

Project: Video Surveillance over IP

Prerequisites

- 1) This is an individual project.
- 2) You can use any programming language you want as long as you do not copy/paste someone else's code.

Submission

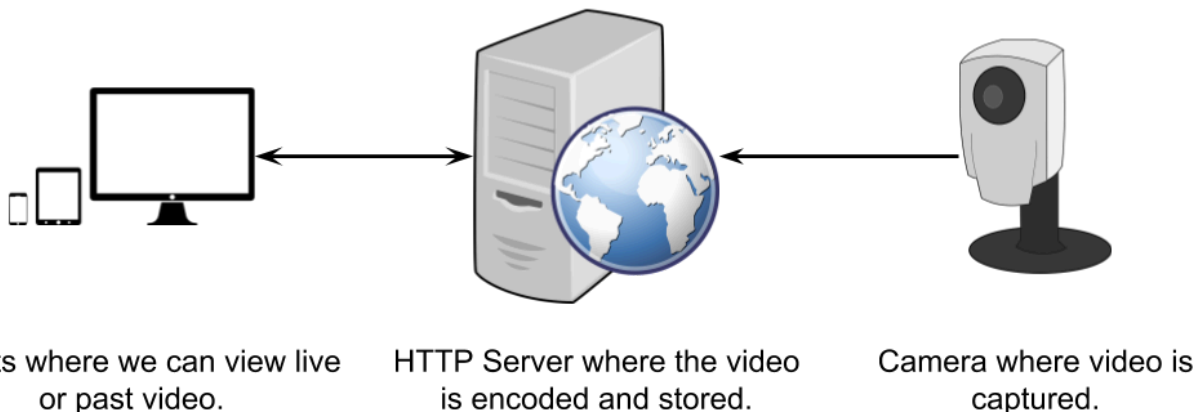
Create a single zip file that includes everything (code, results, report, etc.), name the file `lastname_project2.zip` and upload it in Google Classroom. In the report, list all the SDKs or open-source code you used, where and how you used them, if any.

Demo

We will test your demo online. You will run the server on your side, give us the URL for the Web site and I will test the video in my browser remotely (assuming I can connect to your server). **Failing that, you can play the video on your browser and show it to me through screen sharing.** The demo session will be more like a Q&A session.

Problem to Solve

Here is the problem:



You will need a camera like a webcam (built-in or a USB one) that is attached to your server where you will be running an HTTP server. The goal is to capture real-time video, encode it and store it on the server. Viewers using a Web client (in the browser) should be able to connect to the server and watch the live video.

Here are the pieces you need to develop and make work:

- 1) Connecting to the camera and capturing video. Use FFmpeg.
- 2) Encoding the content to reduce its raw size in real time and storing it. Use FFmpeg.
- 3) Making the encoded content available to serve over HTTP in DASH format (DASH live profile). You can use, e.g., Apache or NGINX for the HTTP server.
- 4) An HTML5 page (media source extensions enabled, see, for example, dash.js) that a desktop or a mobile browser can use to view the encoded content. Check <https://caniuse.com/?search=mse> for compatibility and note that iOS (iPhone) will not work with DASH, you can try an Android phone if you have access to one).
- 5) On this page, the user should minimally have the following controls: rewinding to a specific time, pausing, jumping back to the live point, and taking a screenshot (which will save the current frame as an image like jpeg/png).

The majority of the work above can be easily completed by FFmpeg. You need to find and use the correct scripts. Never hard-code any system-specific parameter such as paths. They should always be a config parameter.

Optionally, you can develop the following for extra points:

- Capturing (from a mic), encoding and viewing not just the video but also the audio.
- Streaming the video adaptively in DASH (e.g., high, medium and low resolution) based on the device and connection speed (as opposed to single-bitrate streaming).
- Motion detection capability on the server to send a notification to the viewers (think of a security camera application).
- Better UI features on the page such as showing thumbnails that will make seeking easier (like YouTube or Netflix).

Important notes:

- Use H.264/AVC for encoding the video and AAC for the audio (if you have it), use segment sizes of 2, 4 and 6 seconds. Try measuring the latency for each segment duration.
- The HTML5 page should not use any plug-ins. Only use HTML5 MSE and JavaScript. Start reading from:
 - https://www.w3schools.com/html/html5_video.asp
 - https://developer.mozilla.org/en-US/docs/Web/HTML/DASH_Adaptive_Streaming_for_HTML_5_Video
 - <https://reference.dashif.org/dash.js/latest/samples/>
 - <https://dashjs.org/>

Note that what we need is **live streaming with timeshifting** capability.

- It is preferred to write your own basic player with HTML5/JavaScript than using someone else's. Even if you do not write the entire code for the HTML5/JavaScript yourself, you are expected to know how the player actually works and what FFmpeg does and so on.