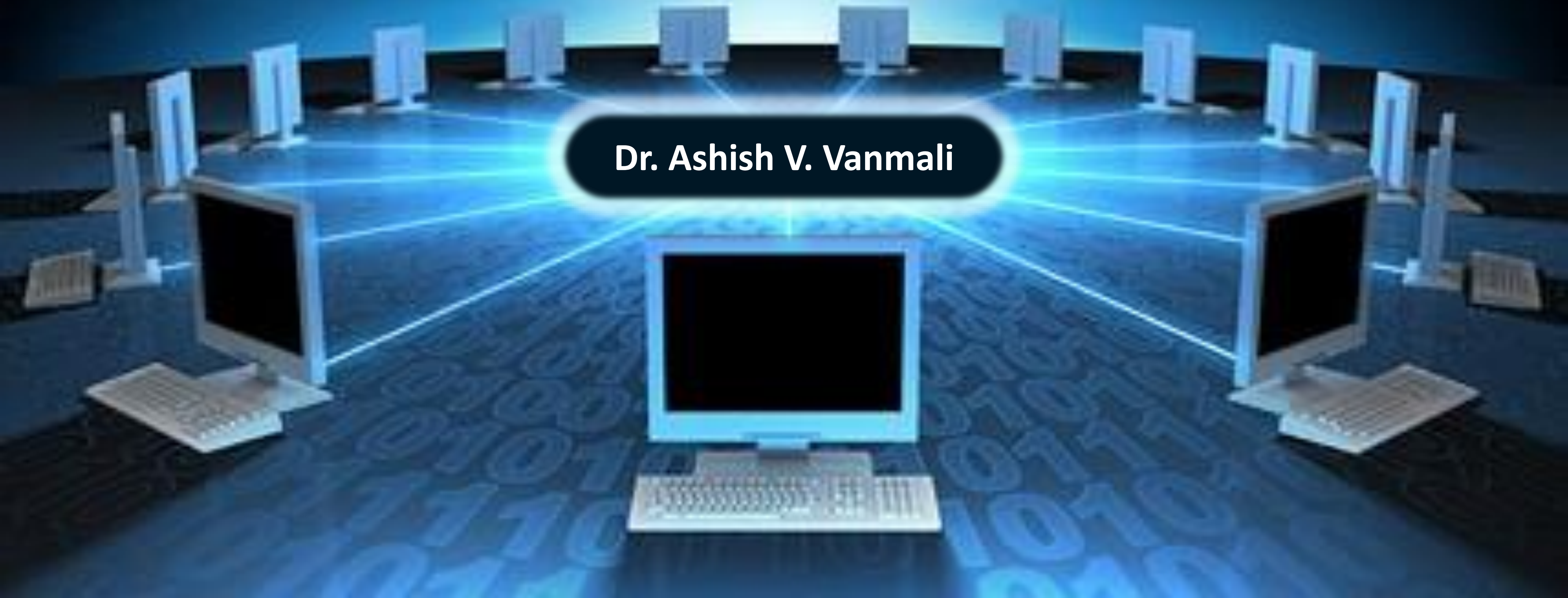
Chapter 5

Presentation Layer & Application Layer

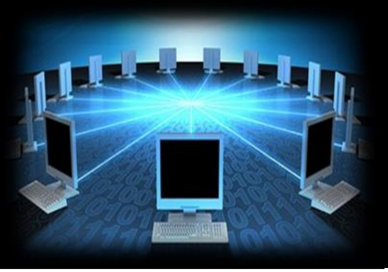
Dr. Ashish V. Vanmali



Outline of the Chapter

- **Presentation Layer**
  - Compression
  - Lossless & Lossy Compression
  - Compression Techniques
- **Application Layer**
  - Standard Client-Server Protocols

Presentation Layer



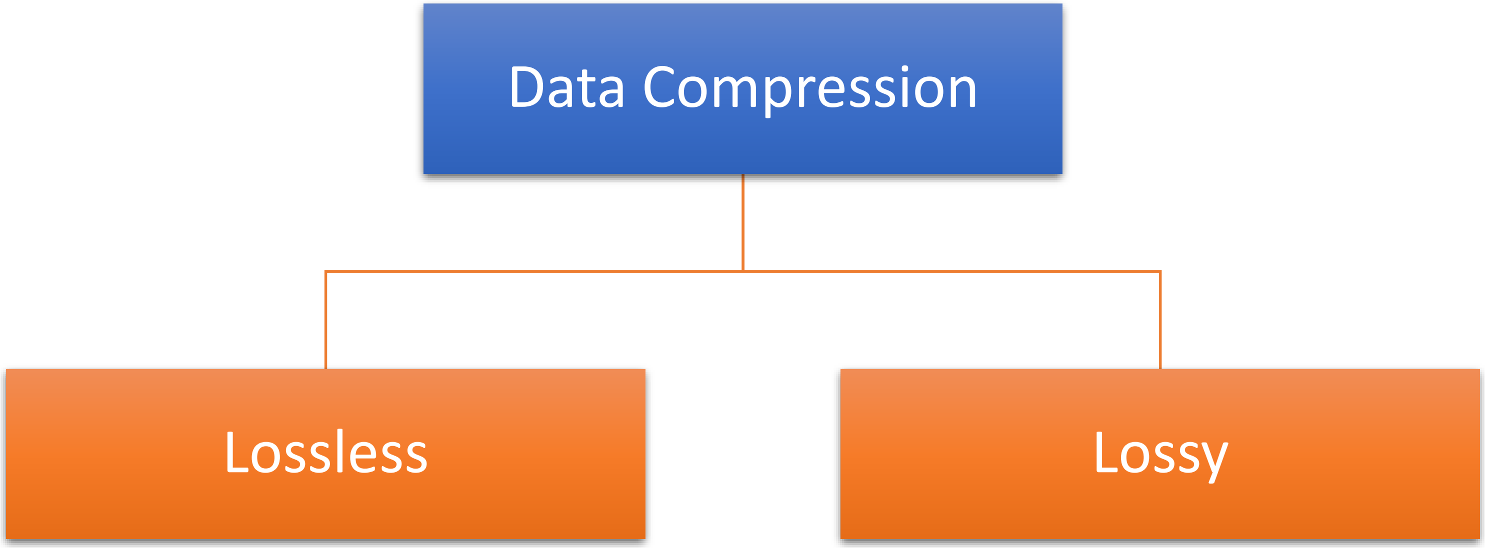
Presentation Layer

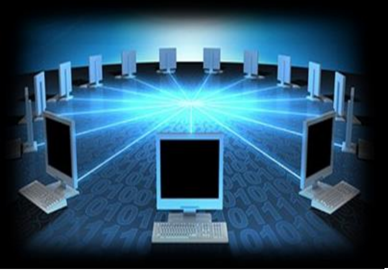
- The Presentation layer provides all the data to the application layer in a presentable format.
- It helps in facilitating tasks such as **encryption and decryption** to ensure and maintain the confidentiality of the data.
- It also helps in facilitating tasks such as **compression and decompression**.

Data Compression

- **Data compression** is a process of reduction in the number of bits needed to represent data.
- Compressing data can save storage capacity, speed up file transfer and decrease costs for storage hardware and network bandwidth.
- Compression is performed by a program that uses a formula or algorithm to determine how to shrink the size of the data.
- Data compression can dramatically decrease the amount of storage a file takes up. For example, in a 2:1 compression ratio, a 20 MB file takes up 10 MB of space.

❑ Types of Data Compression





Data Compression

❑ Lossless Data Compression

- The data is compressed in such a way that when data is decompressed it is exactly same as it was before compression, i.e. there is no loss of data.
- E.g.: Text
- Methods of Lossless Data Compression:
  - Huffman coding
  - Run length encoding
  - Dictionary based encoding

Data Compression

❑ Lossy Data Compression

- The data is compressed in such a way that when data is decompressed it is not exactly same as it was before compression.
- A lossy algorithm removes information that it can not be restore later on.
- E.g.: Still image, audio and video.
- Methods of Lossy Data Compression:
  - JPEG

Data Compression

Terminology

- $N_1 = Size\ of\ uncompressed\ Data$
- $N_2 = Size\ of\ compressed\ Data$

$$Compression\ Ratio = C_R = \frac{N_1}{N_2}$$

- Thus, a representation that compresses a file's storage size from 10 MB to 2 MB has a compression ratio of 10/2 = 5, often notated as an explicit ratio, 5:1

$$Space\ Saving = Redundancy = R_D = 1 - \frac{1}{C_R} = 1 - \frac{N_2}{N_1}$$

- Thus, a representation that compresses the storage size of a file from 10 MB to 2 MB yields a space saving of 1 - 2/10 = 0.8, often notated as a percentage, 80%.

❑ Terminology

- $P_i = Probability\ of\ occurance\ of\ i^{th}\ symbol$
- $L_i = Length\ of\ the\ code\ for\ i^{th}\ symbol$

$$L_{avg} = \sum_i L_i \cdot P_i$$

❑ Terminology

Data	P_i	Code 1	L_i	Code 2	L_i
0	0.19	000	3	11	2
1	0.25	001	3	01	2
2	0.21	010	3	10	2
3	0.16	011	3	001	3
4	0.08	100	3	0001	4
5	0.06	101	3	00001	5
6	0.03	110	3	000001	6
7	0.02	111	3	000000	6

For Code 1

$$L_{avg,1} = 3 \text{ bits}$$

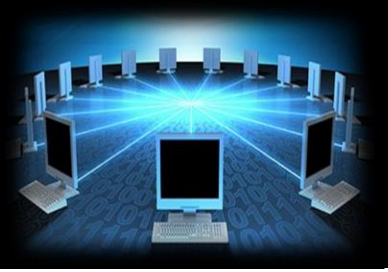
For Code 2

$$L_{avg,2} = 2.7 \text{ bits}$$

$$\therefore C_R = \frac{L_{avg,1}}{L_{avg,\mathbf{2}}} = \frac{3}{2.7} = 1.11$$

and

$$Space\ Saving = 1 - \frac{1}{C_R} = 0.099$$



Huffman Coding

Huffman Coding

- Most popular technique for removing **coding redundancies**.
- **Lossless** coding technique.
- Huffman code is optimum - no code better than Huffman code for any random variable, i.e., Huffman code gives the minimum average length for any random variable.
- Huffman code has average length $\leq H + 1$ where H = Entropy.
- Huffman code is not unique.

Huffman Coding

Code Design

- Arrange the symbols in decreasing order of their probability.
- Assign 0 and 1 as the last bit of the last two symbols.
- Combine the last two symbols as one and assign the sum of their probabilities to the new symbol.
- Repeat this process on the new set of symbol again and again. While assigning a bit to a derived symbol, the bit is **prepended** to the codewords of all the contributing symbols.

Huffman Coding

Example

Encode the symbols with Huffman code with probability od occurrence as given in the table below.

X	Probability
1	0.20
2	0.15
3	0.25
4	0.25
5	0.15

Huffman Coding

Example

X	Code-word	Probability
3		0.25
4		0.25
1		0.20
2		0.15
5		0.15

Huffman Coding

Example

X	Code-word	Probability	
3		0.25	0.30
4		0.25	0.25
1		0.20	0.25
2	0	0.15	0.20
5	1	0.15	

Huffman Coding

Example

X	Code-word	Probability		
3		0.25	0.30	0.45
4	0	0.25	0.25	0.30
1	1	0.20	0.25	0.25
2	0	0.15	0.20	
5	1	0.15		

Huffman Coding

Example

X	Code-word	Probability			
3	1	0.25	0.30	0.45	0.55
4	0	0.25	0.25	0.30	0.45
1	1	0.20	0.25	0.25	
2	0 0	0.15	0.20		
5	0 1	0.15			

Huffman Coding

Example

X	Code-word	Probability				
3	0 1	0.25	0.30	0.45	0.55	1
4	1 0	0.25	0.25	0.30	0.45	
1	1 1	0.20	0.25	0.25		
2	0 0 0	0.15	0.20			
5	0 0 1	0.15				

Huffman Coding

Example

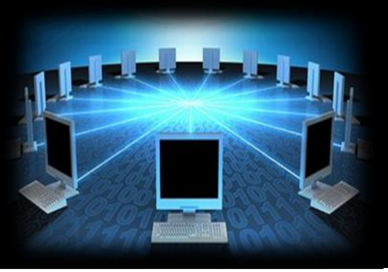
X	Probability	Code word
1	0.20	1 1
2	0.15	0 0 0
3	0.25	0 1
4	0.25	1 0
5	0.15	0 0 1

Example

Construct Huffman code for the given symbols $\{X_1, X_2, \cdots, X_8\}$ with probabilities $P(x) = \{0.07, 0.08, 0.04, 0.26, 0.14, 0.09, 0.07, 0.25\}$.

Also find the coding efficiency.

LZW Coding



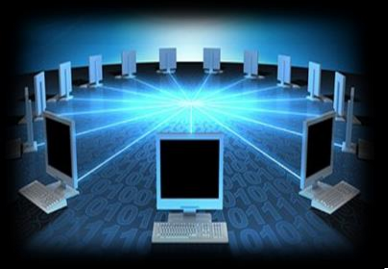
LZW Coding

- **LZW = Lempel-Ziv-Welch**
- Requires no a priori knowledge of the probability function.
- Assigns fixed length code-word to variable length sequences.
- **Lossless** coding technique.
- Uses **dictionary making technique**.
- Sequential encoding / decoding ⇒ reduces delay.
- Used in many imaging file formats e.g. GIF (Graphic interchange format), TIFF (Tagged image file format), PDF (Portable document format)

LZW Coding

Code Design

1. Maintain a list of substrings appeared before.
2. Store the output string from the source in a buffer.
3. Check if the present string at the buffer is there in the list.
   a) If yes, then wait for one more symbol to come into the buffer and then go back to step 3.
   b) If no, then
      i. Find the substring (buffer string excluding the last symbol) in the list and transmit its index.
      ii. Update the next location of the dictionary with buffer string.
      iii. Empty the buffer.



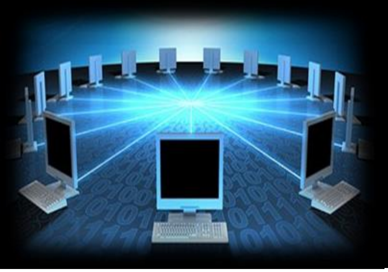
LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Initial Dictionary

Index	Entry
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	m



LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				

Push the current symbol in Buffer

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b					

String

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
$\boxed{b}$		Yes			

String entry is present in dictionary.
Take the next symbol.

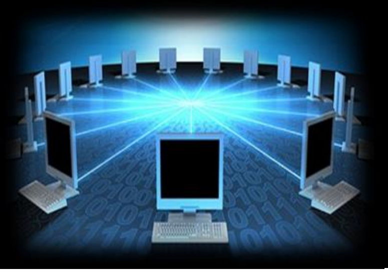
LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
$\boxed{b}$	c	Yes			

Next Symbol



LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
$\boxed{b}$	c	Yes			

No Output

Push the current symbol to buffer

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
$\boxed{\text{b c}}$					

String

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No			

String entry is **Not** present in dictionary.

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
bc		No			

Substring (buffer string excluding the last symbol)

LZW Coding

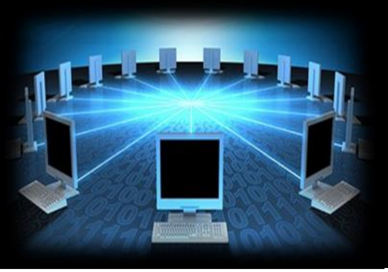
Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
bc		No			1

Substring (buffer string excluding the last symbol)

Send the index of the substring from the dictionary



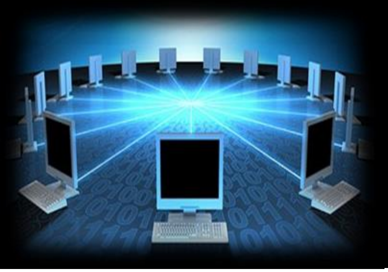
LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
$\boxed{\text{b c}}$		No	9	b c	1

Add the buffer string to the next location of dictionary



LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c					

Flush the encoded substring from buffer.
(It will now contain only the last symbol of the buffer)

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
$\boxed{c}$ String					

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c		Yes			

String entry is present in dictionary.
Take the next symbol.

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
$\boxed{\text{c d}}$					

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
$\boxed{\text{c d}}$		No			

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
$\boxed{\text{c d}}$		No			2

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
$\boxed{\text{c d}}$		No	10	c d	2

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
c d		No	10	c d	2
$\boxed{d}$					

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
c d		No	10	c d	2
d		Yes			

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
c d		No	10	c d	2
$\boxed{d}$		Yes			

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

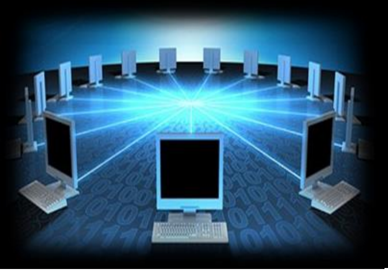
Buffer	Current Symbol	Is the string present	Index	Entry	Output
	b				
b	c	Yes			
b c		No	9	b c	1
c	d	Yes			
c d		No	10	c d	2
d	e	Yes			
$\boxed{\text{d e}}$		No			

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
d e		No	11	d e	3
e	f	Yes			
e f		No	12	e f	4
f	g	Yes			
f g		No	13	f g	5
g	h	Yes			
g h		No	14	g h	6



LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

Buffer	Current Symbol	Is the string present	Index	Entry	Output
h	b	Yes			
h b		No	15	h b	7
b	c	Yes			
b c	m	Yes			
b c m		No	16	b c m	9
m	f	Yes			
m f		No	17	m f	8

LZW Coding

Example

Construct LZW code for sequence **b c d e f g h b c m f b c**

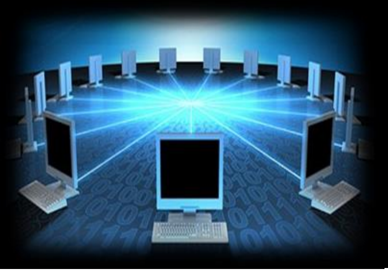
Buffer	Current Symbol	Is the string present	Index	Entry	Output
f	B	Yes			
f b		No	18	f b	5
b	c	Yes			
b c		Yes			9

Output of compressor = 1 2 3 4 5 6 7 9 8 5 9

LZW Coding

Example: Consolidated encoding table

Input	Index	Entry	Output
b	9	b c	1
c	10	c d	2
d	11	d e	3
e	12	e f	4
f	13	f g	5
g	18	g h	6
h	15	h b	7
b c	16	b c m	9
m	17	m f	8
f	18	f b	5
b c			9



LZW Coding

Example: Consolidated decoding table

Input	Decoded Symbol	Index	Entry
1	b	--	--
2	c	9	b c
3	d	10	c d
4	e	11	d e
5	f	12	e f
6	g	13	f g
7	h	18	g h
9	b c	15	h b
8	m	16	b c m
5	f	17	m f
9	b c	18	f b

LZW Coding

Example:

A message is composed of five symbols {A, B, C, D, E}.

Encode the message stream “A A B A A C E A B D A A B A A” using LZW coding.

Run Length Encoding (RLE)

Run Length Encoding (RLE)

- RLE is the simplest method of **lossless** data compression.
- In RLE, **runs** of data (sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.
- This is most efficient on data that contains many such runs, for example, simple graphic images such as icons, line drawings etc.
- Standard compression approach in facsimile (Fax) coding

Example:

Data = WWWWBBBBWWWWWWWWWWCCCDDDDBBWWWWWW

R L E = W 4 B 4 W10 C 3 D 4 B 2 W 6

Run Length Encoding (RLE)

❑ RLE as used in FAX

- FAX is a binary image consisting of only two symbols: 1 (white) and 0 (black).
- One row is coded at a time, independently.
- First bit of every row is assumed to be ‘1’.
- Transmit RLs starting with RL of ‘1’.
- If first bit is not ‘1’, first RL = 0.

Example:

Row1 : 00..(20 times) 11…(65 times) 00..(15 times)

Row2 : 11..(12 times) 00…(72 times) 11..(16 times)

Encoder output

Row1 : 0 20 65 15

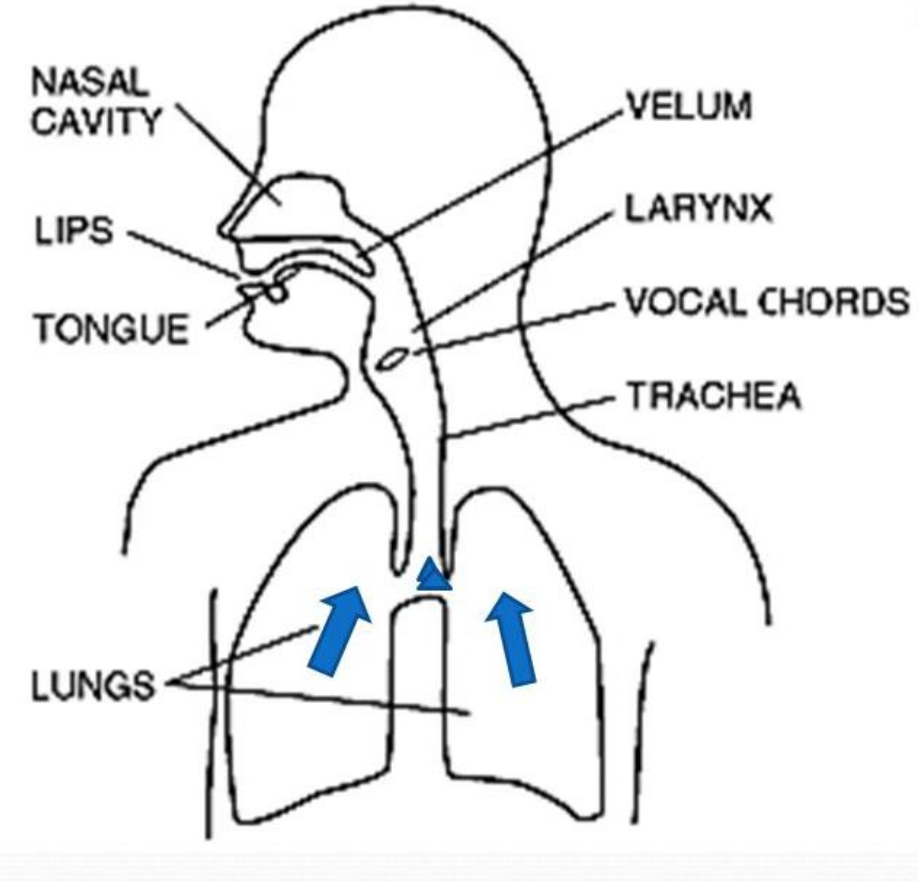
Row2 : 12 72 16

Speech Compression

Speech Compression

❑ What is Speech?

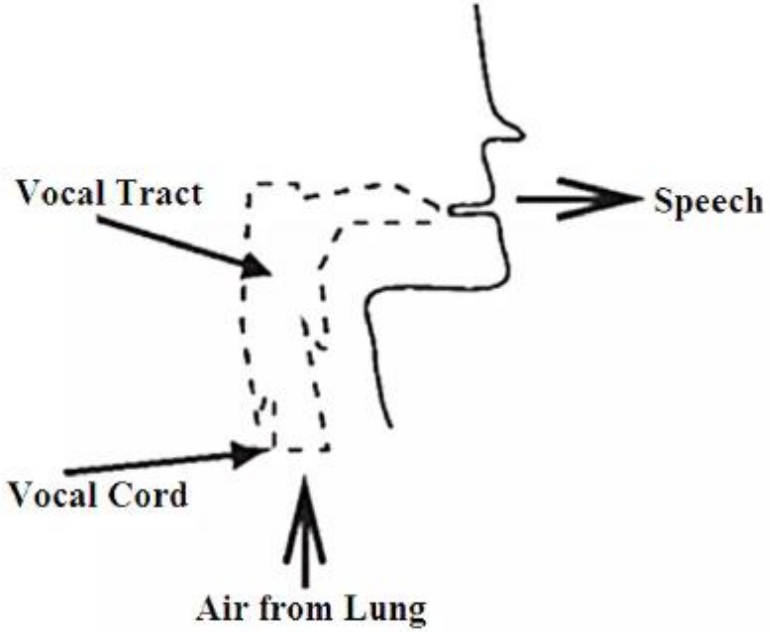
- Air is pushed from your lung through your vocal tract and out of your mouth comes speech.
- For certain voiced sound, vocal cords vibrate (open and close).
- The rate at which the vocal cords vibrate determines the **pitch** of your voice.
- For certain unvoiced sound, vocal cords do not vibrate but remain constantly opened.
- The shape of your vocal tract determines the sound that you make.



Speech Compression

- The aim of speech compression is to reduce the number of bits required to represent speech signals by removing the redundant bits so-that the less bandwidth is required for transmission.

❑ Mathematical Model for Human Speech System



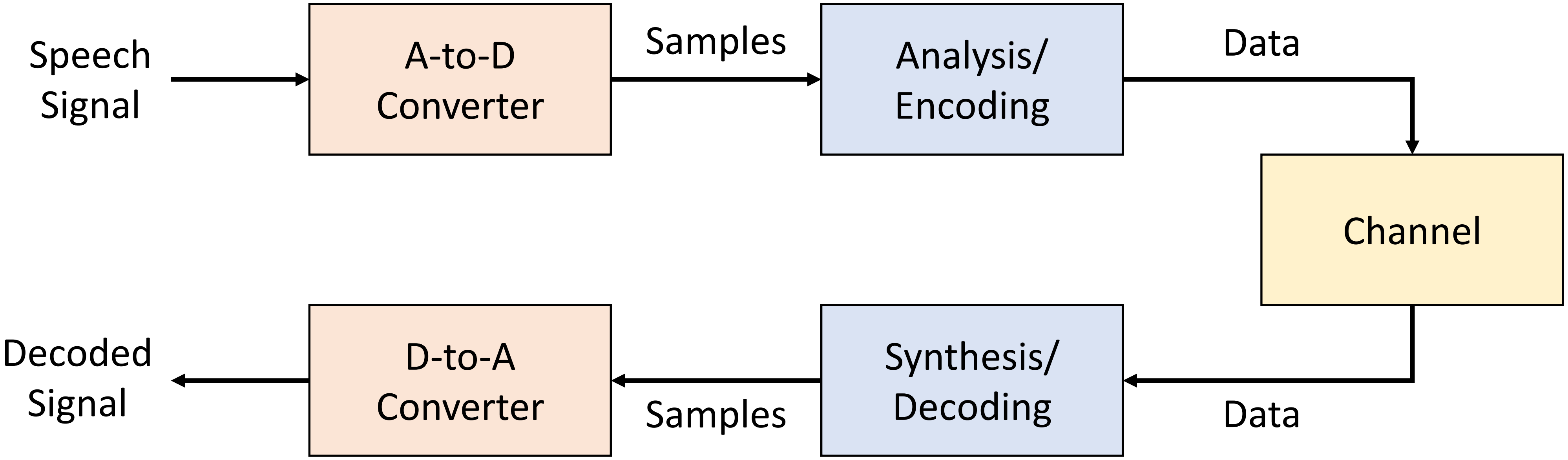
Vocal Tract	⟺	LPC Filter
Air	⟺	Innovations
Vocal Cord Vibration	⟺	Voiced
Vocal Cord Vibration Period	⟺	Pitch Period
Fricatives and Plosives	⟺	Unvoiced
Air Volume	⟺	Gain

Ref: https://www.slideshare.net/ShradheshwarVerma/speech-coding

Speech Compression

❑ Speech Coding

- It is the process to represent speech signal in the digitized form with as few bits as possible.
- It is a speech compression process.

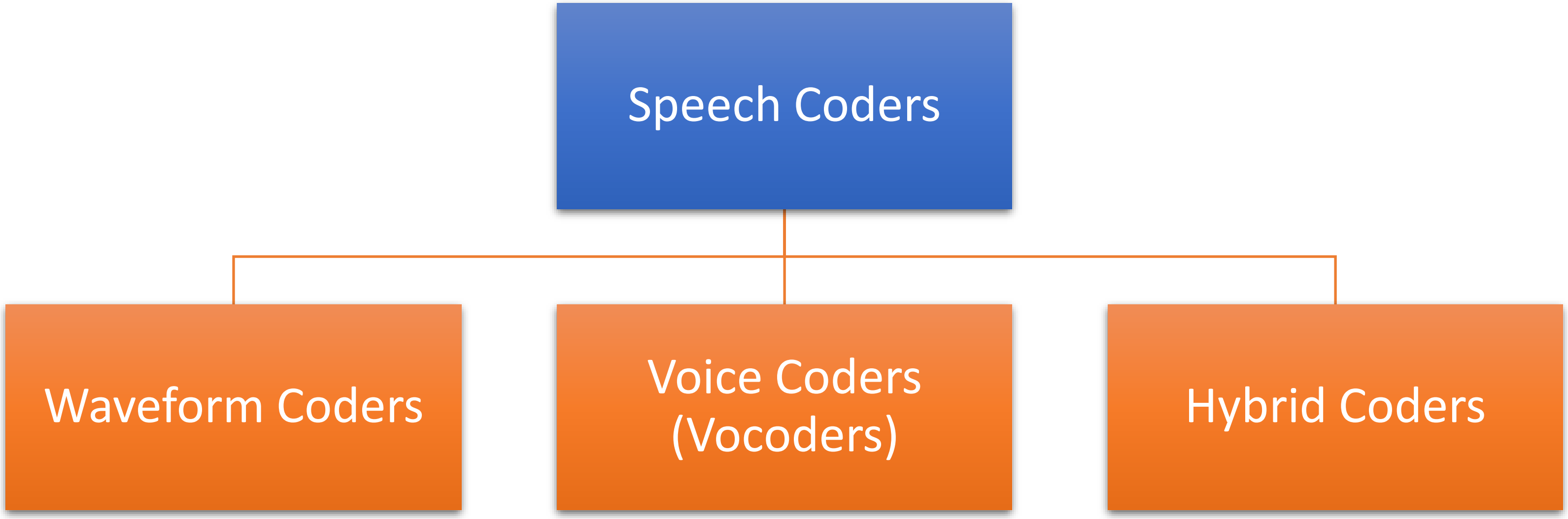


Speech Compression

❑ Speech Coders

- In speech coders, speech is represented in the form of a code and the code is stored or transmitted.
- Speech compression essentially means implementation of speech coders.

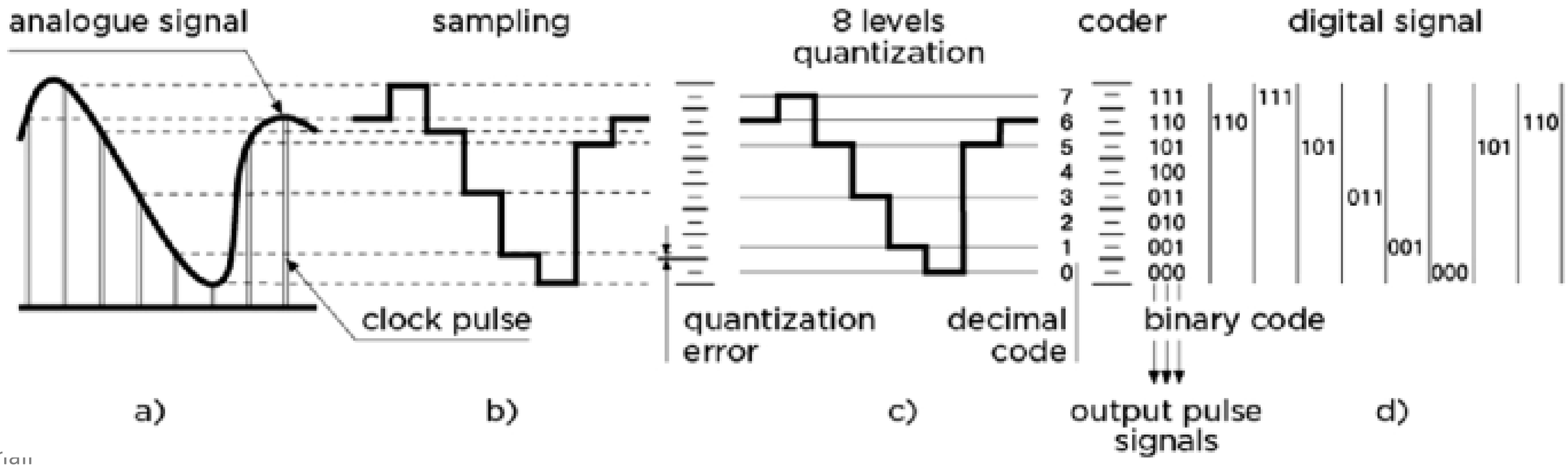
❑ Types of Speech Coders



Speech Compression

Waveform Coders

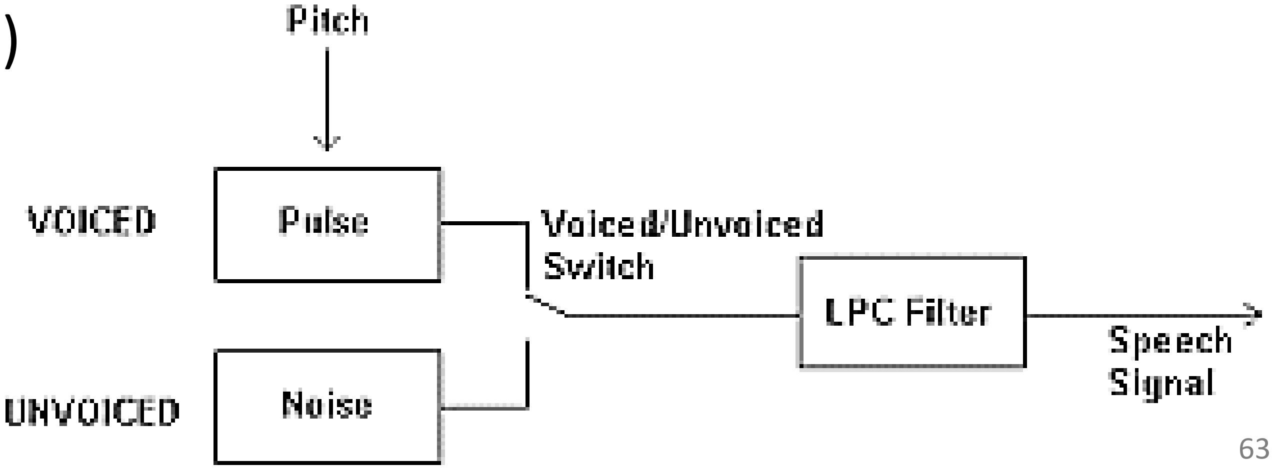
- Convert any analog signal to digital - basically A/D converter.
- Analog signal sampled with twice highest frequency and then quantized into ` n ‘ bit samples.
- Quantization can be using uniform quantization or non- uniform compandor.
- Example: Pulse Code Modulation



Speech Compression

Vocoders

- Models the vocalization of speech
- Speech sampled and broken into frames (~25 msec)
- Instead of transmitting digitized speech
  1. Build model of speech
  2. Transmit parameters of model
  3. Synthesize approximation of speech
- Example: Linear Predictive Coders (LPC)



Speech Compression

Hybrid Coders

- Hybrid coders combine the strength of waveform encoder and vocoder.
- Additional parameters of the model are optimized such that the decoded speech is as close as possible to the original waveform.
- Example: Codebook Excited LPC

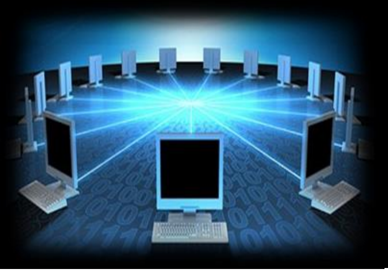


Image Compression

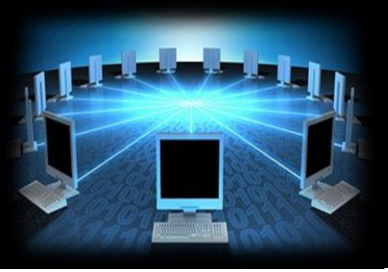
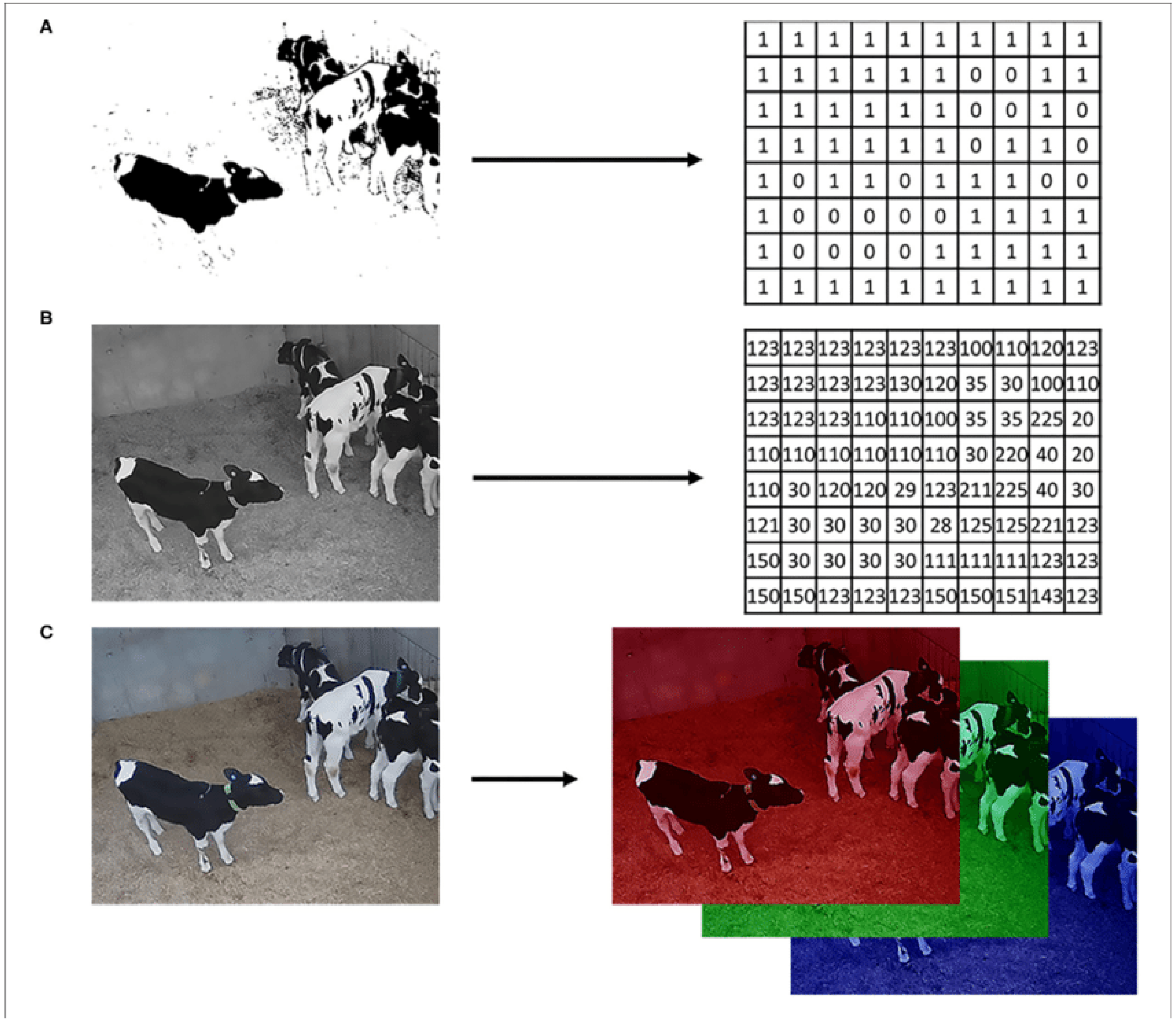


Image Compression

❑ What is Image?

- A digital image is a representation of a real image as a set of numbers that can be stored and handled by a digital computer.
- Digital images are made of picture elements called pixels.
- Binary image = 1 bit/pixel
  Monochrome image = 8 bits/pixel
  Color image = 24 bits/pixel



Ref: https://www.researchgate.net/figure/Digital-image-representation-A-Logical-image-with-values-1-for-white-and-0-for_fig2_344785470

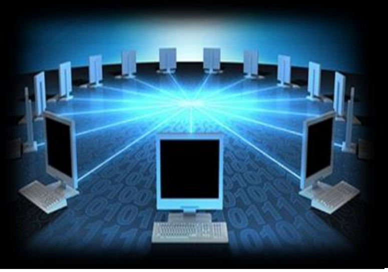


Image Compression

Why Image Compression?

- Huge size of digitized images.
- E.g.
  - Monochrome image (Grayscale image)

    Size = 1024 x 1024

    Storage space required = 1000 x 1000 x 8 = 8 Mb
  - Color image

    Size = 1024 x 1024

    Storage space required = 1000 x 1000 x 24 = 24 Mb
- Image compression Saves storage space / Saves transmission bandwidth ⇒ Saves money.

Image Compression

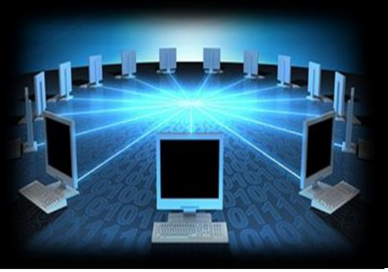
- Image compression exploits three types of redundancies:
  - Coding redundancy
  - Interpixel redundancy
  - Psychovisual redundancy

- **Coding redundancy**
  - Can be removed/reduced using optimum variable length code i.e. assign fewer bits to more probable data and than less probable data.
  - Lossless compression
  - e.g. : Huffman code, Arithmetic code

Image Compression

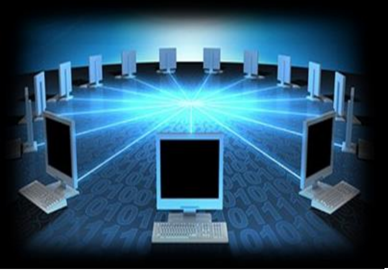
- **Interpixel redundancy**
  - Value of a pixel can be estimated using values of neighbouring pixels.
  - Lossless compression
  - e.g. : Run Length Coding

- **Psychovisual redundancy**
  - Eyes do not respond with equal sensitivity to all visual information.
  - Quantize data : eliminate some of the unneeded (quantitative) information.
  - Lossy compression
  - E.g. : IGS Coding, JPEG



Graphic Interchange Format (GIF)

- Uses LZW coding.
- Requires no a priori knowledge of the probability function.
- Assigns fixed length code-word to variable length sequences.
- **Lossless** coding technique.
- Exploits **interpixel redundancy**.
- Uses **dictionary making technique**.
- Sequential encoding / decoding ⇒ reduces delay.



Graphic Interchange Format (GIF)

Code Design

- Uses a static as well as a dynamic dictionary. (256+256 = 512 entries)
- **Static Dictionary:** Consisting an entry for every character of source = initial dictionary; available to encoder & decoder in advance.
- **Dynamic Dictionary:** Initial dictionary is then expanded depending on the patterns encountered.

Dictionary Location	Entry
0	0
1	1
:	.
255	255
256	—
.	:
511	—

Static Dictionary (locations 0–255)

Dynamic Dictionary (locations 256–511)

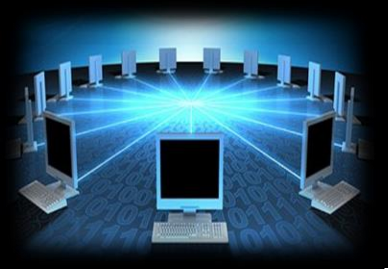
Ref. : R. C. Gonzalez, and R. E. Woods, Digital Image Processing, Pearson

Graphic Interchange Format (GIF)

Encoding of Image

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Ref. : R. C. Gonzalez, and R. E. Woods, Digital Image Processing, Pearson



Graphic Interchange Format (GIF)

Encoding of Image

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

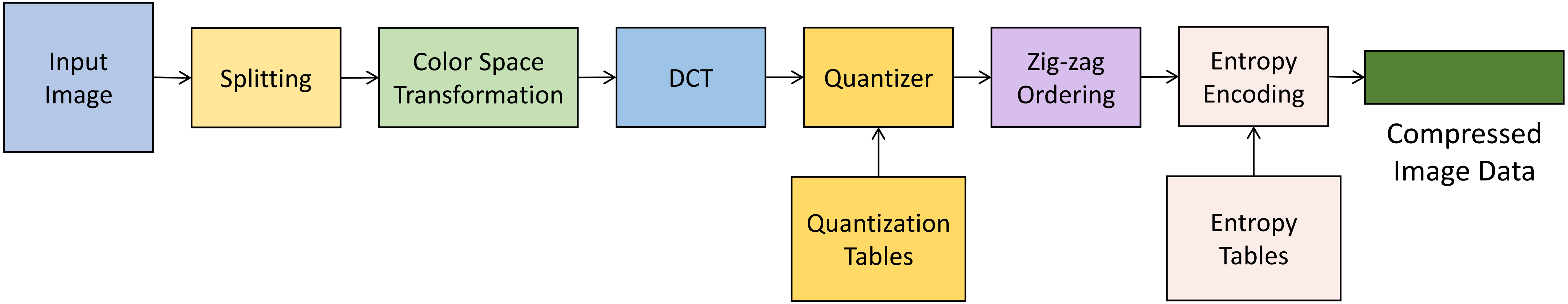
Ref. : R. C. Gonzalez, and R. E. Woods, Digital Image Processing, Pearson

JPEG

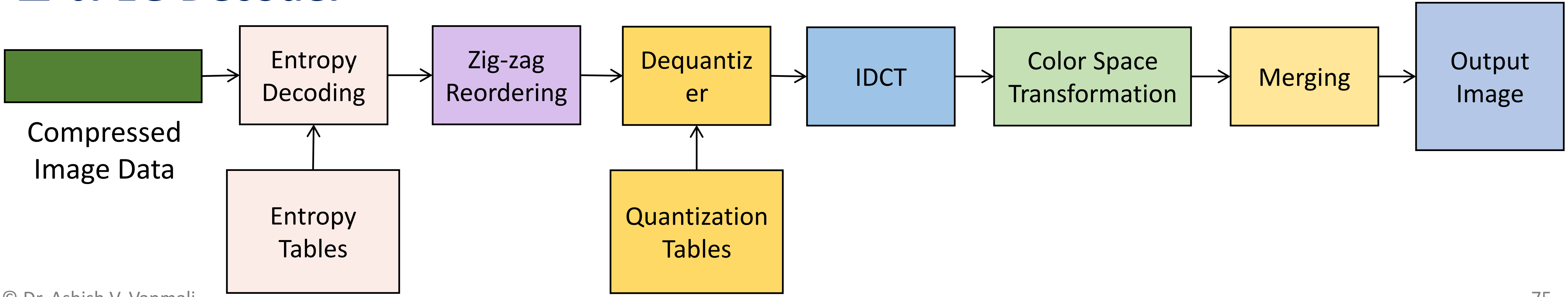
- JPEG ⇒ Joint Photographic Experts Group.
- Developed jointly by both the ITU-T and the ISO .
- First international image compression standard for continuous-tone still images - both grayscale and color images.
- It uses differential coding to form prediction residuals.
- Block-based compression.

JPEG

❑ JPEG Encoder



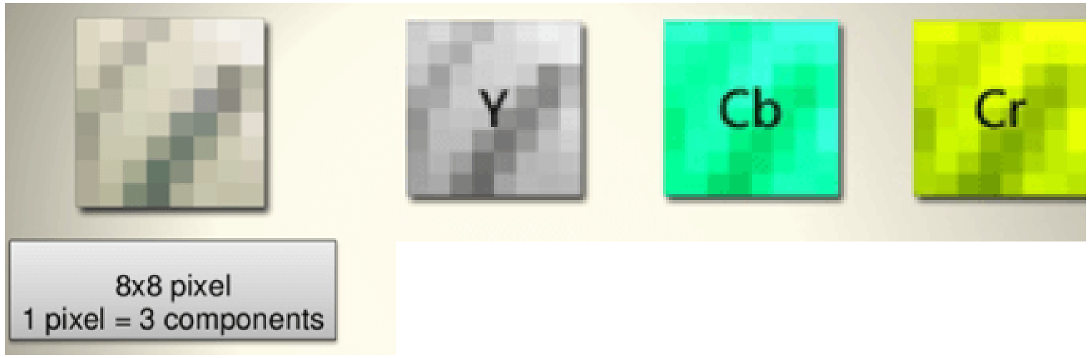
❑ JPEG Decoder



JPEG

- **Splitting:**
  - The input image is divided into a small block which is having 8x8 dimensions.

- **Color Space Transformation:**
  - [R, G, B] color space is transformed to [Y, Cb, Cr] model.
  - Here Y is for brightness, Cb is color blueness and Cr stands for Color redness.
  - We transform it into chromium colors as these are less sensitive to human eyes thus can be removed.



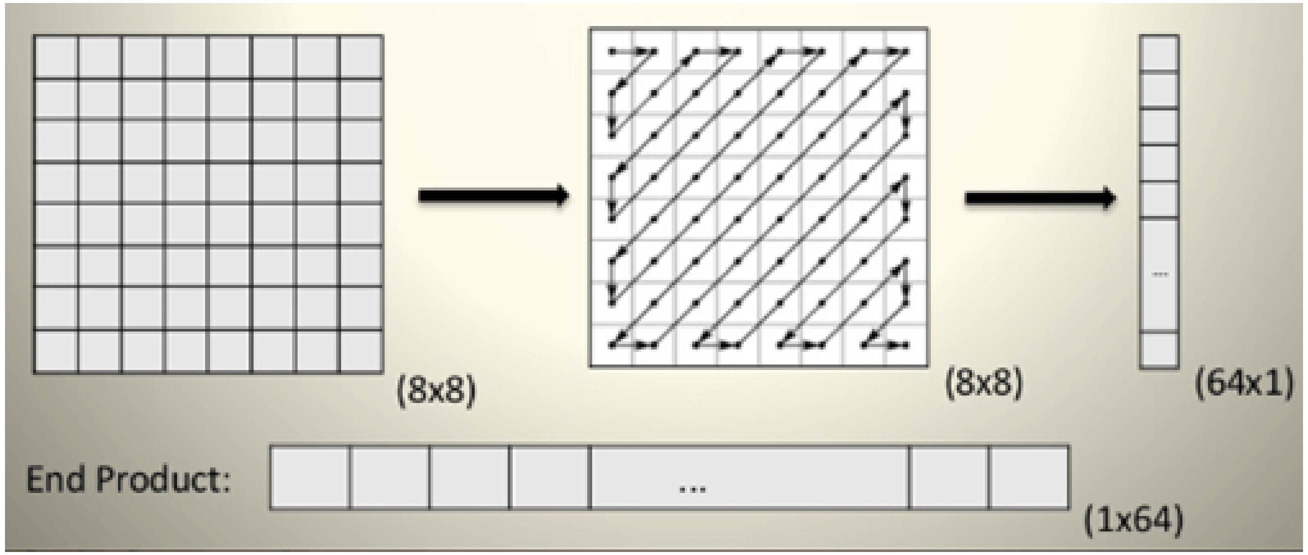
JPEG

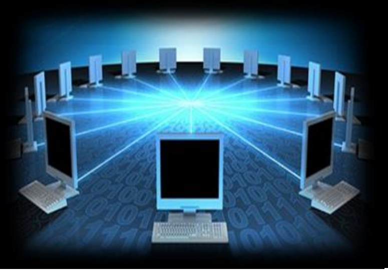
- **DCT:**
  - DCT uses a cosine function and does not use complex numbers.
  - It converts information which are in a block of pixels from the spatial domain to the frequency domain.

- **Quantization:**
  - Humans are unable to see important aspects of the image because they are having high frequencies.
  - The matrix after DCT conversion can only preserve values at the lowest frequency that to in certain point.
  - Quantization is used to reduce the number of bits per sample.

JPEG

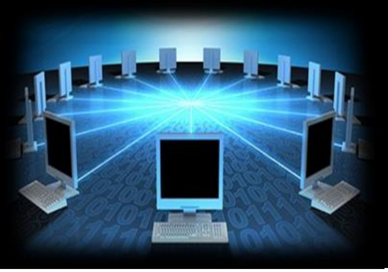
- **Zig-zag Ordering:**
  - The zigzag scan is used to map the 8x8 matrix to a 1x64 vector.
  - Zigzag scanning is used to group low-frequency coefficients to the top level of the vector and the high coefficient to the bottom.
  - To remove the large number of zero in the quantized matrix, the zigzag matrix is used.





JPEG

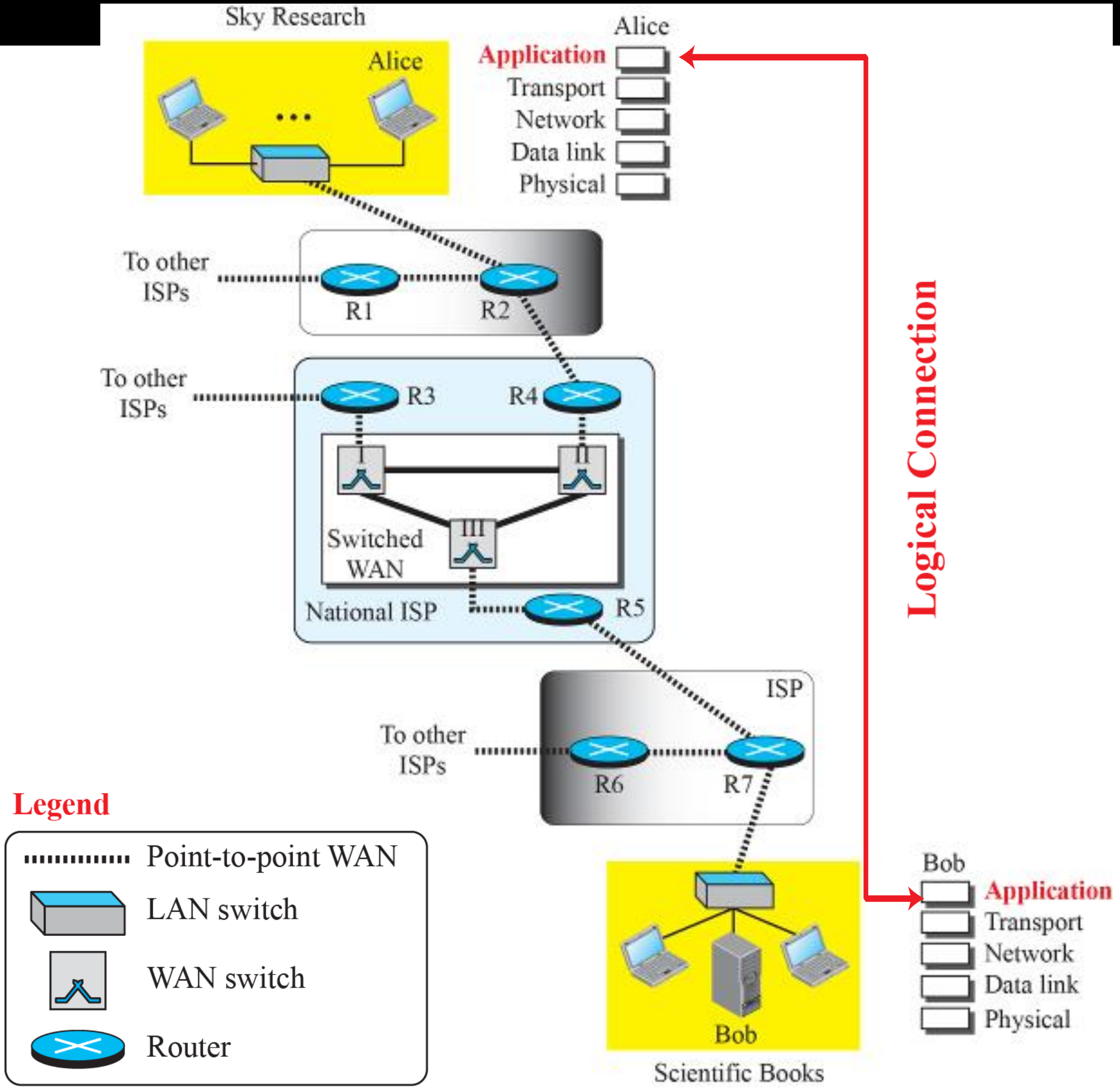
- **Entropy Coding:**
  - First element of the 1x64 vector is DC component. The remaining elements are called AC components.
  - DC component is used to define the strength of colors and AC components define texture.
  - We apply DPCM (differential pulse code modeling) on DC component and code it using Huffman coding.
  - To encode AC components, we apply Run Length Encoding (RLE).
  - This is done because AC components have a lot of zeros in it.
  - The result is a compressed image.



Application Layer

Application Layer

- The application layer provides services to the user.
- Communication is provided using a logical connection, which means that the two application layers assume that there is an imaginary direct connection through which they can send and receive messages.



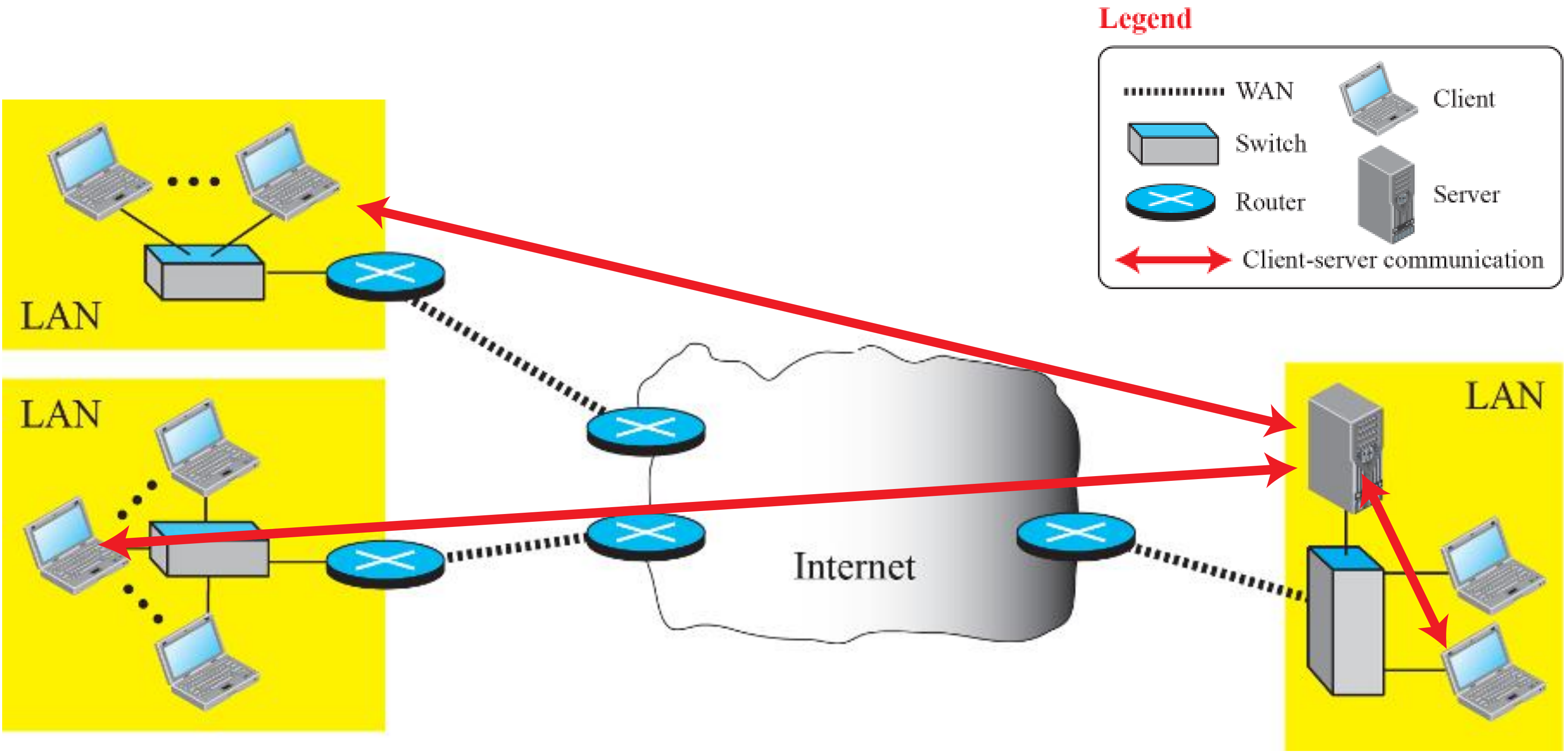
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Application Layer

❑ Application-Layer Paradigm:

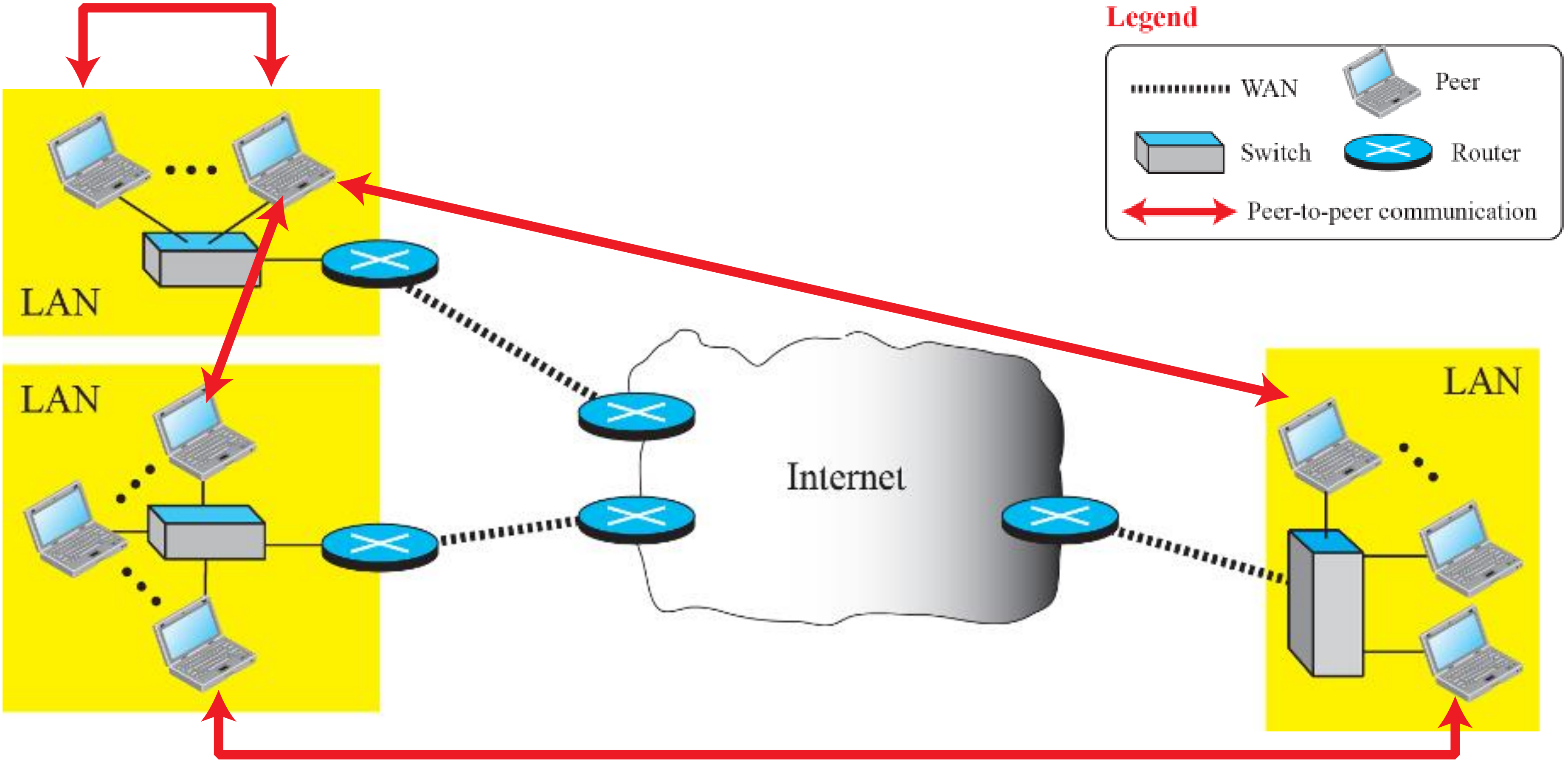
- The two programs (sender and receiver) need to send messages to each other through the Internet infrastructure.
- How they send messages define Application-Layer Paradigms:
  - Traditional Paradigm: Client-Server
  - New Paradigm: Peer-to-Peer
  - Mixed Paradigm

❑ Client-server Paradigm



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

❑ Peer-to-peer (P2P) Paradigm



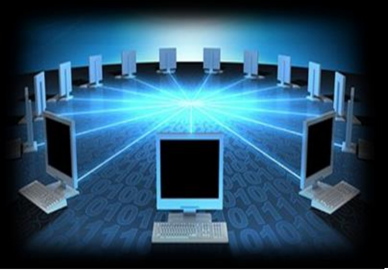
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Application Layer

❑ Client-server Applications

- In client-server paradigm, communication at the application layer is between two running application programs called processes: a client and a server.
- A **client** is a running program that initializes the communication by sending a request; a **server** is another application program that waits for a request from a client.

- Examples of Standard Client-Server Protocols:
  - HTTP
  - FTP
  - Electronic Mail
  - Domain Name System (DNS)
  - SNMP



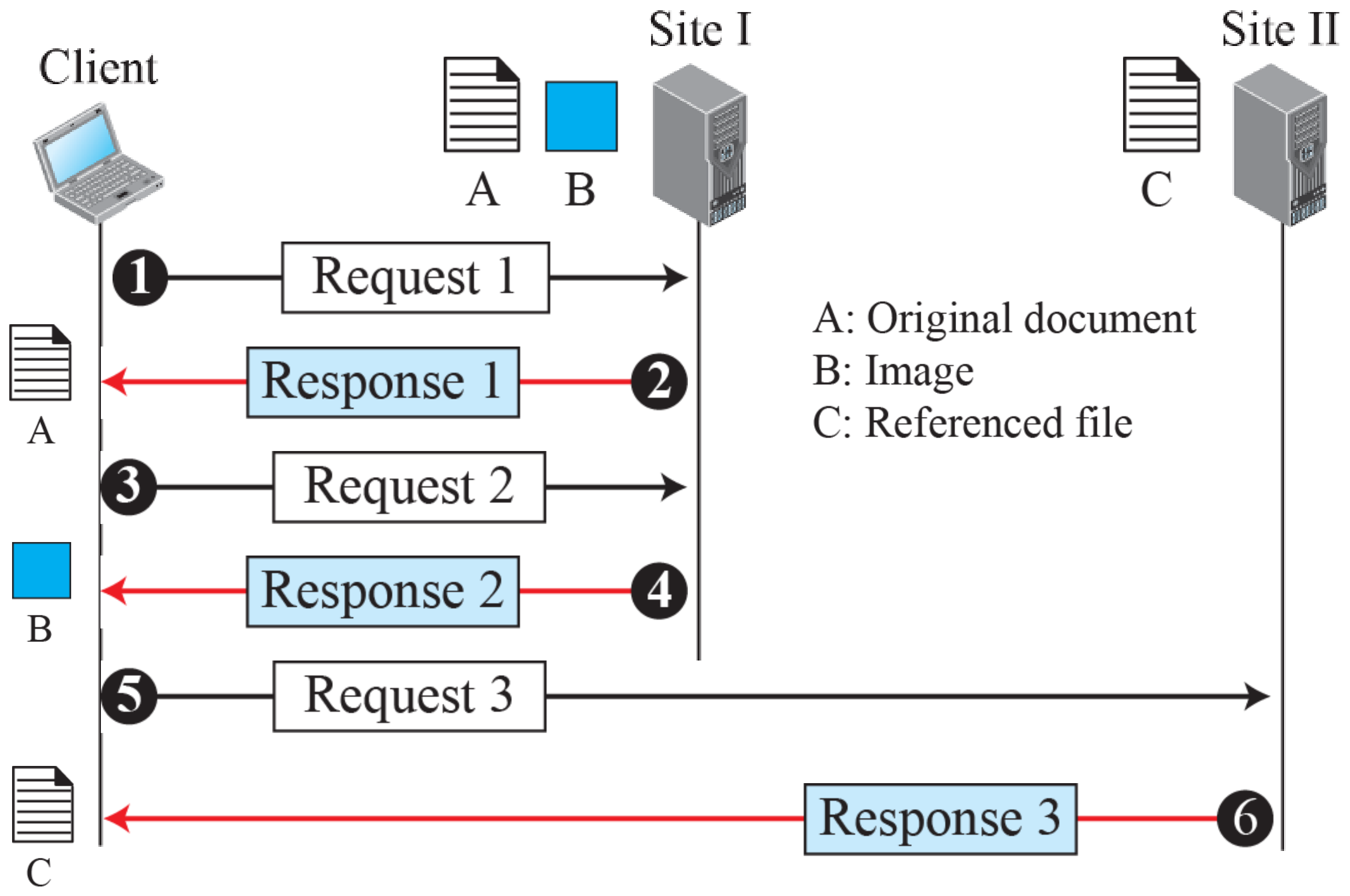
World Wide Web

- **World Wide Web (WWW)** is a repository of information in which the documents, called web pages are distributed all over the world and related documents are linked together.
- Web can be related in two terms distributed and linked.
- Linking of web pages is achieved using a concept called hypertext.

❑ Architecture of World Wide Web

- WWW is a **distributed client-server service**, in which a client using a browser can access a service using server.
- Service provided is distributed over many locations called sites.
- Each site holds one or more web pages, or links to other web pages.

❑ Architecture of World Wide Web

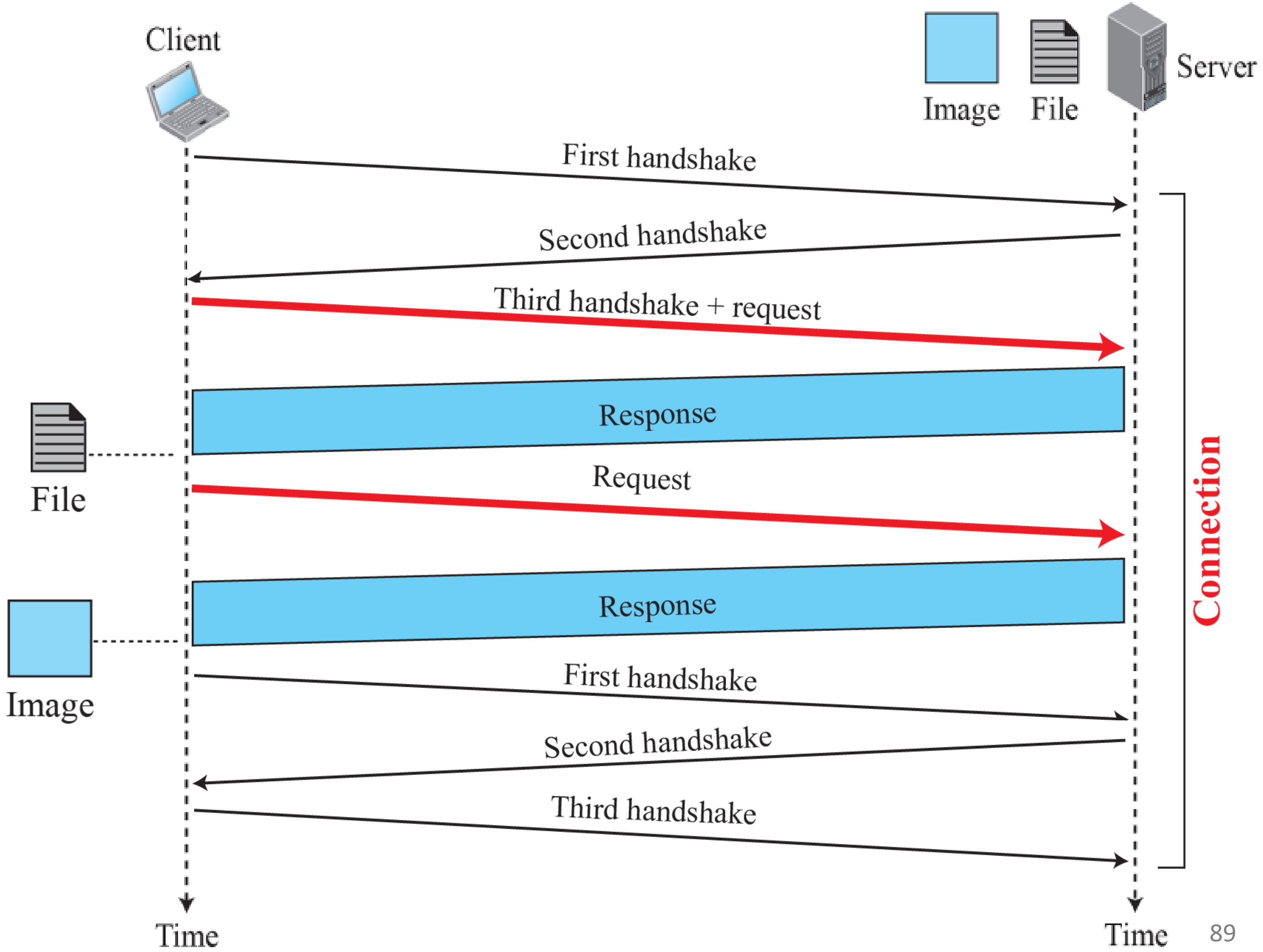


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

3 Way Handshake

- **Step 1 (SYN):** SYN(Synchronize Sequence Number) which informs the server that the client is likely to start communication and with what sequence number it starts segments with.
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with.
- **Step 3 (ACK):** In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer.

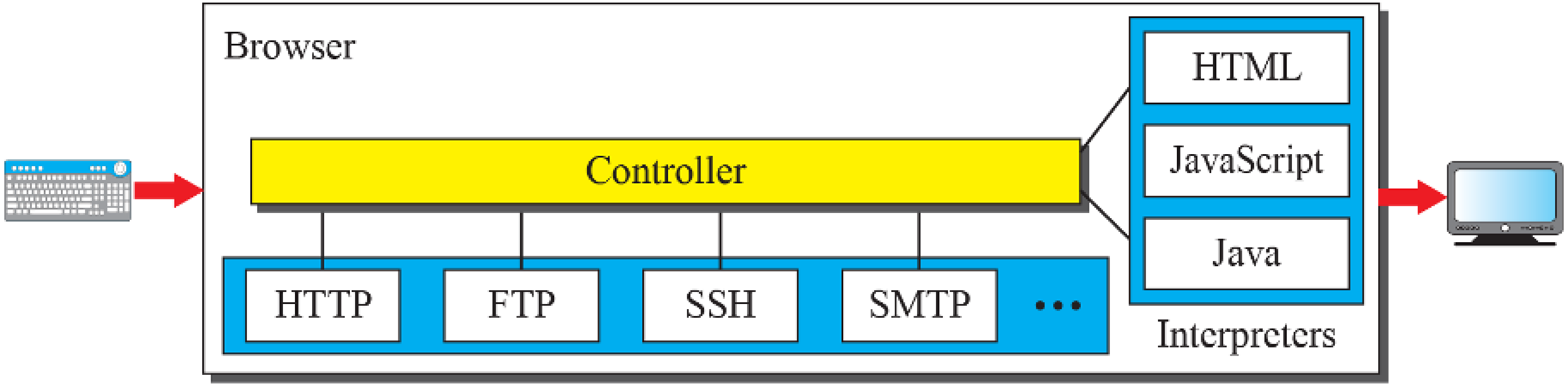
❑ 3 Way Handshake



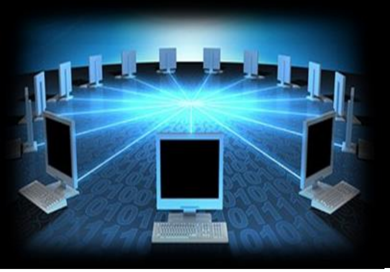
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Web Client (Browser)

- A variety of vendors offer commercial browsers that interpret and display a web page, all of them use nearly same architecture.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



World Wide Web

❑ Uniform Resource Locator (URL)

- A web page needs a unique identifier to distinguished it from other web pages.
- To define a web page 3 identifiers are used: host, port and path.
- Before defining web page, we need to tell the browser what client-server application we want to use i.e. protocol.
- Thus, we need 4 identifiers to define web page.

- protocol://host/path Used most of the time
- protocol://host:port/path Used when port number is needed

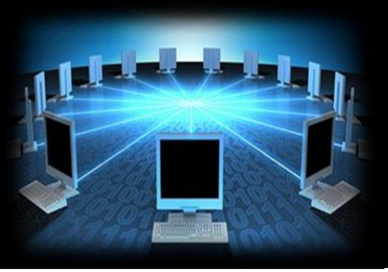
Hypertext Transfer Protocol (HTTP)

- The **Hypertext Transfer Protocol (HTTP)** is a protocol used mainly to access data on the World Wide Web.
- HTTP functions as a combination of FTP and SMTP.
- It is similar to FTP because it transfers files and uses the services of TCP.
- However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server.
- HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages.
- Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser).
- HTTP uses the services of TCP on well-known port 80.

Hypertext Transfer Protocol (HTTP)

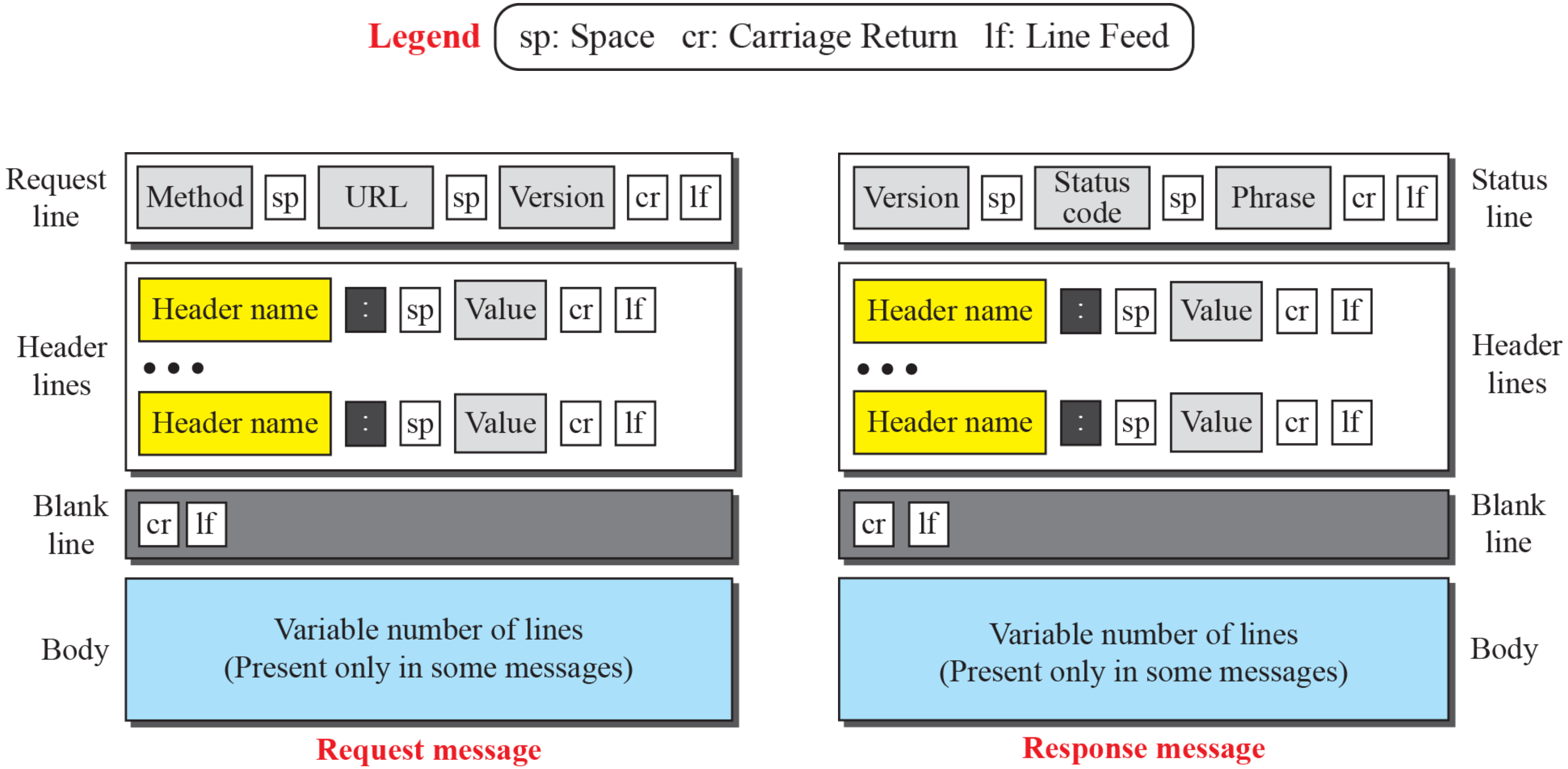
HTTP Request / Response

- Communication between clients and servers is done by **requests** and **responses**:
  1. A client (a browser) sends an HTTP request to the web
  2. A web server receives the request
  3. The server runs an application to process the request
  4. The server returns an HTTP response (output) to the browser
  5. The client (the browser) receives the response

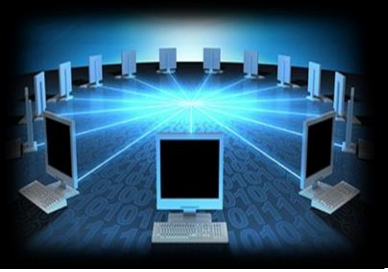


Hypertext Transfer Protocol (HTTP)

❑ HTTP Request / Response



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Hypertext Transfer Protocol (HTTP)

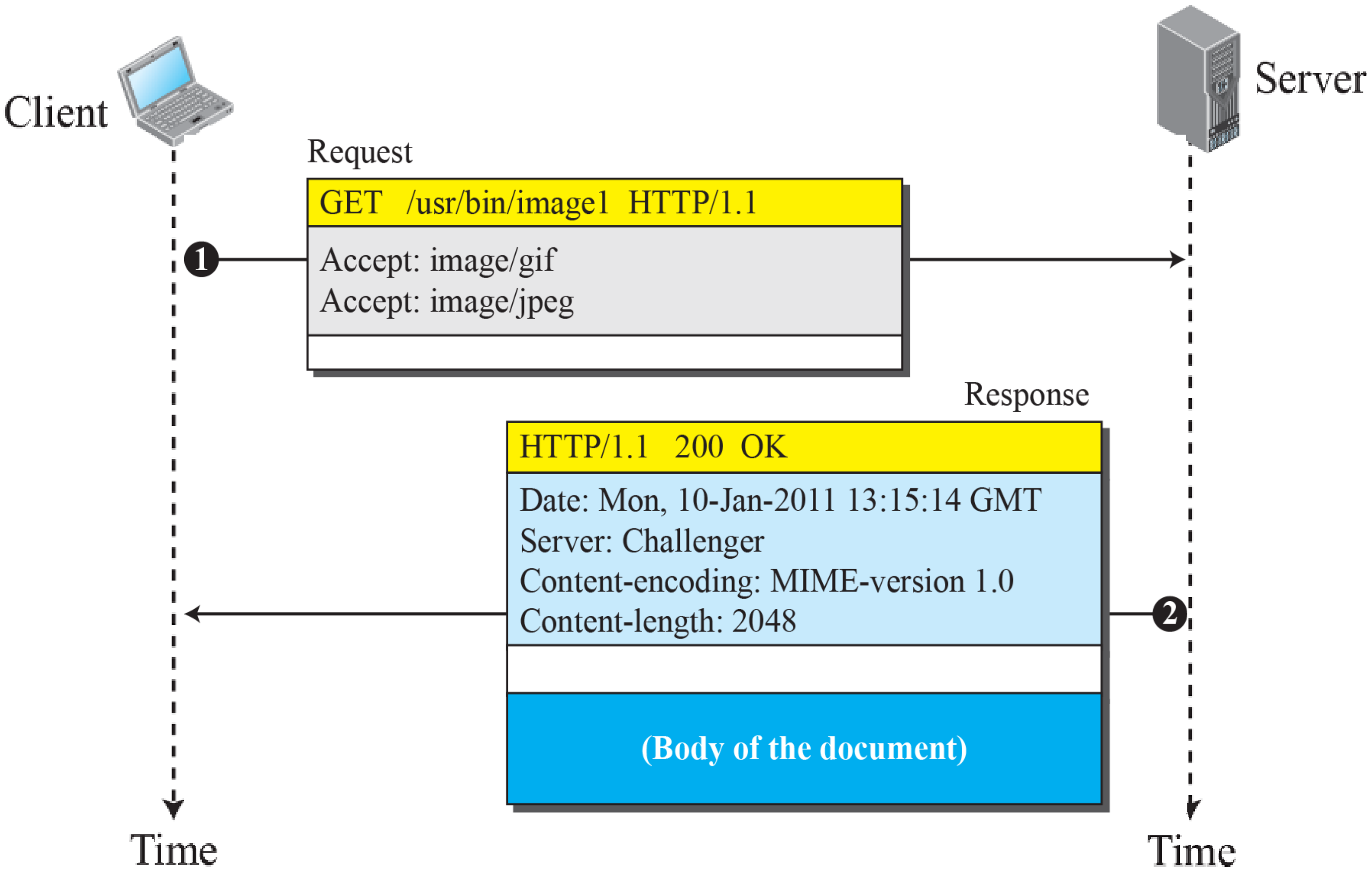
Methods

- In HTTP several request types are defined.
- These request type is categorized into **methods**.

Method	Action
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

Hypertext Transfer Protocol (HTTP)

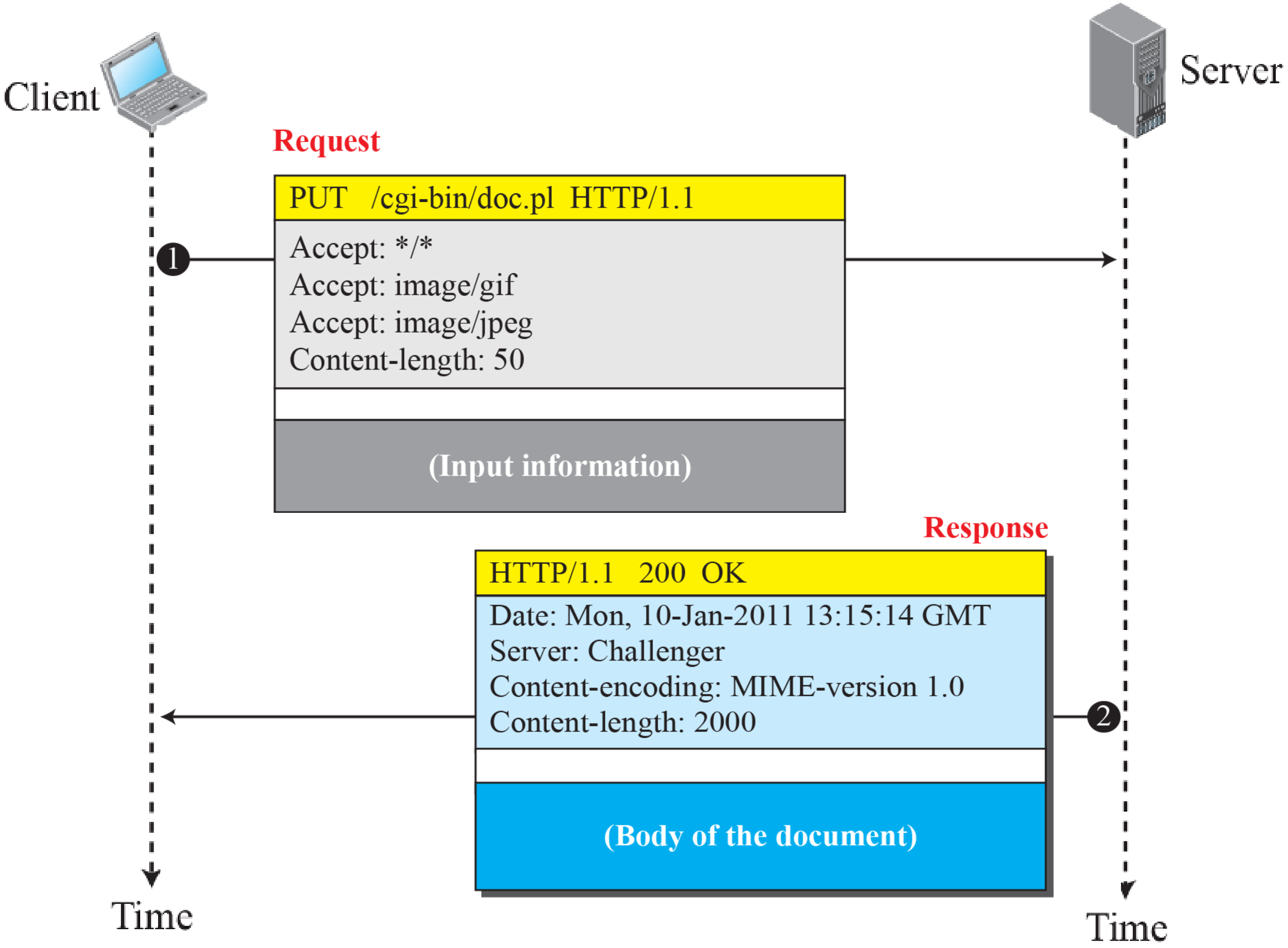
Example 1: GET method to retrieve an image with the path /usr/bin/image1.



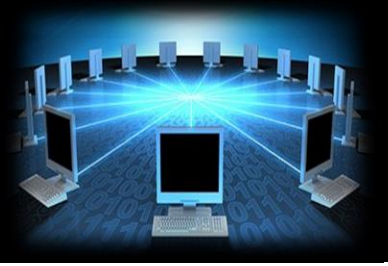
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Hypertext Transfer Protocol (HTTP)

Example 2: The client wants to send a web page to be posted on the server. It uses the PUT method.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



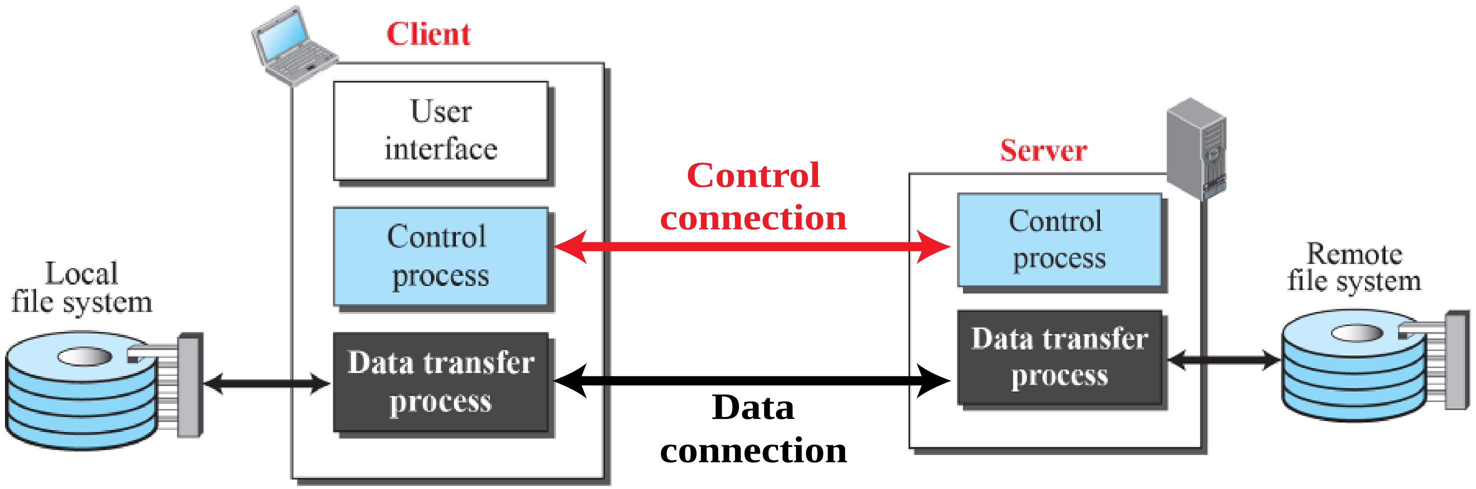
File Transfer Protocol (FTP)

- **File Transfer Protocol (FTP)** is the standard protocol provided by TCP/IP for copying a file from one host to another.
- Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first.
- For example, two systems may use different file name conventions. Two systems may have different ways to represent data.
- All of these problems have been solved by FTP in a very simple and elegant approach.

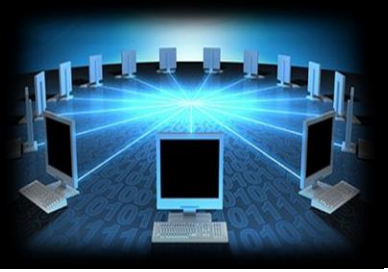
File Transfer Protocol (FTP)

FTP Connections

- FTP differs from other client/server applications in that it establishes two connections between the hosts.
- One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient.
- FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



File Transfer Protocol (FTP)

Communication over Control Connection

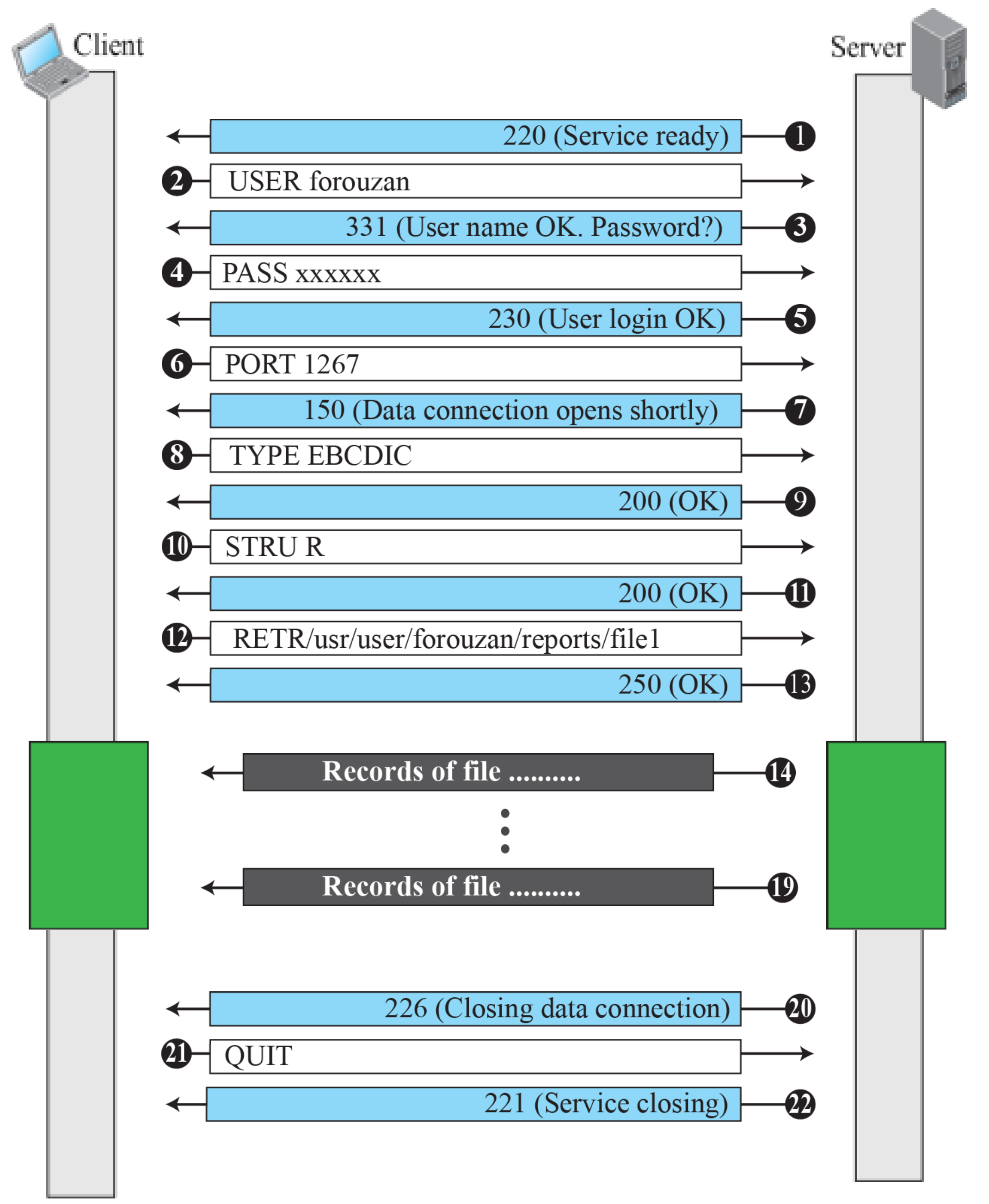
- Communication is achieved through commands and responses.
- It uses the 7-bit ASCII character set.
- Each command or response is only one short line, so we need not worry about file format or file structure.

Communication over Data Connection

- The client must define file type, data structure and transmission mode.
- File type: FTP can transfer one of the file type across the data connection- ASCII, EBCDIC or image.
- Data structure: FTP can transfer one of the following interpretations of the structure of the data: file structure, record structure or page structure.
- Transmission mode: 3 modes- stream mode, block mode or compressed mode.

File Transfer Protocol (FTP)

Example : Using FTP for retrieving a file.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

File Transfer Protocol (FTP)

FTP Security

- Although FTP requires a password, the password is sent in plaintext, which means it can be intercepted and used by an attacker.
- The data transfer connection also transfers data in plaintext, which is insecure.
- For security, Secure Socket Layer (SSL) can be added between FTP application layer and TCP layer. It is called as SSL-FTP.

Electronic Mail

- One of the most popular Internet services is electronic mail (e-mail).
- E-mail allows users to exchange messages.
- The nature of this application, however, is different from other applications.
- In an application such as HTTP or FTP, the server program is running all the time, waiting for a request from a client. When the request arrives, the server provides the service.
- In the case of electronic mail, the situation is different.

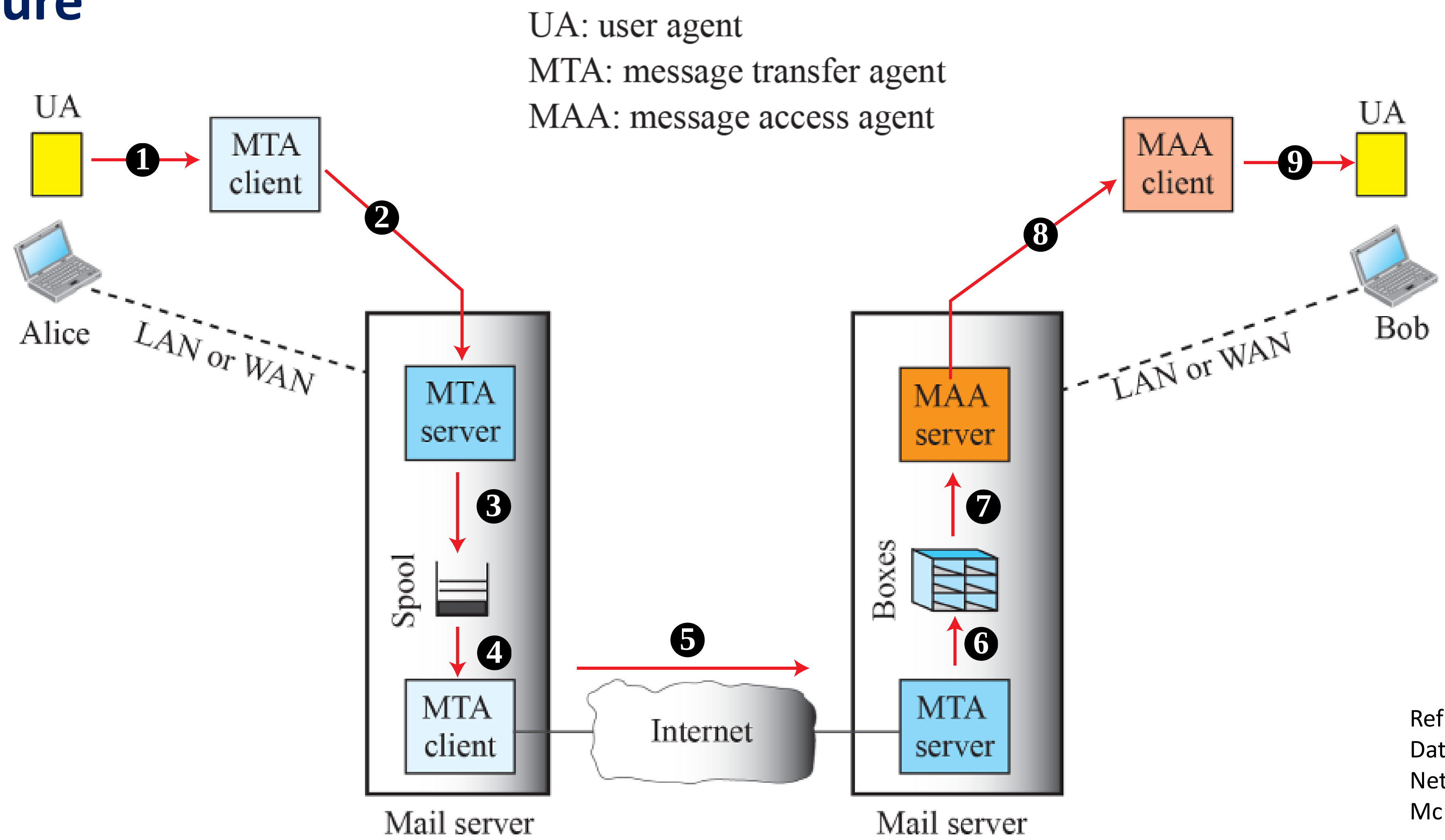
Electronic Mail

Architecture

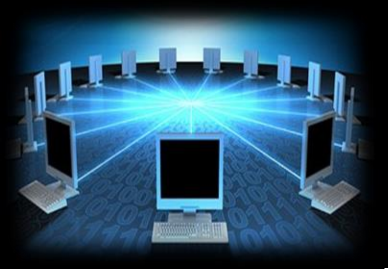
- E-mail is considered a one-way transaction.
- When the sender and the receiver of an e-mail are on the same system, we need only two user agents (UAs).
- When the sender and the receiver of an e-mail are on different systems, we need two UAs and a pair of MTAs (client and server).
- When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).
- When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

Electronic Mail

❑ Architecture



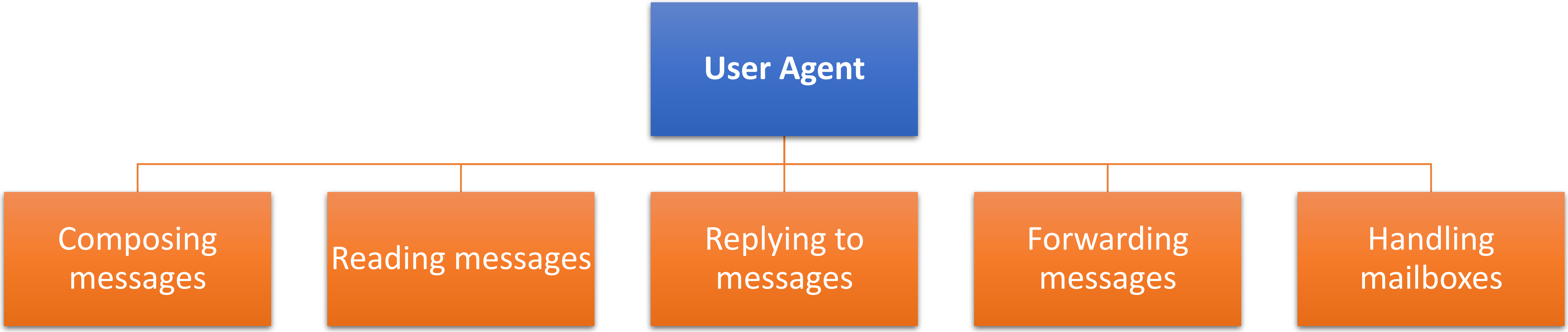
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



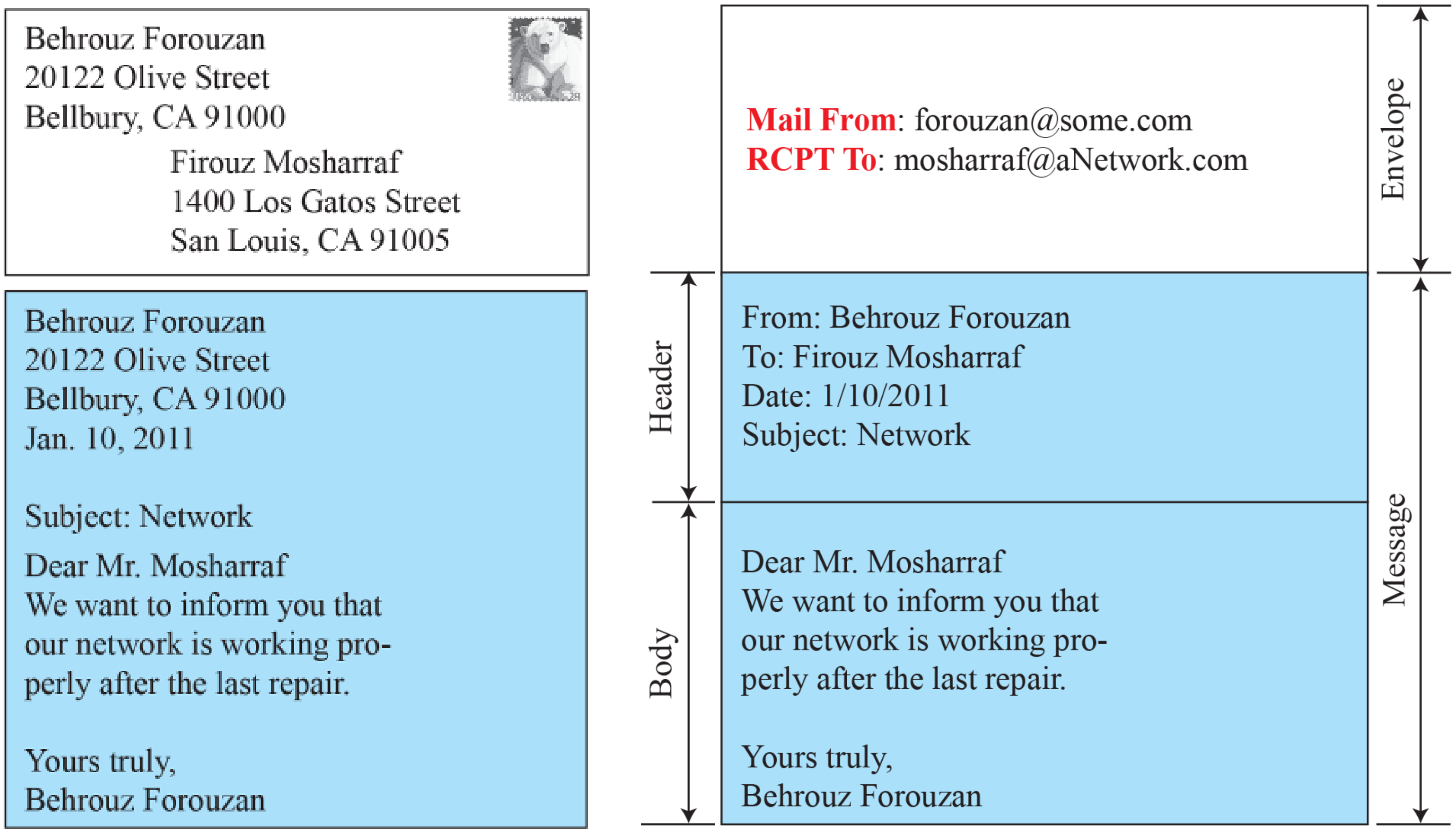
Electronic Mail

❑ Services Provided by a User Agent

- User Agent provides service to the user to make the process of sending and receiving a message easier.



❑ Format of an e-mail

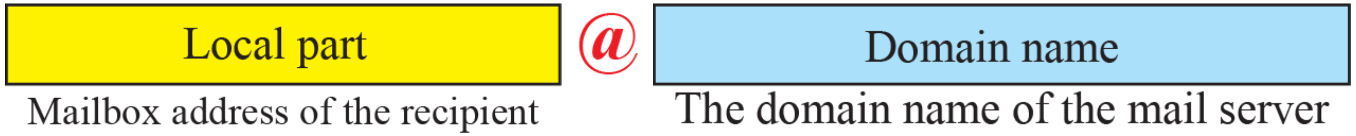


Postal mail **Electronic mail**

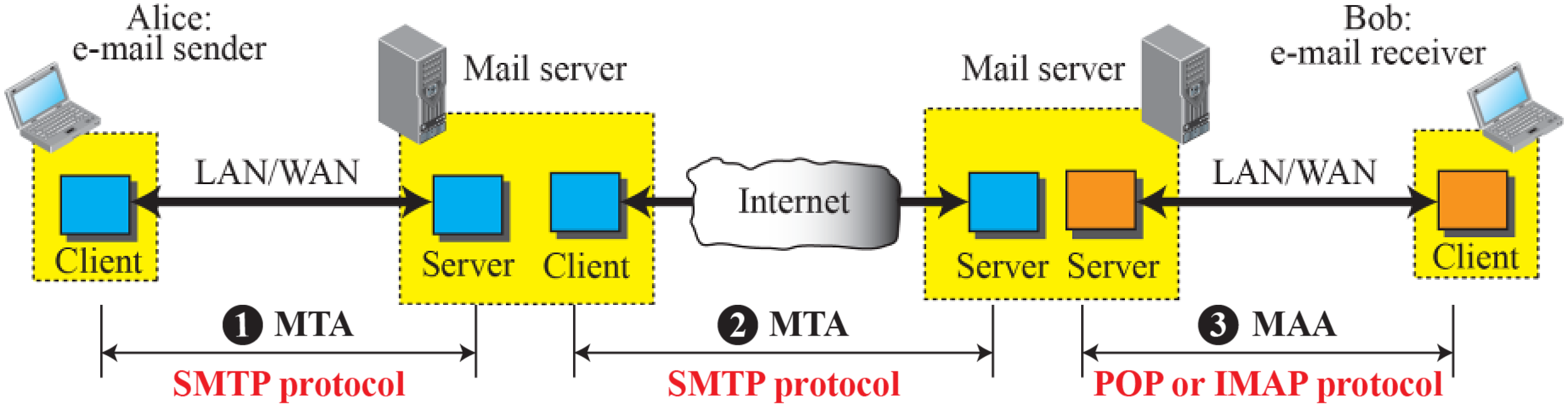
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Electronic Mail

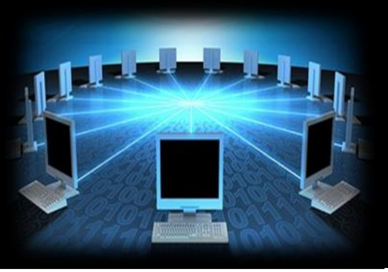
❑ E-mail address



❑ Protocols used in e-mail



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



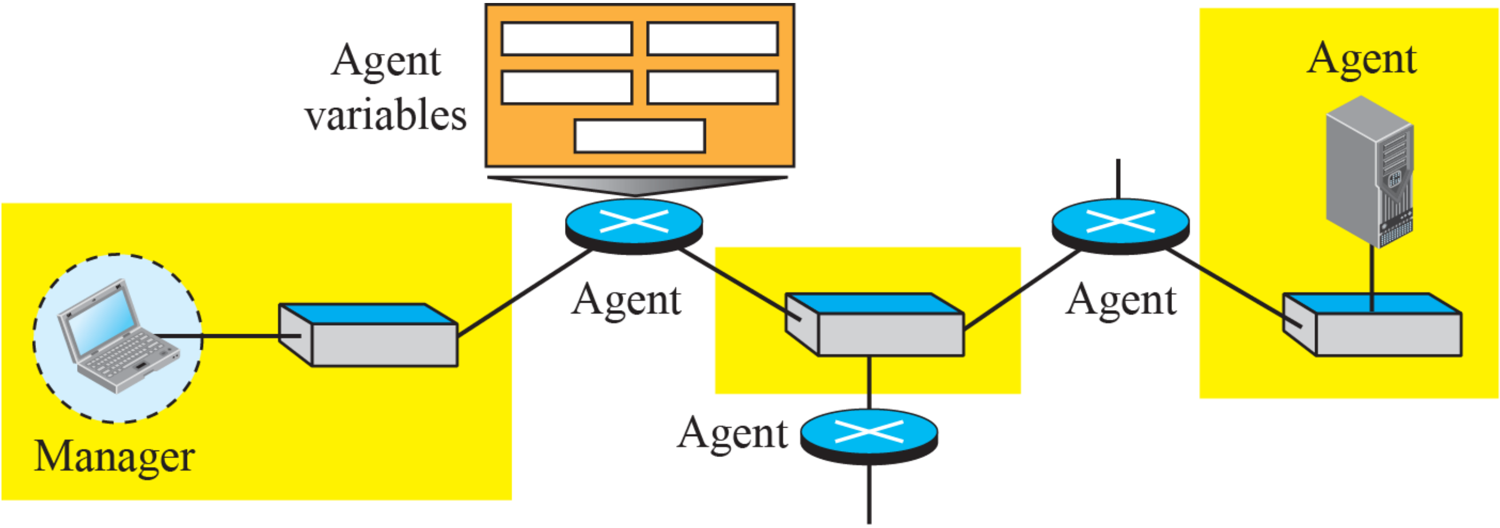
SNMP

- **Network management** as monitoring, testing, configuring, and troubleshooting network components to meet a set of requirements defined by an organization.
- These requirements include the smooth, efficient operation of the network that provides the predefined quality of service for users.
- To accomplish this task, a network management system uses hardware, software, and humans.
- Several network management standards have been devised during the last few decades.
- The most important one **is Simple Network Management Protocol (SNMP)**, used by the Internet.
- SNMP is a framework for managing devices in an internet using the TCP/IP protocol suite.

SNMP

Managers and Agents

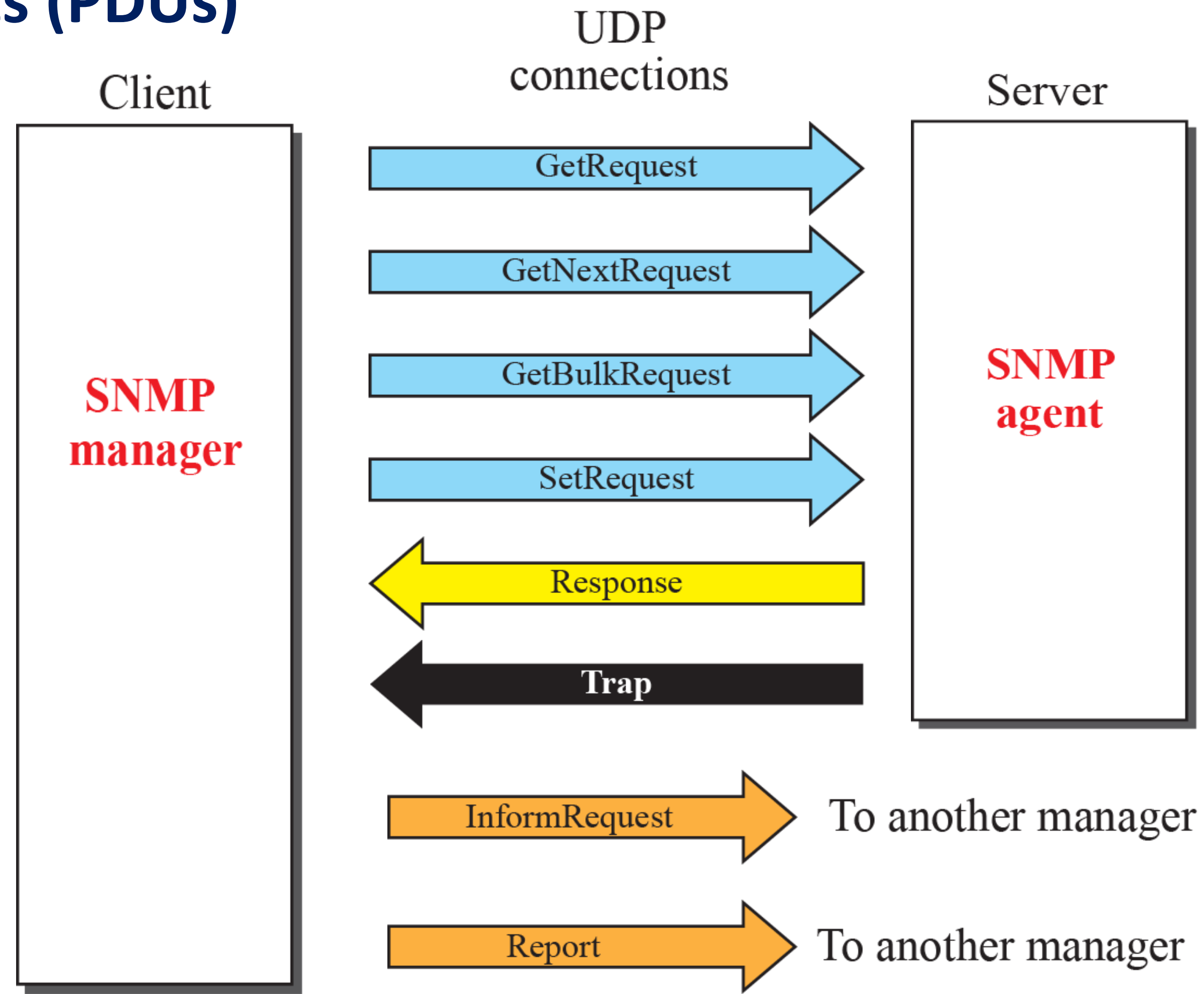
- A management station, called a **manager**, is a host that runs the SNMP client program.
- A managed station, called an **agent**, is a router (or a host) that runs the SNMP server program.
- Management is achieved through simple interaction between a manager and an agent.



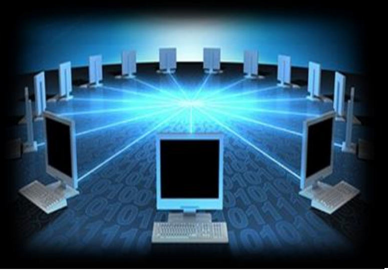
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

SNMP

❑ SNMP Protocol Data Units (PDUs)



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



SNMP

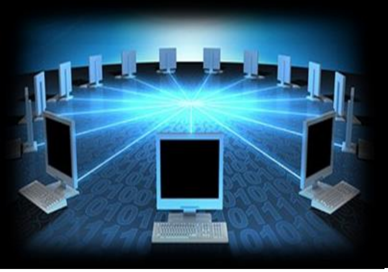
SNMP Protocol Data Units (PDUs)

- **GET Request:** Generated by the SNMP manager and sent to an agent to obtain the value of a variable, identified by its OID, in an MIB.
- **GETNEXT Request:** Sent by the SNMP manager to the agent to retrieve the values of the next OID in the MIB's hierarchy.
- **GETBULK Request:** Sent by the SNMP manager to the agent to efficiently obtain a potentially large amount of data, especially large tables.
- **SET Request:** Sent by the SNMP manager to the agent to issue configurations or commands.
- **RESPONSE:** Sent by the agent to the SNMP manager, issued in reply to a GET Request, GETNEXT Request, GETBULK Request and a SET Request. Contains the values of the requested variables.

SNMP

❑ SNMP Protocol Data Units (PDUs)

- **TRAP:** An asynchronous alert sent by the agent to the SNMP manager to indicate a significant event, such as an error or failure, has occurred.
- **Inform Request:** It was introduced in SNMPv2c, used to identify if the trap message has been received by the manager or not.

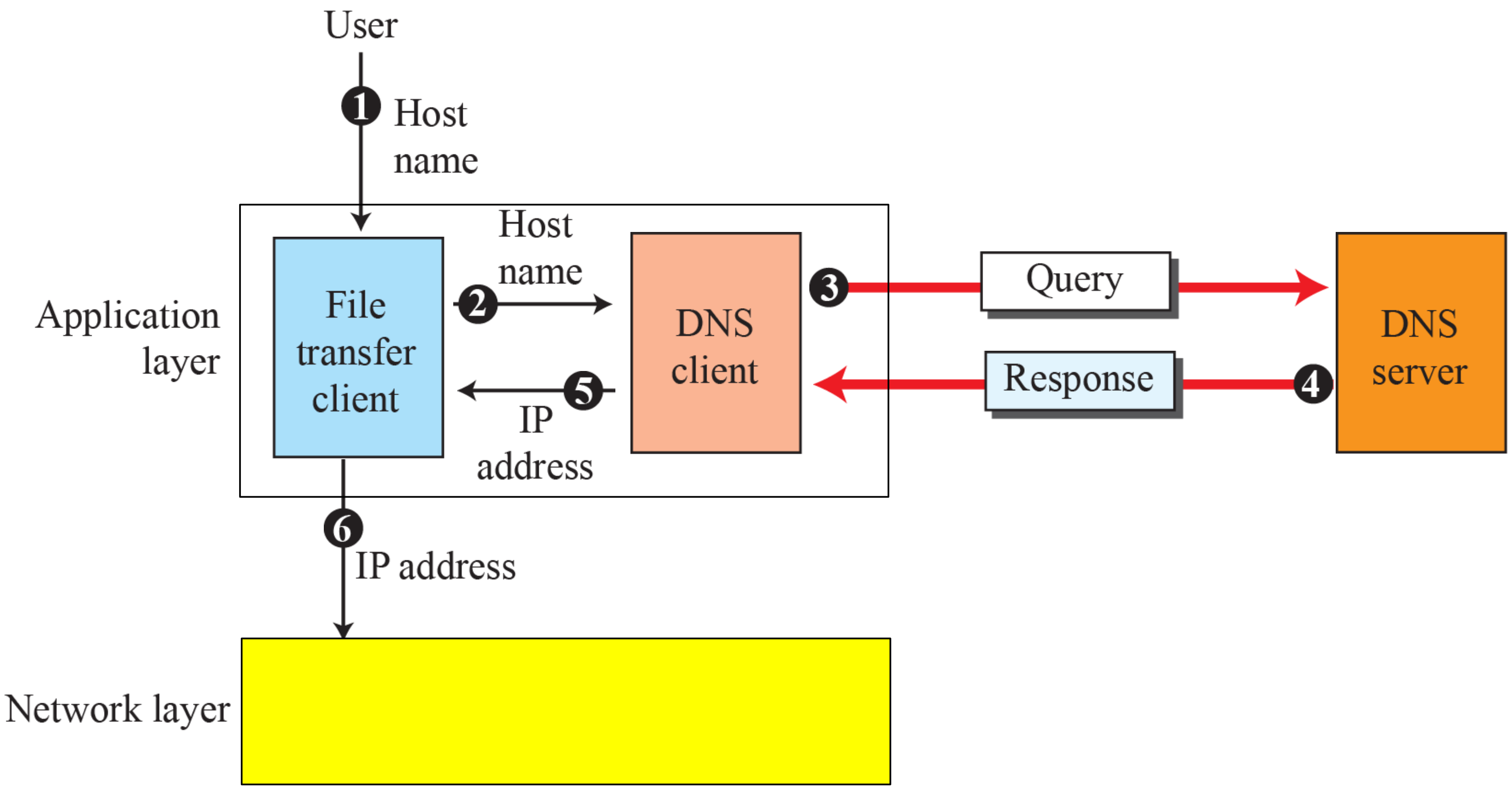


Domain Name System (DNS)

- 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.
- Since the internet is so huge today, a central directory system cannot hold all the mapping.
- In addition, if central computer fails, the whole communication network will collapse.
- Hence the solution is to distribute the information among many computers in the world.
- The host that needs mapping can contact nearest computer holding the needed information.
- This method is used by **Domain Name System (DNS)**.

Domain Name System (DNS)

❑ Flow for DNS Service

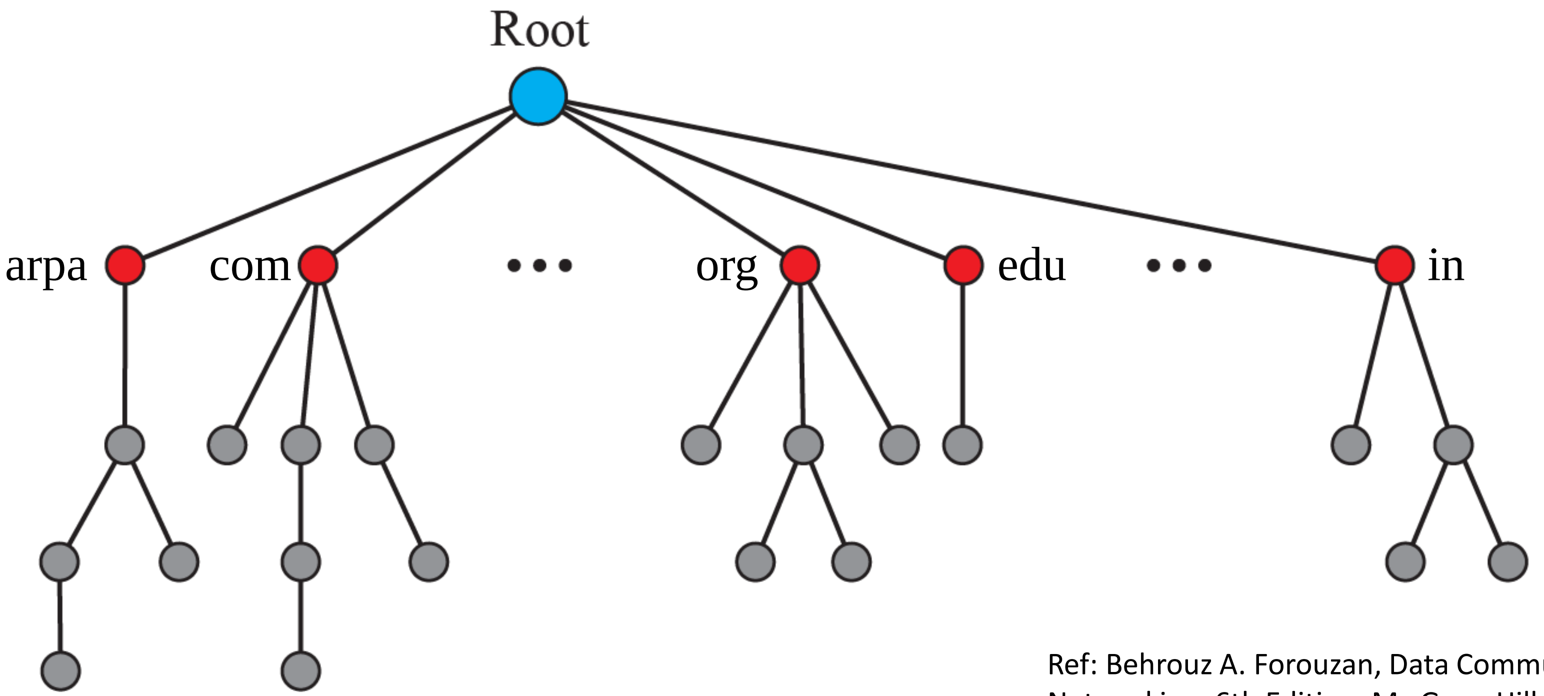


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Domain Name System (DNS)

❑ Domain Name Space

- DNS uses a hierarchical name space; each name is made of several parts.
- 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.

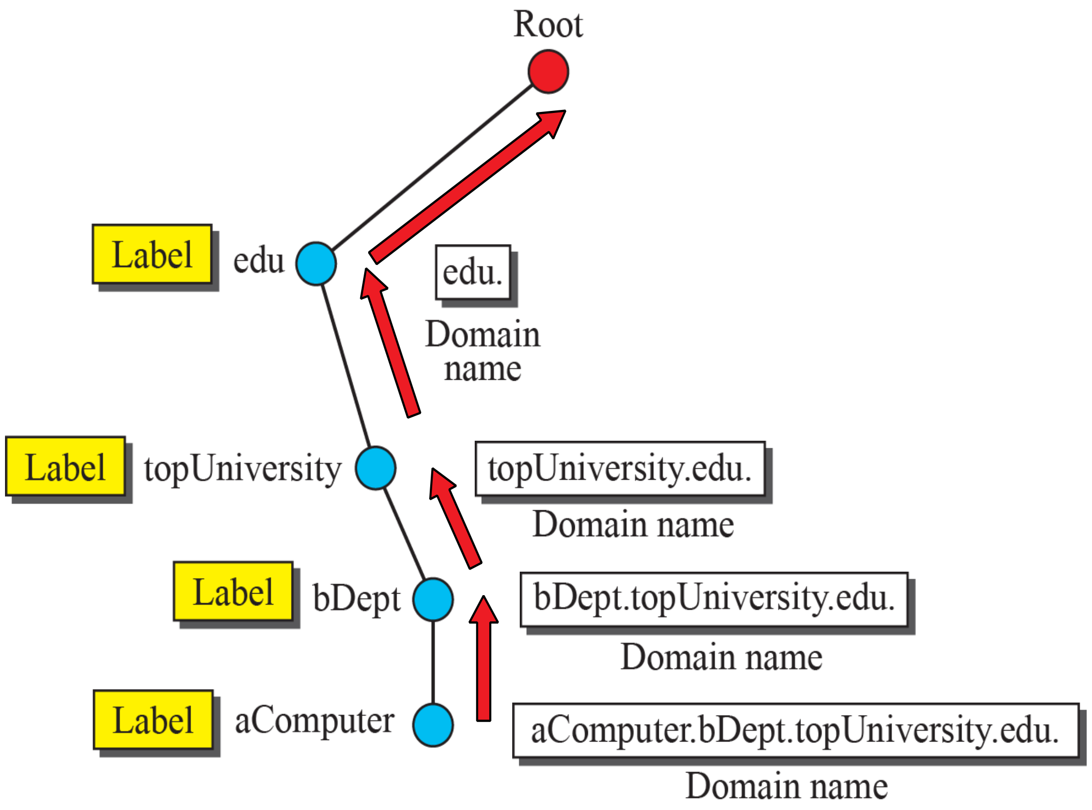


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Domain Name System (DNS)

Domain Names and Labels

- 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.
- A domain may itself be divided into subdomains.
- Each node in the tree has a **label**, which is a string with a maximum of 63 characters.
- DNS requires that children of a have different labels, which guarantees the uniqueness of the domain names.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Domain Name System (DNS)

DNS in the Internet

- In the Internet, the domain name space (tree) is 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.
  - E.g.: .com, .edu, .gov
- **Country Domains:**
  - The country domains section uses two-character country abbreviations. Second labels can be organizational, or they can be more specific, national designations.
  - E.g.: gov.in, ca.us
- **Inverse Domain:**
  - The inverse domain is used to map an address to a name.