Notes:

RTP

1. RTP : Real-Time Protocol
2. It is a protocol used for delivery of real-time media over IP networks.
3. RTP itself does not provide any mechanism to ensure timely delivery or provide other quality-of-service guarantees, but relies on lower-layer services to do so.
4. RTP is generally carried over UDP but it can be run over TCP as well.
5. TCP can be pretty costly due time-sensitive nature and the sheer volume of packets.
6. RTP packets are typically encapsulated within transport protocols like UDP (User Datagram Protocol) to provide the necessary transport and delivery mechanism.
7. UDP provides a lightweight, connectionless transport layer that offers minimal overhead and low latency, making it suitable for real-time applications.
8. It's important to note that RTP itself does not guarantee reliability, delivery order, or congestion control.
9. Additional mechanisms and protocols,
    - such as RTCP (Real-Time Control Protocol), can be used in conjunction with RTP to
        - Monitor the quality of the transmission,
        - Provide feedback,
        - And perform control functions in real-time communication scenarios.

Format of RTP Packets

1. RTP Header: The RTP header provides important information for the correct interpretation and reconstruction of the media at the receiving end. The RTP header has the following structure:
    - Version (2 bits): Indicates the version of RTP being used.
    - Padding (1 bit): Indicates whether padding is present at the end of the RTP packet.
    - Extension (1 bit): Indicates whether an extension header is present.
    - CSRC Count (4 bits): Specifies the number of contributing sources (CSRC) identifiers present in the RTP packet.
    - Marker (1 bit): Used to mark significant events or

boundaries within the stream.
    - Payload Type (7 bits): Identifies the type of media carried by the RTP packet.
    - Sequence Number (16 bits): Provides a sequence number for each RTP packet to help with packet ordering and loss detection.
    - Timestamp (32 bits): Represents the timestamp of the first sample in the RTP packet.
    - SSRC (32 bits): Synchronization source identifier that uniquely identifies the source of the RTP stream.
    - CSRC List (0-15 entries, each 32 bits): Contains identifiers of contributing sources if present.
    - Extension Header (optional): Additional optional header if the extension bit is set.
2. Payload:
    - Following the RTP header, the payload contains the actual media data, such as audio or video.
    - The payload can vary in format and encoding depending on the media type and chosen codecs.
3. RTP packets are typically encapsulated within transport protocols like UDP (User Datagram Protocol) to provide the necessary transport and delivery mechanism.
4. UDP provides a lightweight, connectionless transport layer that offers minimal overhead and low latency, making it suitable for real-time applications.
5. It's important to note that RTP itself does not guarantee reliability, delivery order, or congestion control.
6. Additional mechanisms and protocols,
    - such as RTCP (Real-Time Control Protocol), can be used in conjunction with RTP to
        - Monitor the quality of the transmission,
        - Provide feedback,
        - And perform control functions in real-time communication scenarios.

RTP Header Field Encoding:

- The bits of each header in RTP Packets are joined one after another.
- Broken-up into blocks of 8 (octets: Here octets just says 8, Essentially a byte).
- Then conveted into Hexadecimal representation.

- Exmaple:

```
Version     (2 bits):  10
Padding     (1 bit):   0
Extension   (1 bit):   1
CSRC        (4 bits):  0000
(There are other Headers but left intentionally to give simple exa

With given individual value, we group them together in group of 8
It gives us: 10 0 1 0000 -> 10010000

10010000 conveted to Hexadecimal:
1001 0000 -> 90 (In Hexadecimal)
```

Hexadecimal

- A byte (8 bits, decimal range: 0-255) in Hexadecimal is always represented by 2 Hex Characters.
- It ranges from 00 to FF, where
  - Each Hexadecimal charater represents 4 bits(a small byte aka nibble, decimal range: 0-15)
  - Exmaple:

```
Decimal    Hexadecimal    Binary

0           00              0000 0000
1           01              0000 0001
.....
128         80              1000 0000
.....
255         FF              1111 1111


If we see for 128 as  example:
In decimal it's 128,
In Binary  it's 10000000
In Hexadecimal, we are breaking the Binary representation in block
so after that it comes as 1000 0000 , which after converting indiv
combining both gives us 80 in Hexadecimal.
```

- Reason of breaking the binary representation in blocks of 4 is that, maximum value that a hexadecimal character can represent is 15.
  - So, if binary value is 1111 it can be represented by a single hex character
  - Anything greater (like 1111 0001) cannot be represented by a single hex character

    - So, we break the binary representation in blocks of 4, which can give maximum of (2) 1111 and 1111, which can easily be represented by 2 single hex character
    - Exmaple:

```
1111 1111 -> FF
```

- lower-layer services: UDP (User Datagram Protocol), IP (Internet Protocol), and the network infrastructure.

References:

- https://datatracker.ietf.org/doc/html/rfc3550
- https://nickvsnetworking.com/rtp-more-than-you-wanted-to-know/
- https://nickvsnetwGorking.com/pyrtp-simple-rtp-library-for-python/
- https://book.systemsapproach.org/e2e/rtp.html
- https://en.wikipedia.org/wiki/RTP_payload_formats
- https://webrtcforthecurious.com/
- https://redirect.cs.umbc.edu/~pmundur/courses/CMSC691C/lab5-kurose-ross.html
- https://www.ietf.org/proceedings/67/slides/avt-5.pdf
- https://datatracker.ietf.org/doc/html/rfc5761#page-4
- https://redirect.cs.umbc.edu/~pmundur/courses/CMSC691C/lab5-kurose-ross.html
- https://github.com/vanshp2002/Video-Streaming-using-RTP-RTSP/tree/main
- https://www.interviewcake.com/concept/java/bit-shift