# Υλοποίηση Συστήματος Streaming Client-Server

## 1. Περιγραφή του Συστήματος

Το σύστημα που αναπτύχθηκε αποτελείται από δύο βασικές εφαρμογές: **Streaming Server** και **Streaming Client**, οι οποίες αλληλεπιδρούν μέσω του μοντέλου Client-Server για την επεξεργασία και μετάδοση ροών βίντεο.

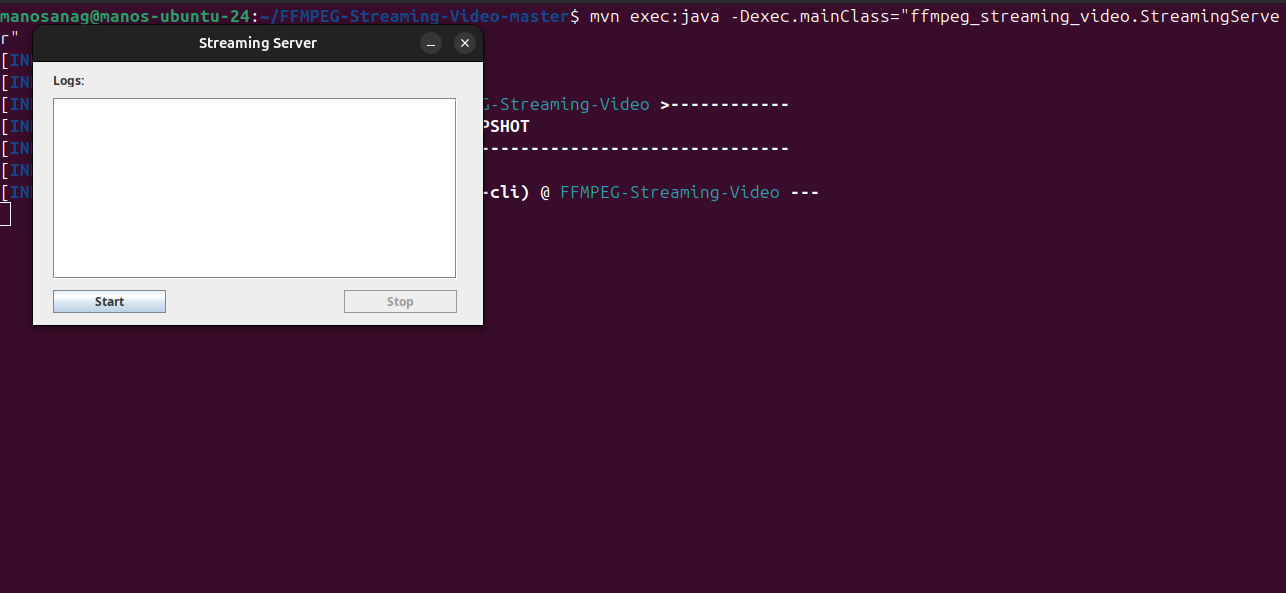
### Streaming Server

* Υλοποιήθηκε σε **Java** χρησιμοποιώντας **Sockets** για επικοινωνία με τους πελάτες.
* Διατηρεί λίστα διαθέσιμων βίντεο σε διάφορα formats (.avi, .mp4, .mkv) και αναλύσεις (240p, 360p, 480p, 720p, 1080p).
* Ελέγχει ποια αρχεία λείπουν και χρησιμοποιεί **FFmpeg** για την αυτόματη δημιουργία των υπολειπόμενων formats και αναλύσεων.
* Διαχειρίζεται αιτήματα από τον Client για επιλογή αρχείου και πρωτοκόλλου μετάδοσης (TCP, UDP, RTP/UDP).
* Υλοποιεί λογική επιλογής αρχείων με βάση την εκτιμώμενη ταχύτητα σύνδεσης του Client (adaptive streaming).
* Χρησιμοποιεί **ProcessBuilder** για να εκκινήσει διαδικασίες FFmpeg ως Video Server.
* Υποστηρίζει **logging** για την καταγραφή ενεργειών και σφαλμάτων.

### Streaming Client

* Υλοποιήθηκε σε **Java** με **Sockets** για επικοινωνία με τον Server.
* Εκτελεί **ταχύτητα σύνδεσης (downlink) test** διάρκειας 5 δευτερολέπτων, χρησιμοποιώντας custom κώδικα κατεβάσματος αρχείων (λόγω προβλημάτων με JSpeedTest).
* Παρουσιάζει στον χρήστη επιλογές για format (.avi, .mp4, .mkv).
* Λαμβάνει από τον Server λίστα αρχείων κατάλληλων με βάση την ταχύτητα σύνδεσης και το format επιλογής.
* Επιτρέπει στον χρήστη να επιλέξει το επιθυμητό αρχείο και το πρωτόκολλο μετάδοσης.
* Χρησιμοποιεί **FFmpeg** μέσω **ProcessBuilder** για την αναπαραγωγή του βίντεο.
* Διαθέτει βασικό **Graphical User Interface (GUI)** με Swing/JavaFX για τη διαχείριση επιλογών.

## 2. Παρουσίαση του Τελικού Υλοποιημένου Συστήματος



### 

### 

### 

### 

### 

### 

### 2.1 Streaming Server

* **Screenshot 1**: Εκκίνηση του Server και σάρωση του φακέλου videos.
* **Screenshot 2**: Δημιουργία αρχείων μέσω FFmpeg (εμφάνιση logs δημιουργίας).
* **Screenshot 3**: Ετοιμότητα για σύνδεση Client.

**Παράδειγμα Εκτέλεσης 1:**

Ο Server ανιχνεύει τα υπάρχοντα αρχεία και δημιουργεί ο,τι λείπει απο μικρότερες αναλύσεις για όλα τα formats:

Creating the missing videos for all available formats with lower resolutions...

Videos created successfuly!

...

Ο Server περιμένει Client:

Listening for requests…

**Παράδειγμα Εκτέλεσης 2:**

Ο Server ανιχνεύει τα υπάρχοντα αρχεία και δημιουργεί ο,τι λείπει απο μικρότερες αναλύσεις για όλα τα formats:

Creating the missing videos for all available formats with lower resolutions...

Videos created successfuly!

...

Ο Server περιμένει Client:

Listening for requests...

### 2.2 Streaming Client

* **Screenshot 4**: Εκκίνηση Client και εκτέλεση Speed Test.
* **Screenshot 5**: Εμφάνιση διαθέσιμων formats και επιλογή από χρήστη.
* **Screenshot 6**: Λήψη λίστας κατάλληλων αρχείων από τον Server.
* **Screenshot 7**: Επιλογή αρχείου και πρωτοκόλλου από τον χρήστη.
* **Screenshot 8**: Αναπαραγωγή του video με FFmpeg.

**Παράδειγμα Εκτέλεσης 1 :**

[DEBUG] 2025-04-29 19:40:05.790 [AWT-EventQueue-0] StreamingClient - Connect button has been pressed

[DEBUG] 2025-04-29 19:40:08.391 [AWT-EventQueue-0] StreamingClient - Download speed: 0.7 Mbps.

[DEBUG] 2025-04-29 19:40:08.392 [AWT-EventQueue-0] StreamingClient - Sending request to server: 0.7 download speed and mkv format

Received request for .mkv format

Available videos:

- Forrest\_Gump-240p.mkv

- The\_Godfather-360p.mkv

...

Μετά την επιλογή:

Using the UDP protocol to stream 'The\_Godfather-360p.mkv'

**Παράδειγμα Εκτέλεσης 2 :**

[DEBUG] 2025-04-29 19:40:05.790 [AWT-EventQueue-0] StreamingClient - Connect button has been pressed

[DEBUG] 2025-04-29 19:40:08.391 [AWT-EventQueue-0] StreamingClient - Download speed: 5.7 Mbps.

[DEBUG] 2025-04-29 19:40:08.392 [AWT-EventQueue-0] StreamingClient - Sending request to server: 5.7 download speed and mp4 format

Received request for .mp4 format

Available videos:

- Forrest\_Gump-720p.mp4

- The\_Godfather-360p.mp4

...

Μετά την επιλογή:

Using the TCP protocol to stream 'Forrest\_Gump-720p.mp4'

## 3. Παρουσίαση Προβλημάτων και Επίλυση

### Προβλήματα Κατά την Υλοποίηση:

1. **Ανεύρεση και χρήση βιβλιοθήκης ταχύτητας (SpeedTest)**
   * Το αρχικό σχέδιο περιλάμβανε τη χρήση της βιβλιοθήκης speed-test-lib.
   * Λόγω ανυπαρξίας της εξάρτησης protocol-http, υλοποιήθηκε custom download tester με HttpURLConnection.
2. **Σφάλματα σύνδεσης σε δημόσιους servers για download tests**
   * Τα παλιά test URLs (π.χ. ikoula.testdebit) δεν λειτουργούσαν.
   * Χρησιμοποιήθηκαν νέα URLs από speedtest.tele2.net.
3. **Συγχρονισμός και σταθερότητα μετάδοσης**
   * Ειδικά με UDP και RTP/UDP απαιτήθηκε σωστό χειρισμό buffer sizes και συντονισμός εκκίνησης Client/Server FFmpeg διαδικασιών.
4. **FFmpeg commands portability**
   * Αναγκαστική χρήση πλήρων paths ή ρύθμιση system PATH στο project για να λειτουργούν τα FFmpeg processes χωρίς σφάλματα.
5. **Διαχείριση GUI και απλότητα**
   * Για απλοποίηση της εργασίας και αποφυγή πολύπλοκου state management, επιλέχθηκε βασικό Swing GUI αντί JavaFX.