Bachelor's Degree in Techniques for Software Development

Networks and Internet Applications

Practical exercise 3: Experiment with streaming

The primary objective of this practical exercise is to provide a practical understanding of streaming video content over the Internet, using the RTMP (*Real-Time Messaging Protocol*) protocol as the basis.

RTMP is a real-time messaging protocol that is present in many of the web video *streaming*, both real-time and stored. RTMP uses TCP as a transport protocol and is used with many different types of media, including live television, video, and Internet telephony services.

In this practical exercise, we present the challenge of configuring an RTMP server to stream and receive stored or real-time video, using different tools and technologies. You will also learn how to stream HLS and DASH streams compatible with the most modern platforms.

You can find information and working details of the RTMP protocol that can help you to carry out this practical exercise in these links:

https://restream.io/blog/rtmp-streaming/
https://teyuto.com/blog/what-is-rtmp-is-rtmp-dead-advantages-disadvantages-alternatives

DASH is explained in section 2.6 of Kurose-Ross, 8th edition (page 173).

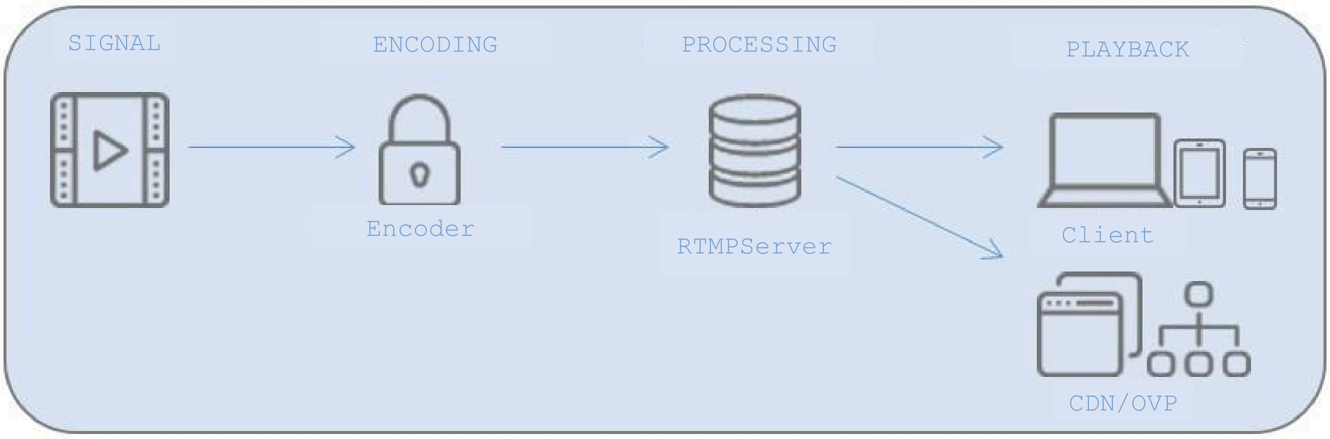
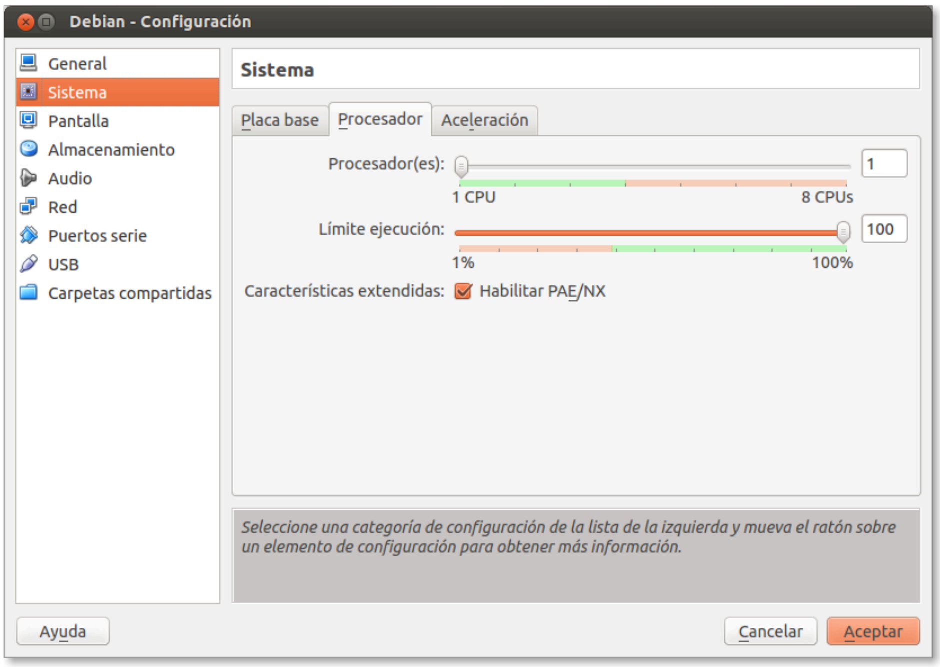


Figure 1. Streaming with RTMP

The figure above shows you the basic operation of the protocol. First, the signal to be sent is encoded using streaming encoders that then send the encoded signal to the server using the RTMP protocol, from which it is distributed directly to the user, to CDNs or online video platforms.

This practical exercise will be built on a Linux[1] server (Debian or Ubuntu), where you will install the necessary software. To do this, you can use a virtual machine (e.g., Virtualbox) or an installation on a real machine. At the end of the document you can find some links to different tutorials that explain how to install VirtualBox, its extensions, and how to install Linux in Virtualbox. For the correct operation of the virtual machines, you may have to enable the PAE/NX in the VirtualBox configuration, as shown in the following screen:



Características extendidas: ✓ Habilitar PAE/NX

Seleccione una categoría de configuración de la lista de la izquierda y mueva el ratón sobre un elemento de configuración para obtener más información.

VirtualBox connects you by default to the Internet through NAT using the Host-only adapter, but you will be interested in working in Bridged adapter mode to be able to connect to the server from any machine in your network. Configure the network in Bridged adapter mode and check that you have Internet access.

Qualification

This practical exercise consists of four parts, which must be performed in sequential order (Part 1 first, then Part 2...). The qualification will be based on the parts delivered:

Part 1: Server Installation and Configuration	*Maximum qualification C-*
Part 2: Stored Streaming	*Maximum qualification C+*
Part 3: Live streaming	*Maximum qualification B*
Part 4: Support for HLS and DASH	*Maximum qualification A*

To obtain the indicated levels of qualification, it is necessary to carry out all the exercises and sections of the indicated parts. If there is any exercise or section not carried out or incomplete, it will imply not obtaining the corresponding rating.

[1] This practical exercise has been tested on a Debian 11.6 server and on Ubuntu

Important: For the requested screenshots, set the operating system prompt to show the day you are on and your username.

First part (maximum qualification C-): Server Installation and Configuration

In this first part, you will install a Nginx open source web server, where you will configure RTMP to be able to stream stored or real-time video.

Steps to follow: installation

Before you begin installing the Nginx web server, upgrade your system:

```
apt update
apt upgrade
```

Installation of the Nginx web server is standard for any Ubuntu/Debian system and as simple as typing the following command into a root console:

```
apt install nginx -y
```

Now you will start Nginx and enable the service to start at system startup (if it gives problems at startup, you can check that you do not have another web server, e.g., apache2, running on port 80):

```
service nginx start
systemctl enable nginx
```

Nginx includes a module to transmit RTMP from a URL with minimal configuration. The Nginx RTMP module is not automatically included with Nginx, but you can install it as an additional package:

```
apt install libnginx-mod-rtmp -y
```

To configure the RTMP server, you need to edit the Nginx master file, using the following command, for example:

```
nano /etc/nginx/nginx.conf
```

At the end of the file, add the following guidelines:

```
rtmp {
        server {
                listen 1935;
                chunk_size 4096;
                allow publish all;

                application live {
                        live on;
                        record off;
                }
        }
}
```

Check that the configuration is correct, using the command:

```
nginx -t
```

Restart the service:

```
service nginx restart
```

If you have a firewall installed, enable the RTMP port:

```
sudo ufw allow 1935/tcp
```

Exercise 1

a) Explain all the RTMP configuration parameters you added in the previous `nginx.conf` configuration file.

b) Add three more RTMP-related configuration parameters and explain why you think they are interesting to take them into account. Show a screenshot of the `nginx.conf` file where they appear.

c) To verify that nginx is working correctly, type the following order in the console and display the result using a screenshot:

```
service nginx status
```

d) On which IP address is the RTMP server running? Check it with the `ifconfig` command or any other similar tool, showing with a screenshot.

e) On which port is the RTMP server running? In what state are you? Check it with the `netstat` command or any other similar tool, displaying it using a screenshot.

f) In any web browser, enter the following URL `http://<your_server_ip>.` Show by screenshot that the web server is properly configured.

Second Part (maximum qualification C+): Streaming Stored Video

In this second part of the practical exercise, you will send an example video to the RTMP server for further network broadcast. If you do not have a video file on your computer, you can download one using the `yt-dlp` command, a command line tool for downloading videos from platforms like YouTube. To use it, you will also need to install the `snap` package manager on your server:

```
apt install snapd
snap install core
snap install yt-dlp
```

Log out, log back in and download some video (a *short* is enough) with the `yt-dlp` command:

```
yt-dlp <video_url> -f mp4
```

Check that you have the downloaded video in your current directory. If you get access denied errors, make sure you have permissions to write to the folder or change them using the `chmod` command.

Steps to follow: Sending stored video to the server

There are several ways to send the video to the RTMP server. One option is to use the `ffmpeg` command, a very popular command line tool. Install it and send the video to the RTMP server by running the following command:

```
apt install ffmpeg

ffmpeg -re -i "<downloaded file name>" -c:v copy -c:a aac -ar 44100 -ac 1 -f
flv -flvflags no_duration_filesize rtmp://localhost/live/stream
```

If the connection is refused, try restarting the service:

```
service nginx restart
```

Exercise 2

a) Explain the meaning of all the options in the previous `ffmpeg` command.

b) Explain why the `live` and `stream` words appear in the URL we are sending the video to.

c) How many frames per second is the video stream?

d) Connect with VLC or any other player to the URL `rtmp://<your_server_ip>/live/stream` while the video is uploading with the `ffmpeg` streaming program. Paste a screenshot showing that the video is playing.

e) The broadcast will end when the video stops. What would need to be done to loop it?

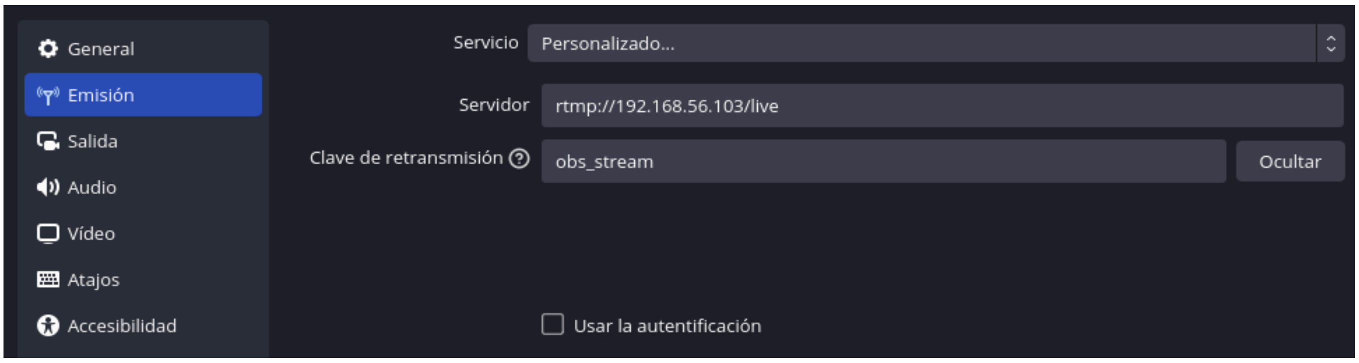
f) What is the average bitrate to which it has been transmitted?

Third Part (Maximum qualification B)
Live video streaming with OBS

In the previous section, you transmitted a static video source from the command line. In this part, you are going to stream real-time video from your PC via the RTMP server. To do this, we will use Open Broadcaster Software (OBS), which is popular for live streaming. You can download and install it for your operating system at `https://obsproject.com/es/download`.

Steps to follow: Creating Live Video

One of the options for creating real-time video is streaming your own desktop. To do this, first configure your streaming service credentials in the OBS `Settings` menu. Navigate to the `Stream->Custom` option and enter the following options, setting your server IP in the Server field, and your UOC user as the Stream key.



Servicio: Personalizado...
Servidor: rtmp://192.168.56.103/live
Clave de retransmisión: obs_stream
Usar la autentificación

The transmission will be available at:

```
rtmp://<your_server_ip>/live/<uoc_user>
```

If it gives you the following error "`failed to initialize video. Your GPU may not be supported, or your graphics drivers may need to be updated.`", you can fix it by putting `LIBGL_ALWAYS_SOFTWARE=1 obs` in the command line.

You do not need to enable authentication, but you do need to configure the video streaming IP address, changing the `allow publish all` line in the Nginx `/etc/nginx/nginx.conf` configuration file:

```
...
allow publish 127.0.0.1;
allow publish <your_local_ip_address>;
deny publish all;
...
```

Check that the configuration test is successful and restart Nginx for the changes to apply:

```
nginx -t
service nginx restart
```

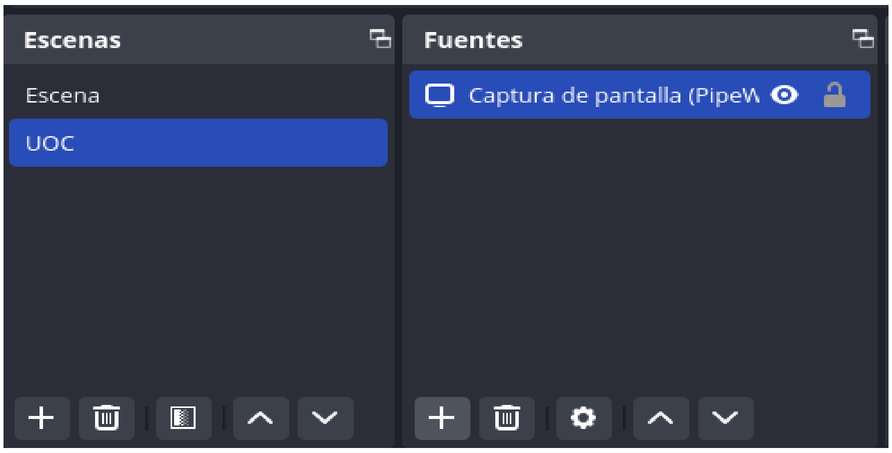
You are now ready to create real-time video to stream over the RTMP server. To do this, the OBS program relies on scenes and fonts to create our streaming design.

With the OBS program open, attach to the windows at the bottom. Scenes are the different compositions of our video. You can use the controls or right-click the Scenes pane to add, delete, or reorder. Similarly, by right-clicking on an existing scene you can rename, duplicate, copy, or display it independently.

Each scene can have different sources that will make up our screen. A font is anything we want to view or hear in our scene: video cameras, app windows, games, screens, audio devices, images, text, web pages... Once the scene is created, you can use the controls or right-click the Sources pane to manage them.

We can create a scene with a single source, for example our webcam; another scene with a game, a slide deck and an explanatory audio recording. OBS uses layered visualization, so you can size the different elements as you see fit.

Let's put these concepts into practice. Add a simple scene in OBS and a *screenshot* type input source:



At the bottom right, tap `Start Streaming` to send your signal to the RTMP server.

Exercise 3

a) To verify that streaming is being streamed in real time, connect to VLC or similar to the URL `rtmp://<your_server_ip>/live/<user_uoc>`. Show it using a screenshot.

b) Perform user actions on the screen where you are running OBS and watch in the streaming you are viewing with VLC or similar. What are you relaying?

c) Think of a video to play that contains at least three scenes with multiple sources. Explain your idea and show through screenshots the OBS settings and streaming playback with VLC.

d) Open the Wireshark program as root before starting the live stream. Paste a screenshot showing the RTMP traffic generated. You can filter by RTMP protocol to see only this interaction. Explain reasonably, packet by packet, the first phase of connection establishment (handshake) with parameter negotiation (up to the audio/video data packets).

e) Stop streaming on VLC and OBS. What packages are generated in Wireshark? Briefly explain the interaction you observe, illustrating it with some screenshot.

The next step is to enable the RTMP statistics page. To do this, you are going to create a virtual host in Nginx:

```
nano /etc/nginx/sites-available/rtmp
```

And add the following content:

```
server {
    listen 8080;
    server_name localhost;

    # rtmp stat
    location /stat {
        rtmp_stat all;
        rtmp_stat_stylesheet stat.xsl;
    }
    location /stat.xsl {
        root /var/www/html/rtmp;
    }

    # rtmp control
    location /control {
        rtmp_control all;
    }
}
```

Create the `/var/www/html/rtmp` directory and copy there the `stat.xsl` file which you can find in GitHub:

```
mkdir /var/www/html/rtmp
cd /var/www/html/rtmp
wget https://raw.githubusercontent.com/arut/nginx-rtmp-module/master/stat.xsl
```

Finally, if you have a firewall installed, you must open another port on your firewall to access the statistics page. In the Nginx configuration file you can see that the port where it is running is 8080, so you have to add an access rule. However, in order for no one else to be able to access the statistics page you should open it only for your own IP address:

```
ufw allow from <your_PC_ip> to any port http-alt
```

Then activate this new configuration:

```
ln -s /etc/nginx/sites-available/rtmp /etc/nginx/sites-enabled/rtmp
```

Check that the configuration test is satisfactory and restart Nginx:

```
nginx –t
service nginx restart
```

Exercise 4

Display a screenshot with your browser's access to the RTMP statistics page: `http://<server_ip>:8080/stat` and update it while streaming video stored with VLC connected. See how streaming statistics change. If not, restart VLC to view them. Answer the following questions:

a) In relation to the audio transmission, indicate:
   1. What codec is being used?
   2. What is the bitrate?
   3. What is the frequency?
   4. And the channel?

b) In relation to video streaming, please indicate:
   1. What codec is being used?
   2. What is the bitrate?
   3. At what resolution is it being transmitted?
   4. At how many frames per second?

Fourth Part (maximum qualification A): Support for HLS and DASH

In this part you will learn how to send a stream over HTTP. There are two protocols you can use to do this: Apple HLS and MPEG DASH. The Nginx-RTMP module supports both standards.

Steps to follow: Streaming Stored Video

To add HLS and DASH support, modify the rtmp block in the `/etc/nginx/nginx.conf` configuration file and add the following content within `application live` and below `record off:`

```
...
hls on;
hls_path /var/www/html/stream/hls;
hls_fragment 3;
hls_playlist_length 60;

dash on;
dash_path /var/www/html/stream/dash;
```

Open the `/etc/nginx/sites-available/rtmp` file and add another server at the end of the file, as well as the type of supported application data:

```
...
server {
    listen 8088;

    location / {
        add_header Access-Control-Allow-Origin *;
        root /var/www/html/stream;
    }

}

types {
    application/dash+xml mpd;
}
```

If you have a firewall installed, enable the new 8088 port:

```
ufw allow 8088/tcp
```

And create a `stream` folder in Nginx `DocumentRoot` to serve the HLS and DASH files:

```
mkdir /var/www/html/stream
```

Check that the configuration test is satisfactory and restart Nginx:

```
nginx -t
service nginx restart
```

Use the `ffmpeg` command to send test videos to the RTMP server by running the following command:

```
ffmpeg -re -i "<downloaded file name>" -c:v copy -c:a aac -ar 44100 -ac 1
-f flv -flvflags no_duration_filesize rtmp://localhost/live/stream
```

If the connection is refused, try restarting the service:

```
service nginx restart
```

Exercise 5

a) Explain all the RTMP configuration parameters you added to the two previous configuration files.

b) Check with the `netstat` command or any other similar tool the services that have been configured, showing it by a screenshot.

c) Connect with VLC to `http://<ip_server>:8088/hls/stream.m3u8` and check that you can see the HLS stream over HTTP. Paste a screenshot.

d) Open Wireshark and see how HLS streaming is streamed. You can filter by port to view the packages we are interested in. Analyze the interaction that occurs, from the start of playback to its stop. You can paste screenshots and explain them. In which field do we see streaming as HLS?

e) Connect with VLC to `http://<ip_server>:8088/dash/stream.mpd` and check that you can view the DASH stream over HTTP. Paste a screenshot.

f) Open Wireshark and see how DASH streaming is streamed. You can filter by port to view the packages we are interested in. Analyze the interaction that occurs, from the start of playback to its stop. You can paste screenshots and explain them. In which field do we see streaming as DASH?

g) Are there differences between HLS and DASH? Discuss them.

References

- **OBS project page:**
  https://obsproject.com/download
- **Videolan project website:**
  http://www.videolan.org
- **VirtualBox documentation:**
  https://www.virtualbox.org/wiki/Documentation
- **How to install VirtualBox:**
  https://www.wikihow.com/Install-VirtualBox
- **How to install the VirtualBox Extension Pack:**
  https://www.youtube.com/watch?v=mwKmxxRbvws
- **How to install Linux on Windows with VirtualBox:**
  https://itsfoss.com/install-linux-in-virtualbox/