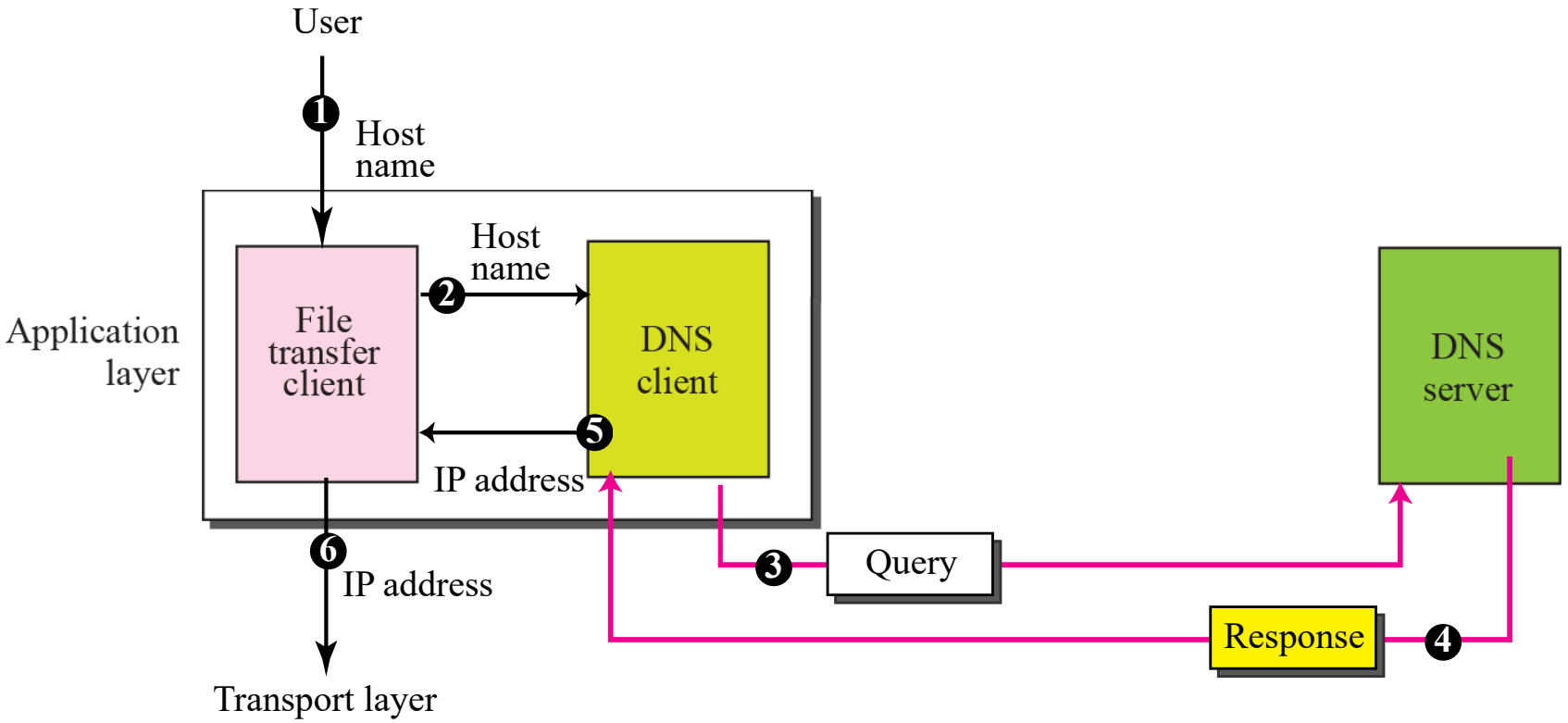
```
আপনি টাইপ করলেন: www.pstu.ac.bd

➡ Root Server → .bd TLD Server → ac.bd Server →
➡ Authoritative Server → দেয়: "IP is 203.112.5.10"
```

19-1 NEED FOR 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, we need a system that can map a name to an address or an address to a name.**

Figure 19.1 *Purpose of DNS*



19-2 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. In other words, 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.

Saved memory full ⓘ

🔹 Flat vs Hierarchical Name Space:

বৈশিষ্ট্য	Flat Name Space	Hierarchical Name Space
কাঠামো	কোনো স্তর নেই, সব নাম একসাথে থাকে	নামগুলো বিভিন্ন স্তরে বিভক্ত (উদাহরণ: ডোমেইন নাম)
উদাহরণ	MAC Address (যেমন: `00:0a:95:9d:68:16`)	DNS (যেমন: `www.pstu.ac.bd`)
অনুসন্ধান	কঠিন, দ্রুত হয় না	সহজ, দ্রুত ও সংগঠিত
নিয়ন্ত্রণ	কেন্দ্রীয়ভাবে নিয়ন্ত্রিত হতে হয়	বিভাজিত নিয়ন্ত্রণ সম্ভব

🔸 সারাংশ:

যেকোনো মেশিন বা সার্ভারের নাম এমনভাবে রাখতে হয় যেন সেটা একেবারে **অনন্য** (Unique) হয় এবং IP ঠিকানার সাথে **সঠিকভাবে যুক্ত** (Bound) থাকে। এজন্য, নামের জায়গা (Namespace) বানাতে হলে সেটা হতে পারে **সমতল** (Flat) অথবা **স্তরবিন্যাসযুক্ত** (Hierarchical)। আধুনিক ইন্টারনেটে আমরা সাধারণত Hierarchical Name Space ব্যবহার করি — যেমন DNS।

Topics Discussed in the Section

- ✓ **Flat Name Space**
- ✓ **Hierarchical Name Space**
- ✓ **Domain Name Space**
- ✓ **Domain**
- ✓ **Distribution of Name Space**

Figure 19.2 *Domain name space*

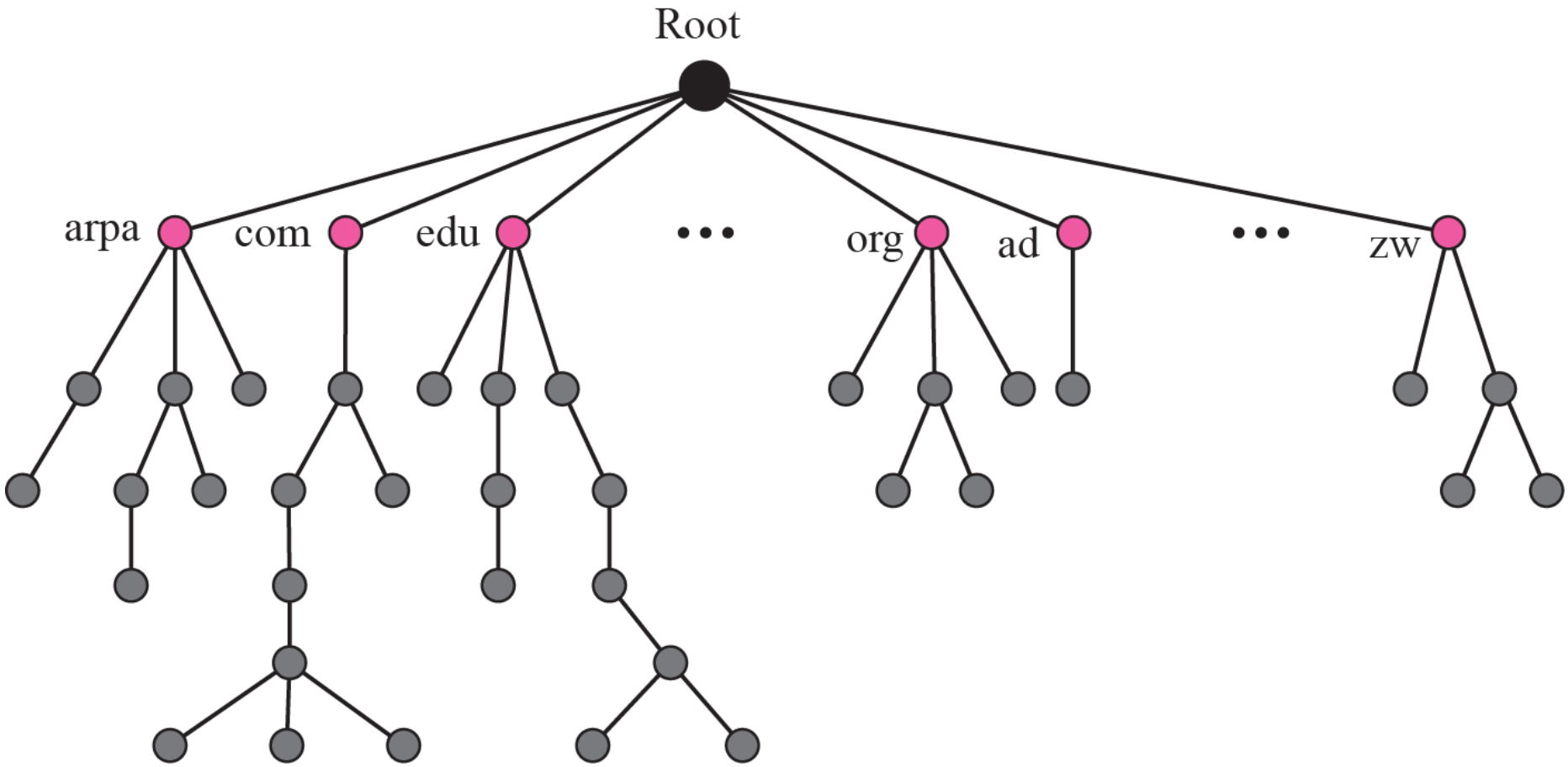


Figure 19.3 *Domain names and labels*

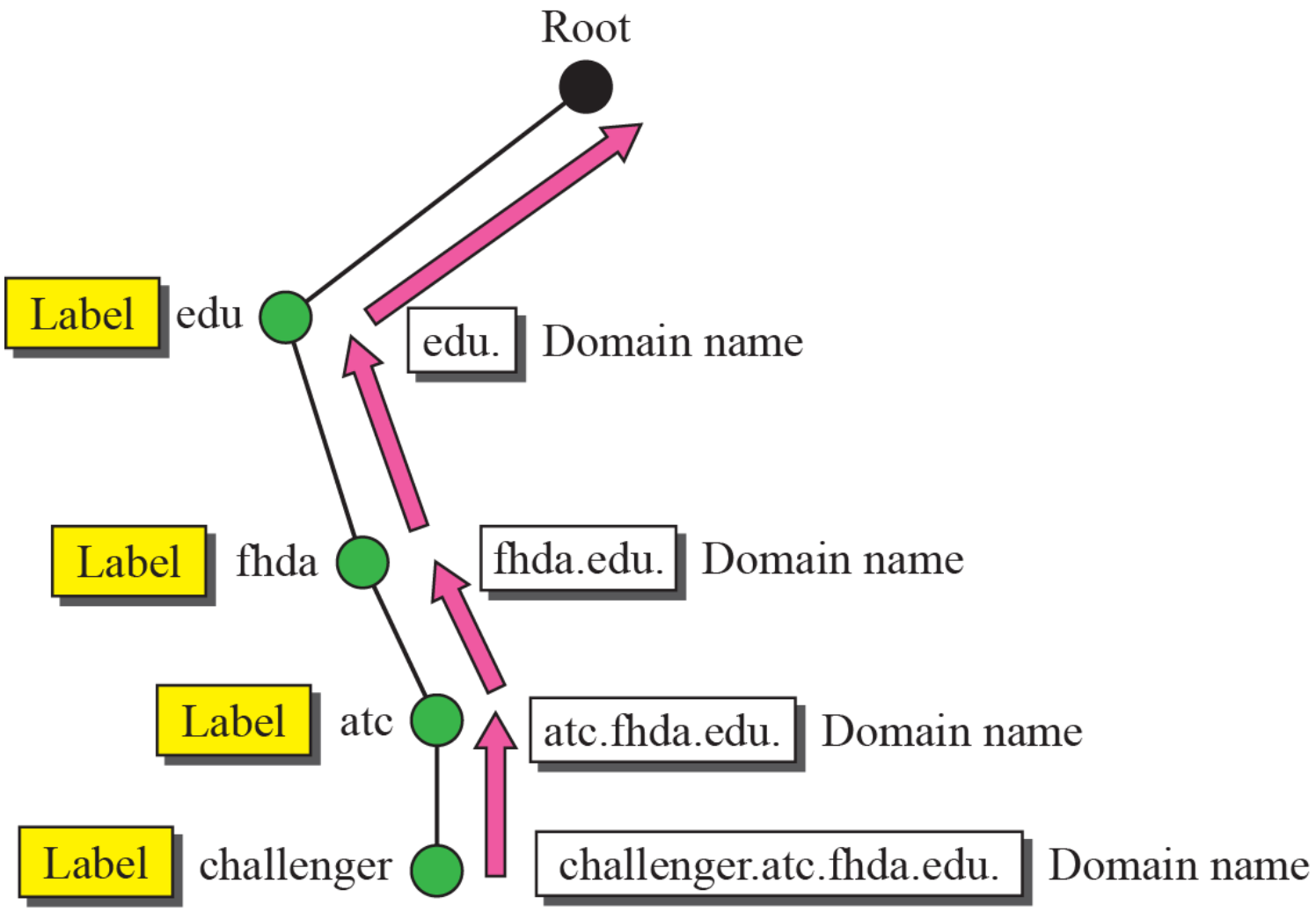
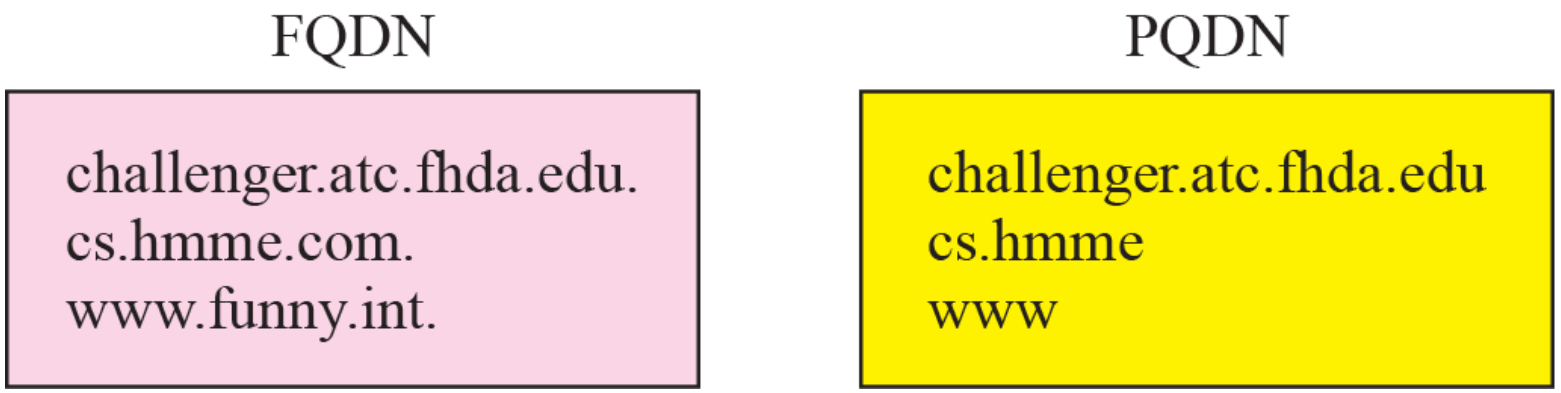


Figure 19.4 *FQDN and PQDN*



Copy Edit

```
www.cs.pstu.ac.bd.
```

This includes:

- `www` → Host
- `cs` → Subdomain (e.g., department)
- `pstu` → Institution/organization
- `ac` → Academic category
- `bd` → Country code top-level domain
- `.` → Root (optional in most interfaces but required in DNS specs)

🔸 PQDN (Partially Qualified Domain Name)

A **PQDN** is a **shortened form** of a domain name that **does not include the full hierarchy**, often missing the top or root domains.

Example:

go Copy Edit

```
www.cs` or `cs.pstu`
```

Fully Qualified Domain Name (FQDN) If a label is terminated by a null string, it is called a **fully qualified domain name (FQDN).** An 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 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.atc.fhda.edu.

Partially Qualified Domain Name (PQDN) 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 *fhda.edu.* site wants to get the IP address of the challenger computer, he or she can define the partial name

challenger

Figure 19.5 *Domains*

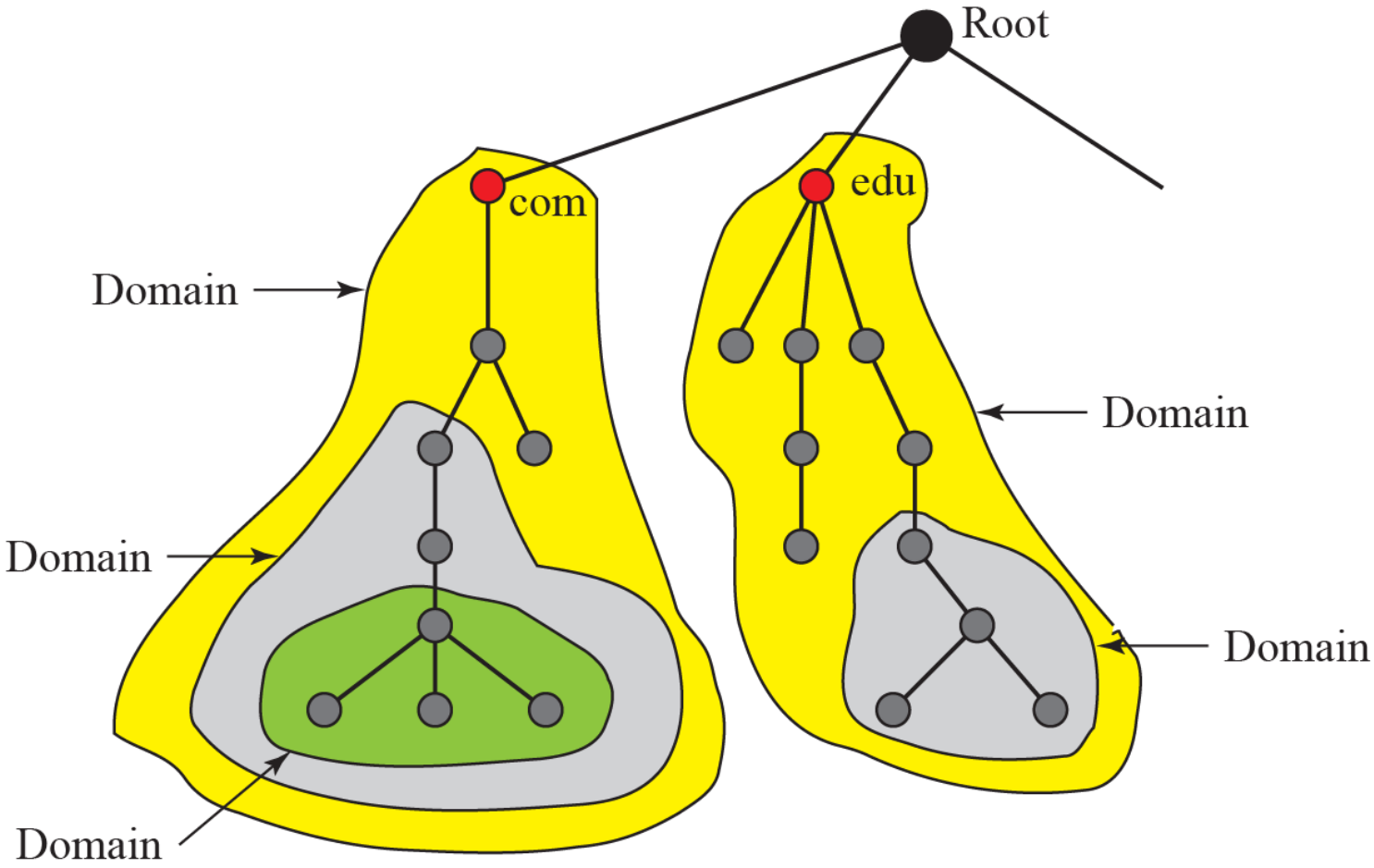


Figure 19.6 *Hierarchy of name servers*

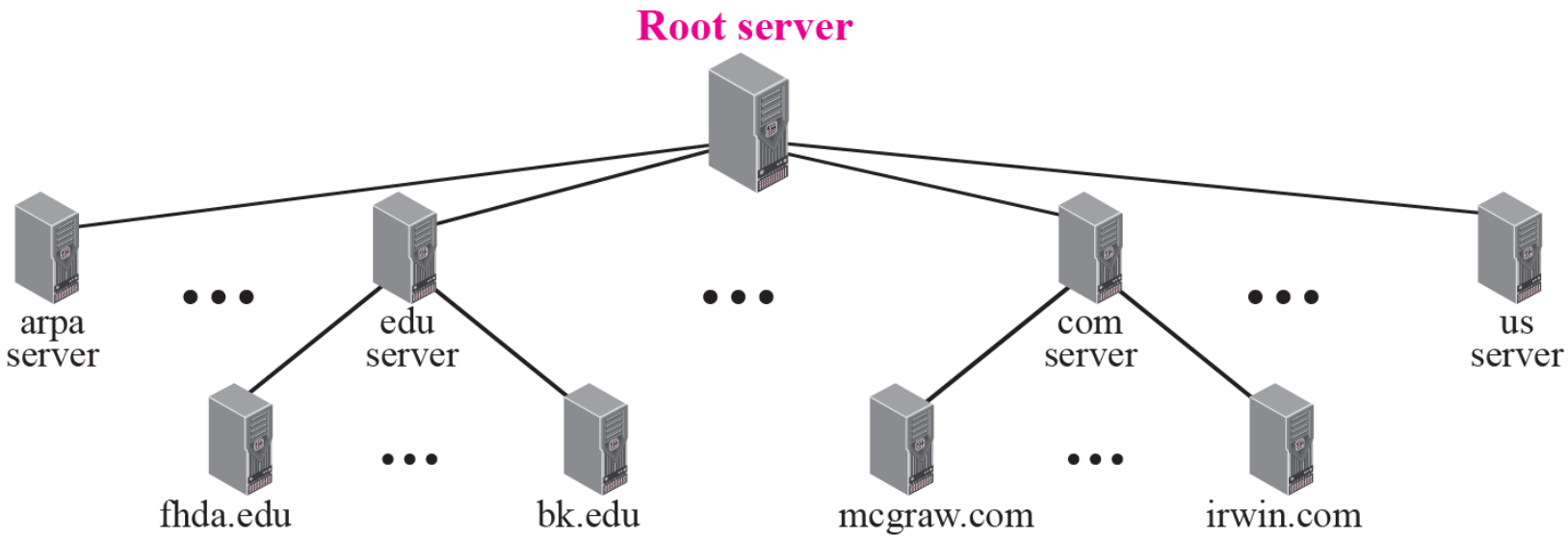
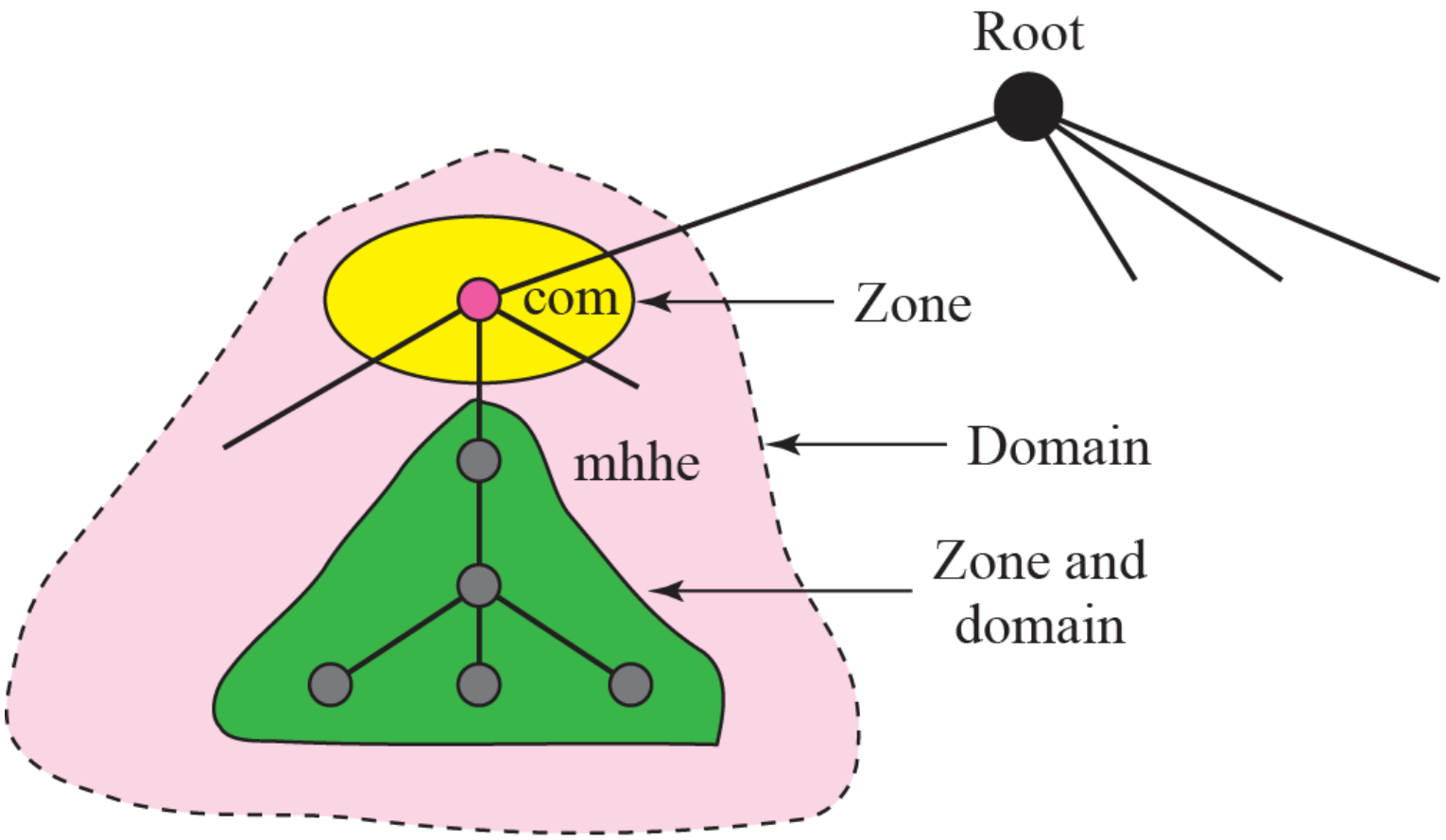


Figure 19.7 *Zones and domains*



🔷 ডোমেইন (Domain):

- DNS-এর একটি **লজিক্যাল গ্রুপ** বা **নাম স্পেস**।
- প্রতিটি ডোমেইন একটি নির্দিষ্ট **নাম হায়ারার্কি** অনুসরণ করে (যেমন: `mhhe.com`, `example.org` ইত্যাদি)।
- প্রতিটি ডোমেইনের মধ্যে এক বা একাধিক **সাব-ডোমেইন** থাকতে পারে।

🔶 জোন (Zone):

- জোন হচ্ছে একটি **ডোমেইনের অংশ**, যা DNS সার্ভার দ্বারা পরিচালিত হয়।
- একটি জোনে **সম্পূর্ণ ডোমেইন বা তার অংশ** থাকতে পারে।
- ডোমেইন **ডেলিগেট** করে সাব-ডোমেইন আলাদা DNS সার্ভারে পরিচালিত হলে, মূল জোন থেকে ওই অংশ আলাদা হয়ে যায়।

↓

pgsql — Saved memory full ⓘ — Copy — Edit

```
          Root
           |
          com          ← Top-level domain
           |
         mhhe           ← Second-level domain (also a zone)
         / \
ftp.mhhe   mail.mhhe    ← সাবডোমেইন
```

এখানে:

- `com` হলো Top-Level Domain (TLD)
- `mhhe.com` হলো Second-Level Domain
- `ftp.mhhe.com`, `mail.mhhe.com` হলো **সাবডোমেইন** (Subdomain)

✅ সাবডোমেইন কোথায়?

চিত্রে `mhhe`-র নিচে যদি `ftp`, `www`, `mail` ইত্যাদি থাকে — এগুলোই **সাবডোমেইন**।

🖼️ চিত্রের গঠন (Figure 19.7):

আপনার ব্যাখ্যা অনুসারে, চিত্রটি দেখায়:

lua Copy Edit

```
        Root
          |
        com    <-- (হলুদ: একটি জোন)
          |
       mhhe      <-- (সবুজ: একটি ডোমেইন এবং জোন, nested)
```

- **mhhe** ডোমেইনটি হল:
- `mhhe.com` এর অংশ
- এটি **নিজেই একটি জোন** হতে পারে, যদি এর জন্য আলাদা DNS রেকর্ড রাখা হয়
- **com** হলো একটি উচ্চস্তরের জোন এবং ডোমেইন, যা:
- **"mhhe" কে অন্তর্ভুক্ত করে**
- **Root** এর অধীনে অবস্থান করে

Saved memory full

🔷 মূল ব্যাখ্যা বাংলায়:

✅ **Primary Server (প্রাইমারি সার্ভার):**

প্রাইমারি সার্ভার হচ্ছে DNS সার্ভার যেটি নিজস্ব **ডিস্ক ফাইল** থেকে ডোমেইনের সকল তথ্য (যেমন: IP address, ডোমেইন নাম) **লোড করে বা সংরক্ষণ করে।**

✅ **Secondary Server (সেকেন্ডারি সার্ভার):**

সেকেন্ডারি সার্ভার মূলত **ব্যাকআপ সার্ভার** হিসেবে কাজ করে। এটি নিজের কোনো মূল কনফিগারেশন ফাইল থেকে তথ্য লোড করে না। বরং এটি **প্রাইমারি সার্ভার** থেকে তথ্য কপি করে নেয়।

🔁 **Zone Transfer (জোন ট্রান্সফার):**

যখন সেকেন্ডারি সার্ভার প্রাইমারি সার্ভার থেকে DNS তথ্য ডাউনলোড করে নেয়, তখন একে বলে **Zone Transfer**। এটি হলো ডোমেইন সংক্রান্ত **DNS রেকর্ড** (যেমন: A record, MX record, NS record) গুলো **এক সার্ভার থেকে আরেক সার্ভারে স্থানান্তর** করার প্রক্রিয়া।

📘 উদাহরণ দিয়ে বোঝানো:

1. **Primary Server:**
   - ধরে নিন `ns1.example.com` হলো প্রাইমারি DNS সার্ভার। এটি নিজের কনফিগ ফাইল থেকে `example.com` ডোমেইনের তথ্য ধারণ করে।
2. **Secondary Server:**
   - `ns2.example.com` হলো সেকেন্ডারি সার্ভার। এটি `ns1.example.com` থেকে zone transfer করে `example.com` ডোমেইনের সব DNS তথ্য কপি করে নেয়।

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.

19-3 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 (see Figure 19.8).

Topics Discussed in the Section

- ✓ **Generic Domains**
- ✓ **Country Domains**
- ✓ **Inverse Domain**
- ✓ **Registrar**

🌐 ডোমেইন নেম স্পেসের তিনটি প্রধান শাখা (বাংলায় ব্যাখ্যা)

DNS-এর স্তরবিন্যাসযুক্ত গাছটি ইন্টারনেটে তিনটি বড় ভাগে বিভক্ত থাকে:

শাখা	কী নিয়ে গঠিত	উদাহরণ	উদ্দেশ্য
Generic Domains (gTLDs)	`.com`, `.org`, `.net`, `.edu`, `.gov`, `.mil`, `.int` ইত্যাদি	`google.com`, `wikipedia.org`	বিষয়ভিত্তিক বা ব্যবহারভিত্তিক গ্লোবাল ডোমেইন নাম। ব্যবসা, অলাভজনক, উচ্চশিক্ষা, সরকার ইত্যাদির জন্য নির্দিষ্ট।
Country Domains (ccTLDs)	ISO-3166 দেশের দু-অক্ষরের কোড-ভিত্তিক TLD	`.bd` (বাংলাদেশ), `.uk` (যুক্তরাজ্য), `.jp` (জাপান)	প্রতিটি দেশের পৃথক ডোমেইন নেম স্পেস; দেশের ভৌগোলিক, আইনগত ও প্রশাসনিক নিয়ম মেনে সাইট নামকরণ ও নিবন্ধন হয়।
Inverse (Reverse) Domain	`in-addr.arpa` (IPv4), `ip6.arpa` (IPv6)	`1.0.168.192.in-addr.arpa` → `router.office.example.com`	IP ঠিকানা থেকে ডোমেইন নাম ফেরত (reverse lookup) পাওয়ার ব্যবস্থা; প্রধানত লগ-অ্যানালিসিস, ইমেইল-ভেরিফিকেশন, নেটওয়ার্ক ডিবাগ-এ কাজে লাগে।

3 Inverse (Reverse) Domain

- **Forward DNS নাম→IP ম্যাপ করে; Inverse DNS ঠিক উল্টো—IP→নাম।**
- IPv4-এর ক্ষেত্রে IP ঠিকানার অক্টেটগুলো উল্টোক্রমে লেখা হয় ও শেষে `in-addr.arpa` যোগ হয়; 192.168.0.1 → `1.0.168.192.in-addr.arpa`.
- IPv6-এ 4-বিট হেক্সাডেসিমাল নিবল উল্টোক্রমে লিখে `ip6.arpa` যোগ করা হয়।
- মেইল-সার্ভার (SMTP), নেটওয়ার্ক টুলস (traceroute, whois) প্রায়ই রিভার্স লুক-আপ ব্যবহার করে যাচাই করে যে IP-এর সঙ্গে ঘোষিত হোস্টনেম মিলছে কিনা।

Figure 19.8 *DNS used in the Internet*

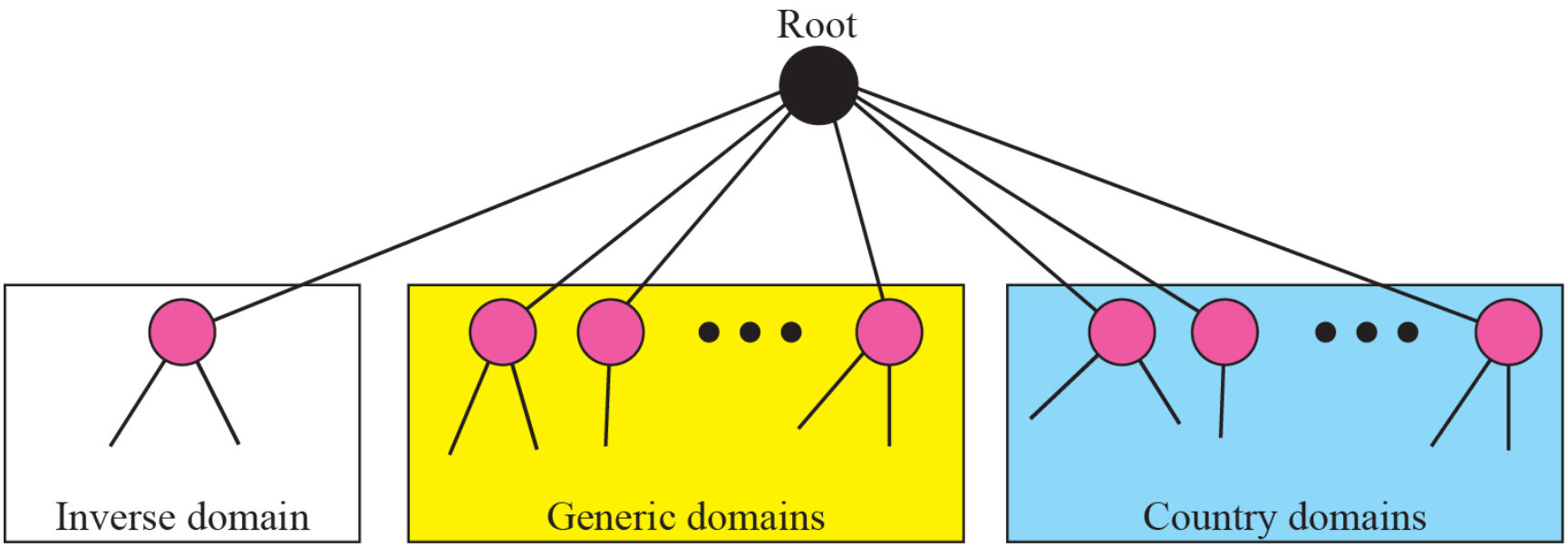


Figure 19.9 *Generic domains*

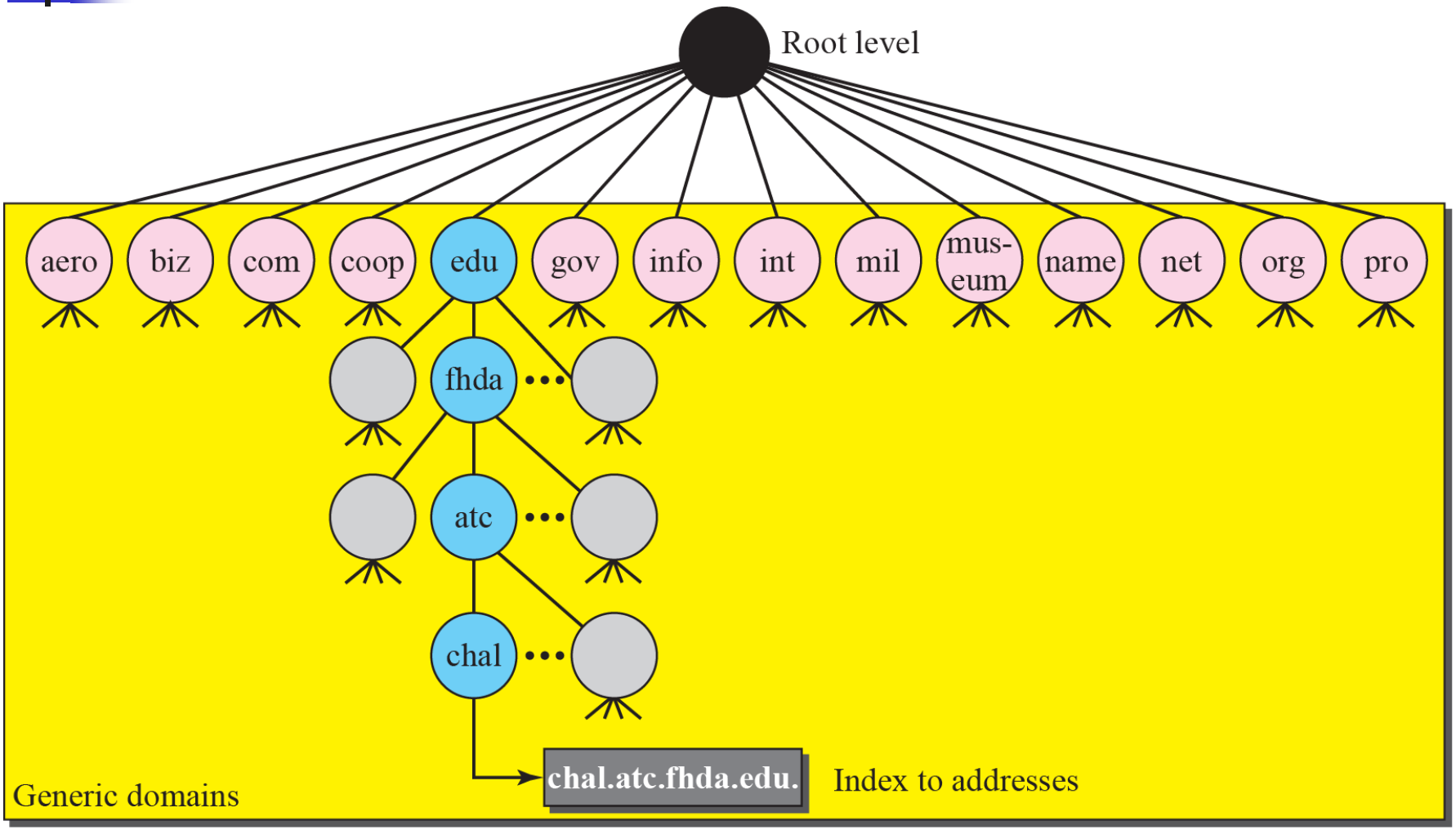


Table 19.1 *Generic domain labels*

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 non-profit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Figure 19.10 *Country domains*

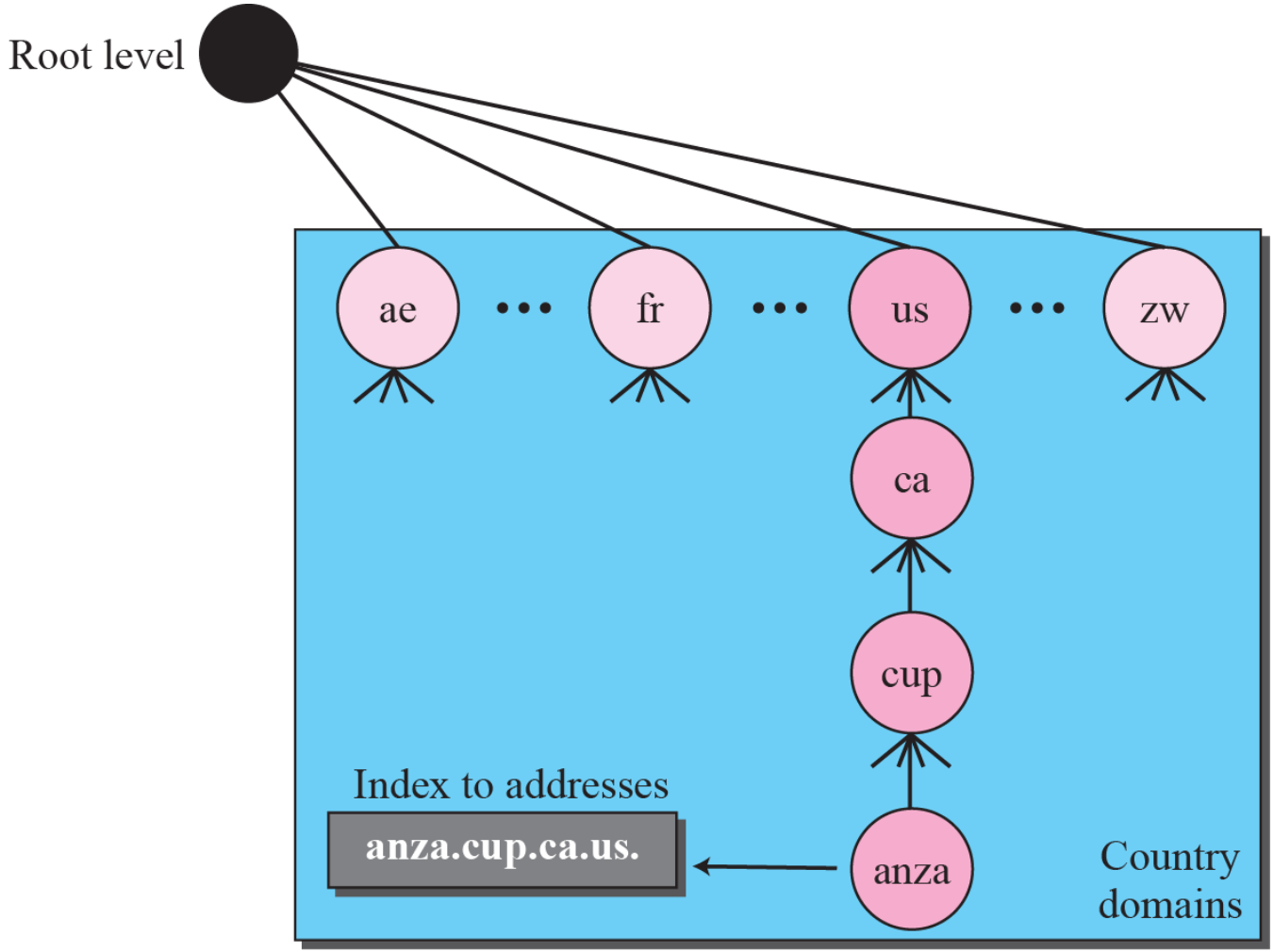
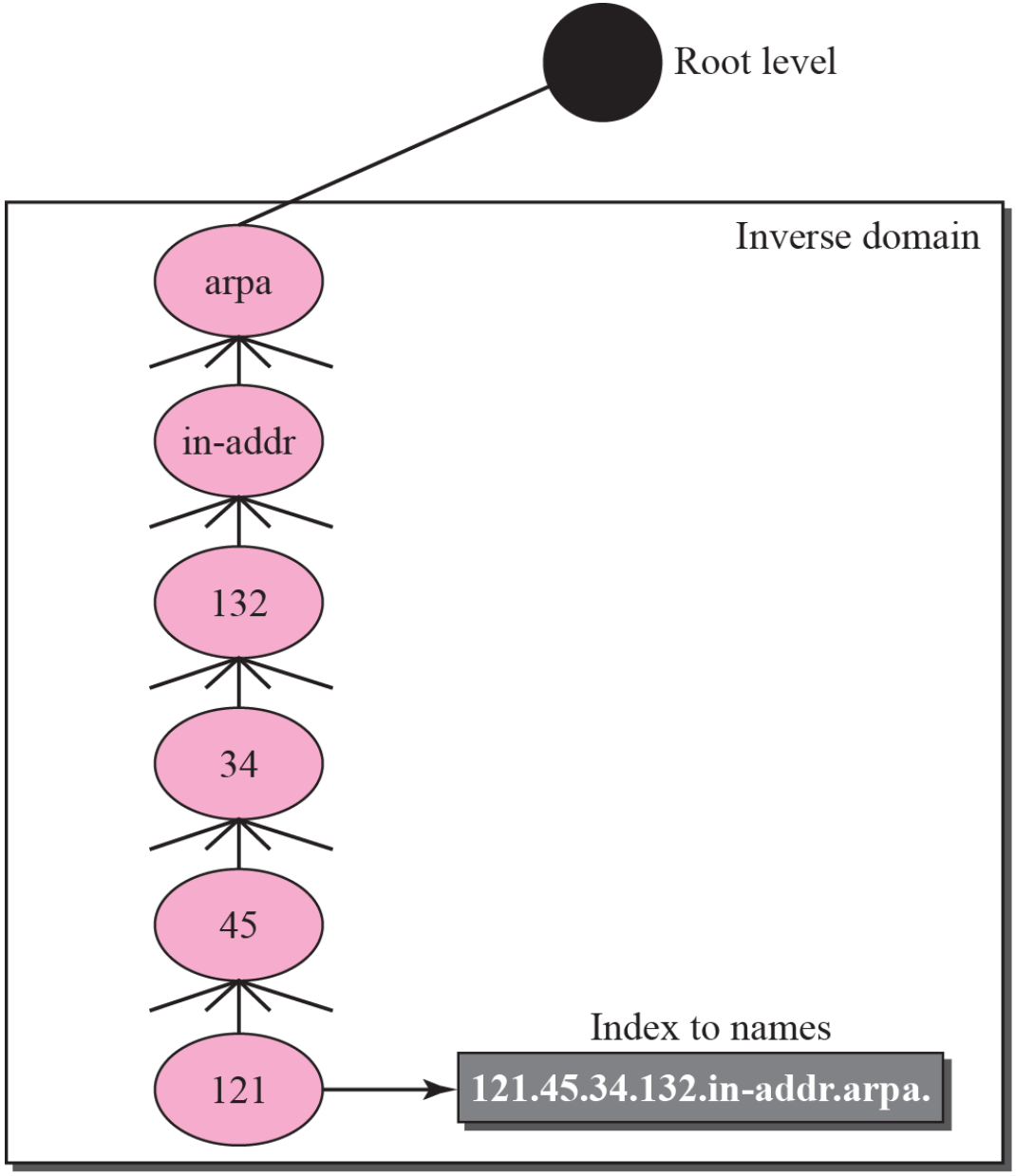


Figure 19.11 *Inverse domain*



19-4 RESOLUTION

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

◆ সংজ্ঞা (Definition):

Saved memory full ⓘ

> "Mapping a name to an address or an address to a name is called name-address resolution."
> অর্থাৎ, কোনো **নাম থেকে IP ঠিকানা খুঁজে বের করা** অথবা **IP ঠিকানা থেকে নাম খুঁজে বের করা**—এই প্রক্রিয়াটিকে বলে **Name-Address Resolution**।

◆ দুই ধরণের Resolution:

ধরন	অর্থ	উদাহরণ
Forward Resolution	নাম → ঠিকানা	`www.google.com` → `142.250.195.100`
Reverse Resolution	ঠিকানা → নাম	`142.250.195.100` → `www.google.com`

◆ ব্যবহৃত প্রোটোকল:

- ইন্টারনেটে **DNS (Domain Name System)** এই Name-Address Resolution এর কাজ করে।
- Reverse Resolution-এর জন্য DNS-এ **PTR (Pointer Record)** ব্যবহৃত হয়।

Topics Discussed in the Section

- ✓ **Resolver**
- ✓ **Mapping Names to Addresses**
- ✓ **Mapping Addresses to Names**
- ✓ **Recursive Resolution**
- ✓ **Iterative Resolution**
- ✓ **Caching**

Figure 19.12 *Recursive resolution*

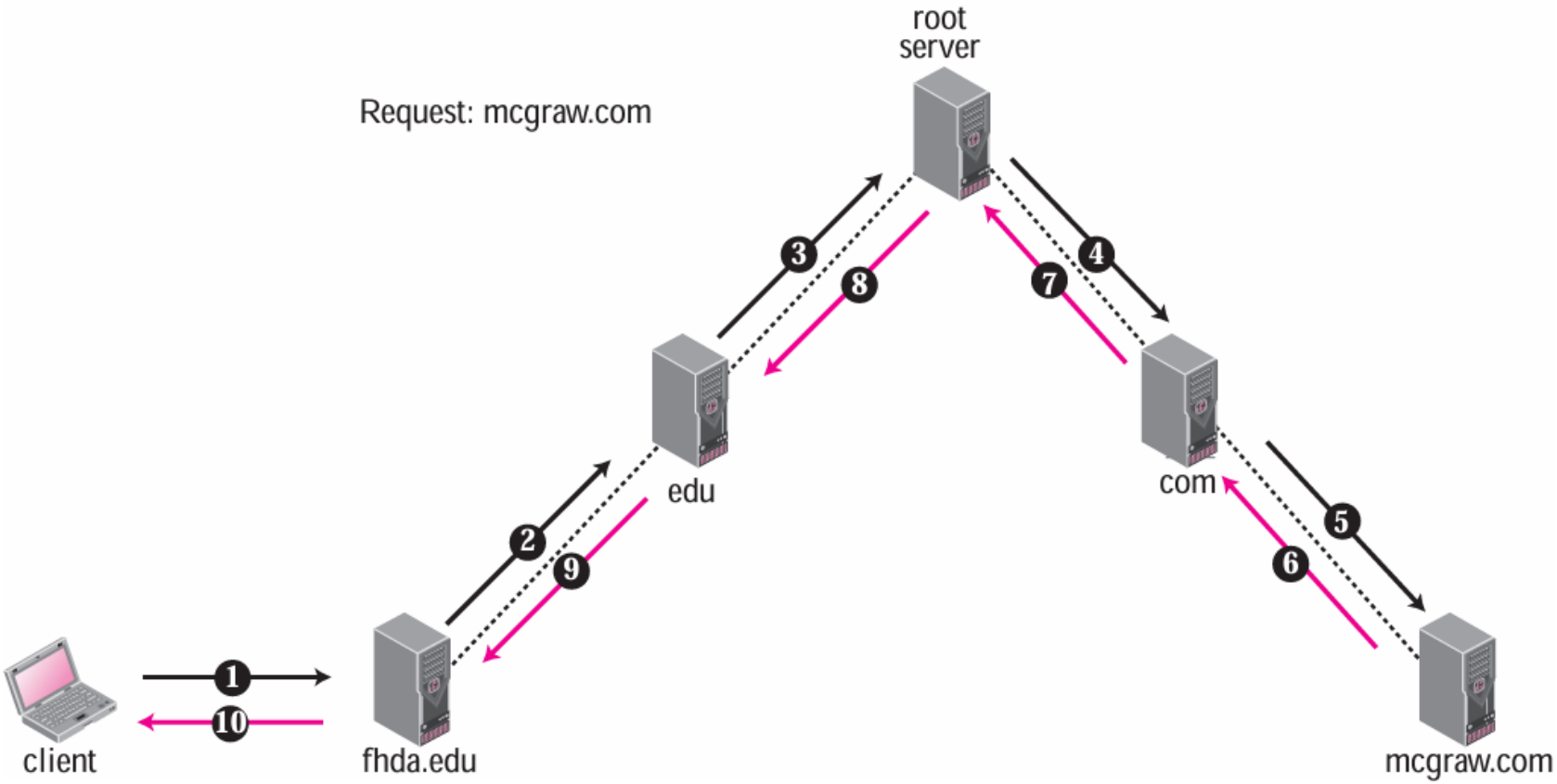
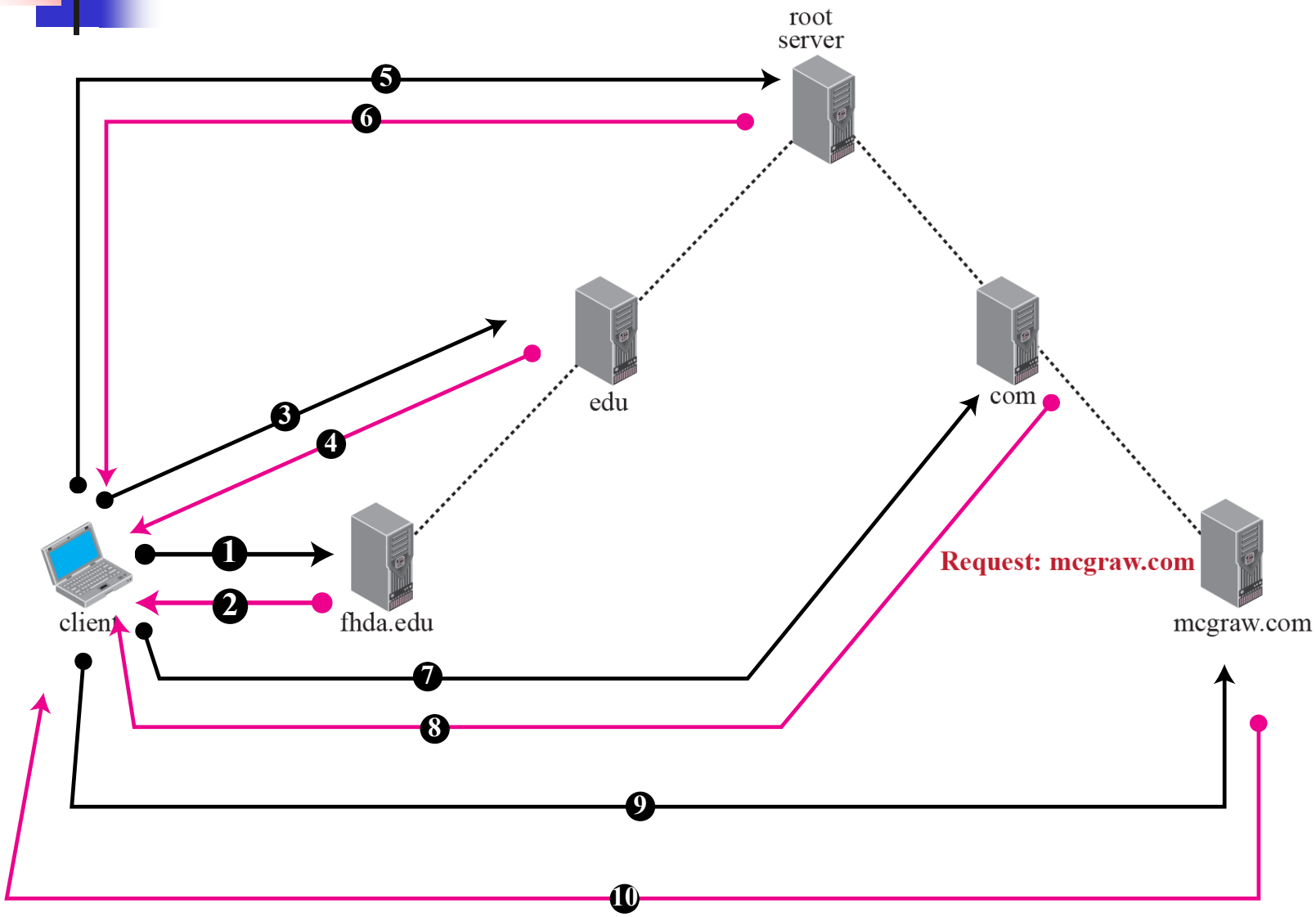


Figure 19.13 *Iterative resolution*



19-5 DNS MESSAGES

DNS has two types of messages: query and response. Both types have the same format. The query message consists of a header and question records; the response message consists of a header, question records, answer records, authoritative records, and additional records (see Figure 19.14).

Topics Discussed in the Section

- ✓ **Header**

Figure 19.14 *Query and response messages*

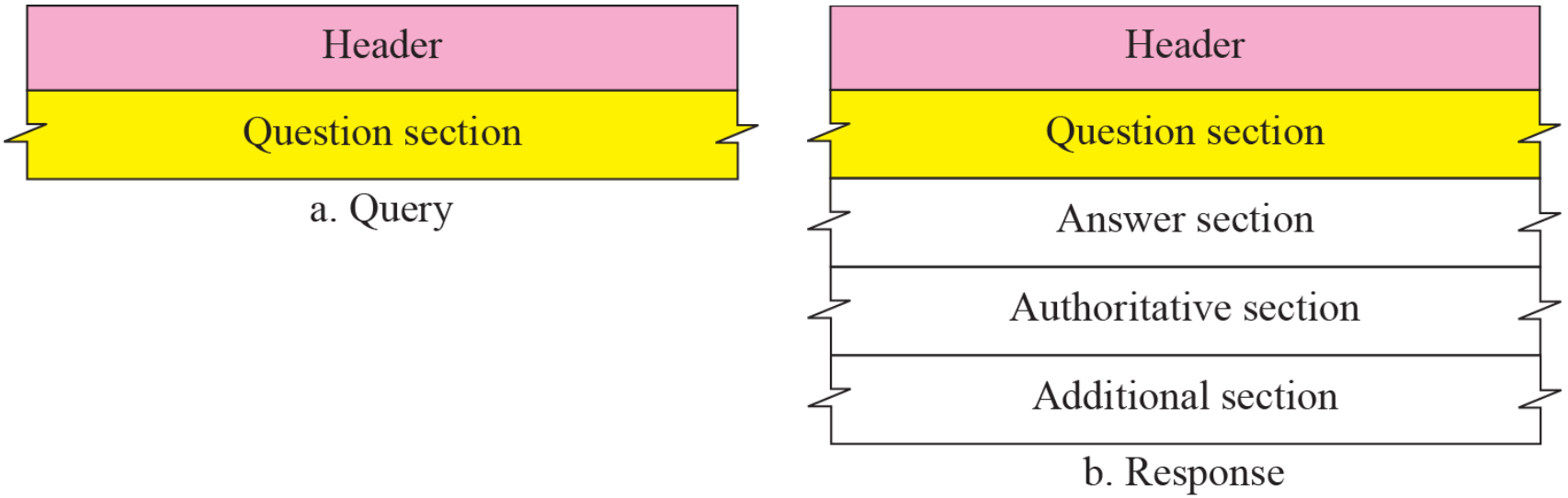


Figure 19.15 *Header format*

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)

- **Number of question records.** This is a 16-bit field containing the number of queries in the question section of the message.
- **Number of answer records.** This is a 16-bit field containing the number of answer records in the answer section of the response message. Its value is zero in the query message.
- **Number of authoritative records.** This is a 16-bit field containing the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message.
- **Number of additional records.** This is a 16-bit field containing the number of additional records in the additional section of a response message. Its value is zero in the query message.

Question Section

This is a section consisting of one or more question records. It is present on both query and response messages. We will discuss the question records in a following section.

Answer Section

This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver). We will discuss resource records in a following section.

Authoritative Section

This is a section consisting of one or more resource records. It is present only on response messages. This section gives information (domain name) about one or more authoritative servers for the query.

Additional Information Section

This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help the resolver. For example, a server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

Figure 19.16 *Flags field*

A brief description of each flag subfield follows.

a. **QR (query/response).** This is a 1-bit subfield that defines the type of message. If it is 0, the message is a query. If it is 1, the message is a response.

b. **OpCode.** This is a 4-bit subfield that defines the type of query or response (0 if standard, 1 if inverse, and 2 if a server status request).

c. **AA (authoritative answer).** This is a 1-bit subfield. When it is set (value of 1) it means that the name server is an authoritative server. It is used only in a response message.

d. **TC (truncated).** This is a 1-bit subfield. When it is set (value of 1), it means that the response was more than 512 bytes and truncated to 512. It is used when DNS uses the services of UDP (see Section 19.8 on Encapsulation).

e. **RD (recursion desired).** This is a 1-bit subfield. When it is set (value of 1) it means the client desires a recursive answer. It is set in the query message and repeated in the response message.

f. **RA (recursion available).** This is a 1-bit subfield. When it is set in the response, it means that a recursive response is available. It is set only in the response message.

g. **Reserved.** This is a 3-bit subfield set to 000.

h. **rCode.** This is a 4-bit field that shows the status of the error in the response. Of course, only an authoritative server can make such a judgment. Table 19.2 shows the possible values for this field.

Table 19.2 *Values of rCode*

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

19-6 TYPES OF RECORDS

As we saw in the previous section, 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.

Topics Discussed in the Section

- ✓ **Question Record**
- ✓ **Resource Record**

Figure 19.17 *Question record format*

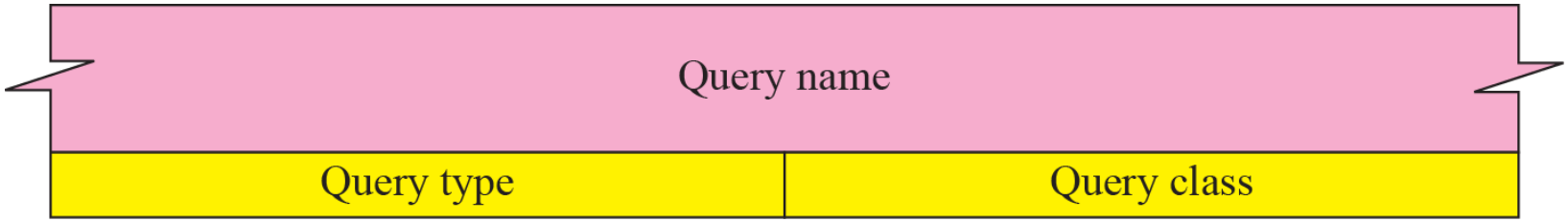


Figure 19.18 *Query name format*

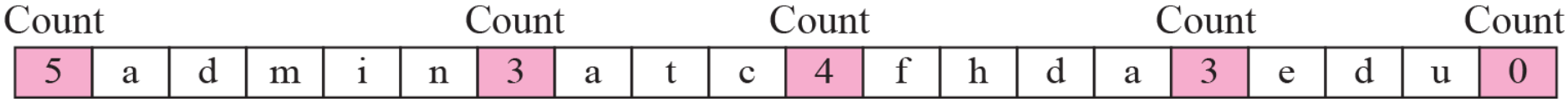


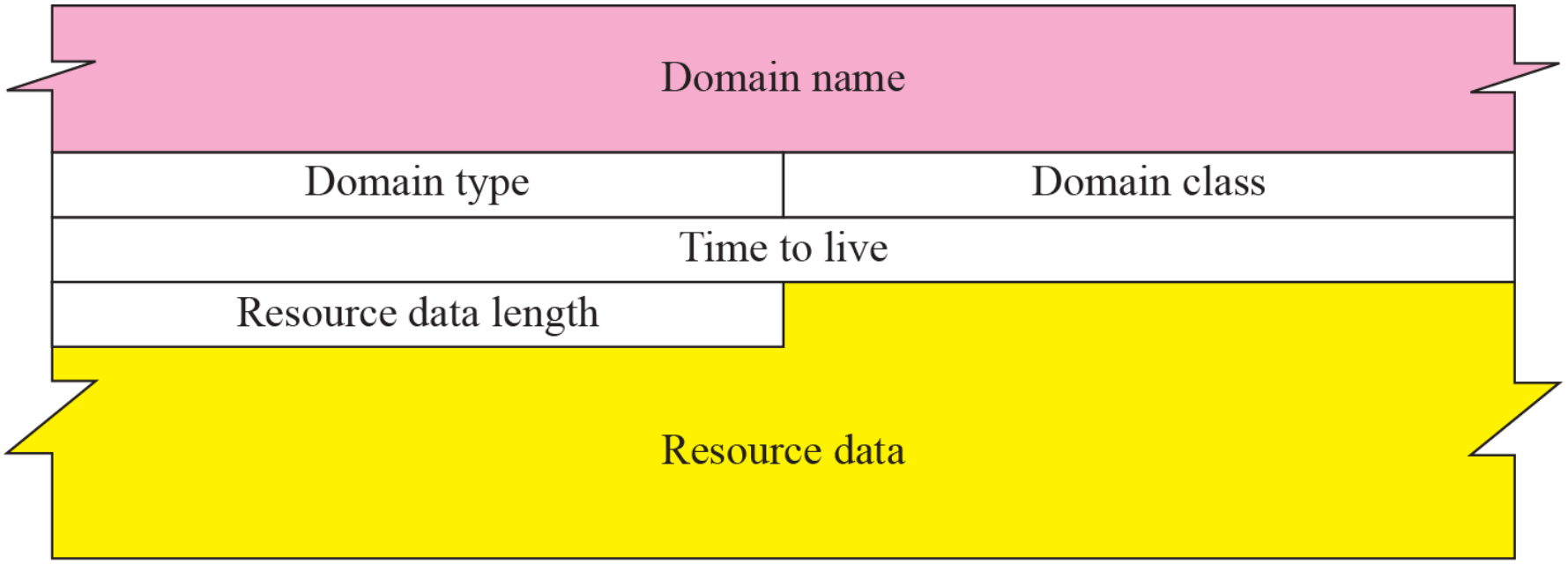
Table 19.3 *Types*

Type	*Mnemonic*	*Description*
1	A	**Address.** A 32-bit IPv4 address. It converts a domain name to an 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.
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 defines the hardware and operating system.
15	MX	**Mail exchange**. It redirects mail to a mail server.
28	AAAA	**Address.** An IPv6 address (see Chapter 26).
252	AXFR	A request for the transfer of the entire zone.
255	ANY	A request for all records.

Table 19.4 *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 19.19 *Resource record format*



✅ Example of an A Record

yaml

Copy Edit

```yaml
Name:           www.example.com.
Type:           A
Class:          IN
TTL:            3600
RDLENGTH:       4
RDATA:          93.184.216.34
```

19-7 COMPRESSION

DNS requires that a domain name be replaced by an offset pointer if it is repeated. For example, in a resource record the domain name is usually a repetition of the domain name in the question record. For efficiency, DNS defines a 2-byte offset pointer that points to a previous occurrence of the domain or part of it. The format of the field is shown in Figure 19.20.

📌 DNS Offset Pointer & Compression (বাংলায় ব্যাখ্যা)

🔹 বিষয়টি কী?

DNS মেসেজে অনেক সময় একটি ডোমেইন নাম **একাধিকবার পুনরাবৃত্তি** (repeat) হতে পারে — যেমন:

- প্রশ্ন রেকর্ডে (Question Record): `www.pstu.ac.bd`
- উত্তর রেকর্ডে (Resource Record): আবার একই `www.pstu.ac.bd`

এই পুনরাবৃত্তি ঠেকাতে DNS **compression technique** ব্যবহার করে, যেখানে পরবর্তীতে নাম লিখার পরিবর্তে **আগে লেখা নামের জায়গায় একটি pointer (off-set)** ব্যবহার করা হয়।

🔹 কিভাবে কাজ করে?

DNS message-এর প্রতিটি অংশ বাইট আকারে থাকে। যদি কোনো ডোমেইন নাম আগে দেখা গেছে, তাহলে সেটির পরবর্তী occurrence-এ সেই নাম আবার লেখার পরিবর্তে:

- একটি **2-বাইটের pointer** ব্যবহার করা হয়
- এই pointer দেখায় **DNS message-এর ভিতরে আগের কোনো নির্দিষ্ট অবস্থান** (offset)-এ যেখানে সেই নামটা আছে।

🔹 Pointer-এর গঠন (Format):

Pointer-এর প্রথম 2 বিট হয় `11`, বাকি 14 বিট হয় offset মান (মানে কত নম্বর বাইটে আগের নামটা শুরু হয়েছে)।

```
diff
```

Copy Edit

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1|  Offset   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 (Total 16 bits = 2 bytes)
```

- 11 = এই অংশটি pointer
- Offset = আগের নামের অবস্থান (0 থেকে 16383 পর্যন্ত যেকোনো পজিশন)

🔸 উদাহরণ:

ধরি DNS প্রশ্ন ও উত্তর এভাবে এসেছে:

```
makefile
```

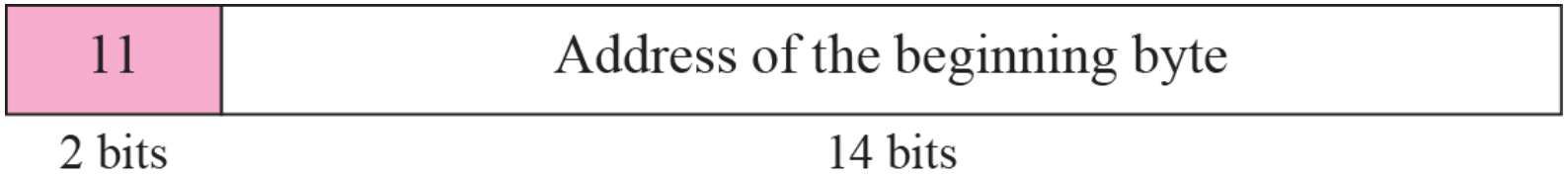
Copy Edit

```
Question:
  www.pstu.ac.bd

Answer:
  www.pstu.ac.bd   A   203.112.5.10
```

এখানে `www.pstu.ac.bd` দ্বিতীয়বার না লিখে, Answer সেকশনে শুধুমাত্র একটি pointer দেওয়া হবে যা Question সেকশনে এই নামটি যেখানে শুরু হয়েছে, সেটি নির্দেশ ক...

Figure 19.20 *Format of an offset pointer*



🔷 1️⃣ Header Section (12 bytes)

ফিল্ড নাম	মান (Hex বা Value)	ব্যাখ্যা (বাংলায়)
Identification	`0x1333`	ক্লায়েন্ট ID (একটি শনাক্তকারী নম্বর)
Flags	`0x0100`	Standard Query, recursion desired
Question Count (QDCOUNT)	`1`	১টি প্রশ্ন রয়েছে
Answer Count (ANCOUNT)	`0`	কোনো উত্তর নেই (এটি query message)
Authority Count (NSCOUNT)	`0`	authoritative তথ্য নেই
Additional Count (ARCOUNT)	`0`	অতিরিক্ত তথ্য নেই

👉 এই ১২ বাইট হলো মেসেজের প্রথম অংশ (header)।

🔍 🔗 Pointer `0xC0 0x0C` ব্যাখ্যা:

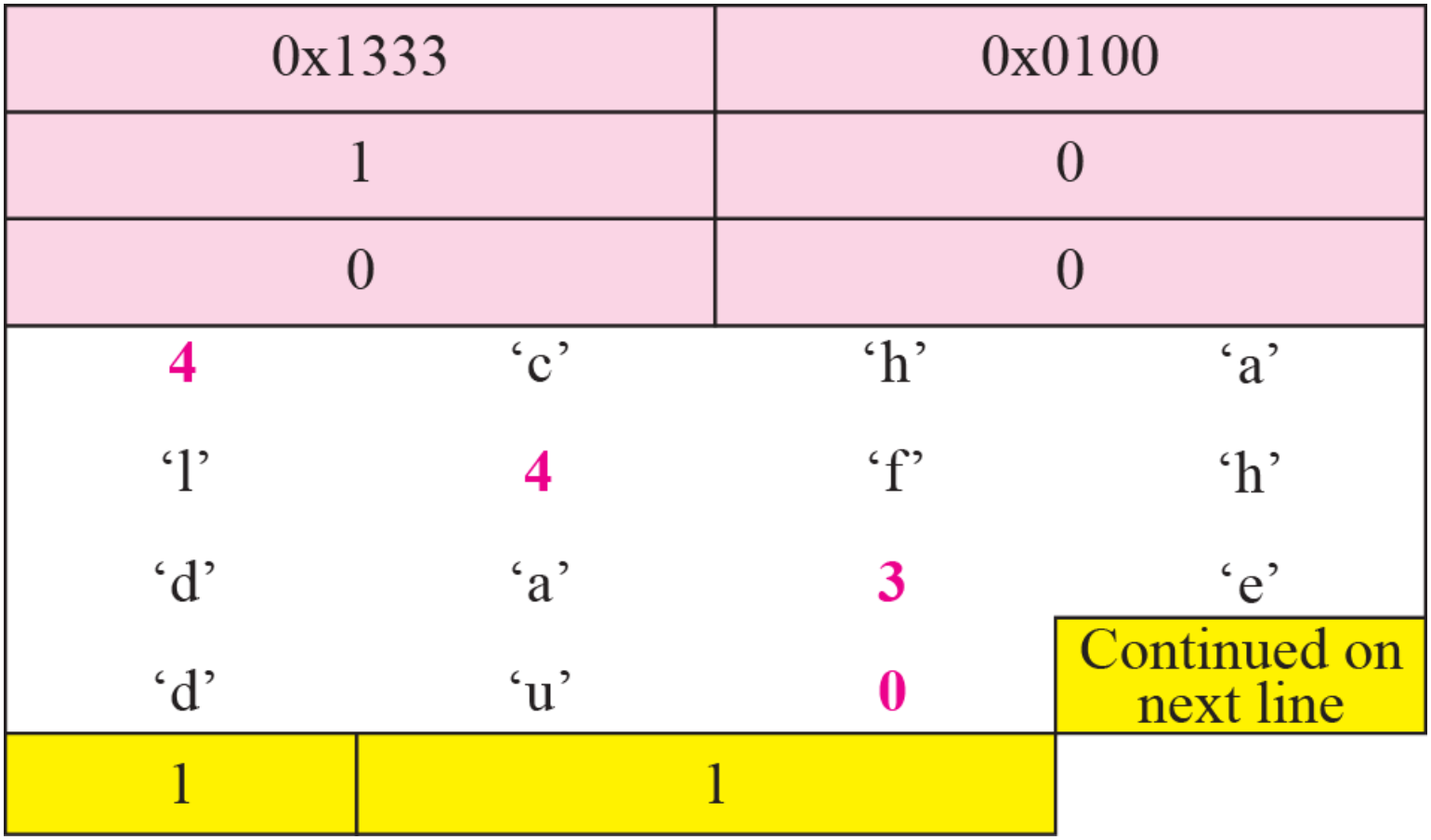
- DNS efficiency এর জন্য, `Name` ক্ষেত্র আবার লেখার বদলে একটি **pointer (C0 XX)** ব্যবহার করা হয়।
- `0xC0` মানে: এটা একটি pointer।
- `0x0C` → Offset 12 → যেখান থেকে QNAME শুরু হয়েছিল।
- অর্থাৎ, এই **Answer Section-এর Name** অংশ **Question Section-এর QNAME** কে নির্দেশ করে।

Example 19.1

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 19.21 shows the query message sent by the resolver. The first 2 bytes show the identifier $(1333)_{16}$. It is used as a sequence number and relates a response to a query. 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 19.21 *Example 19.1: Query message*



🔍 Understanding Example 19.1 – DNS Query Message Format

A DNS **Query Message** typically includes:

Section	Description
Transaction ID	A 16-bit identifier used to match responses with queries. Example: `0x1333`.
Flags	Contains info like whether it's a query/response. Example: `0x0100` means standard query.
QDCOUNT	Number of questions (usually 1).
ANCOUNT	Number of answers (0 in query).
NSCOUNT	Number of authority records (0 in query).
ARCOUNT	Number of additional records (0 in query).
Queries	Contains the domain name, type, and class.

Question Section (Middle part)

This section shows the domain name being queried, encoded in **label format:**

Each label is:

```
css                                                    Copy  Edit

[length][characters]
```

Let's decode it:

Length	Label	Characters
`4`	`ldd`	`l`, `d`, `d`, `c`
`4`	`hfa`	`h`, `f`, `a`, `u`
`3`	`ahe`	`a`, `h`, `e`
`0`	(End)	Terminator of the domain name

So the domain name being queried is:

`lddc.hfau.ahe`

(Each label ends after the number of characters specified by its prefix, and the full name ends with `0`.)

Type and Class Fields (Bottom gray row)

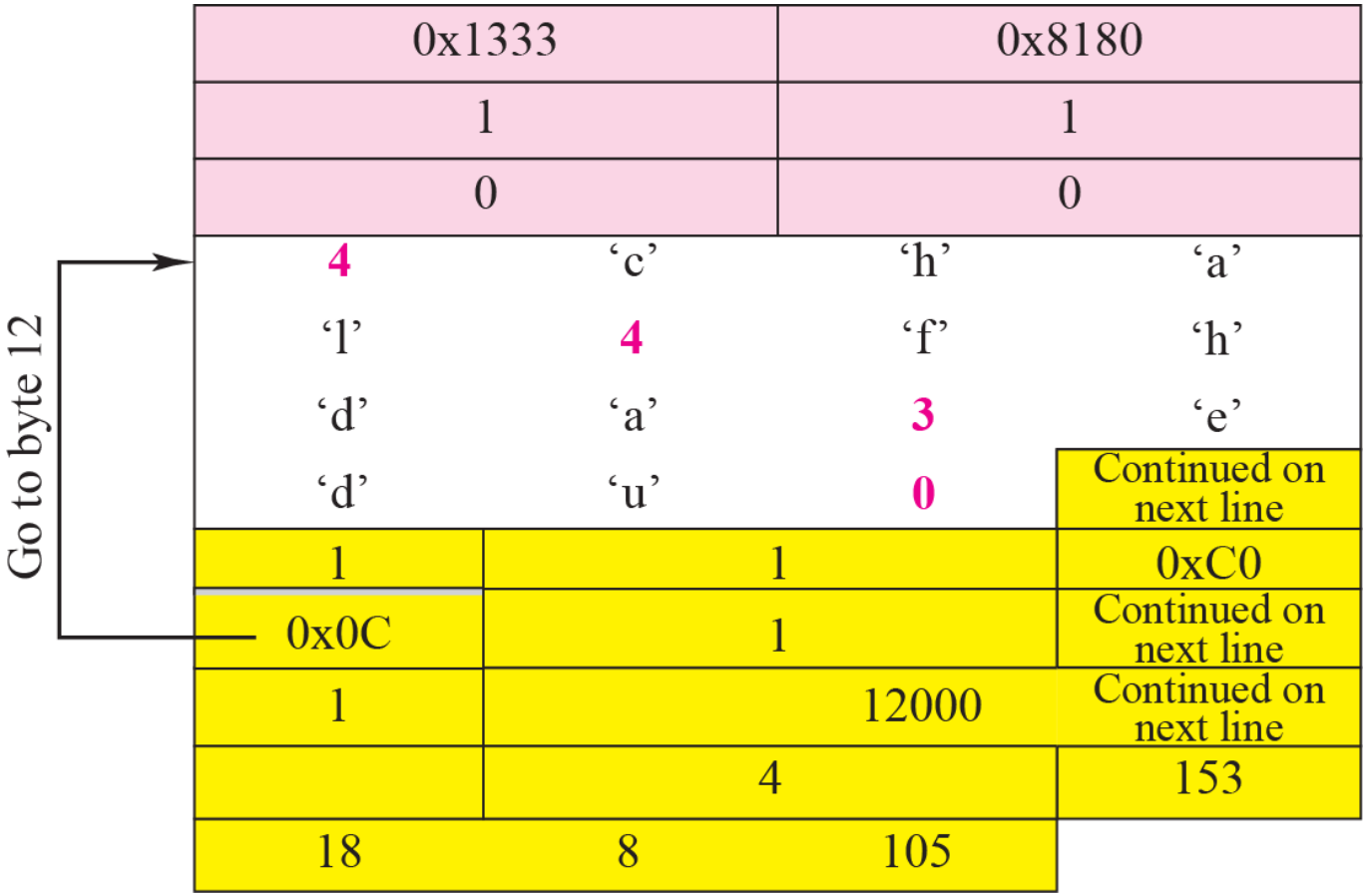
Field	Value	Description
Type	`1`	Type `A`, meaning query for IPv4 address
Class	`1`	Class `IN` (Internet)

Example 19.1 *Continued*

Figure 19.22 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

Figure 19.22 *Example 19.1: Response message*



🔷 1. Header (প্রথম 12 বাইট):

ফিল্ড	মান	ব্যাখ্যা
ID (Identification)	`0x1333`	ক্লায়েন্টের পাঠানো ID — query এর সাথেও ছিল
Flags	`0x8180`	Response, recursion supported, no error
QDCOUNT	`1`	১টি প্রশ্ন ছিল
ANCOUNT	`1`	১টি উত্তর রয়েছে
NSCOUNT	`0`	Authoritative section নাই
ARCOUNT	`0`	Additional section নাই

🔷 2. Question Section:

এখানে ক্লায়েন্ট যে প্রশ্ন করেছে তা পুনরাবৃত্তি হয়:

অংশ	মান (ASCII)	ব্যাখ্যা
QNAME	`'lddc.hfau.ahe'` বা অনুরূপ	লেবেল ফরম্যাটে ডোমেইন (উদাহরণ: `lddc.hfau.ahe`)
QTYPE	`1`	A record (IPv4)
QCLASS	`1`	IN (Internet)

> এখানে ASCII অংশ:

bash Copy Edit

```
4 → 'lddc'
4 → 'hfau'
3 → 'ahe'
0 → শেষ
```

🔷 3. Answer Section:

এখানে DNS সার্ভার এর উত্তর রয়েছে।

ফিল্ড	মান	ব্যাখ্যা
Name	`0xC0 0x0C`	Pointer → offset 12 (QNAME-এর অবস্থান)
Type	`1`	A Record
Class	`1`	IN (Internet)
TTL	`12000`	সময় (সেকেন্ডে) – কতক্ষণ ধরে রাখতে পারবে
RDLENGTH	`4`	ডেটার দৈর্ঘ্য = 4 বাইট
RDATA	`153.18.8.105`	IP Address (উত্তর)

✅ উদাহরণ

ধরুন, DNS message-এর byte 12 থেকে শুরু করে নিচের মতো আছে:

arduino Copy Edit

```
4 'c' 'l' 'd' 'd'
4 'h' 'f' 'a' 'u'
3 'a' 'h' 'e'
0
```

👉 এখন উত্তর অংশে `0xC00C` বলছে: **"ওই ১২ নম্বর বাইটে যাও — ওখানে ডোমেইন নাম পুরোটা লেখা আছে — সেটা এখানে ধরে নাও"।**

Example 19.2

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. Figure 19.23 shows the query message sent from the resolver to the server.

Figure 19.23 *Example 19.2: Inverse query message*

0x1200		0x0900	
1		0	
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	

Example 19.2 *Continued*

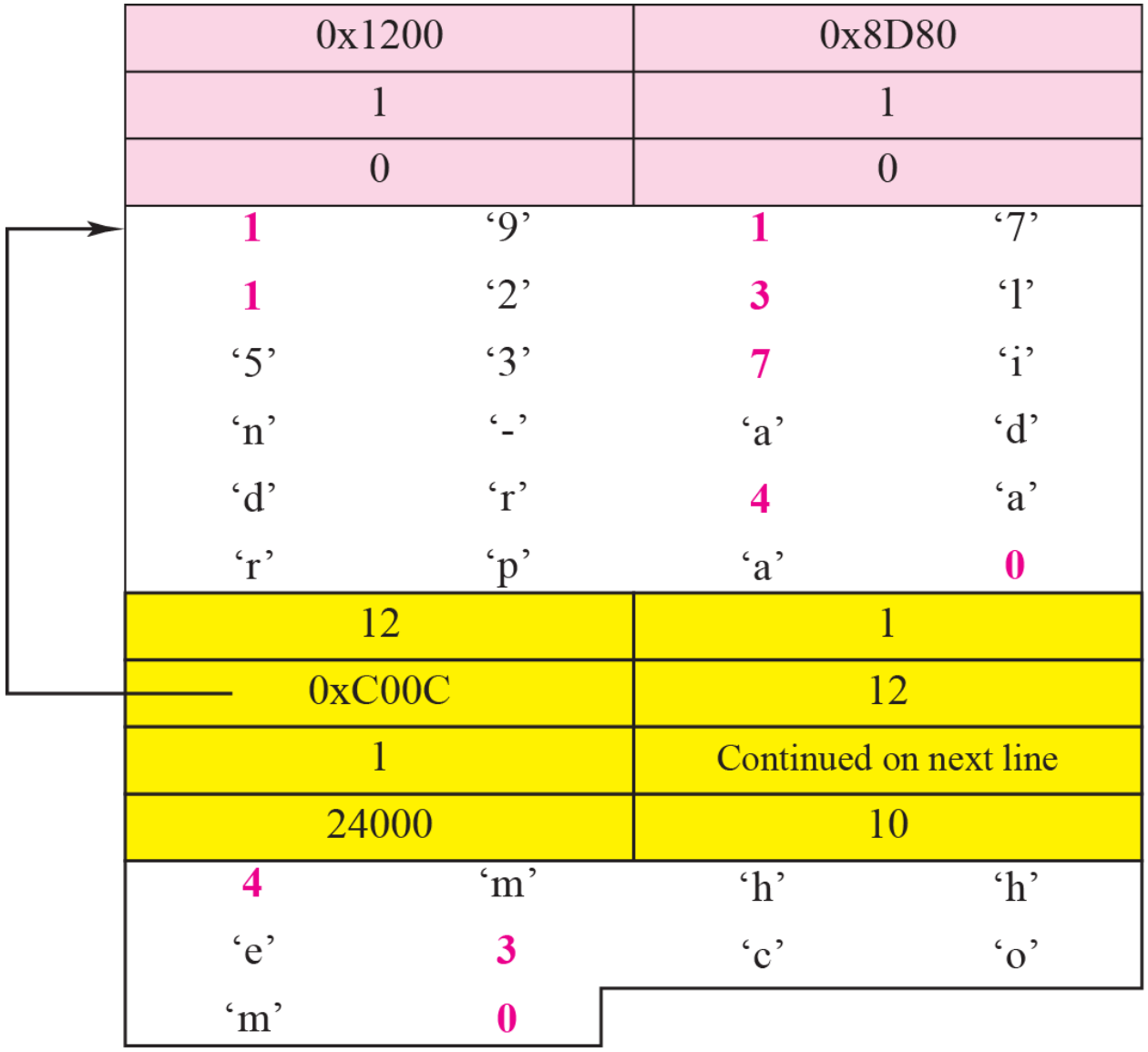
The first 2 bytes show the identifier (0x1200). The flags value is 0x0900 in hexadecimal. In binary it is 0000100100000000, and we divide it into fields as shown below:

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 19.24 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	0	000	0000

Figure 19.24 *Example 19.2: 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**		

✅ সারসংক্ষেপ:

পদক্ষেপ	ফলাফল
`0xC00C`	একটি DNS compression pointer
`0xC00C & 0x3FFF`	কেবল offset অংশ বের করতে
ফলাফল	`0x000C` → Decimal 12
অর্থ	DNS message-এ ১২তম বাইট থেকে ডেটা রি-ইউজ করো

Example 19.3

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.
```

19-8 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:

19-9 REGISTRARS

How are new domains added to DNS? This is done 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.

19-10 DDNS

When the DNS was designed, no one predicted that there would be so many address changes. 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. The DNS master file must be updated dynamically. The Dynamic Domain Name System (DDNS) therefore was devised to respond to this need.

যখন DNS (Domain Name System) ডিজাইন করা হয়েছিল, তখন কেউ ভাবেনি যে এত ঘন ঘন ঠিকানা (IP address) পরিবর্তন হবে। DNS-এ যদি কোনো পরিবর্তন হয় — যেমন, একটি নতুন হোস্ট যোগ করা, কোনো হোস্ট মুছে ফেলা, বা কোনো IP ঠিকানা পরিবর্তন করা — তাহলে সেই পরিবর্তন DNS-এর মাস্টার ফাইলে করতে হয়। এই মাস্টার ফাইলকে স্বয়ংক্রিয়ভাবে (ডাইনামিকভাবে) আপডেট করতে হয়। এই চাহিদা মেটানোর জন্য **Dynamic Domain Name System (DDNS)** তৈরি করা হয়েছে।

19-11 SECURITY OF DNS

DNS is one of the most important systems in the Internet infrastructure; it provides crucial services to the Internet users. Applications such as Web access or e-mail are heavily dependent on the proper operation of DNS. DNS can be attacked in several
Ways.

To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides the message origin authentication and message integrity using a security service called digital signature (See Chapter 29).