NTNU - Department of Information Security and Communication Technology

TTM4250 - Semester Assignment 3: Adaptive Video Streaming

Learning Objectives

- Understand how video is encoded and compressed.
- Understand how video is packaged and served for adaptive video streaming.
- Learn how to properly adjust adaptive video sets to perform well under varying network conditions.

Guidelines

Start by downloading the template Video-lab GNS3 project and import it into your GNS3 instance in your lab VM.

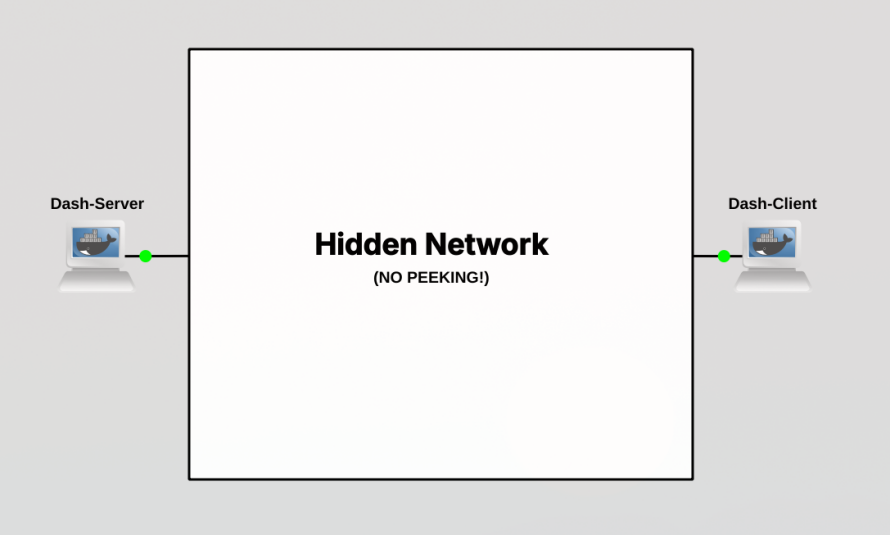


Figure 1: Video Network Template

Introduction

Anna is working for a video content distributor and has been tasked to encode the distributor's masterpiece, Big Buck Bunny. Because the network that the video is distributed over is unstable she needs to ensure that the video-content is streamed adaptively, ensuring that viewers have the best possible experience given the circumstances.

Prerequisites

For this lab we will be interacting with lab components through VNC. To accommodate this, we must adjust our port-forwarding configuration and install a VNC-client. Sahara PCs should have an appropriate VNC-client installed and so for those machines, only the port-forwarding is required.

- Append ports 5900-5910 to your GNS3 ssh configuration. The configuration available at: `https://ttm4250.iik.ntnu.no/latest/wiki/Setup` now has added ports included.

- Install a VNC client. We recommend TigerVNC as it has installations for a wide range of platforms and has been tested for this lab. Sahara PCs should have TigerVNC pre-installed.
- Import the Video-Lab template from the lab-page `https://ttm4250.iik.ntnu.no/latest/labs/video_lab/`

Task 3.1 First Video-Server Setup

a) **Encode the video.**

To properly set up an adaptive video-server we must first encode the raw video into a suitable format. A set of scripts to assist in this task should be available on your VMs. Navigate to the /home/ttm4250 directory and run the following script:

```
python3 HAS_encode.py bbb-first-120s.mp4 encode-set-1 \
--rung 2560x1440:60:4800 --rung 2560x1440:60:400
```

When the encoding is finished, a folder with the required chunks and manifest should be available in "encode-set-1"

b) **Upload the encoded video to the DASH server.**

To upload the encoded video set to the server, use the available upload-stream script. Ensure that the GNS3-project is open and that the DASH Server is started (is on). Find the container-ID of the DASH server using docker ps. Run the upload-stream.sh script from within the /home/ttm4250 directory:

```
# Find the container-ID of the DASH server
docker ps

# Replace CONTAINER_ID with the actual container ID
./upload-stream.sh CONTAINER-ID encode-set-1
```

After the encoded video is uploaded to the server, restart the DASH server in GNS3 to make the video available.

c) **Stream the video from the DASH client.**

To stream the video from the DASH client:

- Ensure that all components of the GNS3-project are started.
- Double-click the DASH client to open a GUI for video streaming.
- Change the video stream URL to point to the encoded and uploaded stream, i.e. http://10.10.10.2/encode-set-1/manifest.mpd
- Set an experiment label, e.g. encode-set-1-run-1
- Click "Start run"

d) **Extract logged video performance data.**

Once the video is done streaming, the player logs the performance of the streamed run. To extract this logged data out of the DASH client container, you can use the collect-metrics script. First, find the container-ID for the DASH client using "docker ps":

```
# Find the container-ID of the DASH client
docker ps

# Replace CONTAINER_ID with the actual container ID
./collect-metrics.sh CONTAINER-ID results
```

The logged metrics of the run should now be available in the "results" directory. To analyse the metrics and receive a summary for all the runs available in the results directory, you can run:

```
python3 analyze-runs.py results
```

This should produce a .csv file which can help compare the different attributes of a run, given an encoding.

Which results in the .csv file might be used as indicators of QoE?

Task 3.2 Re-Encode For Performance

a) **Re-encode the Video**

Anna noticed that the video streaming with the first encoding didn't perform too well. The streamed video was quite blurry and of poor quality. To address the issues, Anna needs help performing a new encoding of the Big Buck Bunny video.

To adjust the encoding, declare new "rungs" of encoding, when using the HAS_encode.py script. A rung is defined by:
HORIZONTAL_PIXELS x VERTICAL_PIXELS : FRAMES_PER_SECOND : VIDEO_BITRATE
Where the video bitrate is specified in kbps.

Please note, that Video encoding is a processing intensive process and it might be efficient to perform this process locally. In that case, follow the suggested steps in the wiki: `https://ttm4250.iik.ntnu.no/latest/wiki/video-streaming/`

To ensure that the video server has enough storage to host the video, the produced encoding set must be smaller than 1.5 times the size of the original video. You must therefore be slightly selective when selecting rungs.

Another restriction on the encoded set is that all the rung resolutions must have an aspect ratio that matches the original video (16:9).

The available options for frames per second are: 24, 30, and 60.

Generate new encoding sets and perform new playback tests. Did the video quality improve between your attempts?

b) **Different Adaptation Strategies**

The default for the DASH player is to use both buffer and throughput measurements to adapt the video-stream. What happens to the performance when adjusting the adaptation mechanism, to either buffer-based or purely throughput-based? What differences do you experience when adjusting the starting throughput or target buffer?

Note: It might be helpful to label the different runs with which adaptation mechanism you have used.

Which types of content are the different adaptation types most suitable for?

Task 3.3 Competitive Encoding

Create at least five different sets of encoding and do at least three runs per encoding set to try to find an ideal set of encodings. `https://ttm4250.iik.ntnu.no/latest/wiki/video-streaming/` describes some principles that might assist in efficient encoding.

For a fair comparison, keep the adaptation strategy and other configurable playback parameters consistent across encoding sets. Perform at least three runs per encoding set and compare the resulting metrics. Discuss the variability between runs and explain how it affects your recommendation.

Make a proposal of which encoding set your group recommends. Ground your recommendation in the collected metrics from your trials. Include a table comparing the metrics of the different encoding sets and highlighting top-performers.

Task 3.4 Reflection

Video streaming is a key application in today's internet. Given that video streaming is a network-intensive application and the conditions of networks are dynamic, technical mechanisms must be in place to ensure efficient and reliable content delivery.

What mechanisms have you applied in this lab? Consider both high-level mechanisms and the underlying components that make these mechanisms work.

Can you think of other network principles that are crucial to realizing video streaming in the internet, that are not covered in this lab?

Submission details

Please submit an archive (.tar.gz or .zip) which contains all files you want to submit. Please have your group number in the file name and the directory name. Submission channel will be shared during the lectures. A report (one single PDF file, named VIDEO-group(num).pdf) containing the following elements is mandatory:

- Your group number on the first page.
- For all questions: state your assumptions, report what you did, describe what you observed, and explain your conclusions and thought process.
- For all references to your encoding sets state which rungs were included.
- If you have used Artificial Intelligence, state which AI tools have been used for what in an AI usage section. Alternatively, you may include an AI declaration form.

Include scripts that were used to compare the results .csv. If you have used scripting or commands other than the provided ones to encode the videos, include these scripts in your delivery.

We can only grade what we find in your submission and what we understand. Please state your assumptions and observations as clearly as possible.

Due date: November 8, 2026, 11:59 PM