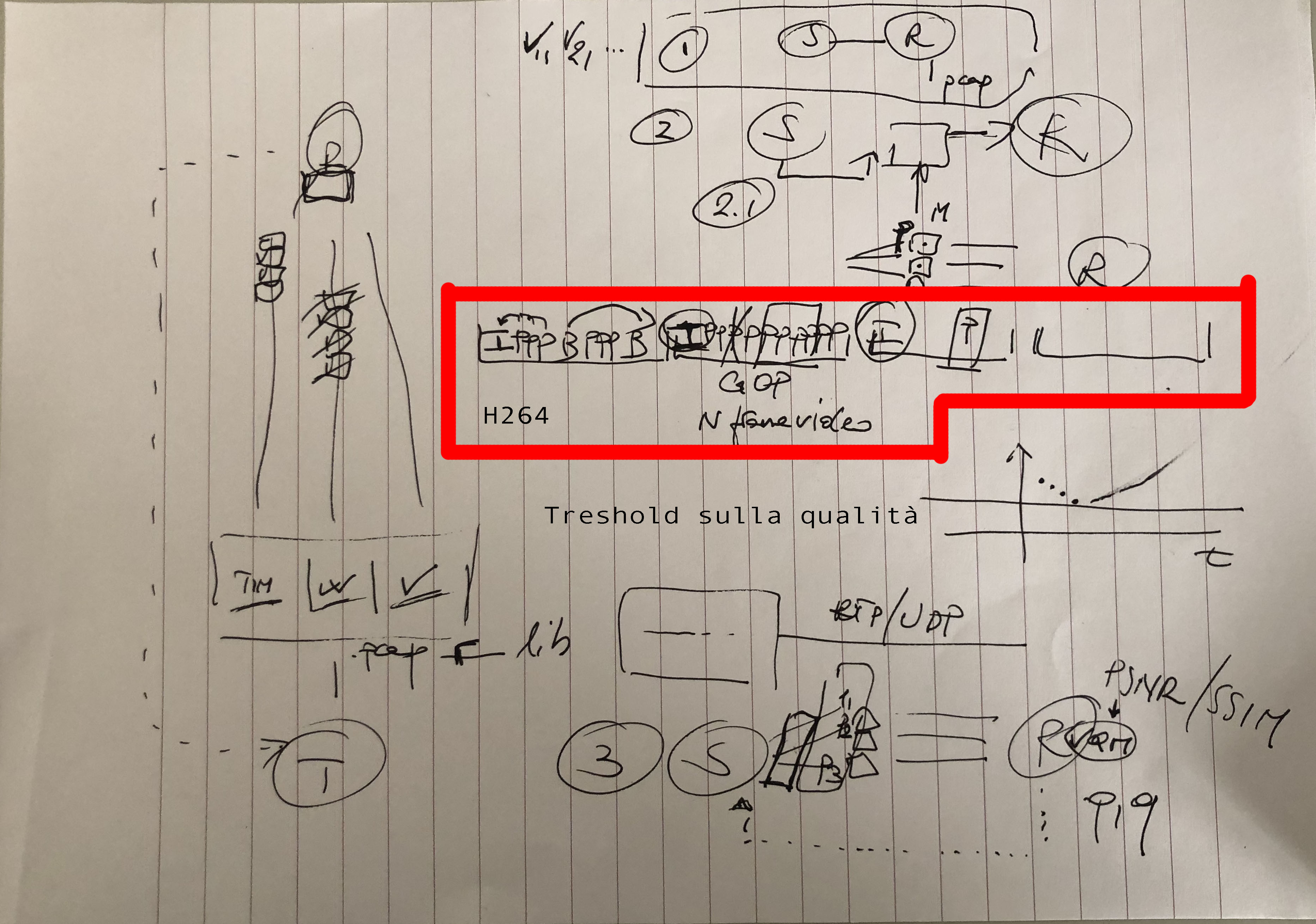
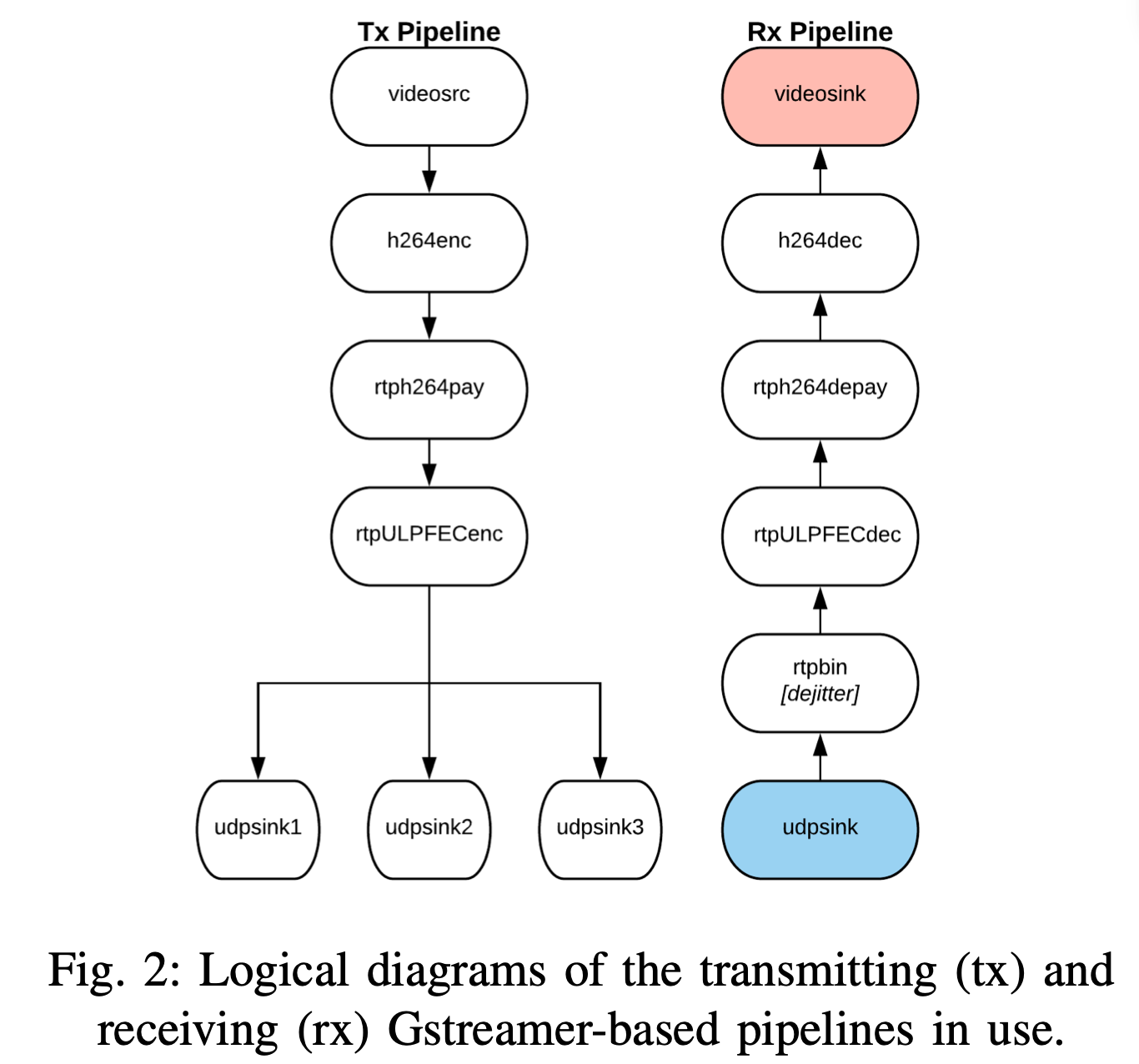
**Riassunto**

Il tirocinio si suddivide in 3 parti:

1. Creazione di un file pcap a partire da un video, creare una sorgente(TX) e il receiver (RX) non è necessario ricompattare, ma a fini dimostrativi è necessario
2. Si iniziano a introdurre gli operatori, non bisogna ancora pensare a una politica di scheduling ma è necessario costruire i moduli per capire quali pacchetti dovranno passare, si potrebbe anche pensare di introdurre una perdita volontaria dei pacchetti
3. Si deve creare un sistema di feedback in cui il receiver invia i P non ricevuti al sender, si deve iniziare a pensare allo scheduler anche in base a questi dati. qui entra anche il trashold, Sarà anche necessario introdurre ml.

**Sviluppo prima parte**

Si è più interessati sulla continuità dell’immagine che sulla effettiva qualità della stessa

Distinzioni tra aree urbane e non(Il miglior operatore potrebbe cambiare, necessità di un feedback)

Vanno ignorati gli udpsink per questa parte. Mi interessa inviare il file, ottenere un pcap(anche ricevendolo tutto insieme e poi aprendolo), e poi aprirlo me ne frego del formato.

Cosa studiare e/o approfondire:

1. Usare python, usare le varie librerie per la realizzazione dello schema
2. Usare Tshark per analisi e creazione dei pacchetti
3. Approfondire rtp

RTP

In [telecomunicazioni](https://it.wikipedia.org/wiki/Telecomunicazioni) l'**RTP** o **Real-time Transport Protocol** è un [protocollo](https://it.wikipedia.org/wiki/Protocollo_di_rete) del [livello applicazioni](https://it.wikipedia.org/wiki/Livello_applicazioni) (e del [livello trasporto](https://it.wikipedia.org/wiki/Livello_di_trasporto)) utilizzato per servizi di comunicazione in [tempo reale](https://it.wikipedia.org/wiki/Real-time) su [Internet](https://it.wikipedia.org/wiki/Internet).

Questo protocollo permette distribuzione di servizi che necessitano di trasferimento in tempo reale, come l'[interattività](https://it.wikipedia.org/wiki/Interattivit%C3%A0) audio e video. Fra questi servizi si trovano anche:

* l'identificazione del payload type
* la numerazione sequenziale
* la marcazione temporale ([timestamp](https://it.wikipedia.org/wiki/Timestamp))
* il monitoraggio.

Solitamente le applicazioni pongono l'RTP sopra l'[UDP](https://it.wikipedia.org/wiki/User_Datagram_Protocol) per le operazioni di [multiplexing](https://it.wikipedia.org/wiki/Multiplexing) e [checksum](https://it.wikipedia.org/wiki/Checksum), anche se può essere usato con altri protocolli di rete e trasporto sottostanti.

**Header**

I pacchetti RTP sono formati da un header di minimo 12 byte seguiti da un payload che dipende dalla specifica applicazione. L'header RTP è formato da:

* **Ver. (Version)**: (2bit) indica la versione del protocollo. La versione corrente è la numero 2.
* **P (Padding)**: (1 bit) indica se è presente un byte di padding alla fine del pacchetto.
* **X (Extension)**: (1 bit) indica la presenza di un Extension header tra l'header standard e il payload.
* **CC (CSRC Count)**: (4 bit) contiene il numero di CSRC identifiers (definiti sotto) che seguono l'header minimo.
* **M (Marker)**: (1 bit) Usato dal livello applicativo e quindi definito dal profilo specifico. Se è settato, significa che il pacchetto ha una qualche speciale rilevanza per il livello applicativo.
* **PT (Payload Type)**: (7 bit) indica il formato del payload e determina la sua interpretazione dall'applicazione. Questo è specifico per ogni profilo RTP.
* **Sequence Number**: (16 bit) il sequence number viene incrementato di uno per ogni pacchetto RTP inviato e permette al ricevente di identificare perdite di pacchetti e ripristinare l'ordine corretto. Il protocollo RTP non prende provvedimenti quanto un pacchetto viene perduto, ma lascia campo libero all'applicazione. In accordo con l'[RFC 3550](https://tools.ietf.org/html/rfc3550), il valore iniziale deve essere casuale per rendere attacchi di tipo [known-plaintext](https://it.wikipedia.org/wiki/Attacco_con_testo_in_chiaro_noto) più difficili.
* **Timestamp**: (32 bit) utilizzato per permettere al ricevente di riprodurre il media ricevuto all'intervallo appropriato. La granularità dipende dall'applicazione ed è definita dallo specifico profilo RTP.

Gli altri elementi hanno poca rilevanza per ora

Inizio sviluppo

Strumenti utilizzati

1. Python