DCF255

Lecture 2 | Data Communication Standards

Agenda

- TCP/IP Protocol Architecture
- TCP/IP Protocol in Brief
- Application Layer and Protocol Specifications
- Message Encoding – UTF-8
- Standard Organizations

TCP/IP

Protocol Architecture

Late 1970's

- In the early days of networking high degree of dissatisfaction among business owners, venders and users
- Different network vendor's used different data formats and data exchange protocols
- Had to buy software and hardware from same vendor.
- "Vender lock-in" resulted in higher prices.
- ISO created task force to develop Open System Inter-Connection (OSI).
- TCP/IP was well established when the task force begin, but it was expected that the OSI would replace it

OSI

- **…we can expect to see continued usage of other open products such as the *TCP/IP protocol suite* (Transmission Control Protocol/Internet Protocol), which provides the basic needs of networking. Use of the official OSI protocols in data communications and networking will continue to gain popularity, and the OSI will continue to be utilized both in the development of OSI protocols and as a gauge for comparison with other protocols.**
- Gerald Cole, *Computer Networking For System Programmers*, John Wiley & Sons, New York, 1990, p.24

TCP/IP Remains Dominant – 5 Layer Protocol Stack

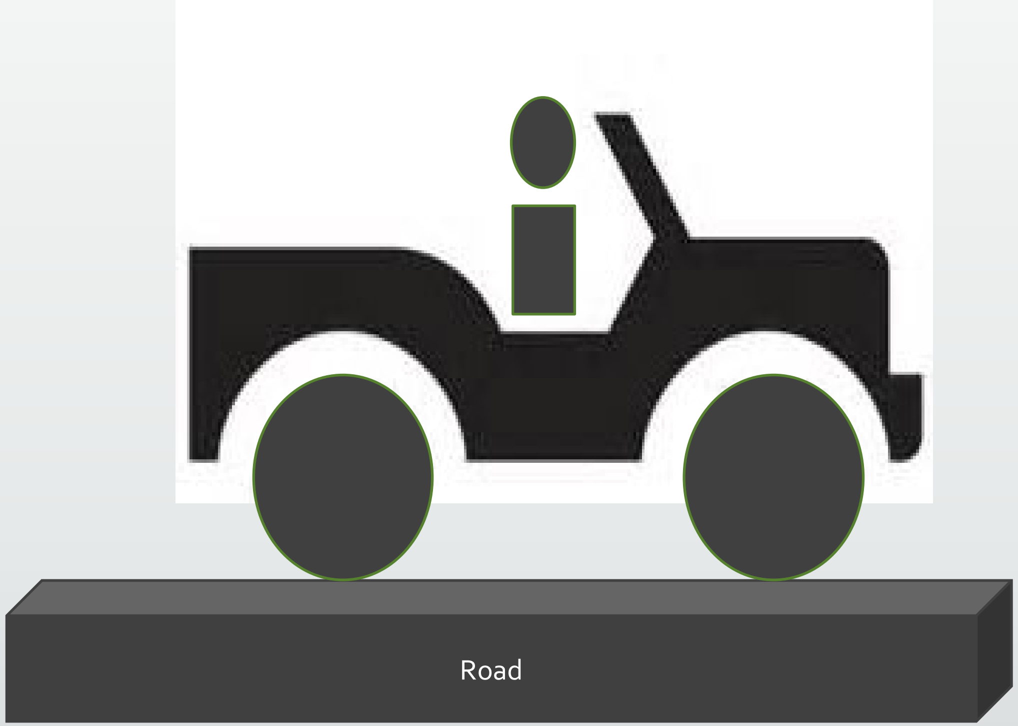
- Development of World Wide Web created exponential growth of the Internet - 1991
- Advances in network speed and reliability made the OSI 7 layer model seem unnecessarily complex
- OSI 7 layer model used to teach networking, but not used in data communications
- Except, physical layer and data link layers have nearly 100% compliance worldwide
- Results in a 5 layer TCP/IP model

TCP/IP

Layered Architecture

Layered Architecture

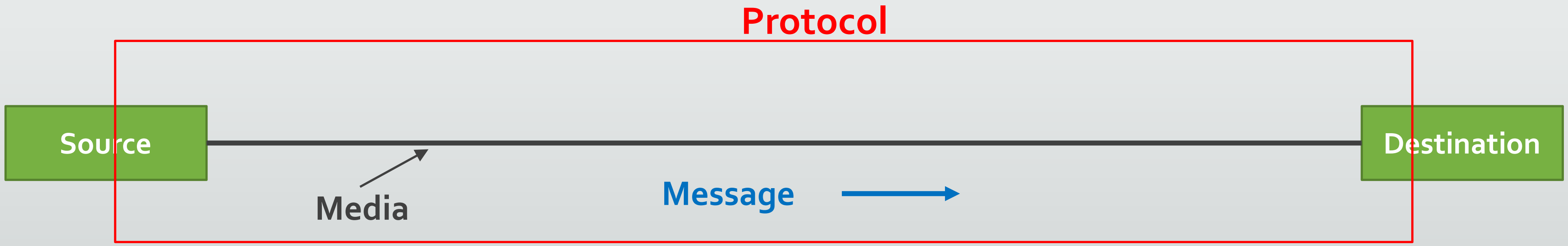
- Each layer operates independently of each other layer
- Each layer below provides a protocol specification which provides a service to the layer above
- The road provides a service for the tires of your car
    - Protocol – how to build a road to support weight of users
- The tires provide a service for the car
    - Protocol – size and thickness of the rubber
- The car provides a service for the driver
    - Protocol - how to drive the car and rules of the road



Layers & Protocol - Data Communication

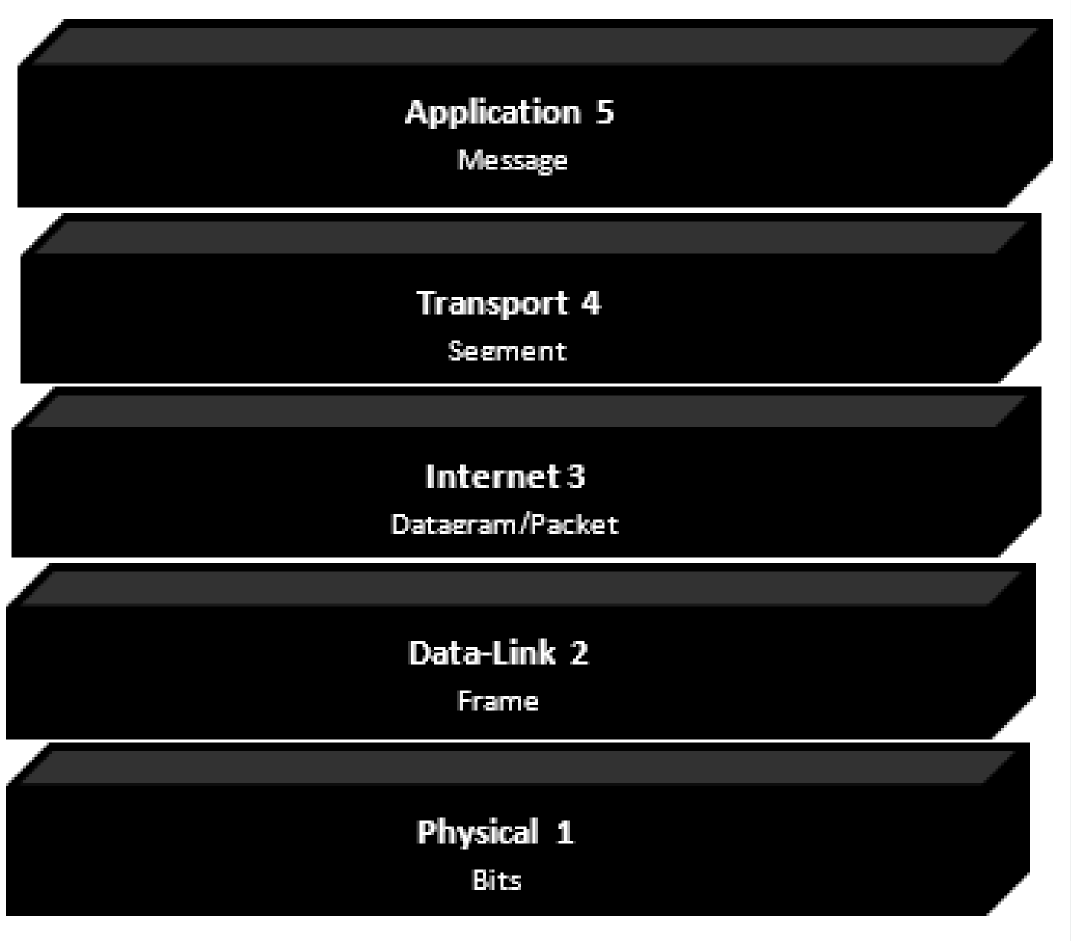
- Layer- a specific piece of software following some pre-defined rules of functioning.
- Unique functions or process are designated to different layers in a communication protocol and layers in conjunction provide the efficient communication.
- Layers are implemented at the source and destination and any piece of hardware which provides interfacing between the two

- Protocol - Collection of Rules for physical establishment of calls, identification, transmission of data, acknowledgement & termination
- Protocol -universal standardization of information exchange over the PC from different vendors, platforms and network configuration
- Protocol - designed in forms of layer.



TCP/IP 5 Layer Model

- Application Layer – controls user interface
  - creates the message, defines the protocol, converts message to digital format. It controls user interface
- Transport Layer -controls data transportability from system to system
  - adds transport header sequence number\source destination ports fragmentation/reassembly, flow control.
- Internet Layer - control efficient routing and paths in the network
  - adds IP header, source\destination IP addresses, routing.
- Data Link Layer - control error free transmission over physical layer
  - adds Data Link header, source\destination MAC address and trailer, FCS for error checking
- Physical provides - interconnection through physical media
  - encoding\decoding and transmits bits across physical medium



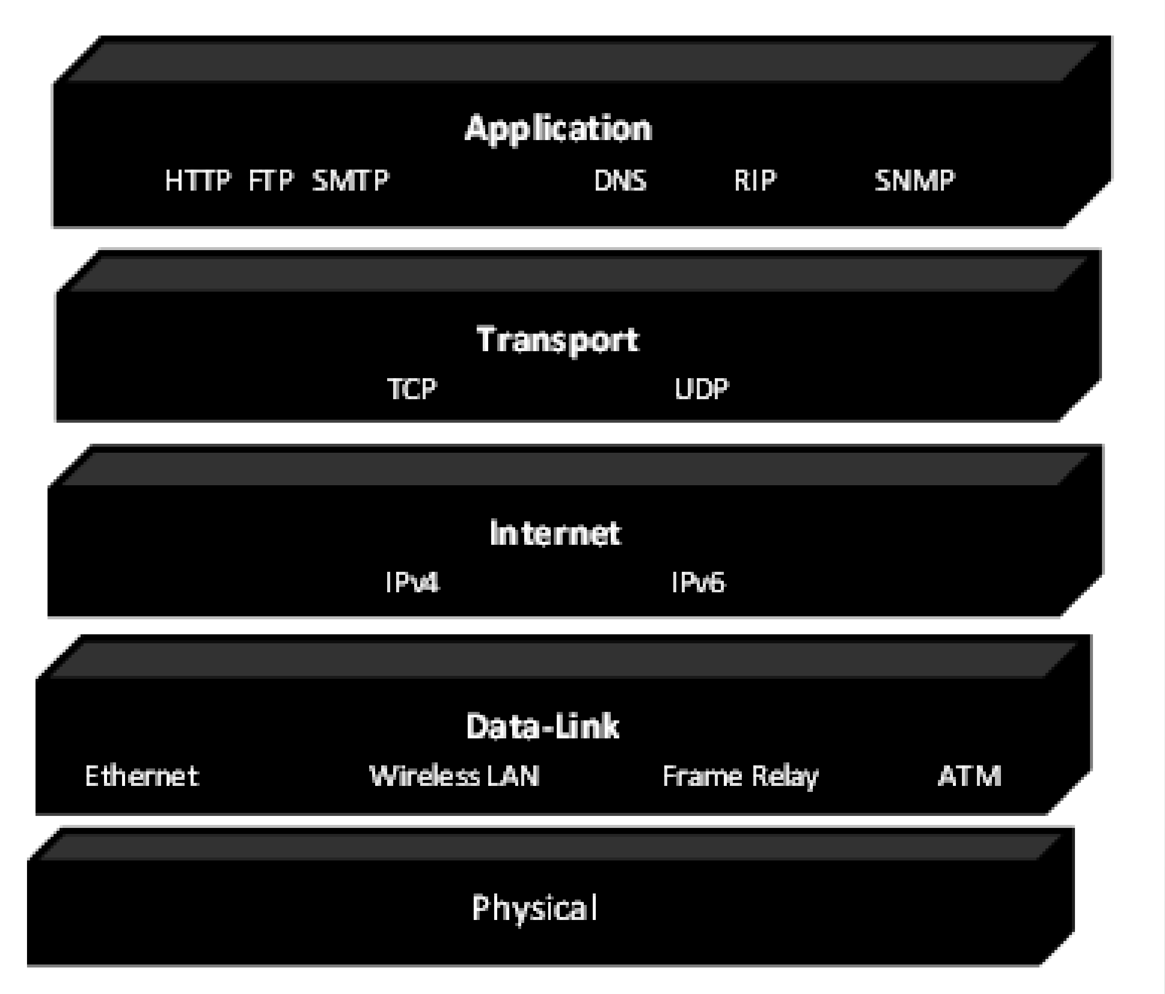
Application Layer

- All Internet applications connect using SOCKETS, and IP address and a port number
  e.g. **64.124.38.2:80**

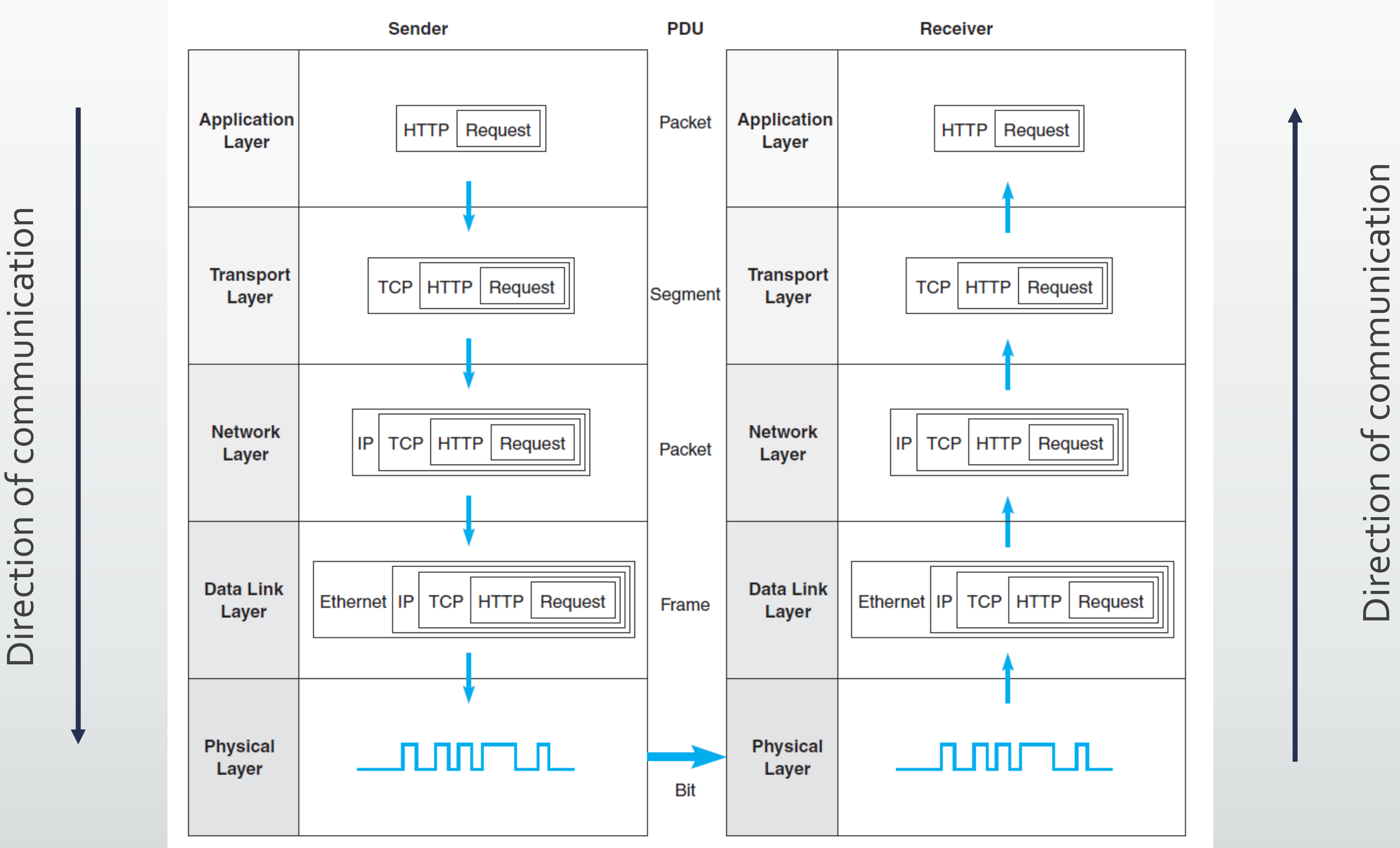
- The Hypertext Transfer Protocol (HTTP) transfers files on the World Wide Web – port 80

- The File Transfer Protocol (FTP) transfers individual files to and from servers –ports 20 and 21

- The Simple Mail Transfer Protocol (SMTP) transfers mail messages and attachments- port 25



Message Transmission Using Layers



Direction of communication

Sender		PDU	Receiver	
Application Layer	HTTP Request	Packet	Application Layer	HTTP Request
Transport Layer	TCP HTTP Request	Segment	Transport Layer	TCP HTTP Request
Network Layer	IP TCP HTTP Request	Packet	Network Layer	IP TCP HTTP Request
Data Link Layer	Ethernet IP TCP HTTP Request	Frame	Data Link Layer	Ethernet IP TCP HTTP Request
Physical Layer		Bit	Physical Layer	

Direction of communication

Application Layer

Protocol Specification

Protocol Specification

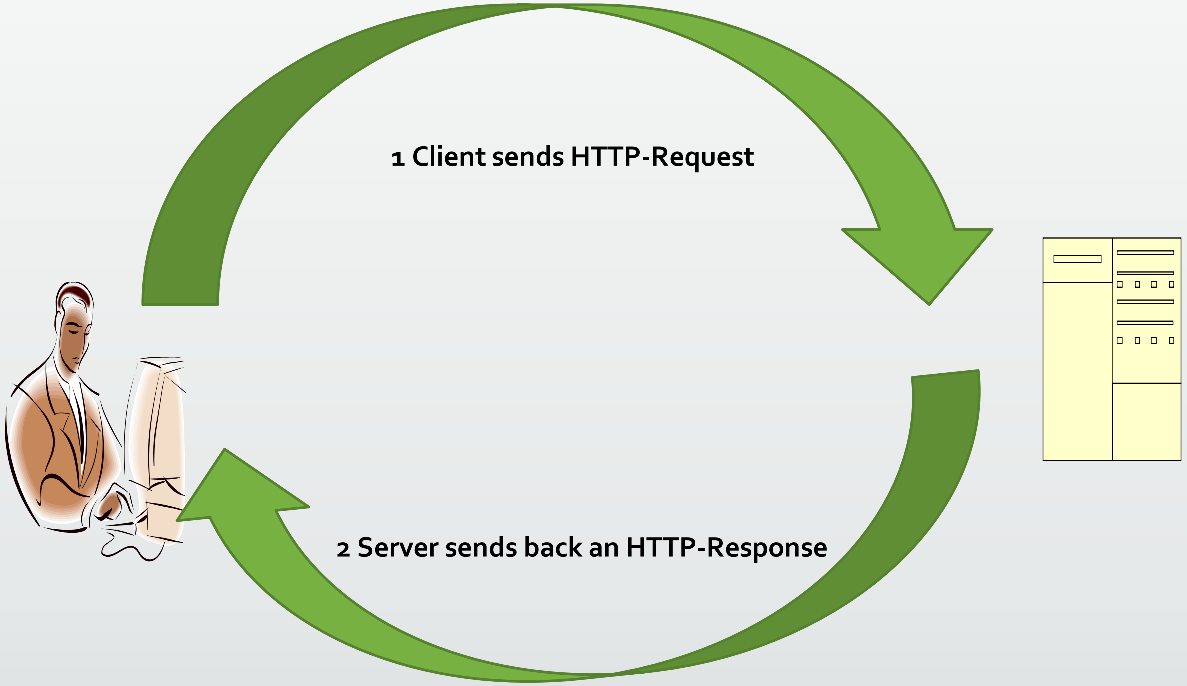
- Message Sequence – Who initiates the communication?
- Message Type – What task or command is to be performed?
- Message Syntax – What is the meaning of the message?
- Type of Connection – What type of connection is needed by the protocol, reliable or unreliable?

Message Sequence - HTTP

- The client sends an HTTP request message to the server

- The server returns an HTTP-response message which answers the client request, or sends an error message

- The server can not initiate a conversation with the client

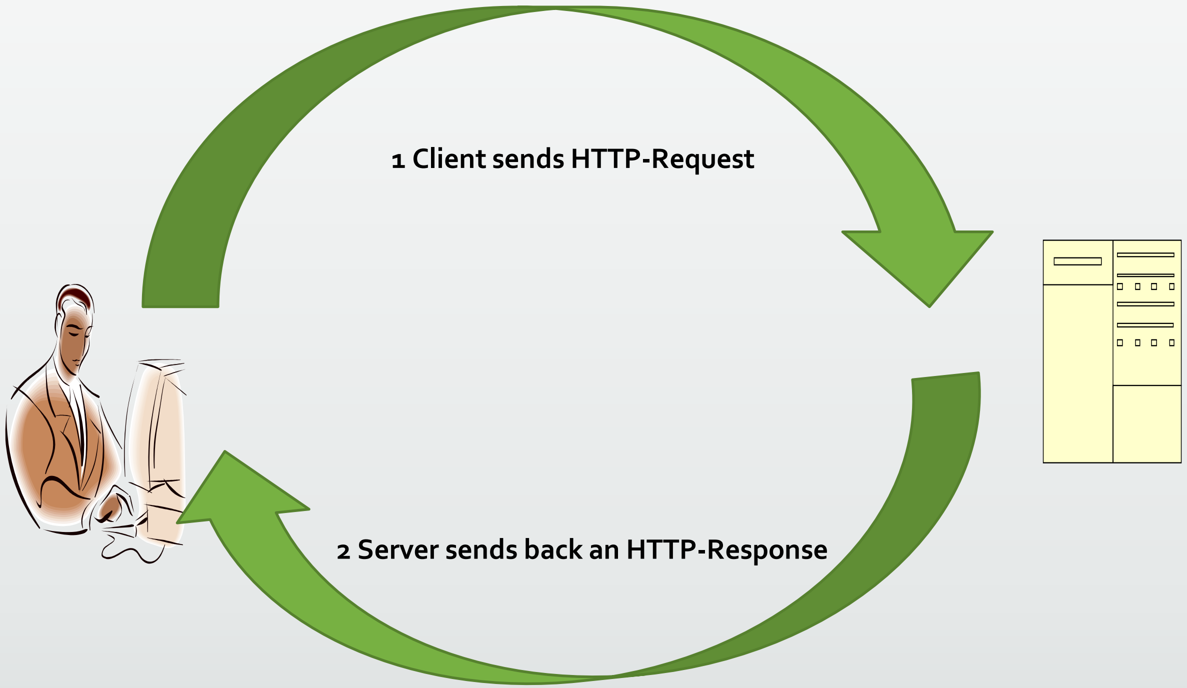


1 Client sends HTTP-Request

2 Server sends back an HTTP-Response

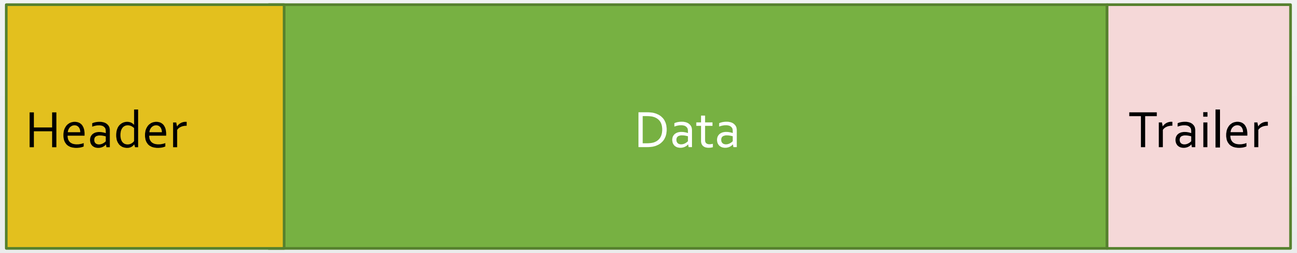
Message Type- HTTP

- GET – commands the server to return a file at the URL – "Get the File"
- POST - uploads a file, usually containing online form data, to the webserver for processing – "Here is the file"
- There are also other file types not discussed here


1 Client sends HTTP-Request
2 Server sends back an HTTP-Response

Message Syntax - HTTP

- Refers to the organization of the message
- Header – something added before the data field
- Data – is the purpose of the message created by the application layer
- Trailer – something added after the data field, usually a FCS for error checking
- Not all packets will have a data field and most do not have a trailer field.

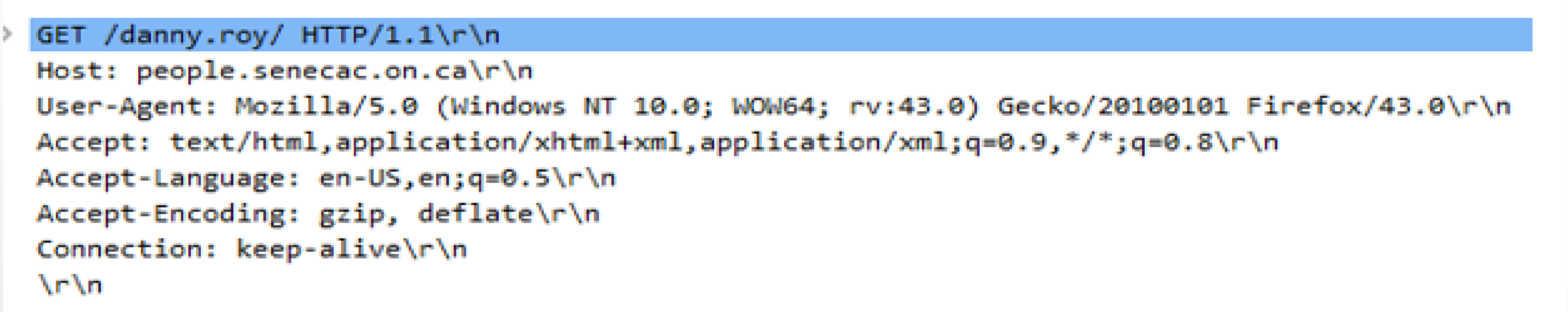


Header

Data

Trailer

HTTP Request Message



```
GET /danny.roy/ HTTP/1.1\r\n
Host: people.senecac.on.ca\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
\r\n
```

1. GET method which requests a path to a file. Notice this line and all other lines end with a character return and a new line [CRLF]

2. host to which the request is sent people.senecac.on.ca

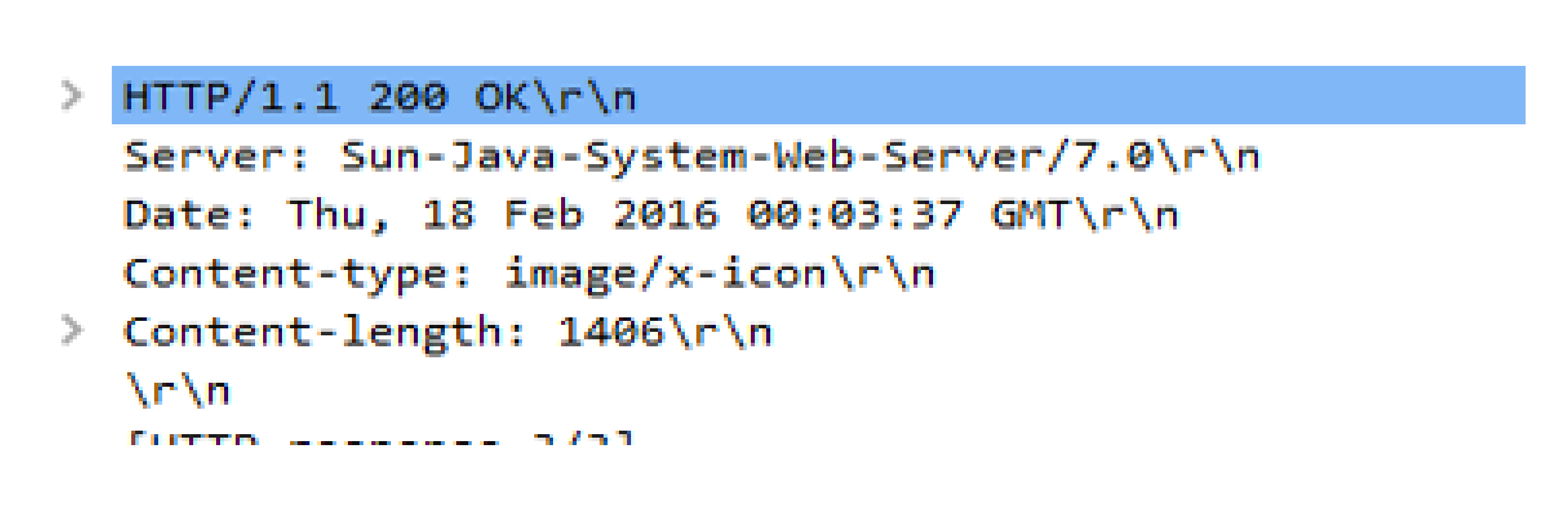
3. host browser and operating system used to send the request

4. language to be used in this case English US

5. type of compression to use, such as gzip or deflate

6. The sixth line specifies the type of TCP connection

HTTP Response Message



```
> HTTP/1.1 200 OK\r\n
  Server: Sun-Java-System-Web-Server/7.0\r\n
  Date: Thu, 18 Feb 2016 00:03:37 GMT\r\n
  Content-type: image/x-icon\r\n
> Content-length: 1406\r\n
  \r\n
  [HTTP response 1/1]
```

1. The first line begins with HTTP/1.1 which indicates the server has a compatible version. The 200 is a success code that the desired file is returned. The browser actually ignores this code; it is designed for humans to indicate the request was successful.

2. The next line indicates the server that returned the request

3. The next line gives the date and time the request was returned.

4. The next heading specifies the content type which was returned, in this case an image file.

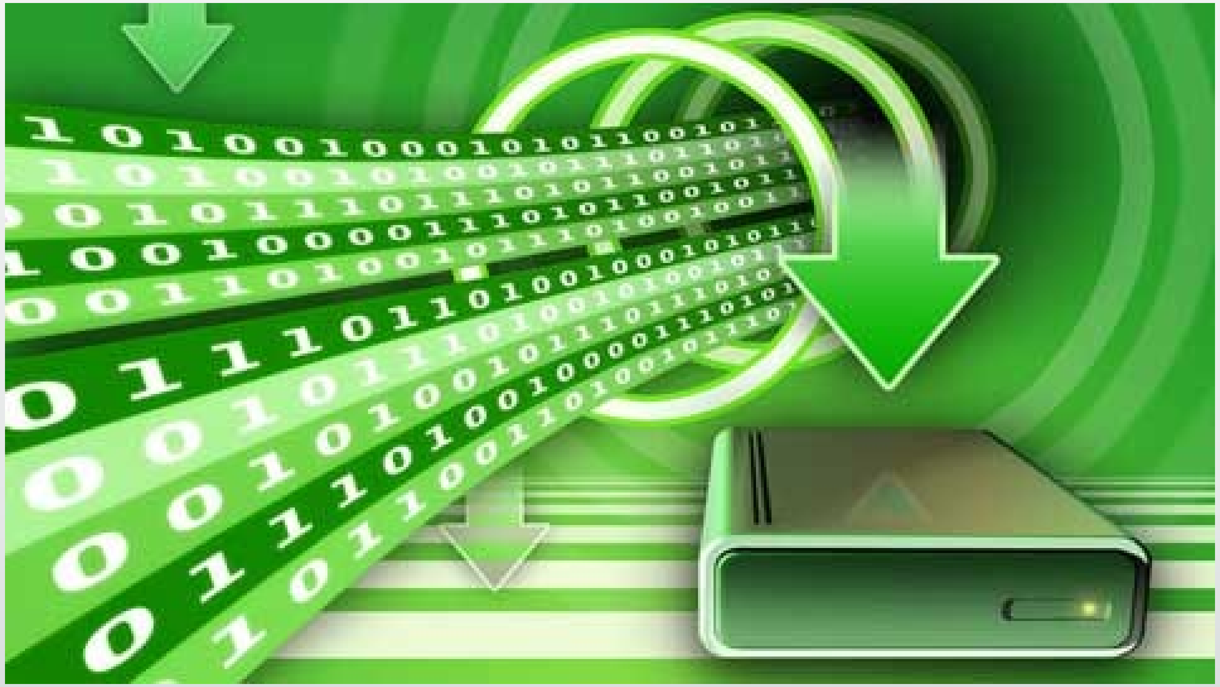
5. Again, the end of the header is marked with 2 blank lines followed by the content returned in the data field (not shown). There is no trailer.

Message Encoding

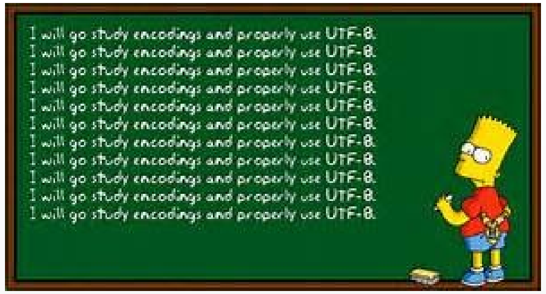
UTF-8

UTF-8 Encoding

- Application layer main job is to convert message into a digital format.
- Application data may be text, image, or video.
- A conversion table is used to convert a byte into a unique numerical value; a process called "encoding". Each encoded value is then converted to a series of 0s and 1s.
- Unicode (Universal Encoding) can represent all of the written languages in the world and is the *de facto* standard for text on the web with over 86.2% of all web pages in 2016 use UTF-8



UTF-8 Encoding


I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.

- UTF-8 (Universal Coded Character Set + Transformation Format – 8-bit.)
- a variable-length encoding system using 1 to 4 “octets” (1 byte is called an octet)
- The first octet representing the first 128 values is identical to ASCII, making UTF-8 a superset of ASCII and safe to use with programming languages that interpret only one byte encoded characters.
- A Unicode character is represented by writing “U+” followed by a hexadecimal number
- When one byte is used for ASCII values the high order bit begins with a “0”. Characters above 127 in value will use more than one byte
- The number of “1s” in the first byte indicates the number of octets used for encoding, followed by a “0”. Each subsequent byte has a “continuation marker” of “10” at the beginning of the byte. Refer to the table below.

UTF-8 Encoding

Bits	Decimal Range	First Code Point	Last Code Point	Bytes used	Byte1	Byte2	Byte3	Byte 4
7	0-127	U+0000	U+007F	1	0xxxxxxx			
11	128-2,047	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	2,048-65,535	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	65,536-1,112,064	U+10000	U+1FFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- UTF-8 is "self-synchronizing". The high order bits of every byte determine the type and number of bytes for each character. Since there is no overlap of values, a receiving computer can reevaluate a transmission by backing up at most three bytes to determine the start of a character.

UTF-8 Encoding: Examples

- To encode UTF-8 you need to understand decimal to digital and hexadecimal conversion
- .Example 1: we will encode the, ASCII character “a”, decimal value of **97**
- Example 2: we will encode the Unicode value U+20AC, representing the Euro symbol, **€.**

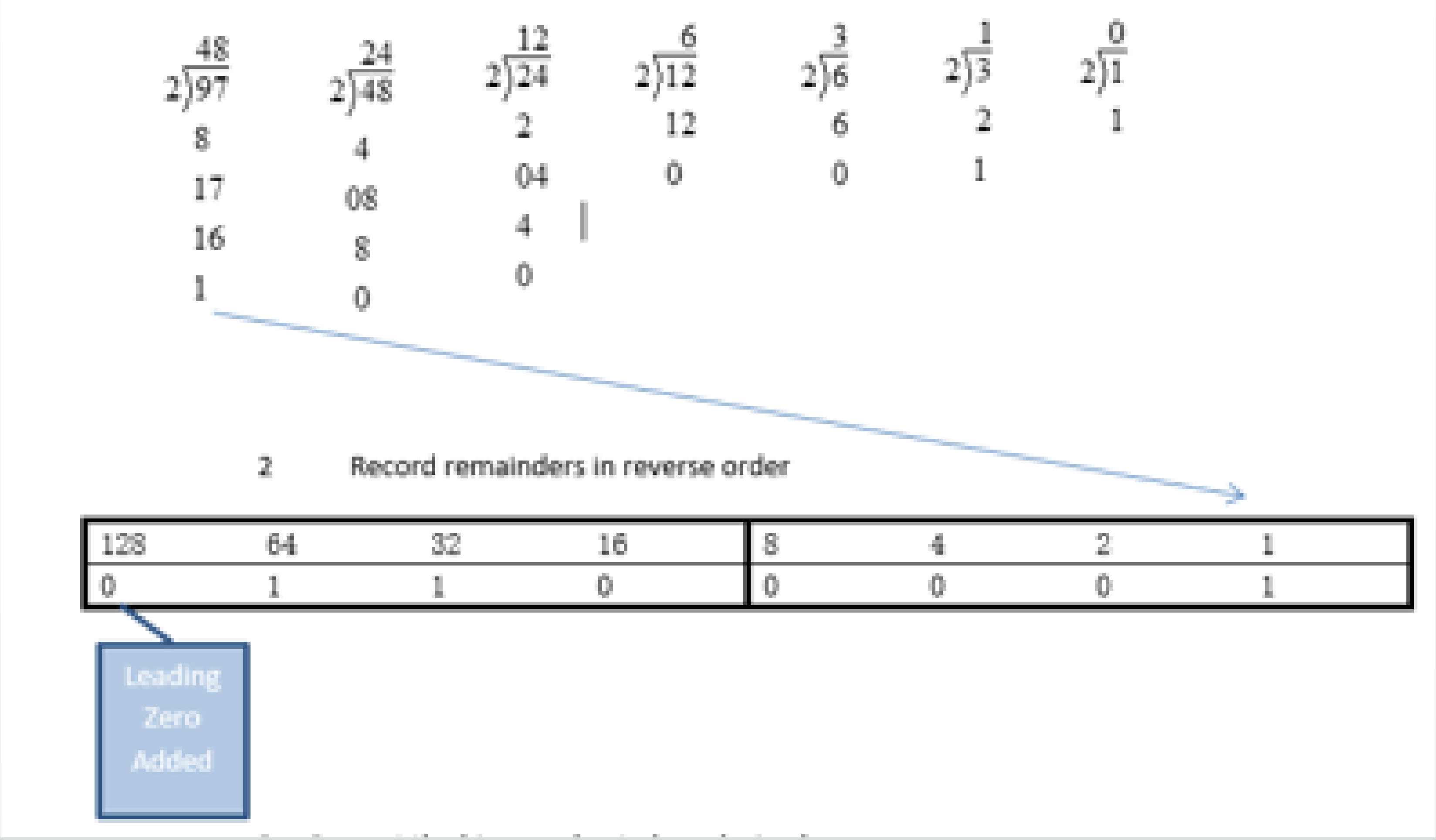
Steps:

1. Convert the number to binary by dividing by 2, till 0 or 1 in the remainder
2. Write the remainders in reverse order
3. Convert the binary value to hexadecimal

Example 1: Converting Decimal Value to Hexadecimal

1. Convert the number to binary by dividing by 2

2. Write the remainders in reverse order



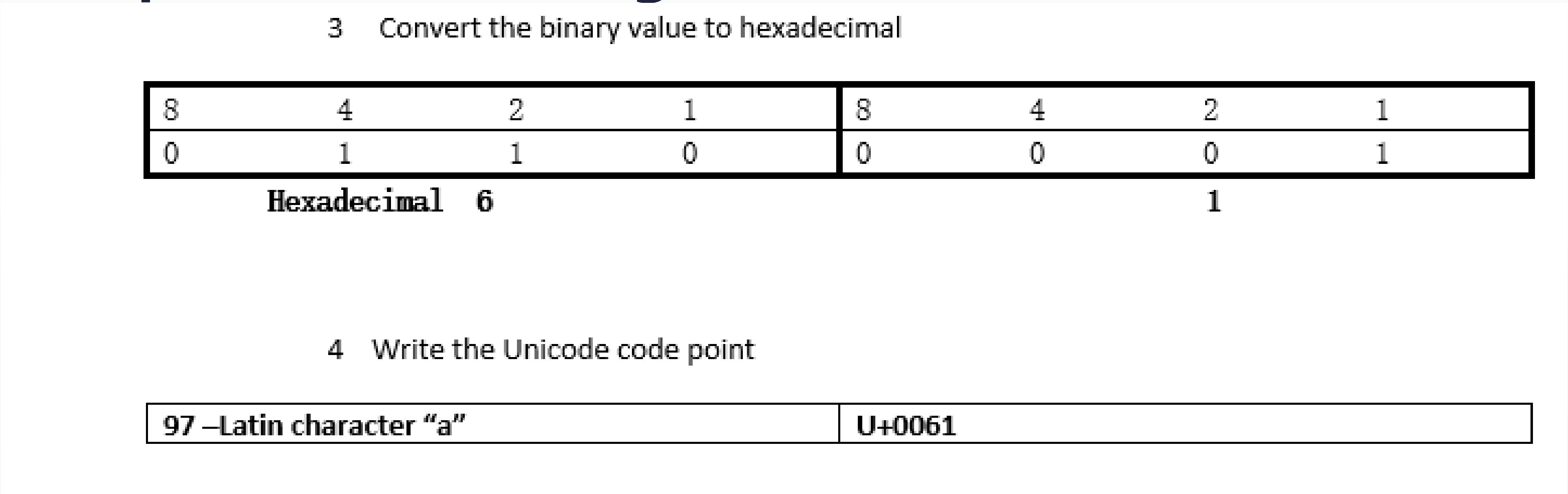
$$\begin{array}{r} 48 \\ 2\overline{)97} \\ 8 \\ 17 \\ 16 \\ 1 \end{array} \quad \begin{array}{r} 24 \\ 2\overline{)48} \\ 4 \\ 08 \\ 8 \\ 0 \end{array} \quad \begin{array}{r} 12 \\ 2\overline{)24} \\ 2 \\ 04 \\ 4 \\ 0 \end{array} \quad \begin{array}{r} 6 \\ 2\overline{)12} \\ 12 \\ 0 \end{array} \quad \begin{array}{r} 3 \\ 2\overline{)6} \\ 6 \\ 0 \end{array} \quad \begin{array}{r} 1 \\ 2\overline{)3} \\ 2 \\ 1 \end{array} \quad \begin{array}{r} 0 \\ 2\overline{)1} \\ 1 \end{array}$$

2 Record remainders in reverse order

128	64	32	16	8	4	2	1
0	1	1	0	0	0	0	1

Leading Zero Added

Example 1: Converting Decimal Value to Unicode



3 Convert the binary value to hexadecimal

8	4	2	1	8	4	2	1
0	1	1	0	0	0	0	1

Hexadecimal 6 **1**

4 Write the Unicode code point

97 –Latin character "a"	**U+0061**

3. Convert the binary value to hexadecimal by grouping the binary value into 4 bits and determining each 4 bit value

4. Write the Unicode code point for the hexadecimal value – **97 decimal = U+0061**

Example 2: Converting Unicode Value U+20AC to UTF-8 Encoding

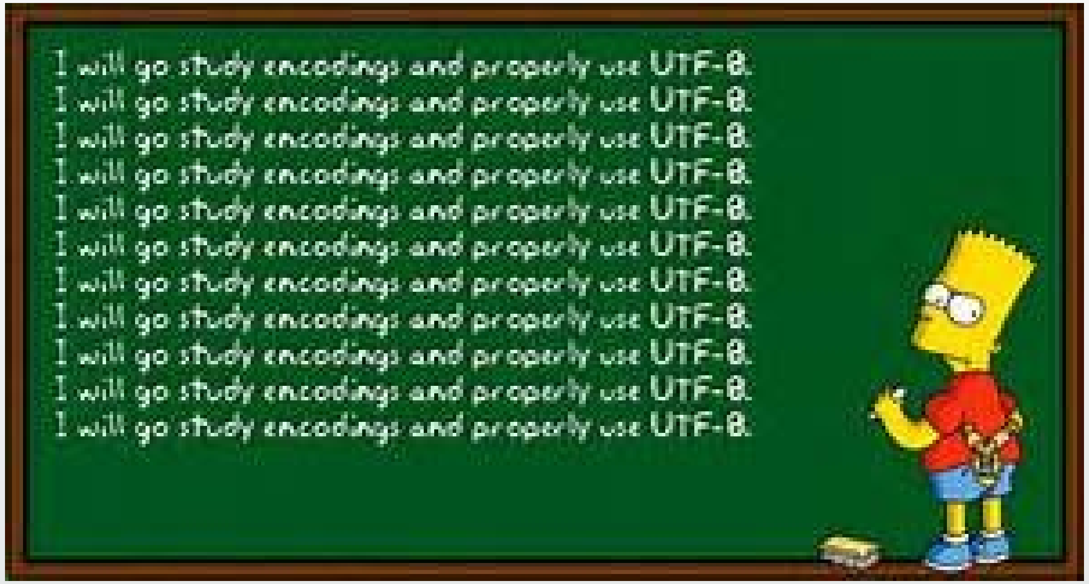
- Unicode characters are represented by hexadecimal notation
- The value U+20AC is only the bit sequence or the code point to represent the Euro symbol "€"
- It is not the hexadecimal UTF-8 encoded string created by the application layer.

1. Convert the code point to binary
2. Encode the binary value into a UTF-8 multi-byte sequence and
3. Then convert the sequence back to hexadecimal.

Example 2: Converting Unicode Value U+20AC to UTF-8

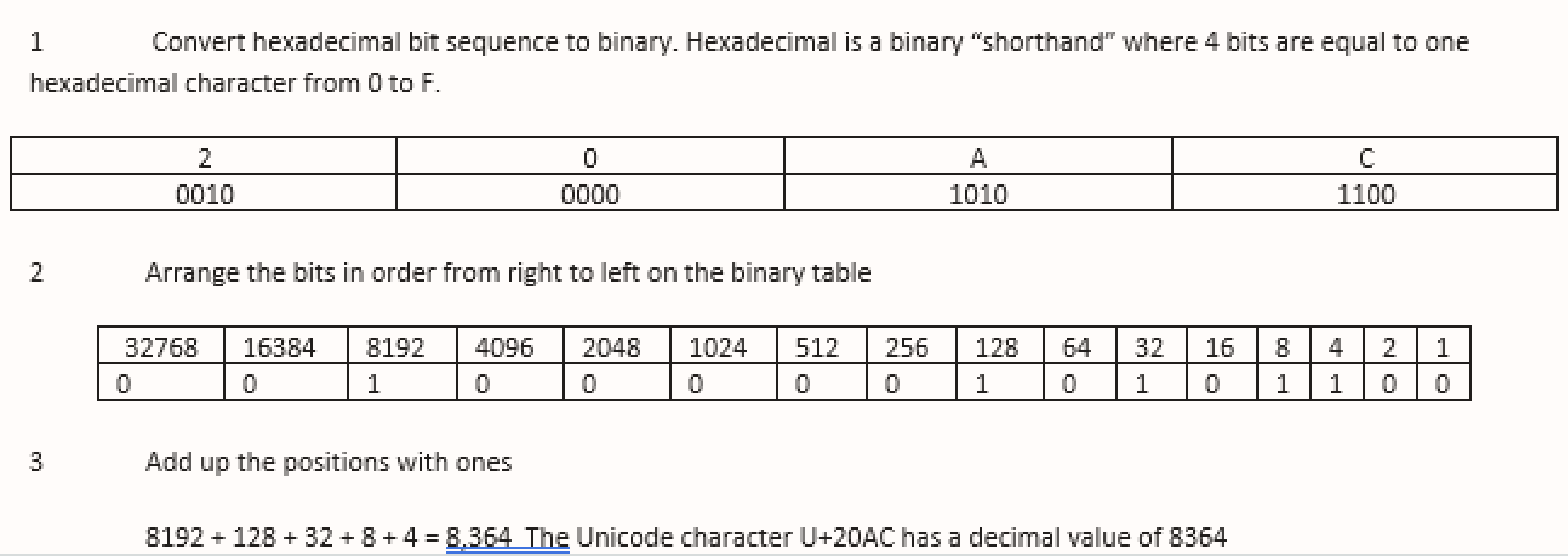
- To convert the code point to UTF-8 encoding, we must first :

1. Convert the code point to binary
2. Encode the binary value into a UTF-8 multi-byte sequence and
3. Then convert the sequence back to hexadecimal.


I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.
I will go study encodings and properly use UTF-8.

Example 2: Converting Unicode Value U+20AC to UTF-8

If you need to convert hexadecimal to decimal follow the steps below:



1 Convert hexadecimal bit sequence to binary. Hexadecimal is a binary "shorthand" where 4 bits are equal to one hexadecimal character from 0 to F.

2	0	A	C
0010	0000	1010	1100

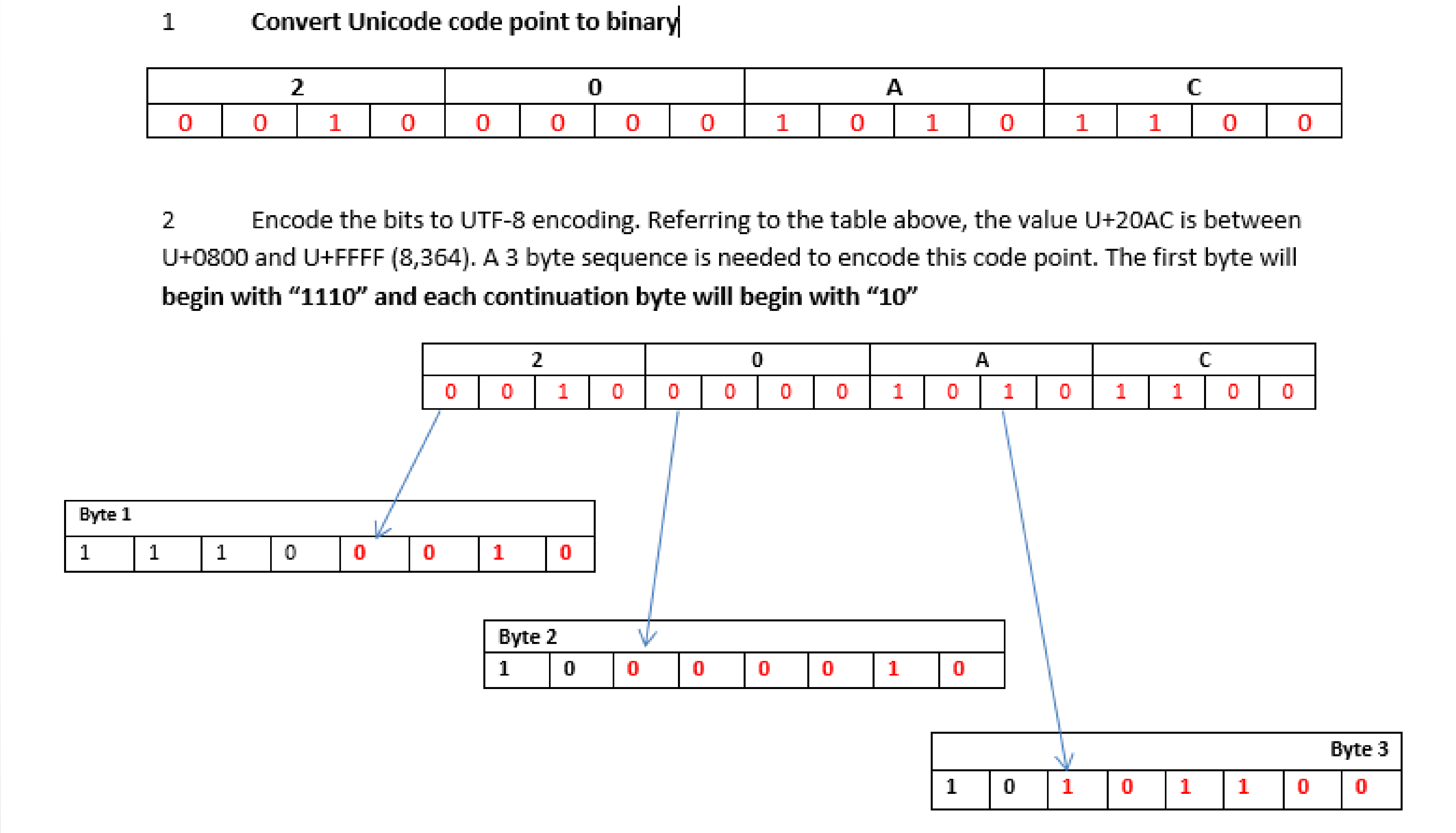
2 Arrange the bits in order from right to left on the binary table

32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0

3 Add up the positions with ones

8192 + 128 + 32 + 8 + 4 = 8,364 The Unicode character U+20AC has a decimal value of 8364

Example 2: Converting Unicode Value U+20AC to UTF-8



1 Convert Unicode code point to binary

2				0				A				C			
0	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0

2 Encode the bits to UTF-8 encoding. Referring to the table above, the value U+20AC is between U+0800 and U+FFFF (8,364). A 3 byte sequence is needed to encode this code point. The first byte will **begin with “1110” and each continuation byte will begin with “10”**

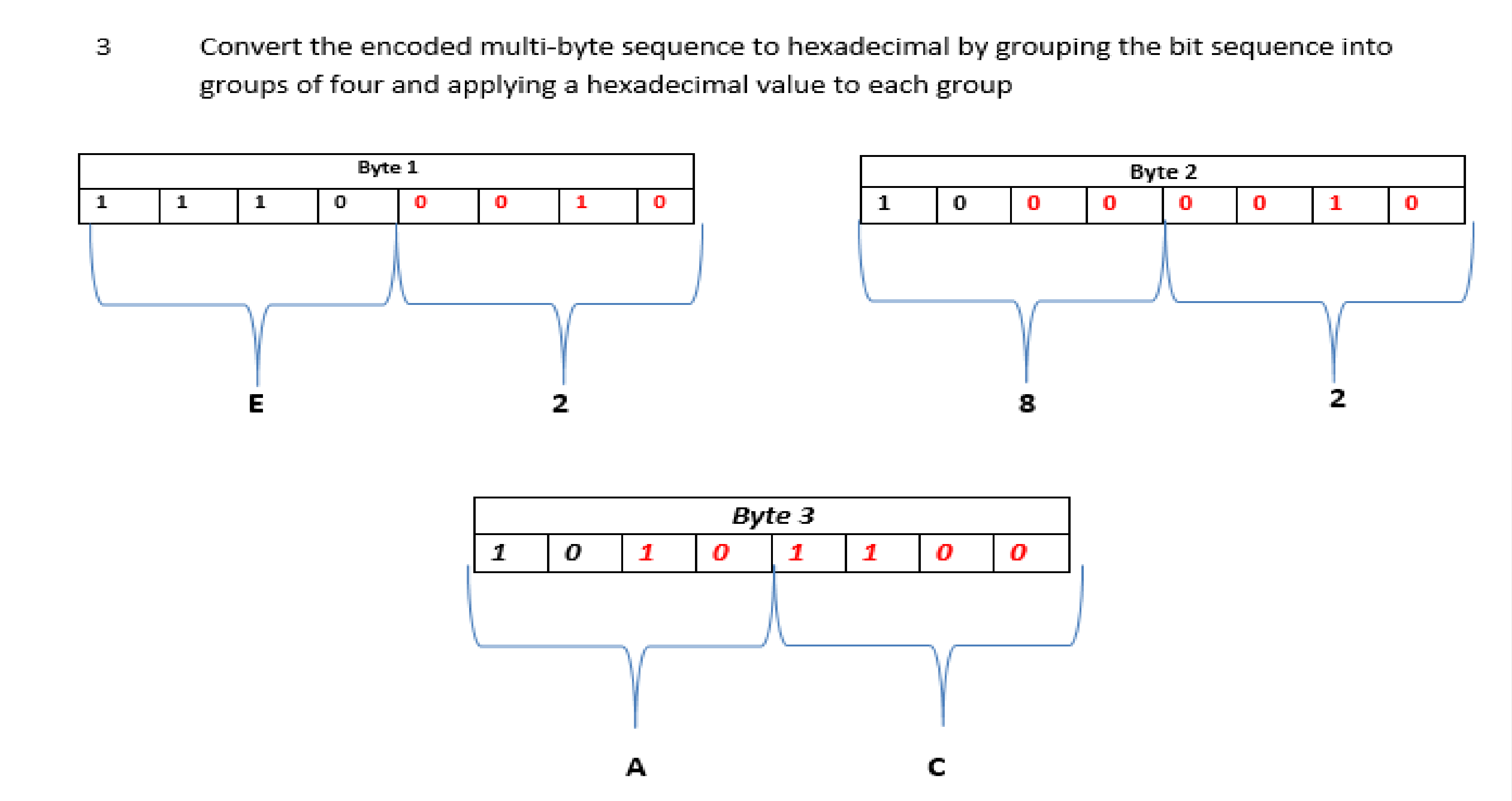
2				0				A				C			
0	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0

Byte 1							
1	1	1	0	0	0	1	0

Byte 2							
1	0	0	0	0	0	1	0

							Byte 3
1	0	1	0	1	1	0	0

Example 2: Converting Unicode Value U+20AC to UTF-8



3 Convert the encoded multi-byte sequence to hexadecimal by grouping the bit sequence into groups of four and applying a hexadecimal value to each group

Byte 1							
1	1	1	0	0	0	1	0

E 2

Byte 2							
1	0	0	0	0	0	1	0

8 2

Byte 3							
1	*0*	*1*	*0*	*1*	*1*	*0*	*0*

A C

The Unicode character U+20AC has a hexadecimal UTF-8 encoding of **0xE2 82 AC**

Encoding Alternatives

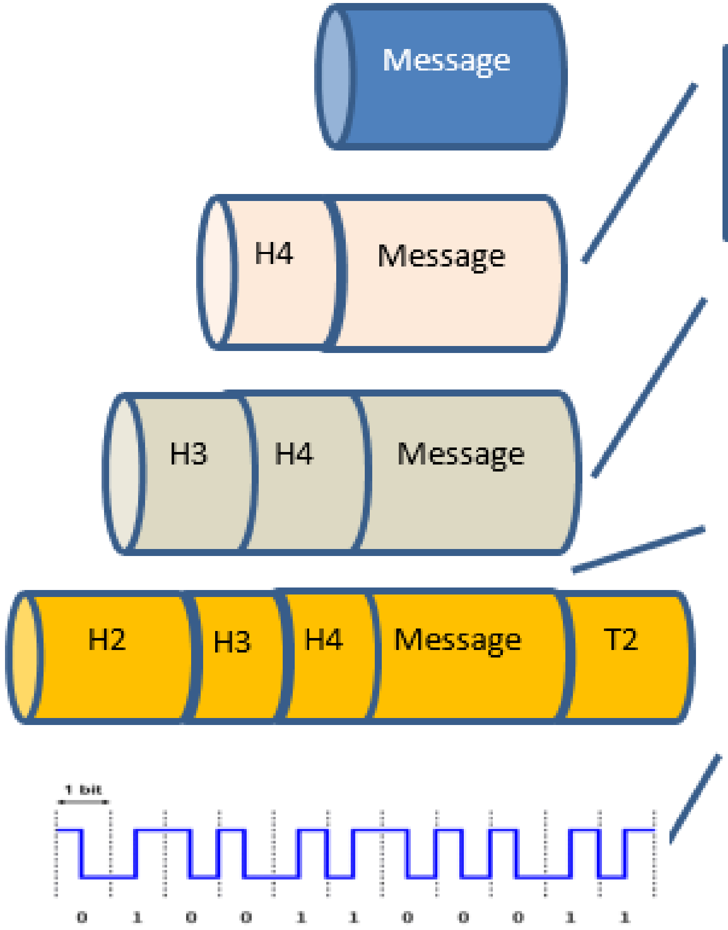
- Sometimes the application layer will use a number of bits to represent alternatives, such as a field representing the type of message, or the protocol to use

- Many applications today involve voice and video. To encode an analog signal a special electrical circuit is used called a CODEC (Encoding\Decoding). This circuit measures the amplitude of the wave and converts it into a digital value, called encoding. When the digital signal is converted back to an analog signal, to play through a speaker, the signal is decoded.

Bits	Alternatives	Examples
1	2	Yes\No Male\Female
2	4	Married, Single, Divorced, Widowed
3	8	Number of Company Departments
4	16	Top 10 Google Searches
8	256	Number of Users in company
16	65,536	Number of colour levels
32	4,294,967,296	Number of IPv4 Addresses

TCP/IP Protocol Stack in Action

Application	5
Transport	4
Internet	3
Data-Link	2
Physical	1



Message encapsulated as data of the TCP packet

Message encapsulated as data of the IP datagram

Message encapsulated as data of the Ethernet frame

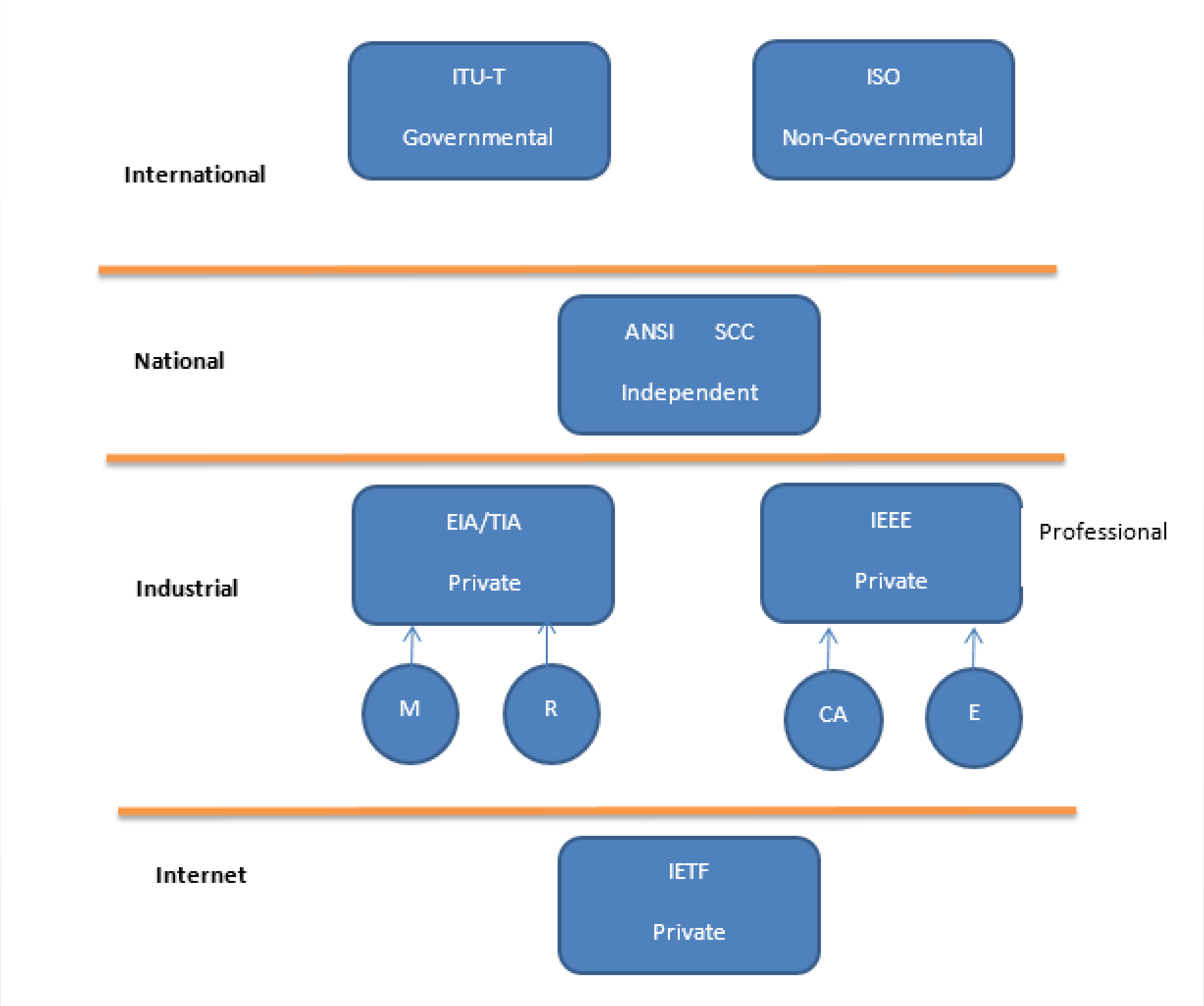
Conversion of bits into outgoing signals

Standard Organizations

International, National, Industry, Professional and Internet

Standard Organizations

- ITU-T is an agency of the United Nations
- ISO is an independent organization made up of all the national standard organizations
- ANSI and Standard Council of Canada (SCC) – independent national bodies often work together
- EIA\TIA are separate industry organizations that often form joint standards. IEEE is the largest professional organization comprised of engineers
- IETF is an independent body responsible for the Internet



Summary

1. In the late 70's the ISO developed a 7 layered architecture called Open System Inter-Connection; this model has not replaced TCP/IP, but has modified it by adding the physical and data links to create a 5 layer TCP/IP protocol stack.

2. The TCP/IP protocol stack is a layered model with each layer working independently of the other layers. Each lower layer also provides a service to the layer above. This modular approach allows changes in cabling with affecting the higher layers.

3. Each layer also has a protocol specification. The application layer protocol specification affects the sequence, type and syntax of the messages used in data communications.

4. The most important job of the application layer is to convert the message into a digital format. The most common format today is UTF-8 which is 100% compatible with ASCII 8 bit character set and is self synchronizing.

5. Lastly, standards are created by standard organizations which exist at the national, international, industrial, professional and Internet areas. Standards must set in advance of data communication and networking, which requires standard organizations to work together