Lesson 3: End-to-End Data

Van-Linh Nguyen

Fall 2024

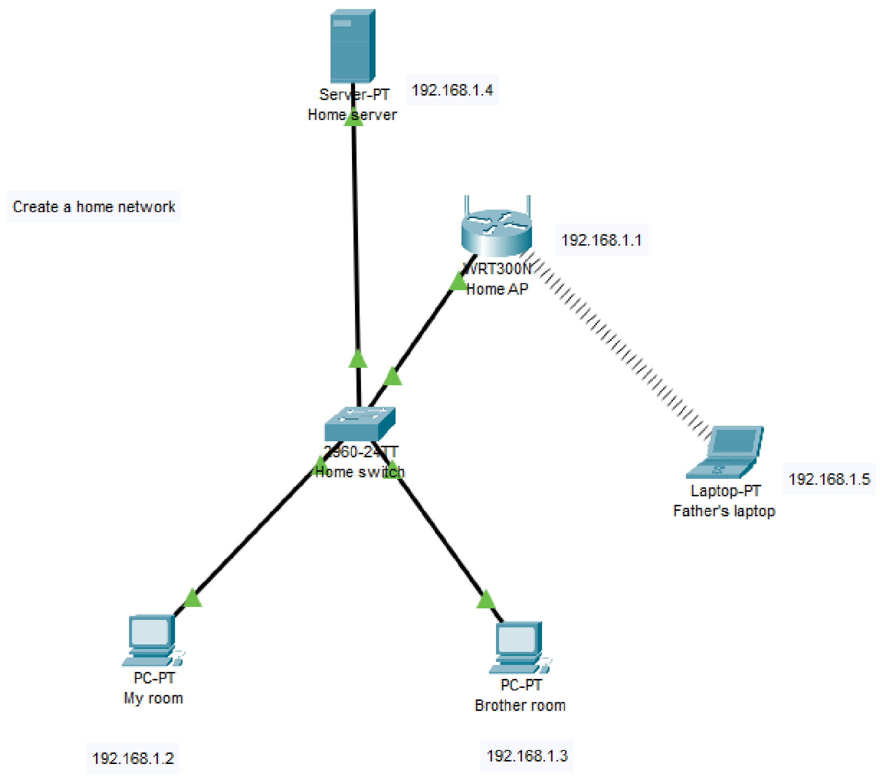
國立中正大學 Cybersecurity Lab

Outline

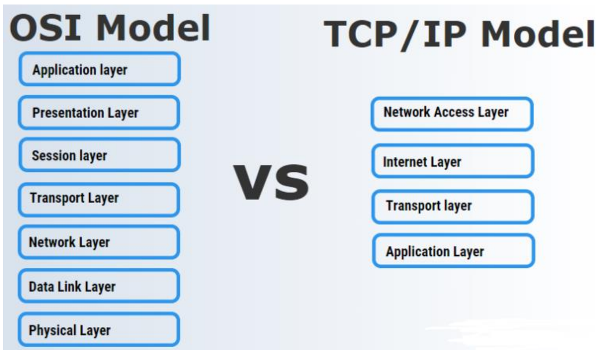
- Create a network by Packet Tracer [to recall the last lesson]
- How to present data in networks
- Different formats of data
    1. Multimedia data

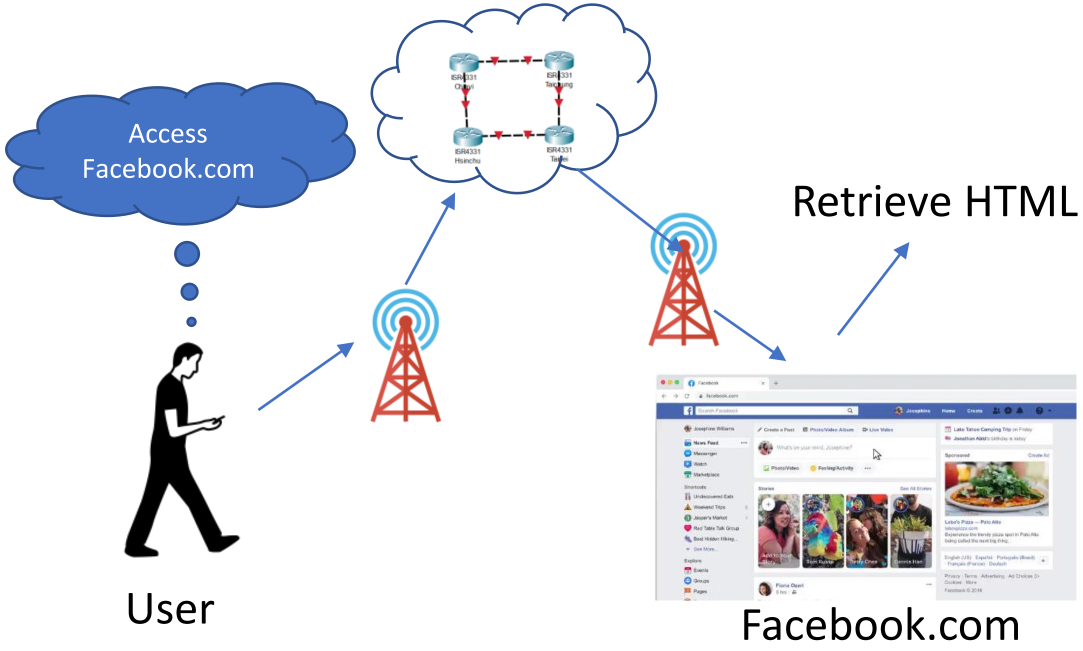
Build a simple lab to recall the last lesson

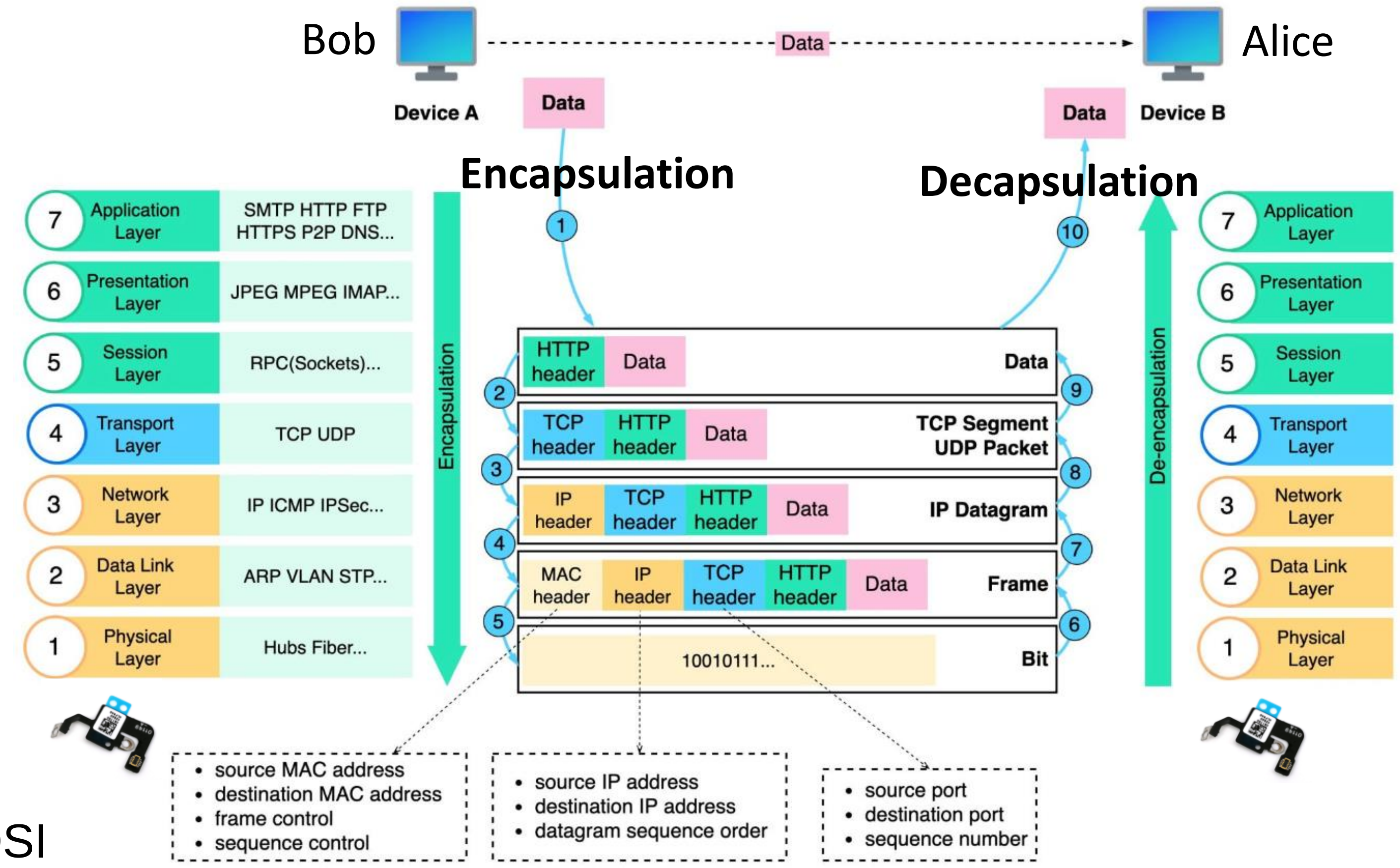
- Create a home network as the beside figure configuration
- Test whether the laptop can ping any PC/server
- Test whether PCs can access Web on the Home server
- Test and see the packet format



Encapsulation/Decapsulation







https://blog.bytebytego.com/

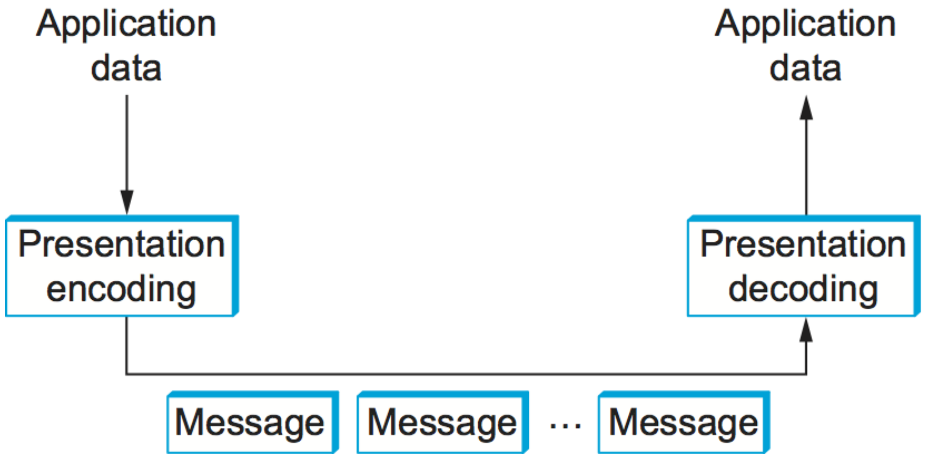
TCP/IP Model is more popular in industry than OSI
OSI is for academic research

Presentation Formatting

- Transform application data into a form that is suitable for transmission over a network and vice versa

- The sending program translates the data into a message that can be transmitted over the network, this calls **presentation encoding** or **data encoding**

- On the receiving side, the application translates the arriving message into a representation that it can then process: **presentation decoding** or **data decoding**



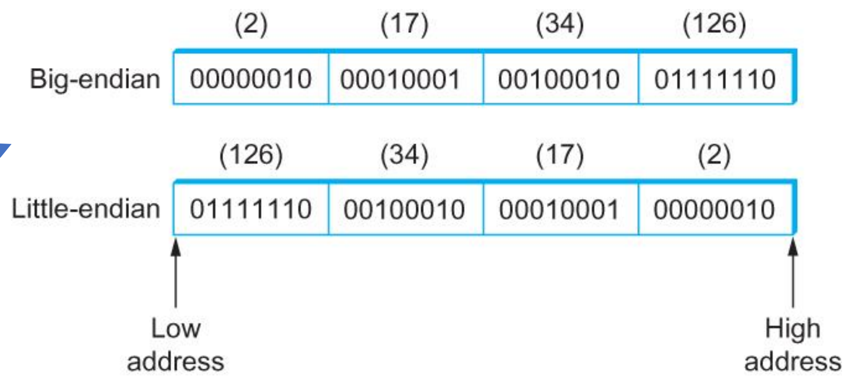
Presentation formatting involves encoding and decoding application data

Encoding is sometimes called argument *marshalling*

What makes this formatting challenging?

- Different devices (Laptop, PCs, Mobile) may run 16-bit, 32-bit, 64-bit architecture
- Device may use different formatting styles: Big-endian and little-endian byte

PCs/Intel x86



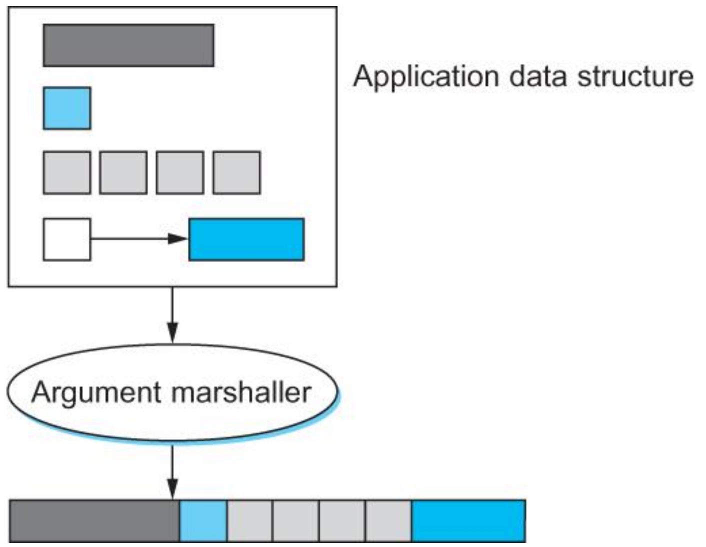
- Application programs in the transmitter and the receiver are written in different languages (C/C++, Python, .Net) or even compiled by different compilers

- Transmitting a structure from one machine to another is challenging

Encoding: build a standard format

- Data Types: design data types the system is going to support
  - ✓ Low-level: integers, floating-point numbers, and characters
  - ✓ Mid-level: structures and arrays
  - ✓ High-level: pointers, memory allocation

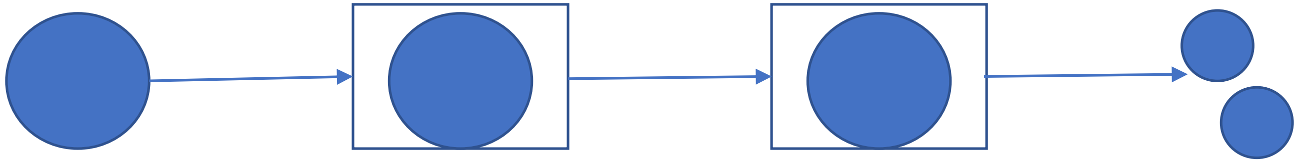
Converting data type: e.g., int(34.5)
Packing data: e.g., a[12,13]
Linearizing: Pointer list



Argument marshalling: converting, packing, and linearizing

Encoding: Conversion Strategy

- Two general options: *canonical intermediate form* and *receiver-makes-right*
- *Canonical intermediate form:*
  1. The sending host translates from its internal representation to this external representation before sending data
  2. The receiver translates from this external representation into its local representation when receiving data



- *Receiver-makes-right :*
  1. The sender transmit data in its own internal format (does not convert but pack data type)
  2. The receiver is then responsible for translating the data from the sender's format into its own local format
  3. The problem with this strategy is that every host must be prepared to convert data from all other machine architectures

Encoding: The best thing to do?

How can a MacBook laptop communicate with a Window desktop?

How to know the receiver is a Window desktop?

- The best option: Using a common external format (standard for all)
- But the computer system world has too many architectures, formatting which are often tailored to specific manufacturers

Encoding: Tag

- How the receiver knows what kind of data is contained in the message it receives?

- A tag is any additional information included in a message—beyond the concrete representation of the base types—that helps the receiver decode the message

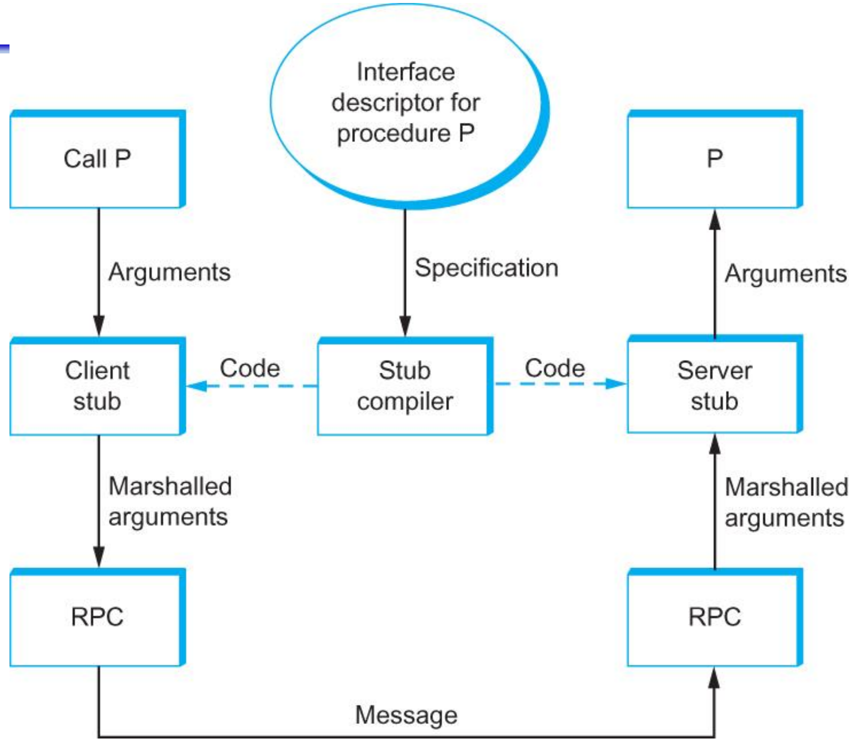
A 32-bit integer encoded in a tagged message

Tag in luggage

- The alternative, of course, is not to use tags. How does the receiver know how to decode the data in this case? Fixed program

Encoding: Stubs

- Definition: the piece of code that implements encoding
- Stubs is invented to support Remote Procedure Call (RPC) protocol
- On the server side, the stub converts the message back into a set of variables that can be used as arguments to call the remote procedure



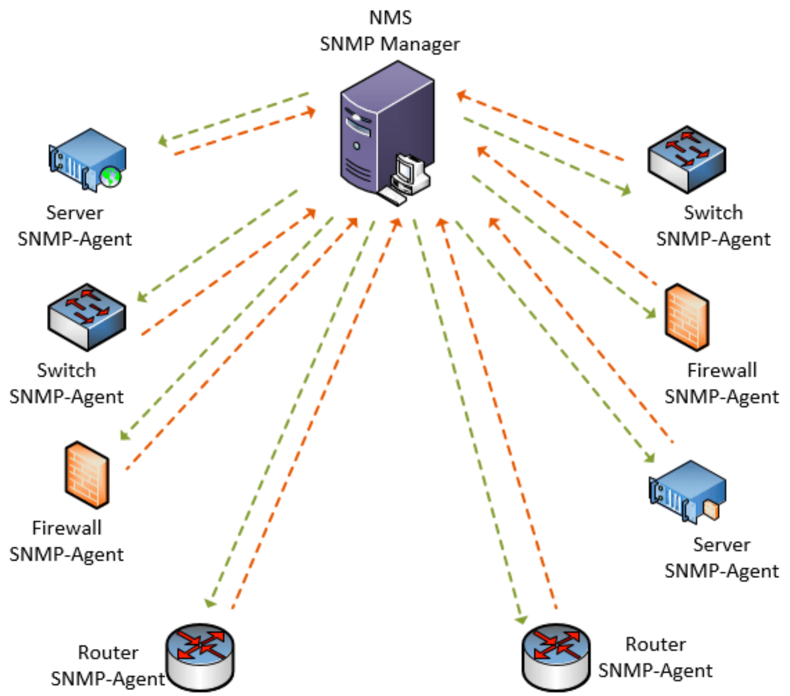
Stub compiler takes interface description as input and outputs client and server stubs.

Presentation formatting example: ASN. 1

- Abstract Syntax Notation One (ASN.1) is an ISO standard that defines, among other things, a representation for data sent over a network

- The representation-specific part of ASN.1 is called the Basic Encoding Rules (BER).

- One of the claims to fame of ASN.1 BER is that it is used by the Internet standard Simple Network Management Protocol (SNMP).

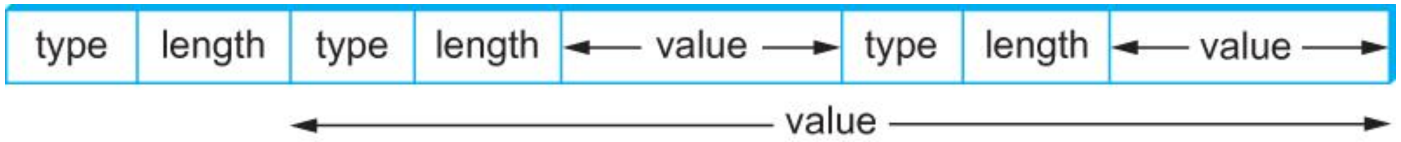


Presentation formatting example: ASN. 1

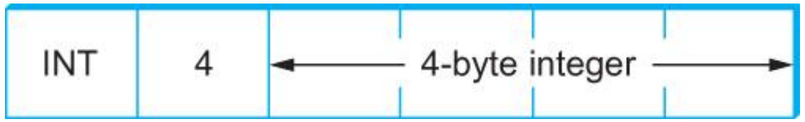
- ASN.1 represents each data item with a triple of the form

  < tag, length, value >

- The tag is typically an 8-bit field, although ASN.1 allows for the definition of multi-byte tags.
- The length field specifies how many bytes make up the value;
- Compound data types, such as structures, can be constructed by nesting primitive types



Compound types created by means of nesting in ASN.1/BER

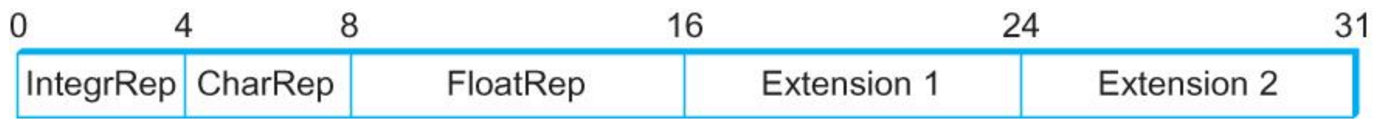


ASN.1/BER representation for a 4-byte integer

國立中正大學 Cybersecurity Lab

Presentation formatting example: NDR

- Network Data Representation (NDR) is the data-encoding standard used in the Distributed Computing Environment
- Unlike XDR and ASN.1, NDR uses receiver-makes-right. It does this by inserting an architecture tag at the front of each message; individual data items are untagged.
- NDR uses a compiler to generate stubs.
  - This compiler takes a description of a program written in the Interface Definition Language (IDL) and generates the necessary stubs
  - IDL looks pretty much like C, and so essentially supports the C type system.



NDR's architecture tag

Presentation formatting example: Markup Languages – XML

- XML syntax looks much like HTML.
- XML syntax provides for a nested structure of tag/value pairs
- A XML format can be both processed by programs and read by humans
- Parsers can be used across different XML-based languages

```xml
<?xml version="1.0"?>
<employee>
<name>John Doe</name>
<title>Head Bottle Washer</title>
<id>123456789</id>
<hiredate>
<day>5</day>
<month>June</month>
<year>1986</year>
</hiredate>
</employee>
```

XML example [data + sitemap]

```
<?xml version="1.0" encoding="UTF-8"?>
- <EmployeeData>
  - <employee id="34594">
      <firstName>Heather</firstName>
      <lastName>Banks</lastName>
      <hireDate>1/19/1998</hireDate>
      <deptCode>BB001</deptCode>
      <salary>72000</salary>
    </employee>
  - <employee id="34593">
      <firstName>Tina</firstName>
      <lastName>Young</lastName>
      <hireDate>4/1/2010</hireDate>
      <deptCode>BB001</deptCode>
      <salary>65000</salary>
    </employee>
  </EmployeeData>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
 xmlns:image="http://www.google.com/schemas/sitemap-image/1.1">
  ▼<url>
     <loc>https://www.gymshark.com/pages/about-us</loc>
     <lastmod>2021-08-13T00:09:33-07:00</lastmod>
     <changefreq>weekly</changefreq>
   </url>
  ▼<url>
     <loc>https://www.gymshark.com/pages/shop-men</loc>
     <lastmod>2021-02-19T09:03:08-08:00</lastmod>
     <changefreq>weekly</changefreq>
   </url>
  ▼<url>
     <loc>https://www.gymshark.com/pages/shop-women</loc>
     <lastmod>2021-02-19T09:03:41-08:00</lastmod>
     <changefreq>weekly</changefreq>
   </url>
  ▼<url>
     <loc>https://www.gymshark.com/pages/terms-and-conditions</loc>
     <lastmod>2020-11-12T07:14:25-08:00</lastmod>
     <changefreq>weekly</changefreq>
   </url>
  ▼<url>
     <loc>https://www.gymshark.com/pages/blackout-terms-and-conditions</loc>
     <lastmod>2018-09-24T05:24:41-07:00</lastmod>
     <changefreq>weekly</changefreq>
   </url>
```

XML Schema

- The definition of a specific XML-based language is given by a *schema*
- There are a number of schema languages defined for XML

Src:EduTechWiki

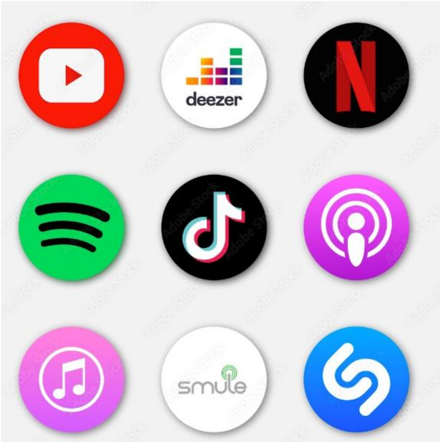
```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="character" type="charType"/>
  <xs:element name="book" type="bookType"/>
  <xs:element name="title" type="xs:string"/>
  <xs:element name="author" type="xs:string"/>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="friend-of" type="xs:string"/>
  <xs:element name="since" type="xs:string"/>
  <xs:element name="qualification" type="xs:string"/>
  <xs:complexType name="charType">
    <xs:sequence>
      <xs:element ref="name"/>
      <xs:element ref="friend-of"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="since"/>
      <xs:element ref="qualification"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="bookType">
    <xs:sequence>
      <xs:element ref="title"/>
      <xs:element ref="author"/>
      <xs:element ref="character"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Multimedia Data

- Multimedia data, comprising audio, video, and still images, now makes up the majority of traffic on the Internet by many estimates.

- Part of what has made the widespread transmission of multimedia across networks possible is advances in compression technology.

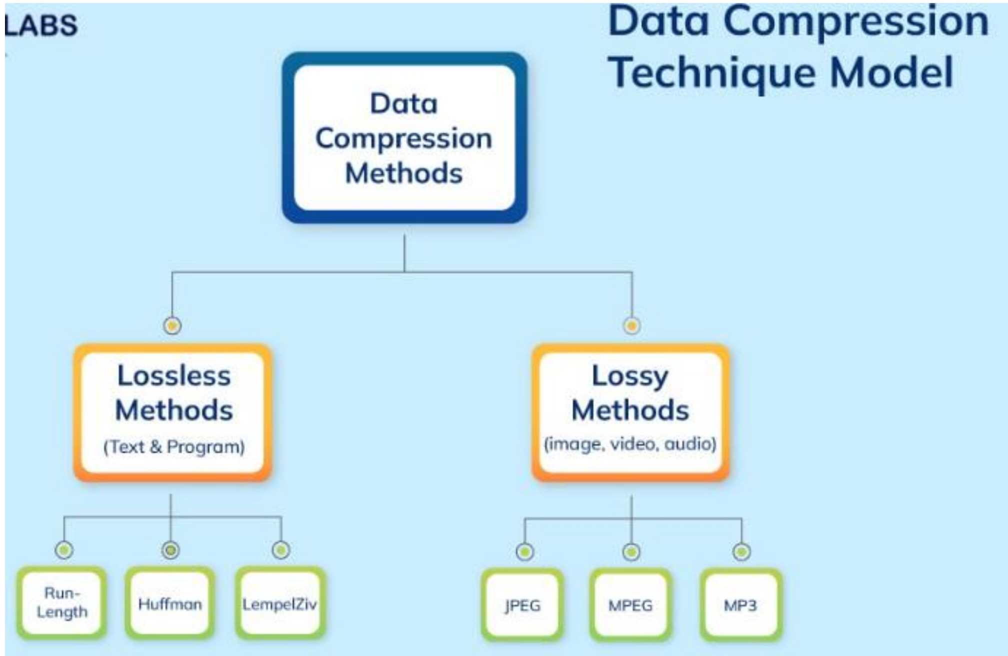
- Because multimedia data is consumed mostly by humans using their senses—vision and hearing—and processed by the human brain, there are unique challenges to compressing it.



Multimedia Data

4K Video: 8GB
+ Lossless: 6GB
+ Lossy: 200 MB

- You want to try to keep the information that is most important to a human, while getting rid of anything that doesn't improve the human's perception of the visual or auditory experience.
- Hence, both computer science and the study of human perception come into play.
- In this section we'll look at some of the major efforts in representing and compressing multimedia data.



Multimedia Data

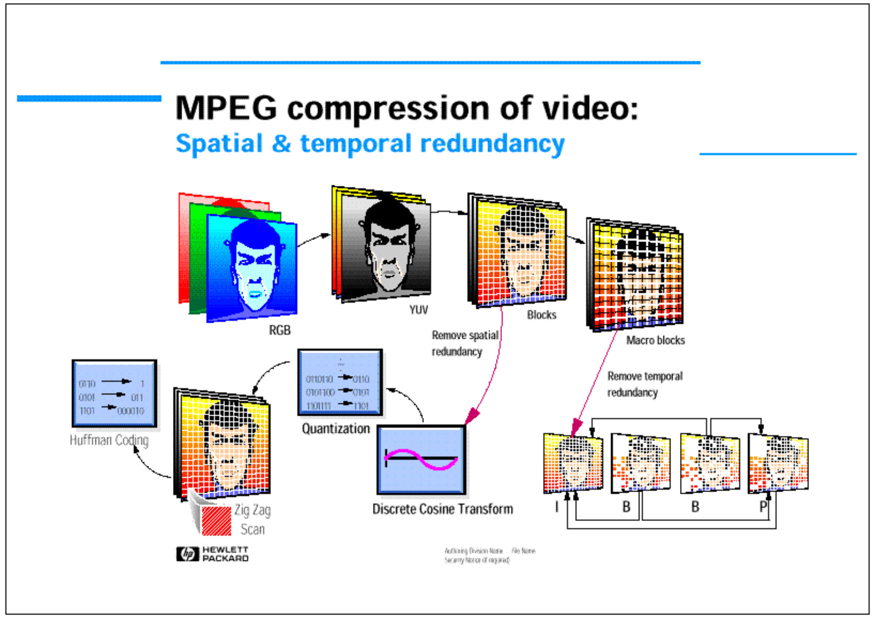
- To get a sense of how important compression has been to the spread of networked multimedia, consider the following example.

- A high-definition TV screen has something like 1080 × 1920 pixels, each of which has 24 bits of color information, so each frame is 1080 × 1920 × 24 = 50Mb and so if you want to send 24 frames per second, that would be over 1Gbps.
-
- That’s a lot more than most Internet users can get access to, by a good margin.
- By contrast, modern compression techniques can get a reasonably high quality HDTV signal down to the range of 10 Mbps, a two order of magnitude reduction, and well within the reach of many broadband users.
- Similar compression gains apply to lower quality video such as YouTube clips—web video could never have reached its current popularity without compression to make all those entertaining videos fit within the bandwidth of today’s networks.

Multimedia Data

- To get a sense of how important compression has been to the spread of networked multimedia, consider the following example.

- A high-definition TV screen has something like 1080 × 1920 pixels, each of which has 24 bits of color information, so each frame is 1080 × 1920 × 24 = 50Mb and so if you want to send 24 frames per second, that would be over 1Gbps.



國立中正大學 Cybersecurity Lab

The importance of compression in multimedia

- If you have an Internet connection with 20Mbps, could you watch 4K on Youtube?

- Why does Youtube support multiple resolutions?

Lossless Compression Techniques

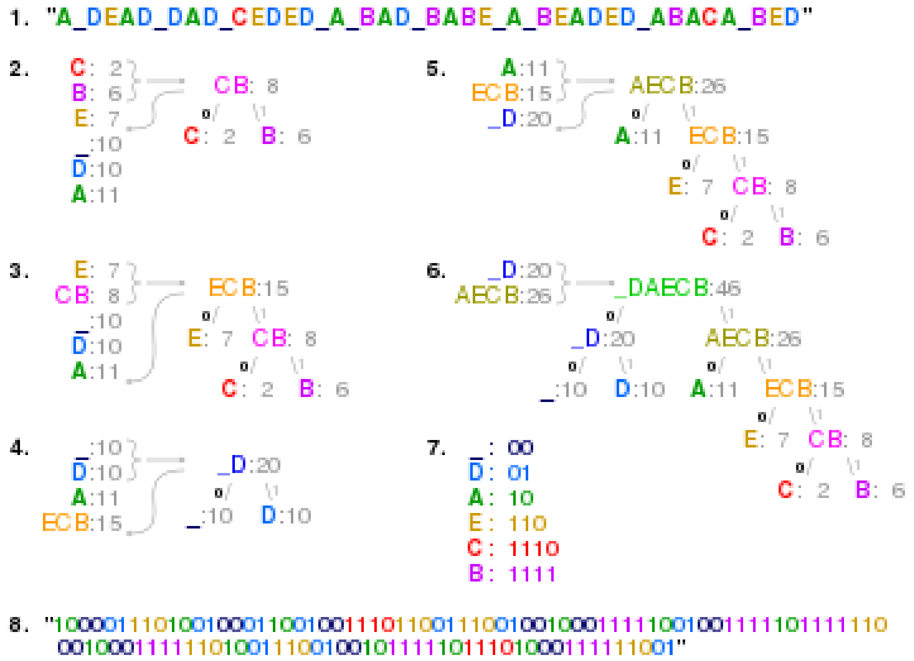
- Compression is inseparable from data encoding
- That is, in thinking about how to encode a piece of data in a set of bits, we might just as well think about how to encode the data in the smallest set of bits possible.
- For example, if you have a block of data that is made up of the 26 symbols A through Z, and if all of these symbols have an equal chance of occurring in the data block you are encoding, then encoding each symbol in 5 bits is the best you can do (since $2^5 = 32$ is the lowest power of 2 above 26).
- If, however, the symbol O occurs 50% of the time, then it would be a good idea to use fewer bits to encode the R than any of the other symbols.

Tom told Jerry that he will go to his home to have dinner tonight!

Technique 1: Huffman codes

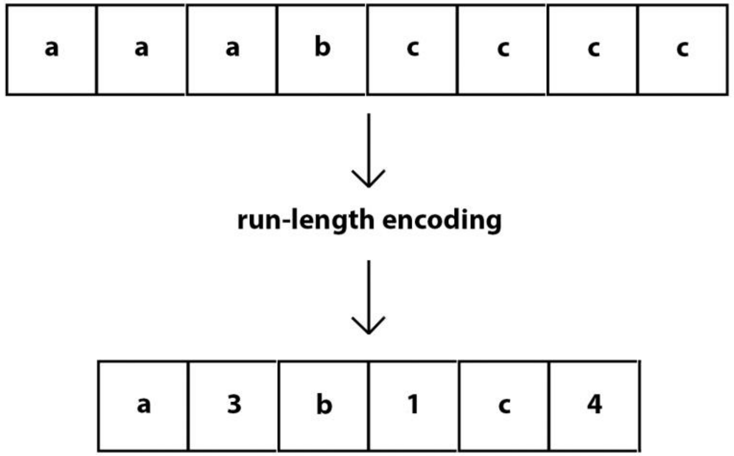
- In general, if you know the relative probability that each symbol will occur in the data, then you can assign a different number of bits to each possible symbol in a way that minimizes the number of bits it takes to encode a given block of data.

- This is the essential idea of *Huffman codes, one of the important early developments in data compression.*



Technique 2: Run length Encoding

- Run length encoding (RLE) is a compression technique with a brute-force simplicity.
- The idea is to replace consecutive occurrences of a given symbol with only one copy of the symbol, plus a count of how many times that symbol occurs—hence the name "run length."
- For example, the string AAABCCCC would be encoded as A3B1C4.



Technique 3: Differential Pulse Code Modulation

- Another simple lossless compression algorithm is Differential Pulse Code Modulation (DPCM).
- The idea here is to first output a reference symbol and then, for each symbol in the data, to output the difference between that symbol and the reference symbol.
- For example, using symbol A as the reference symbol, the string AAABBCDDDD would be encoded as A0001123333 since A is the same as the reference symbol, B has a difference of 1 from the reference symbol, and so on.

AAABBCDDDD → A0001123333

Technique 4: Dictionary based Methods

- The final lossless compression method we consider is the dictionary-based approach, of which the Lempel-Ziv (LZ) compression algorithm is the best known.
- The Unix compress and gzip commands use variants of the LZ algorithm.
- The idea of a dictionary-based compression algorithm is to build a dictionary (table) of variable-length strings (think of them as common phrases) that you expect to find in the data, and then to replace each of these strings when it appears in the data with the corresponding index to the dictionary.

Example

To the swinging and the ringing
of the bells, bells, bells-
of the bells, bells, bells, bells
Bells, bells, bells-
To the rhyming and the chiming of the bells!

Pattern	Reference	Binary value
To	0	0000
the	1	0001
swinging	2	0010
and	3	0011
ringing	4	0100
Of	5	0101
bells	6	0110
Bells	7	0111
rhyming	8	1000
chiming	9	1001
,	10	1010
-	11	1011
!	12	1100

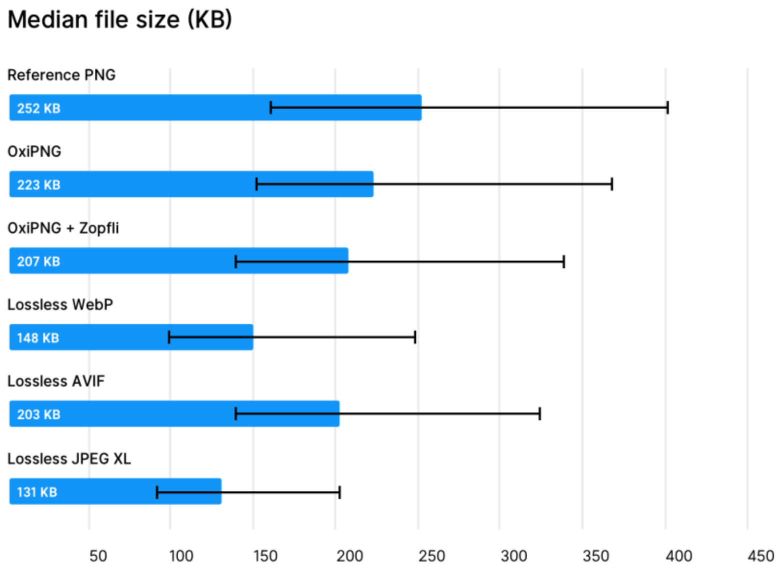
→ 01231451610610……

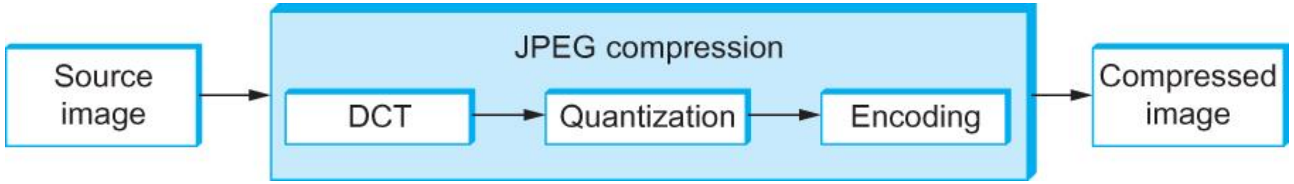
Src: yatishparma

Image Representation and Compression

- Given the increase in the use of digital imagery in recent years—this use was spawned by the invention of graphical displays, not high-speed networks—the need for standard representation formats and compression algorithms for digital imagery data has grown more and more critical.

- In response to this need, the ISO defined a digital image format known as JPEG, named after the Joint Photographic Experts Group that designed it. (The "Joint" in JPEG stands for a joint ISO/ITU effort.)



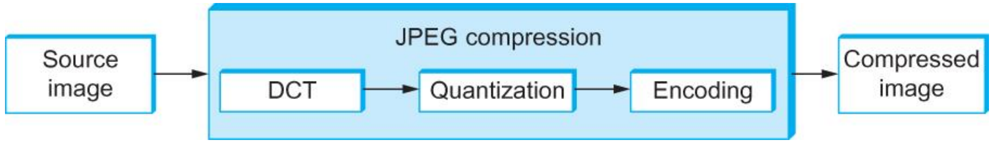


Block diagram of JPEG compression

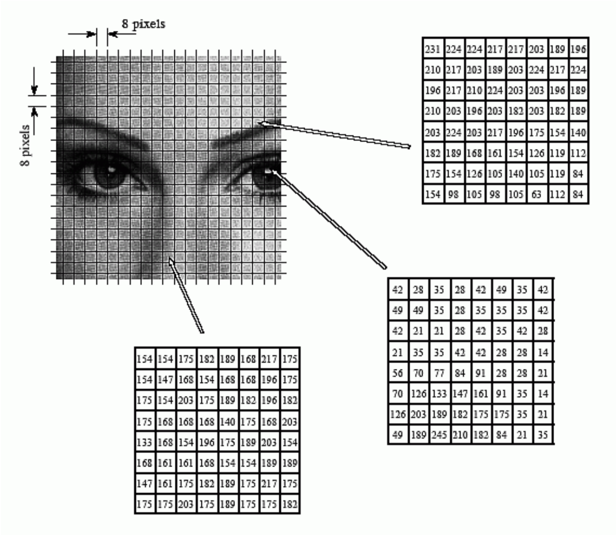
JPEG: Joint Photographic Experts Group

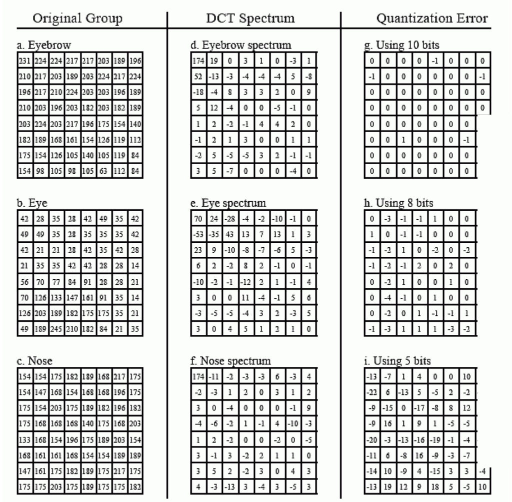
Check: https://www.dspguide.com/ch27/6.htm

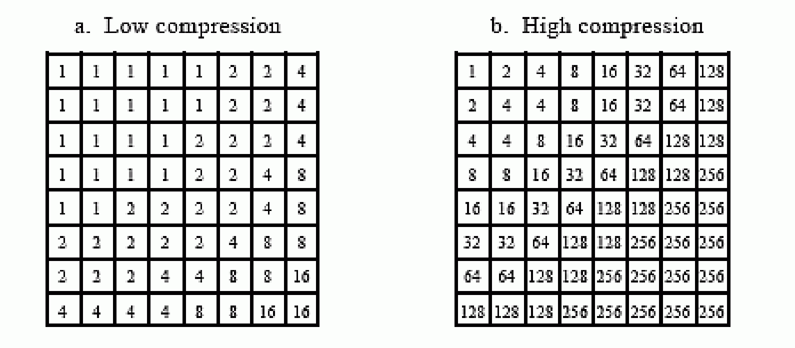
- JPEG is the most widely used format for still images in use today.
- At the heart of the definition of the format is a compression algorithm, which we describe below.
- Many techniques used in JPEG also appear in MPEG, the set of standards for video compression and transmission created by the *Moving Picture Experts Group.*



Block diagram of JPEG compression







a. Low compression

1	1	1	1	1	2	2	4
1	1	1	1	1	2	2	4
1	1	1	1	2	2	2	4
1	1	1	1	2	2	4	8
1	1	2	2	2	2	4	8
2	2	2	2	2	4	8	8
2	2	2	4	4	8	8	16
4	4	4	4	8	8	16	16

b. High compression

1	2	4	8	16	32	64	128
2	4	4	8	16	32	64	128
4	4	8	16	32	64	128	128
8	8	16	32	64	128	128	256
16	16	32	64	128	128	256	256
32	32	64	128	128	256	256	256
64	64	128	128	256	256	256	256
128	128	128	256	256	256	256	256

Discrete Cosine Transform (DCT)

JPEG

- DCT Phase
  - DCT is a transformation closely related to the fast Fourier transform (FFT). It takes an
  - 8 × 8 matrix of pixel values as input and outputs an 8 × 8 matrix of frequency coefficients.
  - You can think of the input matrix as a 64-point signal that is defined in two
  - spatial dimensions (x and y); DCT breaks this signal into 64 spatial frequencies.

DCT, along with its inverse, which is performed during decompression, is defined by the following formulas:

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x,y) \times \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$pixel(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)DCT(i,j) \times \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

where pixel(x, y) is the grayscale value of the pixel at position (x, y) in the 8×8 block being compressed; N = 8 in this case

JPEG

- Quantization Phase
  - The second phase of JPEG is where the compression becomes lossy.
  - DCT does not itself lose information; it just transforms the image into a form that makes it easier to know what information to remove.
  - Quantization is easy to understand—it's simply a matter of dropping the insignificant bits of the frequency coefficients.
  - The basic quantization equation is

    QuantizedValue(i, j) = IntegerRound(DCT(i, j)/Quantum(i, j))

    Where

$$\text{IntegerRound}(x) = \begin{cases} \lfloor x + 0.5 \rfloor & \text{if } x \geq 0 \\ \lfloor x - 0.5 \rfloor & \text{if } x < 0 \end{cases}$$

  - Decompression is then simply defined as

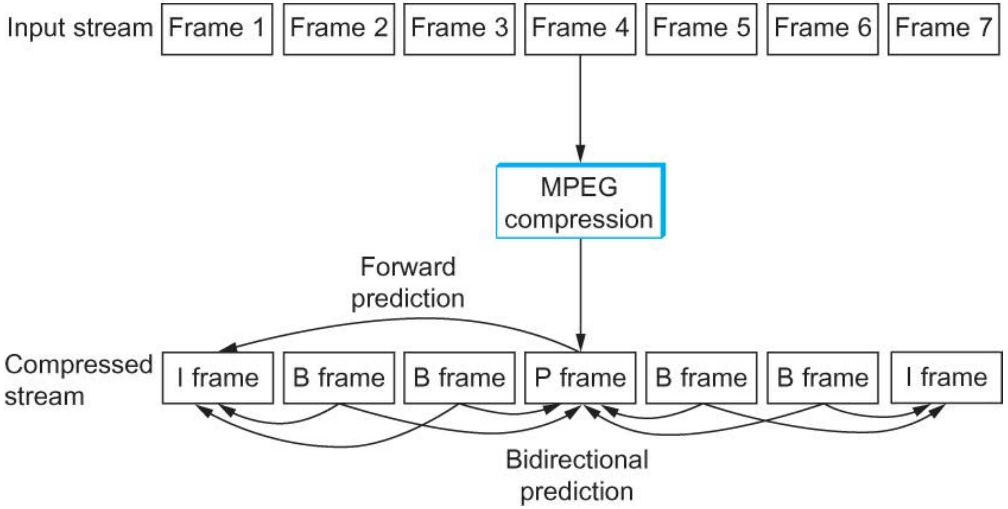
    DCT(i, j) = QuantizedValue(i, j) × Quantum(i, j)

JPEG

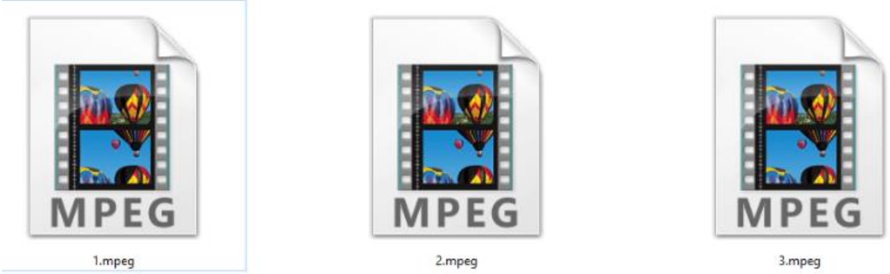
- Encoding Phase
  - The final phase of JPEG encodes the quantized frequency coefficients in a compact form.
  - This results in additional compression, but this compression is lossless.
  - Starting with the DC coefficient in position (0,0), the coefficients are processed in the zigzag sequence.
  - Along this zigzag, a form of run length encoding is used—RLE is applied to only the 0 coefficients, which is significant because many of the later coefficients are 0.
  - The individual coefficient values are then encoded using a Huffman code.

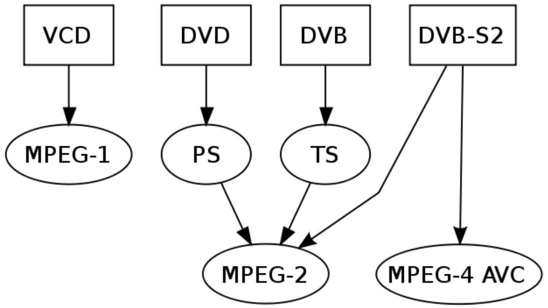
MPEG: Moving Picture Experts Group

- To a first approximation, a moving picture (i.e., video) is simply a succession of still images—also called *frames or pictures—displayed at some* video rate.
- Each of these frames can be compressed using the same DCT-based technique used in JPEG.



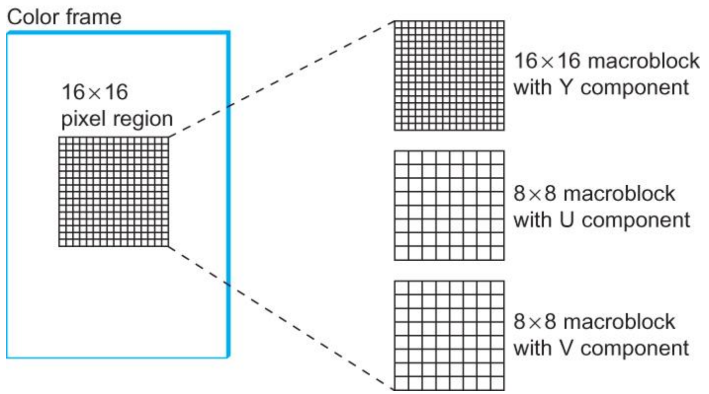
Sequence of I, P, and B frames generated by MPEG.





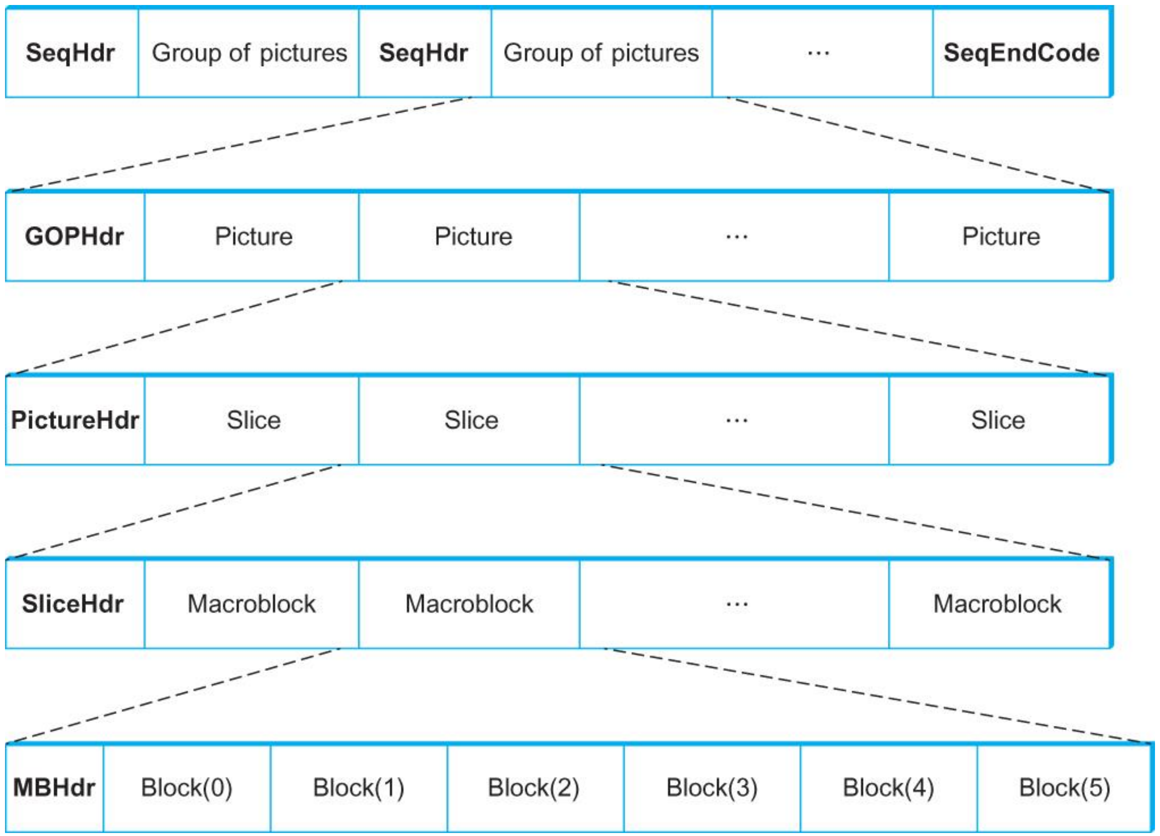
MPEG compression

- Frame Types
  - MPEG takes a sequence of video frames as input and compresses them into three types of frames, called *I frames (intrapicture), P frames (predicted picture), and B frames* (bidirectional predicted picture).
  - Each frame of input is compressed into one of these three frame types. I frames can be thought of as reference frames; they are self-contained, depending on neither earlier frames nor later frames.

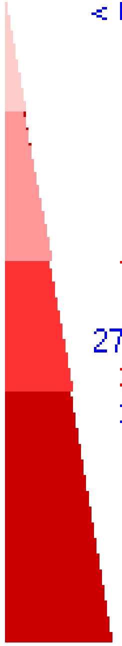


Each frame as a collection of macroblocks

MPEG compression



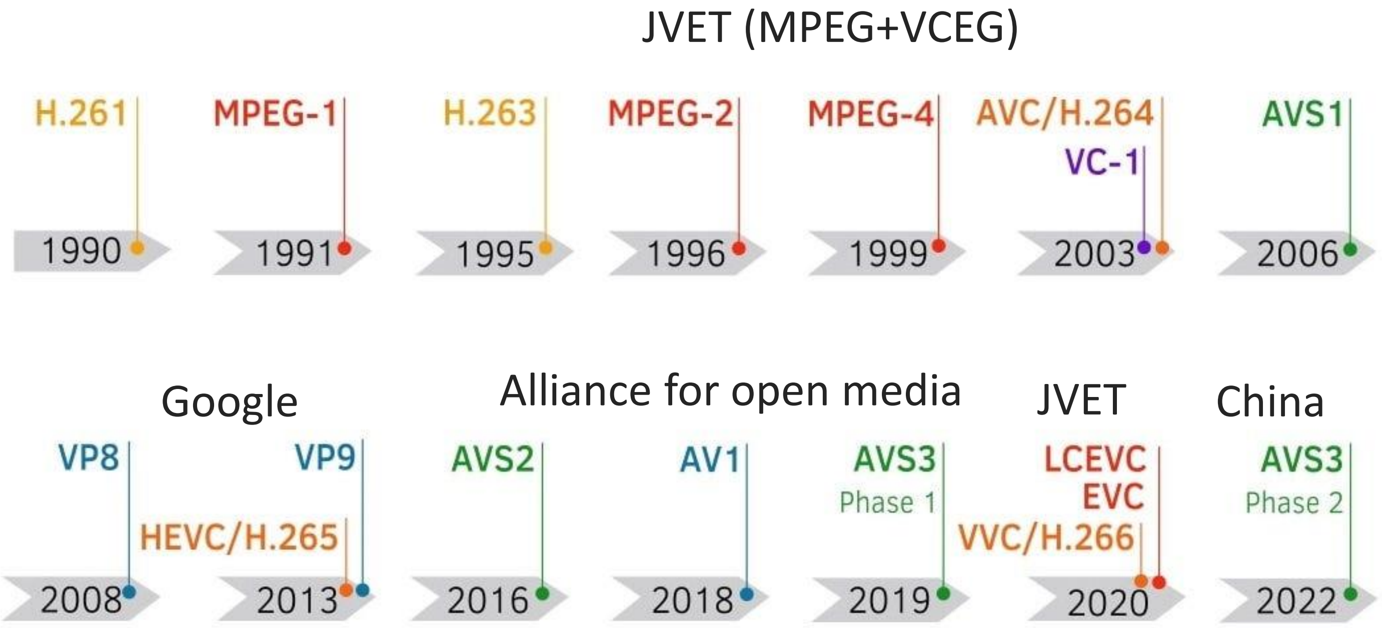
Format of an MPEG-compressed video stream



< 0.384 Mbps	Video conference	(MPEG-4)
<1.5 Mbps	Video in a window	(MPEG-1)
1-2 Mbps	VHS quality full screen	(MPEG-2)
2-3 Mbps	Broadcast NTSC	(MPEG-2)
4-6 Mbps	Broadcast PAL	(MPEG-2)
8-10 Mbps	Professional PAL	(MPEG-2)
12-20 Mbps	Broadcast HDTV	(MPEG-2)
27.5-40 Mbps	DVB satellite multiplex	(MPEG-2 Transport)
32-40 Mbps	Professional HDTV	(MPEG-2)
34-50 Mbps	Contribution TV	(MPEG-2-I)
140 Mbps	Contribution HDTV	(MPEG-2-I)
168 Mbps	Raw NTSC	(uncompressed)
216 Mbps	Raw PAL	(uncompressed)
270 Mbps	Raw contribution PAL	(uncompressed)
1-1.5 Gbps	Raw HDTV	(uncompressed)

MPEG family

- MPEG-4 video, AVC/H.264, MP4 are all parts of the MPEG-4 group
- MP4 file format is a media container. It determines a method of storage rather than a compression algorithm



History of video codecs developments

Graphical Interchange Format (GIF)

- GIF uses the RGB color space, and starts out with 8 bits to represent each of the three dimensions of color for a total of 24 bits.
- Rather than sending those 24 bits per pixel, however, GIF first reduces 24-bit color images to 8-bit color images.
- This is done by identifying the colors used in the picture, of which there will typically be considerably fewer than 2^{24}, and then picking the 256 colors that most closely approximate the colors used in the picture.
- There might be more than 256 colors, however, so the trick is to try not to distort the color too much by picking 256 colors such that no pixel has its color changed too much.

PNG: Portable Network Graphics

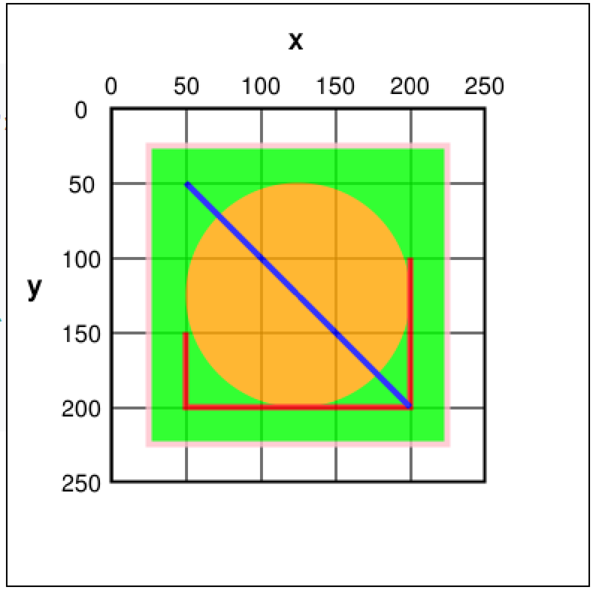
- Portable Network Graphics is a raster-graphics file format that supports lossless data compression
- PNG is an improved, non-patented replacement for GIF
- PNG is designed the format for transferring images on the Internet, not for professional-quality print graphics
- PNG often has a transparent background

SVG: Scalable Vector Graphics

- SVG is an XML-based vector image format for defining two-dimensional graphics, having support for interactivity and animation
- SVG files store images via mathematical formulas based on points and lines on a grid → you can scale up the image without losing quality → good for Responsive design (work well on both mobile + PCs)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"
<svg width="391" height="391" viewBox="-70.5 -70.5 391 391" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
<rect fill="#fff" stroke="#000" x="-70" y="-70" width="390" height="390"/>
<g opacity="0.8">
    <rect x="25" y="25" width="200" height="200" fill="lime" stroke-width="4" stroke="pink" />
    <circle cx="125" cy="125" r="75" fill="orange" />
    <polyline points="50,150 50,200 200,200 200,100" stroke="red" stroke-width="4" fill="none" />
    <line x1="50" y1="50" x2="200" y2="200" stroke="blue" stroke-width="4" />
</g>
</svg>
```

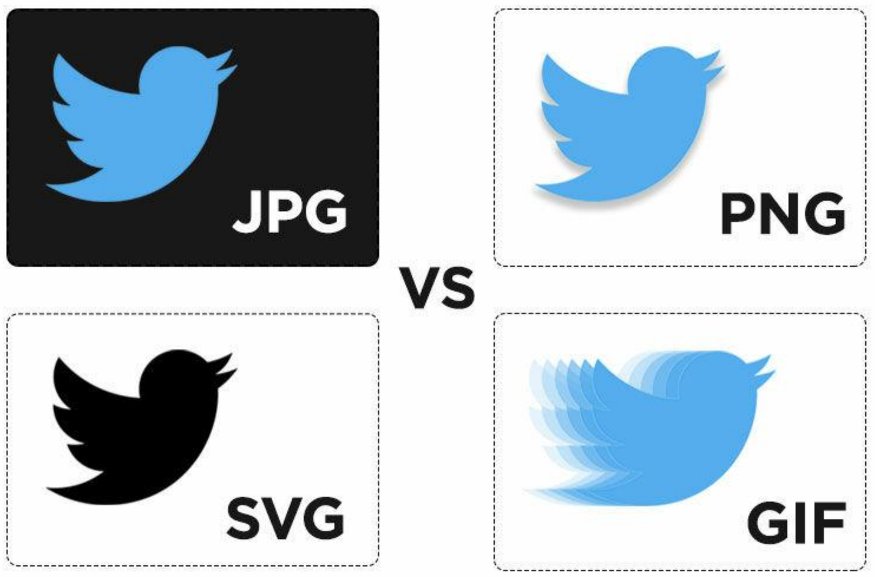
Src: Wikipedia



You can view its code but hard to save

JPEG, PNG, SVG, GIF

	JPG	GIF	PNG	SVG
VECTOR				✓
RASTER	✓	✓	✓	
TRANSPARENCY			✓	✓
ANIMATION		✓	✓	✓
LOSSY	✓			



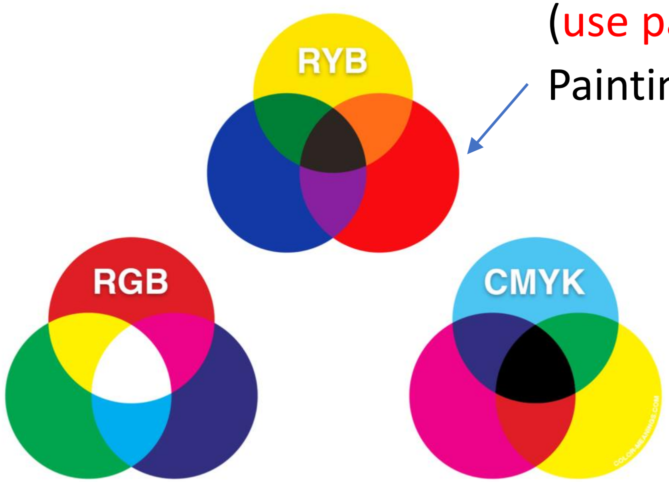
Src: PNGStore

Image size and color model

- Digital images are made up of pixels (hence the megapixels quoted in digital camera advertisements), e.g., iPhone 14 front camera: 48-megapixel
- Each pixel represents one location in the two-dimensional grid that makes up the image, and for color images, each pixel has some numerical value representing a color.

- There are lots of ways to represent colors, referred to as *color spaces:* the one most people are familiar with is RGB (red, green, blue), CMYK (Cyan, Magenta, Yellow and Black) , RYB(red, yellow, and blue).

(use paint, pigment)
Painting, Art design



(use ink)
Printing

(use light)
Video Graphics Array (VGA)
PC/laptops/Digital display

國立中正大學 Cybersecurity Lab

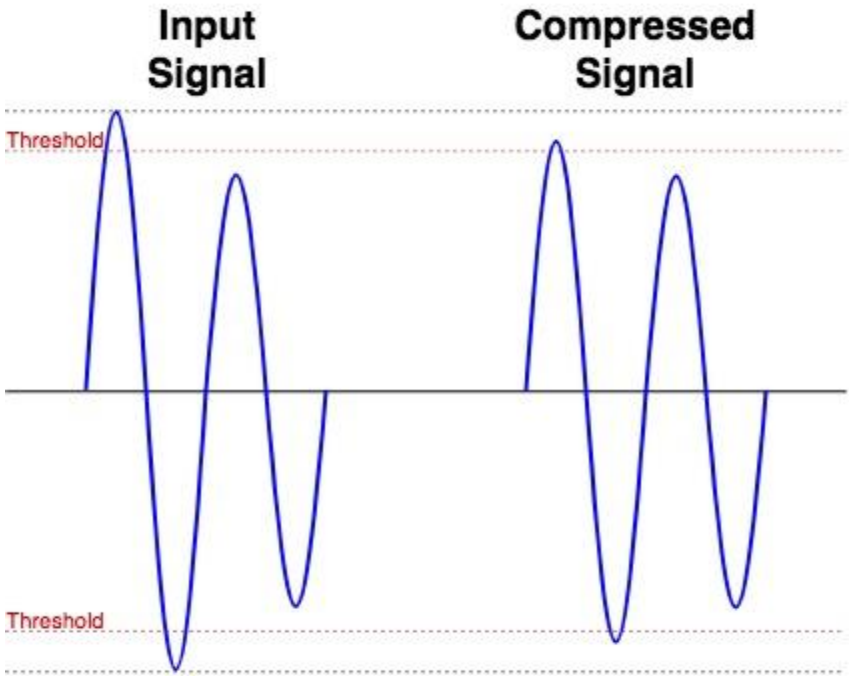
Audio compression file

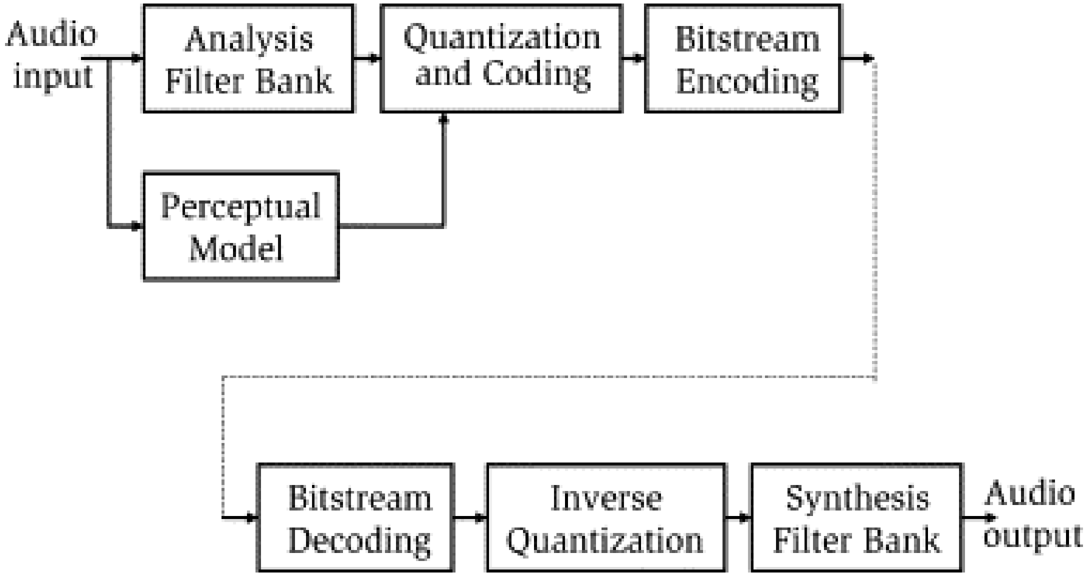
- **Lossless audio files** contain 100% of the audio data: they offer the highest-quality sound, but they also result in the largest file sizes
  - ✓ The two most popular formats of this kind audio are WAV (Waveform Audio Format) and AIFF (Audio Interchange File Format)

- **Compressed lossless audio files**: are designed to squeeze audio data into a smaller file size.
  - ✓ Two examples of this kind audio file formats are FLAC (Free Lossless Audio Codec) and Apple Lossless
- **Compressed lossy audio files:** are made by removing certain types of audio data to shrink the file size
  - ✓ Two examples of this kind file format are AAC (Advanced Audio Coding) and MP3

Compress audio signal

Lossy audio compression transforms the signal into a frequency representation by employing psychoacoustic models which remove information from the signal that is not perceptible to human listeners

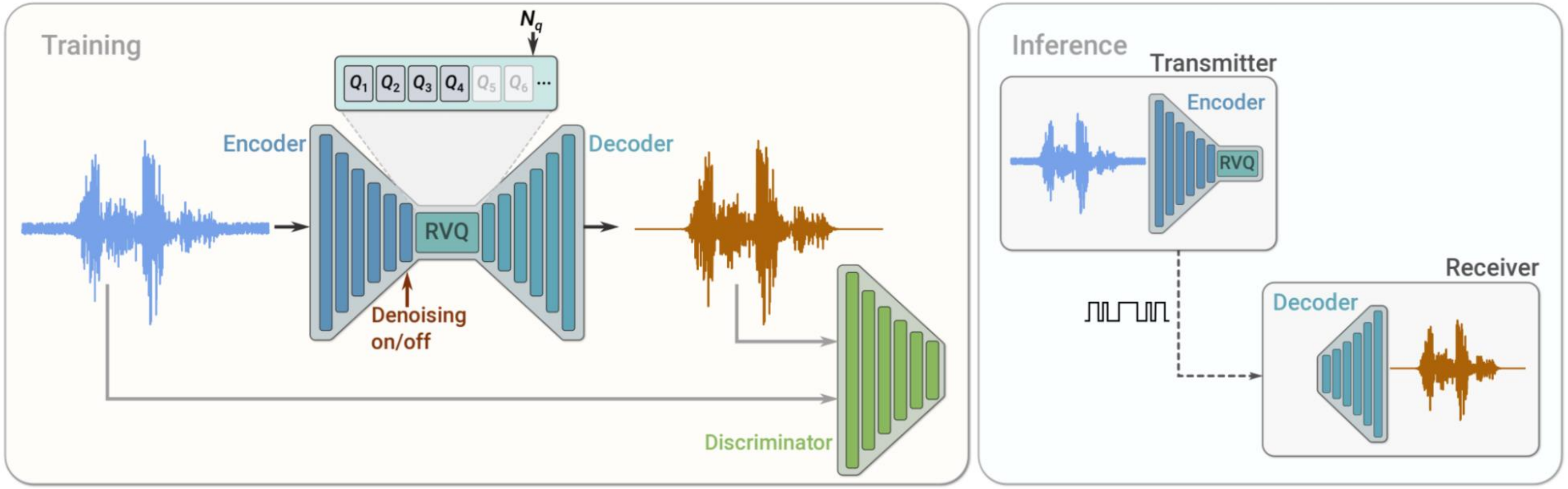




Src: Mymicrophone

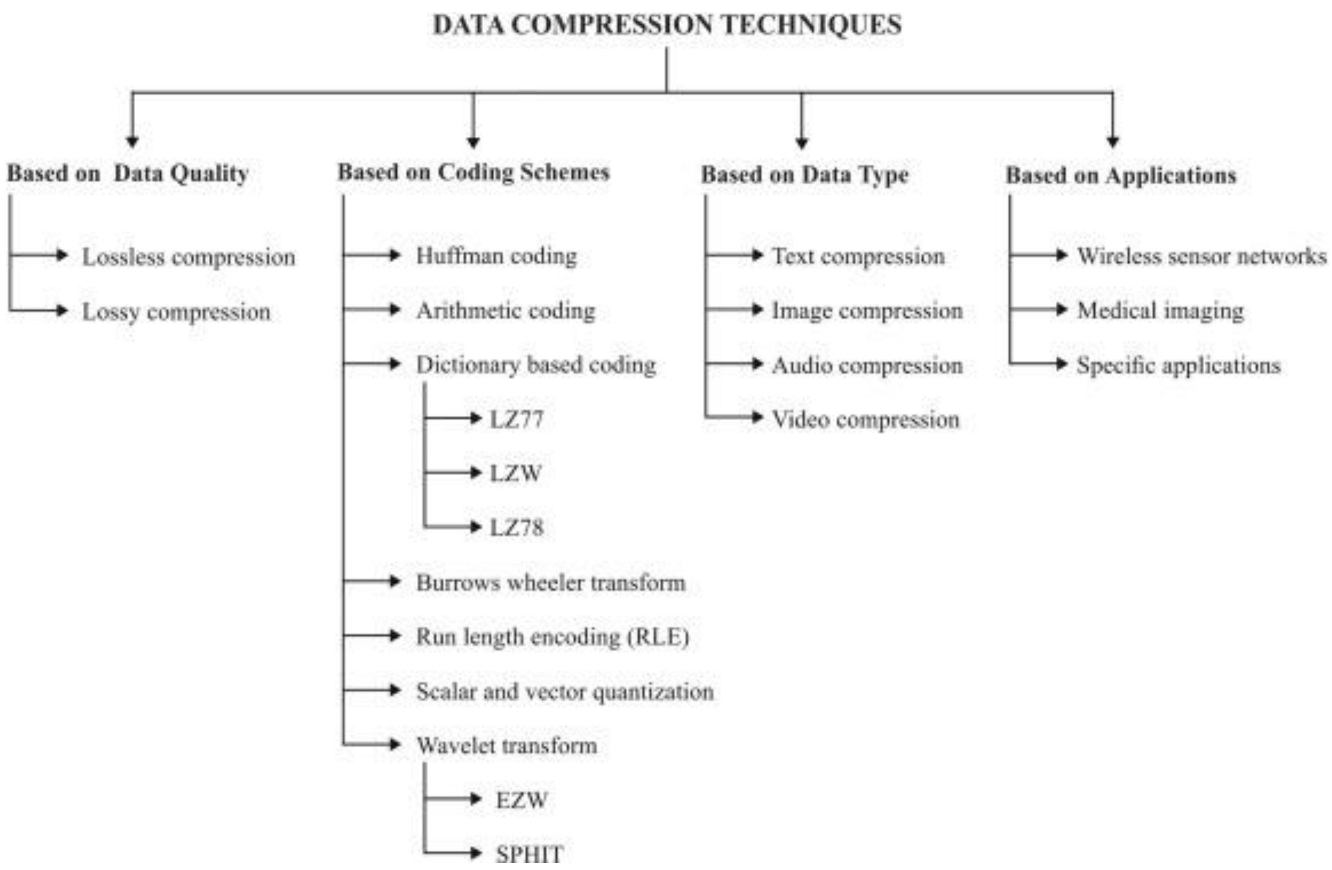
State-of-the-art compression technique

- SoundStream: An End-to-End Neural Audio Codec → Using AI
- During training, the encoder, quantizer and decoder parameters are optimized using a combination of reconstruction and adversarial losses, computed by a discriminator, which is trained to distinguish between the original input audio and the reconstructed audio.
- During inference, the encoder and quantizer on a transmitter client send the compressed bitstream to a receiver client that can then decode the audio signal.



Src: Google

Summary



A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications, 2021