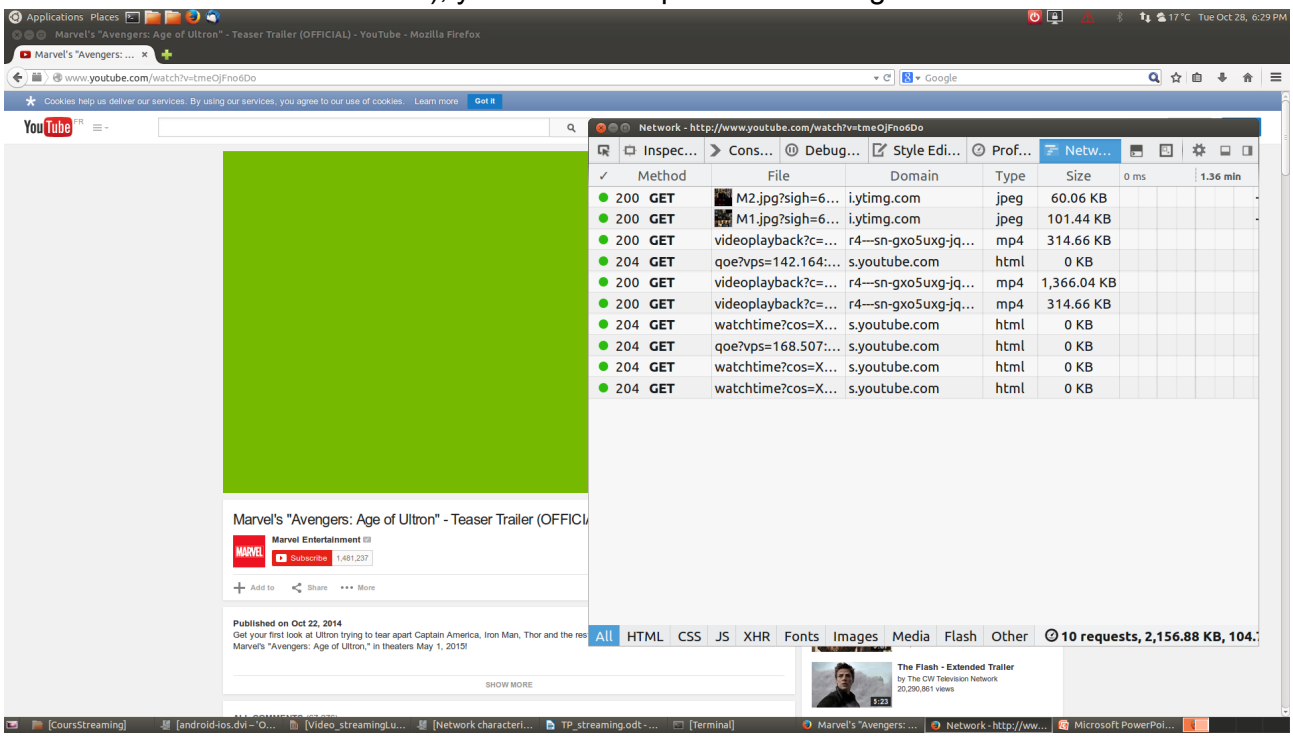
Lab HTTP streaming

Open Youtube web site with Firefox. Then open the Web developer console from Firefox (in the Tools menu and choose network). To better carry the subsequent analysis, detach the console from the window.

Select and start viewing a popular video (any music clip, just check that there is indeed a video, not only a still image).
In the steady state (once the possible small objects got downloaded and only the main multimedia content is transfered), you must end up with something like:



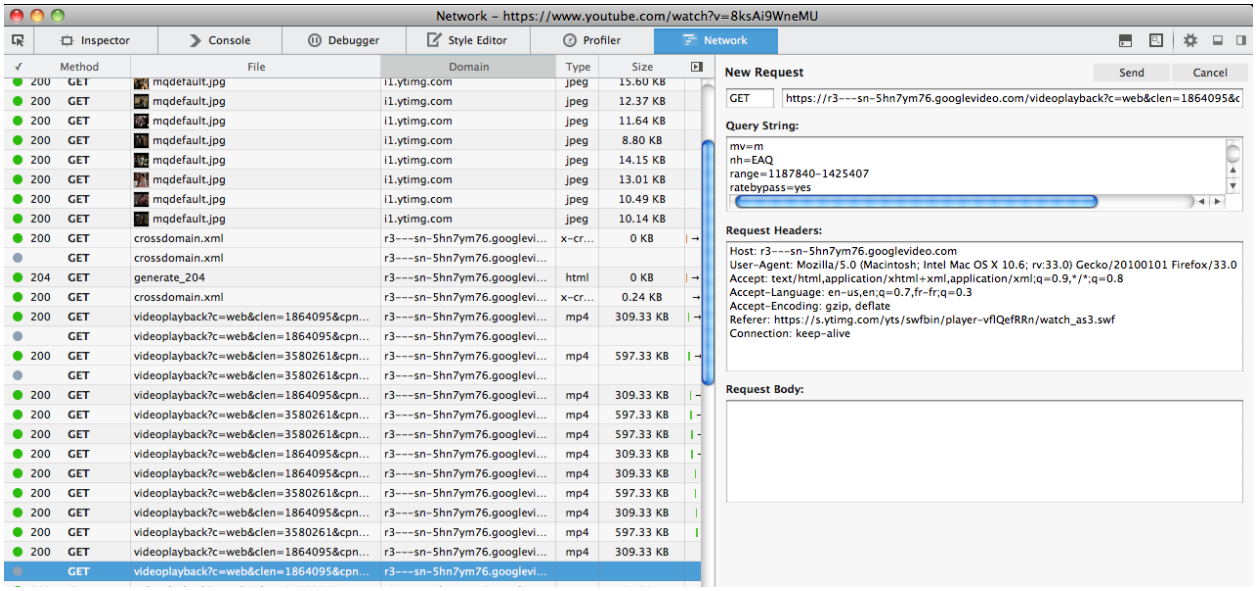
Each line of the network console represents a HTTP request+HTTP response, the details are visible when clicking on.

Task 1: derive the video server name of this video from the Network tab by looking for the GET request for video object (hint: look for the highest sized HTTP exchanges).

Task 2: Can you distinguish between audio and video streams ? Are they serviced by the same server? What are the container and encoding types?

Task 3: How is data transferred:

1. From a single or several servers?
2. A single or several GET requests? If several, analyze and comment the type of request by inspecting the request header, the response header and the *params* tab (what about the *range*?).



Task 4: More into the details:

1. Are the GET requests serviced over a single TCP connection or several connections
2. Resolve the IP of the video server. Is it far or close from you machine?

Task 5: Resolve the IP of the video server from another DNS resolvers using https://www.whatsmydns.net/. Is there a different name-to-IP mapping depending on the location?

Task 6: In the article “Vivisecting YouTube...”, it is referred to lscaches as the primary servers that are mapped on different servers. Take one of them, as they still exist:
v23.lscache1.c.youtube.com
Resolve it with https://www.whatsmydns.net/
What can you say?

Task 7: pick several videos. Do you always obtain the same video server? Is the mapping of the video to a specific server apparently constant or not?

Task 8: Carry out a similar analysis with Dailymotion :

- how many servers serve a video?
- what is the type of encoding?
- is it always the same server servicing a video?