

Streamer en direct un flux goPro sur internet

Hédi ZITOUNI - Léo BONGIORNO

But

Pour ce projet, nous souhaitons mettre en place un système qui nous permettra de diffuser en direct le flux d'une GoPro sur Internet. La diffusion sur Internet peut se faire de différentes façons en fonction de son utilité, diffusion sur des sites de streaming live (comme Youtube, Twitch...) ou pour des utilisations plus personnelles (Caméra de surveillance). La GoPro pourrait également être remplacée par une autre caméra qui dispose des mêmes fonctionnalités, mais nous nous sommes concentré sur la GoPro pour ce projet.

Mise en situation (scénario)

Les GoPros de dernière génération (depuis Hero 3 Session) disposent d'une connectivité Wi-Fi qui nous permet de visualiser le live depuis un ordinateur. Le challenge est donc de créer / configurer un système qui capte ce flux pour le diffuser ensuite selon l'utilité.

Nous nous sommes procuré une carte UDOO, qui est un micro-ordinateur (comme un raspberry pi), et qui va faire cet intermédiaire entre la GoPro et le service de streaming. Grâce à cela, nous pouvons imaginer de nombreux scénarios pour notre projet.

La GoPro pourrait se transformer en caméra de surveillance pour surveiller d'éventuelles intrusions chez soi ou pour surveiller ces enfants / animaux de compagnies.

La GoPro pourrait également garder son statut de caméra sportive, mais en diffusant en direct sur YouTube / Twitch ces sessions de sports grâce à un micro-ordinateur monter sur batterie / puce 3G.

Technologie que nous utilisons

Pour le projet, nous avons à notre disposition une carte [UDOO](#), avec comme système d'exploitation [udoobuntu](#) qui est un dérivé d'ubuntu, mais en plus léger et moins coûteux en énergie. Cette carte possède deux interfaces réseaux, ce qui est utile et primordiale dans notre projet. Une interface réseau pour la GoPro, et l'autre pour accéder à internet.

Nous avons également une caméra GoPro (Hero 3 wifi). La particularité des GoPros wifi est qu'elle diffuse un réseau wifi autour d'elle. En se connectant à ce réseau Wi-Fi, nous pouvons visionner en direct ce que filme la GoPro en se connectant à cette adresse, (si on dispose des bonnes librairies) : <http://10.5.5.9:8080/live/amba.m3u8>

Déroulement du projet

Première approche : Tout faire avec la carte UDOO

La première étape consistait à récupérer le flux diffusé par la GoPro par Wi-Fi.

VLC

Nous avons avant tout testé sur un ordinateur Windows pour essayer de récupérer le flux. Pour récupérer ce flux, il suffit de se connecter au réseau Wi-Fi de la GoPro, de lancer VLC (Lecteur multimédia), Média->ouvrir un flux, et de rentrer l'adresse suivante : <http://10.5.5.9:8080/live/amba.m3u8>

Avec quelques secondes de décalage, le visionnage du stream de la GoPro marchait parfaitement.

Grâce à cela nous savions que le flux était accessible à cette adresse. Après avoir installé VLC sur [udoobuntu](#), nous avons retesté l'expérience directement depuis la carte mais le résultat était différent. Une seule image du flux s'affichait et n'était jamais modifiée.

En analysant les logs et les messages d'erreur que la console nous donnait, nous n'avons pas réussi à identifier le problème exactement et nous sommes passé à autre chose.

OBS (Open broadcaster Software)

Une deuxième solution que nous avons testé avec réussite sur Windows était d'utiliser un outil de streaming tel que [OBS](#) qui permettait de récupérer le flux et de le streamer directement.

L'adresse de la goPro sert en fait de flux d'entrée comme une image ou une simple video pour mettre en forme ce que l'utilisateur de la plateforme de stream verra.

Le problème est que [udoobuntu](#) manque beaucoup de librairie par rapport à ubuntu. Nous avons passer énormément de temps à chercher celles manquantes et surtout comment les installer.

[FFMPEG](#) en faisait partie, et ça été très compliqué de se le procurer. Très peu de documentation ni de forum sont présents pour expliquer comment l'installer et pourquoi c'est si difficile de l'obtenir pour le système udoobuntu.

Finalement, nous avons réussi à l'installer, nous avons donc pu continuer à chercher les autres librairies manquantes pour OBS jusqu'à ce qu'il ne se lance toujours pas sans nous donner aucune information. Il faut savoir que OBS est libre de droit et a été porté sur de nombreux système linux, malheureusement pas spécifiquement pour udoobuntu.

Nous avons donc dû abandonner l'idée.

GStreamer

Une autre solution se basait sur [Gstreamer](#). C'est un framework multimédia, permettant de traiter des données audio et video, qui fonctionne grâce à une suite de `!`.

La première étape, comprendre le fonctionnement des `!`. C'est exactement comme les pipes sur Unix, le flux de sortie de la commande précédente arrive en entrée de la commande suivante.

Par exemple :

```
gst-launch1.0 souphttpsrc
location=http://10.5.5.9:8080/live/amba.m3u8 ! autovideosink
```

Cette commande permet de récupérer le flux de la GoPro (on reconnaît l'adresse). La sortie de `souphttpsrc` (le flux video) est envoyé en entrée de `autovideosink` (qui choisit un lecteur video pour afficher l'entrée à l'écran).

Il existe des fonctions pour à peu près tout faire, demultiplexer, multiplexer, formater, réduire la taille...

Deuxième étape, trouver les bonnes fonctionnalités à mettre entre les `!` pour décoder correctement le flux.

A ce stade, nous avons réussi à obtenir la première image du flux de la goPRO grâce à une fonctionnalité qui essaye de former les `!` automatiquement : `playbin`

```
gst-launch-1.0 playbin uri=http://10.5.5.9:8080/live/amba.m3u8
```

<code>playbin</code>	Autoplug and play media from an uri	<code>playback</code>	<code>gst-plug-ins-base</code>
----------------------	-------------------------------------	-----------------------	--------------------------------

N'ayant qu'une seule frame, nous avons essayé de former la suite de commande nous même. Après quelques recherches nous avons constaté qu'il existe une multitude de décodeurs utiles suivant le format du flux, ou de l'audio/video entrant.

D'après la multitude des decodeurs possibles, **decodebin** nous a paru le plus évident à utiliser.

decodebin	Autoplug and decode to raw media	playback	gst-plugins-base
------------------	----------------------------------	----------	------------------

En effet il est censé s'adapter au format de l'entrée

```
gst-launch-1.0 souphttpsrc
location=http://10.5.5.9:8080/live/amba.m3u8 ! decodebin !
autovideosink
```

Cependant le résultat était identique, une seule frame.

Nous avons continué nos recherches et vu qu'un demultiplexeur pour flux hls existait

hlsdemux	HTTP Live Streaming demuxer	hls	gst-plugins-bad
----------	-----------------------------	-----	-----------------

```
gst-launch-1.0 souphttpsrc
location=http://10.5.5.9:8080/live/amba.m3u8 ! hlsdemux !
decodebin ! autovideosink
```

Mais malheureusement le résultat restait le même. Une seule frame s'affichait et ne changeait pas.

playbin	Autoplug and play media from an uri	playback	gst-plugins-base
---------	-------------------------------------	----------	------------------

```
gst-launch-1.0 playbin uri=http://10.5.5.9:8080/live/amba.m3u8
```

Toutes les commandes utilisées sur ce flux ne modifiaient aucunement la sortie.

Après réflexion nous nous sommes penchés sur le flux en lui même. Et des recherches plus profondes nous ont appris que le flux de la goPro n'est pas un HLS standard. Il a été reporté comme "broken".

En effet son flux est découpé en segments très courts. Malheureusement gstreamer, comme VLC ne récupèrent seulement que le premier segment qui représente une fraction de seconde. (voir bibliographie)

Nous n'avons trouvé aucune documentation ni forum qui explique comment palier ce problème. Nous avons bien compris qu'utiliser n'importe quel logiciel allait arriver au même résultat et qu'il devait certainement manquer une librairie (puisque ça marche sur d'autres système d'exploitation). Udoobuntu est un système pas très connu et donc avec peu de documentation pour notre problème de flux HLS.

Nous voulions installer ubuntu et voir si le problème venait de là, mais la carte Udoos n'est pas compatible avec cet OS.

Sur la documentation de la carte, plusieurs OS sont disponibles notamment Archlinux mais son installation n'a non plus pas fonctionné.

Deuxième approche : Utiliser seulement la carte UDOO

Après des heures à chercher des solutions pour tout réaliser directement avec la carte UDOO et après s'être heurter continuellement à des murs, nous avons décidé de voir les choses autrement.

Puisque nous n'avons pas réussi à streamer directement avec la carte UDOO sur un service de streaming type Twitch/YouTube, nous avons cherché un moyen pour rendre le flux de la GoPro accessible tant bien que mal depuis Internet. Une solution nous a sauté aux yeux, qui est la redirection de port.

REDIRECTION DE PORT

Comme vu précédemment la GoPro diffuse un réseau Wi-Fi. En s'y connectant, nous pouvons voir le flux grâce à un logiciel (et surtout de bonnes librairies) comme VLC sous Windows/OSX/Ubuntu.



Notre but principal étant de diffuser sur Internet le flux de la GoPro, notre idée a été d'étendre cette accès à Internet.

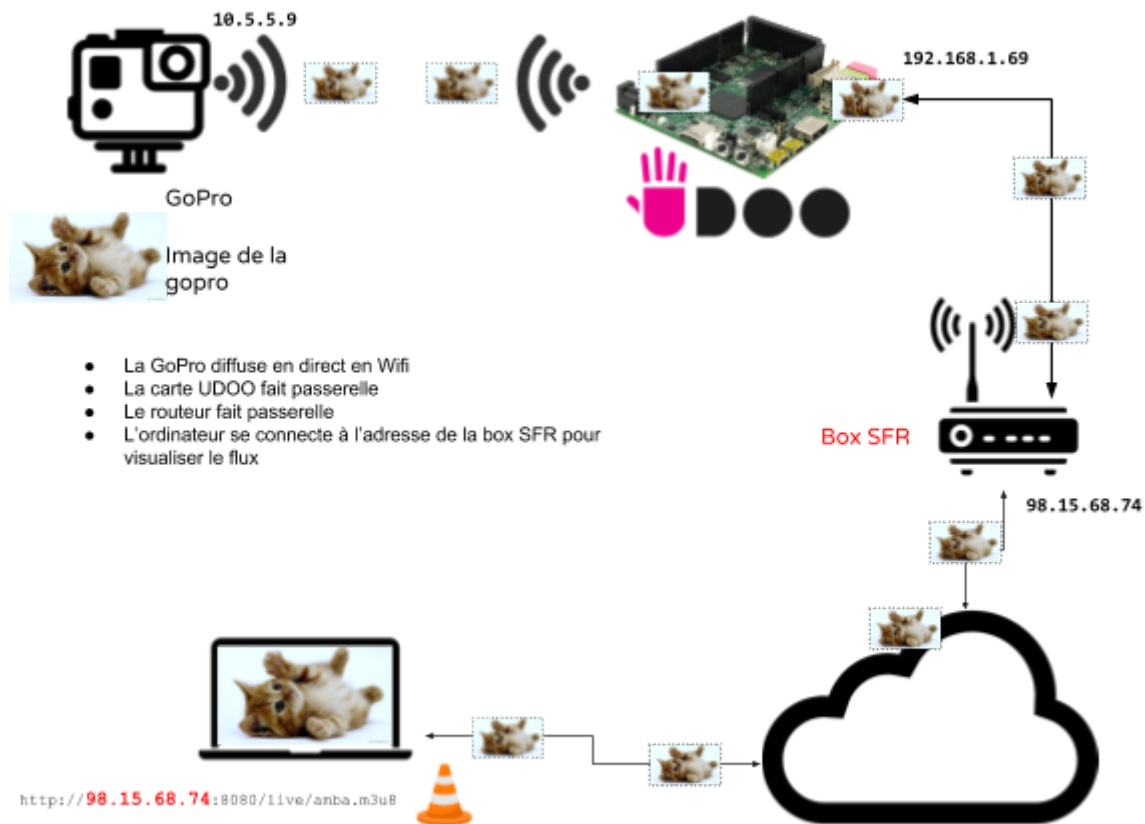
La redirection de port que nous avons donc mis en place sur notre projet permet de faire passerelle entre la carte UDOO et la GoPro. En faisant cela, le flux de la GoPro devenait

accessible depuis un ordinateur en utilisant l'adresse de la carte UD00 (et non de la GoPro).

(Pour faire une redirection de port sur un système linux, nous nous sommes aidé de ce site: <https://askubuntu.com/questions/751467/forward-port-to-local-ip-port>)



Pour que l'accès au flux de la gopro se fasse depuis Internet, nous avons également fait une redirection de port depuis une box fibre domestique (appelons la box SFR) et le flux de la GoPro devenait maintenant accessible pour n'importe qui connaissant l'adresse IP de la box SFR.



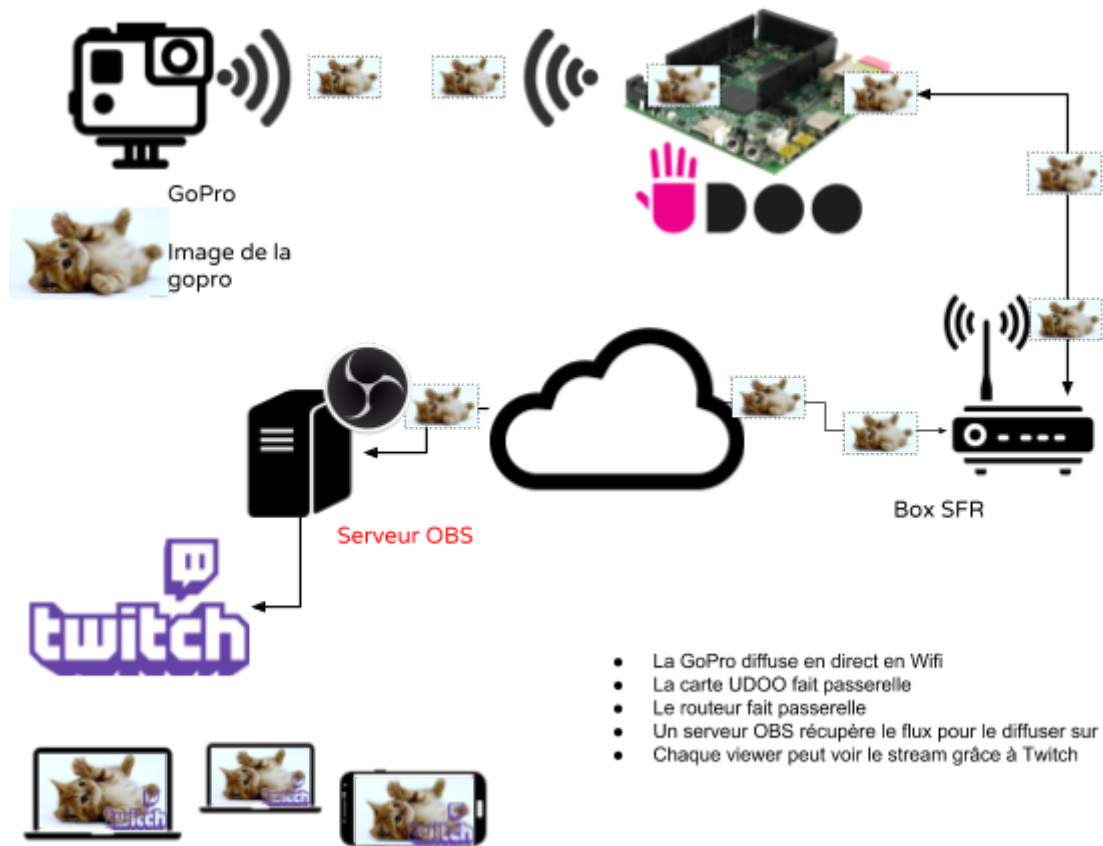
Une fois en place, n'importe quel client peut accéder au flux de la GoPro en lançant VLC, Média → Ouvrir un flux réseau, et en rentrant l'adresse : `http://@ip_box_sfr:port_defini/live/amba.m3u8`

Serveur OBS

Nous n'avons cependant pas abandonné l'idée de streamer sur Twitch ou YouTube car nous trouvons l'utilisation beaucoup plus simple pour le client et plus adaptable pour l'utilisation que l'on pourrait en faire.

Nous avons donc pensé à mettre en place un ordinateur "serveur" qui récupérerait ce flux et le streamerait sur une plateforme directement. Ce "serveur" doit cependant disposer des bons codecs / librairies...

L'ordinateur serveur, grâce à OBS, va donc récupérer le stream de la GoPro et le diffuser en direct sur Twitch/YouTube.



Récapitulatif

Première approche :

GoPro -> Udoos -> Twitch

Tentative infructueuse car le format de stream de la GoPro (HLS) n'est pas standard et qu'il manque des bibliothèques sur l'OS de Udoos.

Deuxième approche :

GoPro -> Udoos -> Internet

Solution qui marche en faisant une redirection de port sur Udoos et en connectant Udoos à Internet. Cependant cette solution n'est pas user-friendly car complexe à utiliser pour le client.



Troisième approche :

GoPro -> Udoos -> Serveur OBS -> Twitch

Solution qui est la plus facile à utiliser pour un client (un viewer du stream) car il peut visionner le stream directement sur un site web, Twitch.

Cependant, cette solution est coûteuse car nécessite un serveur OBS qui tourne en plus de la carte Udoos.

Bibliographie :

liste decoder gstreamer:

<https://gstreamer.freedesktop.org/documentation/plugins.html>

Bug flux goPro :

<http://gstreamer-devel.narkive.com/ckvctpJ8/streaming-gopro-with-gstreamer-1-0>

https://bugzilla.gnome.org/show_bug.cgi?id=740461

Archlinux Udo0 :

<https://archlinuxarm.org/platforms/armv7/freescale/udoo>

Faire une redirection de port sur Linux :

<https://askubuntu.com/questions/751467/forward-port-to-local-ip-port>

```
sudo sysctl net.ipv4.ip_forward=1
sudo iptables -t nat -A PREROUTING -p tcp --dport 8080 -j DNAT --to-destination
10.5.5.9:8080
sudo iptables -t nat -A POSTROUTING -j MASQUERADE
```

OBS (Open Broadcaster Software) :

OBS est un logiciel libre permettant de streamer et/ou d'enregistrer des vidéos en direct ou non. De nombreuses fonctionnalités sont présentes et je vous invite à visiter le site pour plus d'information.

<https://obsproject.com/fr>

HLS :

HLS (ou HTTP Live Streaming) est un protocole de streaming basé sur le protocole HTTP. (Plus d'info : https://fr.wikipedia.org/wiki/HTTP_Live_Streaming)

Twitch :

Plateforme de streaming en direct destiné principalement aux les jeux vidéos.