

Universidad Técnica Federico Santa
María
Campus Casa Central
Valparaíso, Chile



“TRANSMISIÓN DE CÁMARA WEB A TELEVISIÓN DIGITAL DE FORMA REMOTA”

Televisión digital bajo norma ISDB-Tb

DESARROLLADO POR:

Sebastián Castillo Araya

CORREO :

sebastian.castilloar@sansano.usm.cl

FECHA:

03 Enero 2023

Índice

1	Resumen	2
2	Problema	2
3	Solución	3
3.1	Diseño de solución	3
3.2	Implementación	6
3.2.1	Configuración de playout	6
3.2.2	Configuración programas	10
3.3	Escenario de prueba y resultados	10
4	Conclusión	13
5	Código fuente	13
6	Referencias	13

Índice de figuras

1	Relación entre programas.	3
2	Comunicación en pasos para lograr transmisión.	4
3	Diagrama Cliente Webcam	4
4	Diagrama Servidor redireccionador	5
5	Diagrama Cliente Playout	5
6	Interfaz log in	6
7	Nuevo elemento multimedia	6
8	Configuración contenido multimedia	7
9	Nuevo servicio	7
10	Configuración nuevo servicio	8
11	Asociación a nuevo servicio	8
12	Nuevo transport stream	8
13	Configuración general transport stream	9
14	Configuración norma ISDB-T	9
15	Configuración Capa B con su servicio	10
16	Transport stream con capa A y B configurada	10
17	Paso 1: Ejecución ServidorRedireccionador	11
18	Paso 2.1: Ejecución ClienteWebcam con cam integrada	11
19	Paso 2.2: Ejecución ClienteWebcam con cam usb	11
20	Paso 3: Ejecución ClientePlayout	12
21	Paso 4: Lista de Webcams en ServidorRedireccionador	12
22	Paso 5.1: Transmisión webcam integrada en pc	12
23	Paso 5.2: Cambio de cámara	12
24	Paso 5.3: Transmisión webcam externa usb	13

1. Resumen

Se plantea transmitir señal de video de webcam desde cualquier punto a servidor playout, el cual se encarga de retransmitir el una señal de TV bajo norma ISDB-Tb para posteriormente por parte del usuario poder visualizar el contenido en televisión digital. Para esto se mejora una solución existente, la cual permite únicamente transmitir webcam a servidor playout desde un lugar fijo, y además se agrega la capacidad de recibir más de una cámara web seleccionable. Para dicha mejora se plantea una solución en la que se utiliza un servidor intermediario, quien recibe el video de webcam desde cualquier punto y lo retransmite a un dispositivo capaz de comunicarse directamente con el playout, esto para atravesar posibles NATs. Además se agrega que servidor dé la opción al usuario de escoger qué flujo de video retransmitir. La solución implica a tres programas desarrollados en lenguaje C, donde los contenidos presentes en la solución abarcan desde uso de creación de procesos, hilos, sincronización entre ellos, hasta uso de clientes y servidores tanto UDP como TCP.

2. Problema

El **objetivo** por alcanzar es transmitir el video emitido por una webcam **desde cualquier punto** y poder visualizarlo a través de la televisión digital. **¿Cómo se puede transmitir estando fuera del laboratorio?** Para lograrlo se transmite el video de la webcam al playout, este modula el video y lo emite por aire, la señal emitida es recibida por el decodificador el cual recupera el video demodulado para que finalmente pueda ser observado en la televisión. Esta solución ya existe y es usada como base en este proyecto, pero **tiene una limitación**, pues solo funciona si es que el computador que está emitiendo el video de la webcam tiene comunicación directa con el playout, lo cual **sólo se cumple si es que ambos dispositivos están bajo la misma subred**. Ahora toca responder la pregunta **¿Por qué sólo se cumple si ambos están bajo la misma red?** Porque como muchos dispositivos presentes en Internet, el **playout se encuentra escondido detrás de una NAT**, lo que provoca que no podamos contactar de forma directa al dispositivo si es que se le desea ubicar desde fuera de la subred. Por lo tanto, la **problemática a solucionar** se resume en buscar **cómo puede llegar el video de la webcam desde fuera de la red al playout**.

3. Solución

3.1. Diseño de solución

La solución está compuesta por tres programas:

- Un **cliente webcam** que produce un video por webcam y lo envía a servidor redireccionador.
- Un **servidor redireccionador** que permite seleccionar y redireccionar los paquetes de video hacia el cliente payout según la webcam que se haya escogido. Además mantiene registro de los posibles clientes webcam disponibles.
- Un **cliente payout** que se encuentra en la misma red, eventualmente “nateada”, que el dispositivo payout y recibe video desde servidor redireccionador y posteriormente reenvía a payout que permite la reproducción en la TV.

La relación entre los programas se presenta en la figura 1.

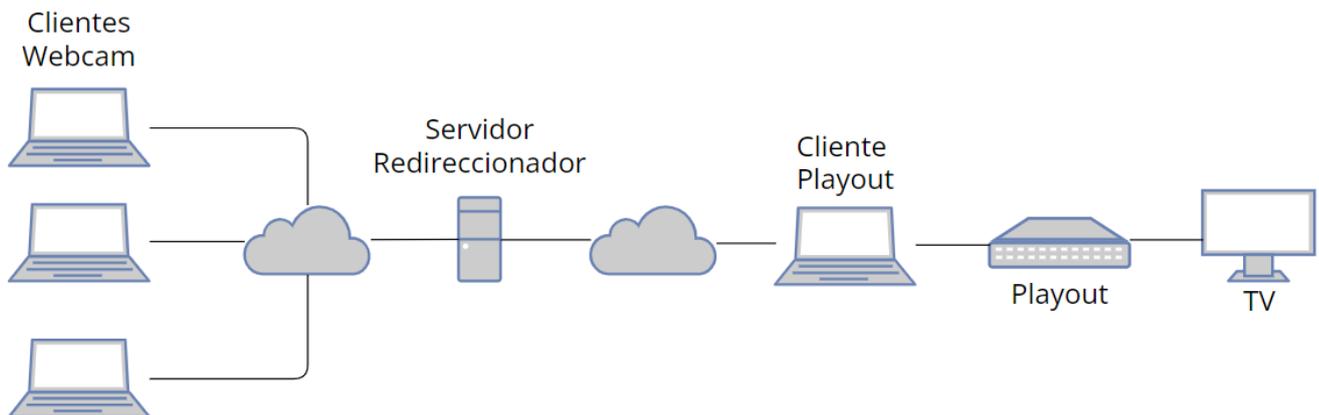


Figura 1: Relación entre programas.

Toda la comunicación entre ambos clientes ocurre con el servidor como intermediario, tal como indica la figura 2.

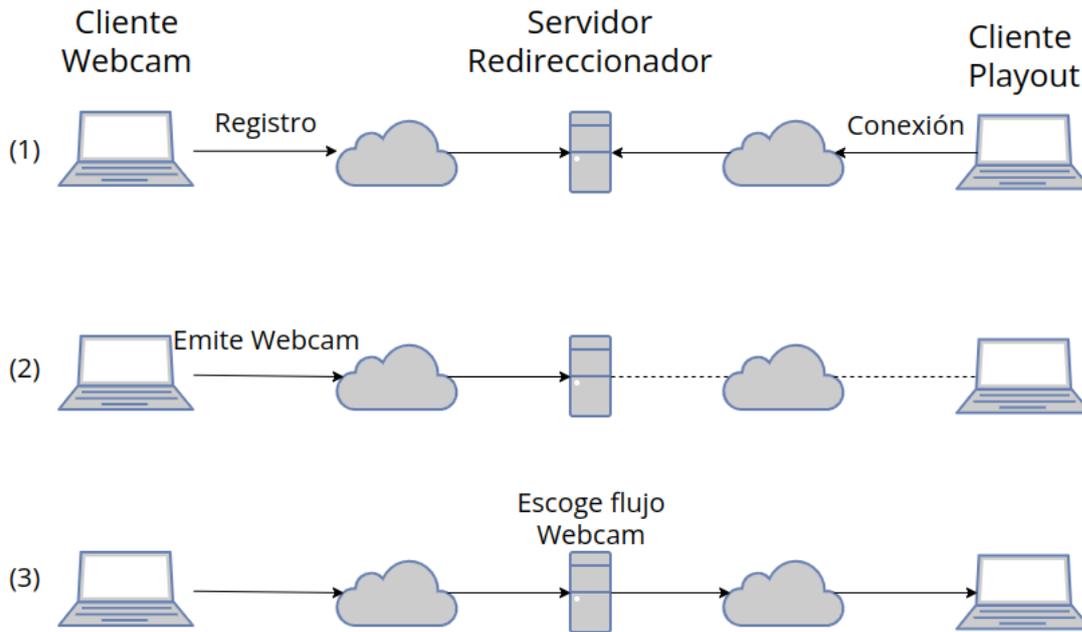


Figura 2: Comunicación en pasos para lograr transmisión.

La estructura general del **Cliente Webcam**, graficada por la figura 3, consiste en un sólo hilo que se conecta con el servidor, al cual le debe indicar su nombre y puerto por el cuál enviará video de webcam. Una vez realizada la conexión el programa ejecuta un script bash con el comando ffmpeg para emitir el video mediante UDP al servidor.

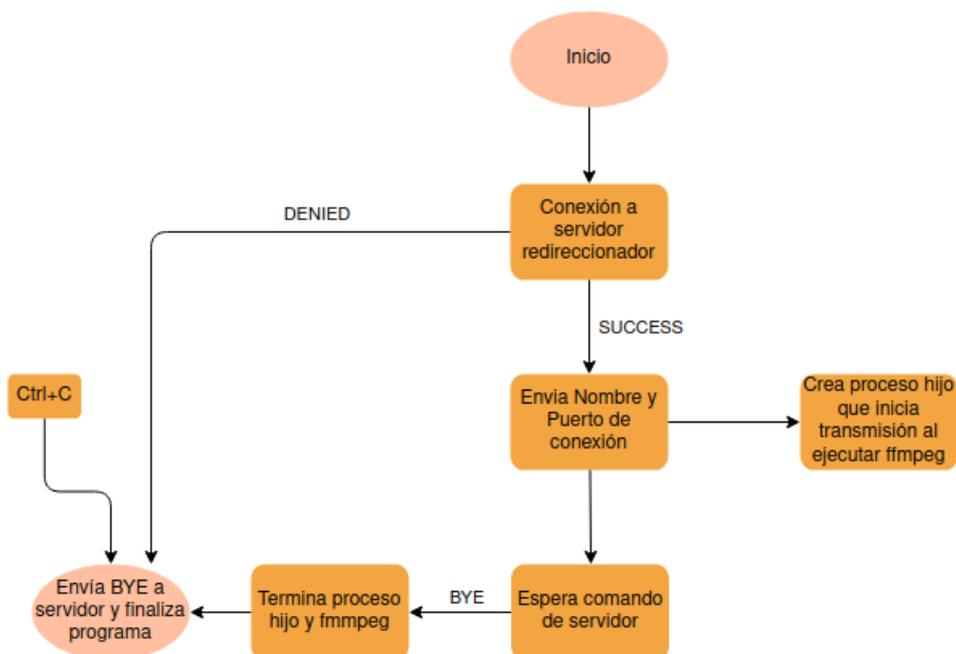


Figura 3: Diagrama Cliente Webcam

La estructura general del **Servidor Redireccionador**, descrito gráficamente por la figura 4, consta

de dos hilos para mantener comunicación con ambos tipos de clientes y un arreglo global con los clientes webcam conectados. Para los clientes webcam es necesario recibirlos y revisar un espacio en el arreglo global para agregarlos o denegarles la conexión en caso de ser necesario, además de estar atento a la desconexión de alguno de los activos para removerlos. Cuando cliente playout se conecta a servidor, este último crea un hilo en el cual comienza inmediatamente a redireccionar paquetes al cliente playout. Mientras esto ocurre, dependiendo las webcams registradas, se muestran las opciones para que se escoja que webcam retransmitir.

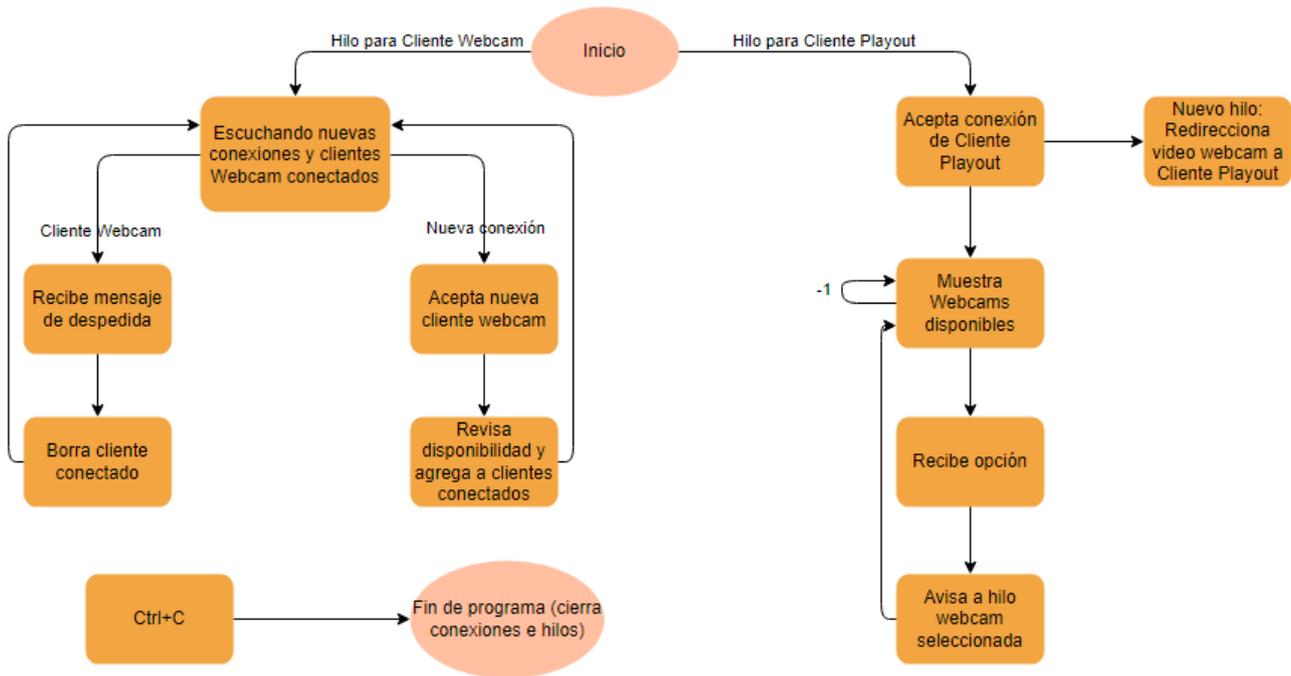


Figura 4: Diagrama Servidor redireccionador

La estructura de **Cliente Playout**, dada por figura 5, consta de un hilo principal el cual realiza la conexión con servidor redireccionador y luego crea otro hilo en el cual se reciben los paquetes de video de servidor y a la vez son transmitidos finalmente a servidor playout.

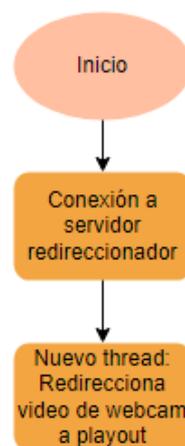


Figura 5: Diagrama Cliente Playout

3.2. Implementación

Para la implementación es necesaria la preparación del servidor playout y la de los programas diseñados.

3.2.1. Configuración de playout

Mediante el navegador de preferencia se accede a la interfaz de configuración que proporciona el playout (<http://10.2.51.11>), con los siguientes datos de identificación:

usuario: eitv y contraseña: playout (figura 6).



Figura 6: Interfaz log in

Ya ingresado a la interfaz, se debe agregar un nuevo elemento multimedia. Para esto se hace click en Add New TS dentro de la sección A/V (figura 7).

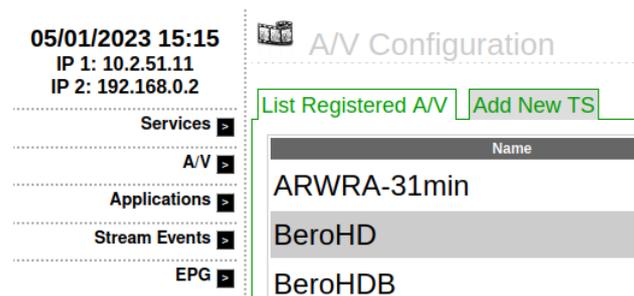


Figura 7: Nuevo elemento multimedia

La configuración para el nuevo elemento debe seguir lo dado por la figura 8. La IP debe corresponder a la del playout (a la fecha en que se realiza este informe corresponde a 10.2.51.11) y para el puerto se tiene más libertad, únicamente se debe registrar pues se usará como destino en el terminal de ClientePlayout. Respecto a los demás datos a rellenar deben corresponder a los que se utilizarán en el script bash.

188 or 204 bytes TS Stream (CBR)

Name:

Source: File IP

File:
(Maximum Size: 1000 MB)
FTP:

IP Interface: Address:
Port:
Protocol: UDP RTP

PID Video:
Video Stream Type:

PID Audio:
Audio Stream Type:

PID PCR:

TOT time decode: Synchronize system time with TOT from this input

Bitrate Video (kbps):
Bitrate Audio (kbps):
Bitrate TS (bps):

Figura 8: Configuración contenido multimedia

Si siguiendo con la configuración, el archivo creado se procede a adjuntar a un Service. Para esto se va al apartado de Services y click en Add New Service (figura 9). Ya creado se configura (figura 10) y luego se asocia el archivo A/V creado (figura 11) al servicio.

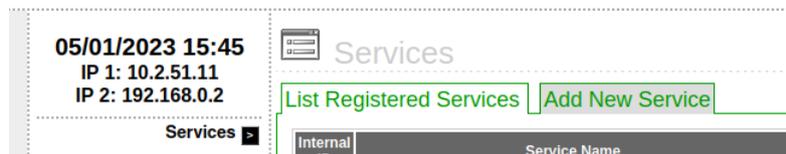


Figura 9: Nuevo servicio

General

Name	<input type="text" value="Service_Webcam_SC"/>
Provider	<input type="text" value="FFMPEG"/>
Service Type	<input type="text" value="[0x01] Serviço de televisão digital"/>
EIT Schedule Flag	<input type="checkbox"/>
EIT Present Following Flag	<input type="checkbox"/>
H-EIT Flag	<input type="checkbox"/>
M-EIT Flag	<input type="checkbox"/>
L-EIT Flag	<input type="checkbox"/>

Custom

The following parameters only need to be configured if custom values are needed. The value "0" (zero) can be left for defaults.

Program PIDs

Primary Video PID	<input type="text" value="180"/> (decimal)
Video Descriptors	<input type="text"/> (tag:length:byte:tag:length:byte ...)
Primary Audio PID	<input type="text" value="0"/> (decimal)
Audio Descriptors	<input type="text"/> (tag:length:byte:tag:length:byte ...)
PCR PID	<input type="text" value="180"/> (decimal)

Figura 10: Configuración nuevo servicio

05/01/2023 15:45
IP 1: 10.2.51.11
IP 2: 192.168.0.2

A/V Playlist for Service 'Service_Webcam_SC'

Name	A
Associate Stream	
Stream	<input type="text" value="AV_Webcam_SC"/>
<input type="button" value="Associate"/>	
<input type="button" value="Back"/>	

Figura 11: Asociación a nuevo servicio

Si siguiendo con la configuración de playout ahora se procede a crear un transport stream. Para eso se va al apartado de transport stream y click en Add New TS (figura 12)

05/01/2023 15:34
IP 1: 10.2.51.11
IP 2: 192.168.0.2

Transport Stream

• Please, fill in the Name field.

Name

Figura 12: Nuevo transport stream

Se procede a rellenar los parámetros de acuerdo a nuestro objetivo, indicando interfaz de salida, frecuencia de transmisión, norma a utilizar, modulación entre otros más (Figura 13 y 14).

05/01/2023 15:51
IP 1: 10.2.51.11
IP 2: 192.168.0.2

Transmission Configuration

Transport Stream: TS_Webcam_SC

Configuration | DVB-C | DVB-T | DVB-S | ISDB-T

Services ▾
A/V ▾
Applications ▾
Stream Events ▾
EPG ▾
EWBS ▾
Transport Stream ▾
Log ▾
Configuration ▾

Transmission Mode: Terrestrial Delivery System (ISDB-T) ▾
Out Interface: Modulator ▾
Network ID: 1
Network Name: TS_Webcam_SC
Transport Stream ID: 1
Original Network ID: 1

Broadcast Information Table (BIT)

Affiliation ID: 0
1st Descriptors Loop: (tag:length:byte:tag:length:byte ...)

Save

Figura 13: Configuración general transport stream

Transport Stream: TS_Webcam_SC

Configuration | DVB-C | DVB-T | DVB-S | ISDB-T

ISDB-T Transmission data successfully updated!

Broadcast type: 13-segment TV Broadcast ▾
Frequency: Channel 23 ▾ 527142857 Hz
Area Code: 17
Virtual Channel (Remote Control Key ID): 17
TS Name: EITV Playback
Transmission Mode: Mode 3: 8k ▾
Guard Interval: 1/16 ▾
Partial Reception: Enabled partial reception in the Layer A (implies in Layer A with only 1 segment)
Emergency: Enabled emergency flag

Hierarchical Layers	Layer A	Layer B	Layer C
Segments	1 ▾	12 ▾	0 ▾
Modulation	QPSK ▾	16-QAM ▾	16-QAM ▾
Convolutional Rate	2/3 ▾	5/6 ▾	3/4 ▾
Time Interleaving Length	4 ▾	4 ▾	4 ▾
Bitrate (bps)	440560	13216800	0
Total Bitrate (bps)	13657360		

Save

Figura 14: Configuración norma ISDB-T

AVISO: Se configuró el streaming para utilizar la capa B de la norma ISDB-Tb, para que pueda ser reproducido en TV (a través de set top-box), pero aún así es necesario adjuntar un servicio en la Capa A. En la figura 15 se adjunta servicio a capa B, el procedimiento para la capa A es igual. Se aprovechó un servicio ya creado previamente para adjuntar a transport stream (figura 16).

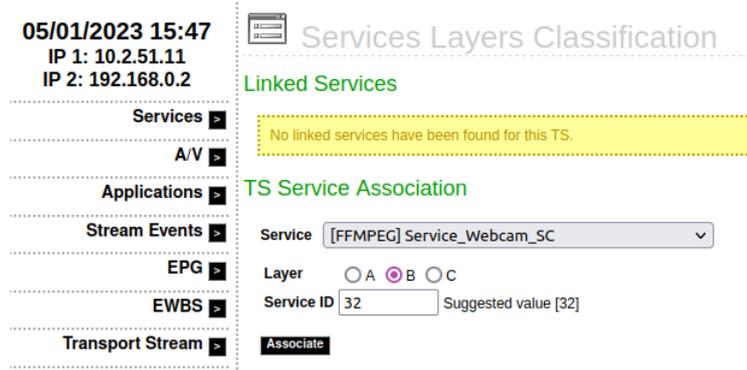


Figura 15: Configuración Capa B con su servicio



Figura 16: Transport stream con capa A y B configurada

Con esto se da por finalizada la configuración de playout, tan solo queda dar play a transport stream y seguir con los siguientes pasos.

3.2.2. Configuración programas

Los tres programas están diseñados en lenguaje C, por lo que se requiere tener un compilador para preparar los programas. Un compilador recomendado es gcc.

Para la ejecución correcta de script bash se requiere un computador con **Linux y con webcam**. Para conocer el nombre de dispositivo de video ejecutar comando `v4l2-ctl--list-devices`, del paquete `v4l-utils`. El script permite dos posibles nombres al momento de iniciar ejecución, `"/dev/video0"` y `"/dev/video2"`, los cuales corresponden a cámara integrada y webcam usb al momento de realizar el script. Si el nombre de los dispositivos al momento de ejecutar comando difiere con los nombrados, cambiar código de script por los correspondientes para dicho equipo.

Para la transmisión de video también es necesario programa "ffmpeg", ya que este se encarga de extraer video de webcam y enviarlo en paquetes UDP a servidor.

3.3. Escenario de prueba y resultados

Se realiza prueba para observar resultado de solución diseñada. Esta fue realizada en el Laboratorio Leonard Kleinrock de la universidad, utilizando la red del laboratorio y la proporcionada a

los estudiantes. El programa ClienteWebcam es ejecutado en computador conectado a red externa del playout, el programa ServidorRedireccionador es ejecutado en servidor aragorn y programa ClientePlayout es ejecutado en computador conectado a la misma red que servidor playout.

Para un funcionamiento correcto, se ejecuta inicialmente programa ServidorRedireccionador en aragorn (Figura 17). Una vez ejecutado, el programa se pondrá en espera a la escucha de clientes.

```
sebastian.castilloar@aragorn:~/ProyectoEL0330$ ./ServerRelay.o
-_Recibiendo registro de Webcams por puerto: 47203
-_Esperando conexiones con Webcams
-_Esperando conexion con cliente Playout...
Hilo UDP activado
```

Figura 17: Paso 1: Ejecución ServidorRedireccionador

El siguiente paso es ejecutar ClientesWebcam (figura 18 y figura 19). Para la prueba se ejecutan dos clientes, uno que emite webcam integrada a puerto 47200 y otro cliente que emite webcam externa usb a puerto 47201.

```
sebita@sebita-HP-Notebook:~/Workspace/practica$ ./ClienteWebcam.o aragorn.elo.utfsm.cl 47203 0
TCP Server host NAME: aragorn.elo.utfsm.cl
TCP Server INET ADDRESS: 200.1.17.195:47203
Connecting...
Connection established
Porfavor ingresa tu nombre...
Cam_PC-HP
Nombre enviado: Cam_PC-HP
Porfavor ingresa puerto al que envias video...
47200
Child process created
Listening to server...

Starting video streaming via ffmpeg...
```

Figura 18: Paso 2.1: Ejecución ClienteWebcam con cam integrada

```
sebita@sebita-HP-Notebook:~/Workspace/practica$ ./ClienteWebcam.o aragorn.elo.utfsm.cl 47203 1
TCP Server host NAME: aragorn.elo.utfsm.cl
TCP Server INET ADDRESS: 200.1.17.195:47203
Connecting...
Connection established
Porfavor ingresa tu nombre...
Cam_USB
Nombre enviado: Cam_USB
Porfavor ingresa puerto al que envias video...
47201
Child process created
Listening to server...

Starting video streaming via ffmpeg...
```

Figura 19: Paso 2.2: Ejecución ClienteWebcam con cam usb

Ya conectados los ClientesWebcam a ServidorRedireccionador, se ejecuta ClientePlayout (Figura 20).

```
sebita@sebita-HP-Notebook:~/Workspace/practica$ ./ClientePlayout.o  
aragorn.elo.utfsm.cl  
- _Conectando con servidor...  
- _Conexion establecida  
  
Insert BYE : to close the client  
UDP redirect START...
```

Figura 20: Paso 3: Ejecución ClientePlayout

Si todo va correctamente, en vista de ServidorRedireccionador debería verse lo de la figura 21

```
- _Aceptando nueva conexion Webcam...  
- _Nombre nuevo cliente Webcam: Cam_PC-HP  
- _Aceptando nueva conexion Webcam...  
- _Nombre nuevo cliente Webcam: Cam_USB  
- _Cliente Playout aceptado satisfactoriamente  
- _Escoja entre los siguientes flujos de video:  
- _Opcion 1 ---> Cam_PC-HP, puerto 47200  
- _Opcion 2 ---> Cam_USB, puerto 47201  
-_-1 ---> Actualizar lista
```

Figura 21: Paso 4: Lista de Webcams en ServidorRedireccionador

Para probar el funcionamiento del programa, se escoge el cliente Cam_PC-HP (figura 22).



Figura 22: Paso 5.1: Transmisión webcam integrada en pc

A continuación se cambia la cámara (figura 23) y se escoge cliente Cam_USB (figura 24).



Figura 23: Paso 5.2: Cambio de cámara



Figura 24: Paso 5.3: Transmisión webcam externa usb

4. Conclusión

Se logra diseñar una solución a la problemática, pudiendo transmitir video de webcam desde fuera de la red de playout, mejorando así el estado inicial de la solución base. Por otro lado, la manipulación de más de una cámara también se logra de manera satisfactoria.

5. Código fuente

[Repositorio GitHub](#)

6. Referencias

- “Manual Transmisión ISDBT V2019” de Gonzalo Callender y Roberto Muñoz. Con revisión de Eric Borzone.
- “Manual Transmisión Cámara USB ISDBT V2019” de Jose Luis Rojel. Con revisión de Eric Borzone.
- “Transmisión de cámara web a televisión digital - Proyecto ELO323” de Sebastián Castillo, Sebastián Madariaga y Darael Badilla.
- “Solución Cliente-Servidor para transmisión de Webcam por TVD - Proyecto ELO330” de Sebastián Castillo y Nicolás Miranda.