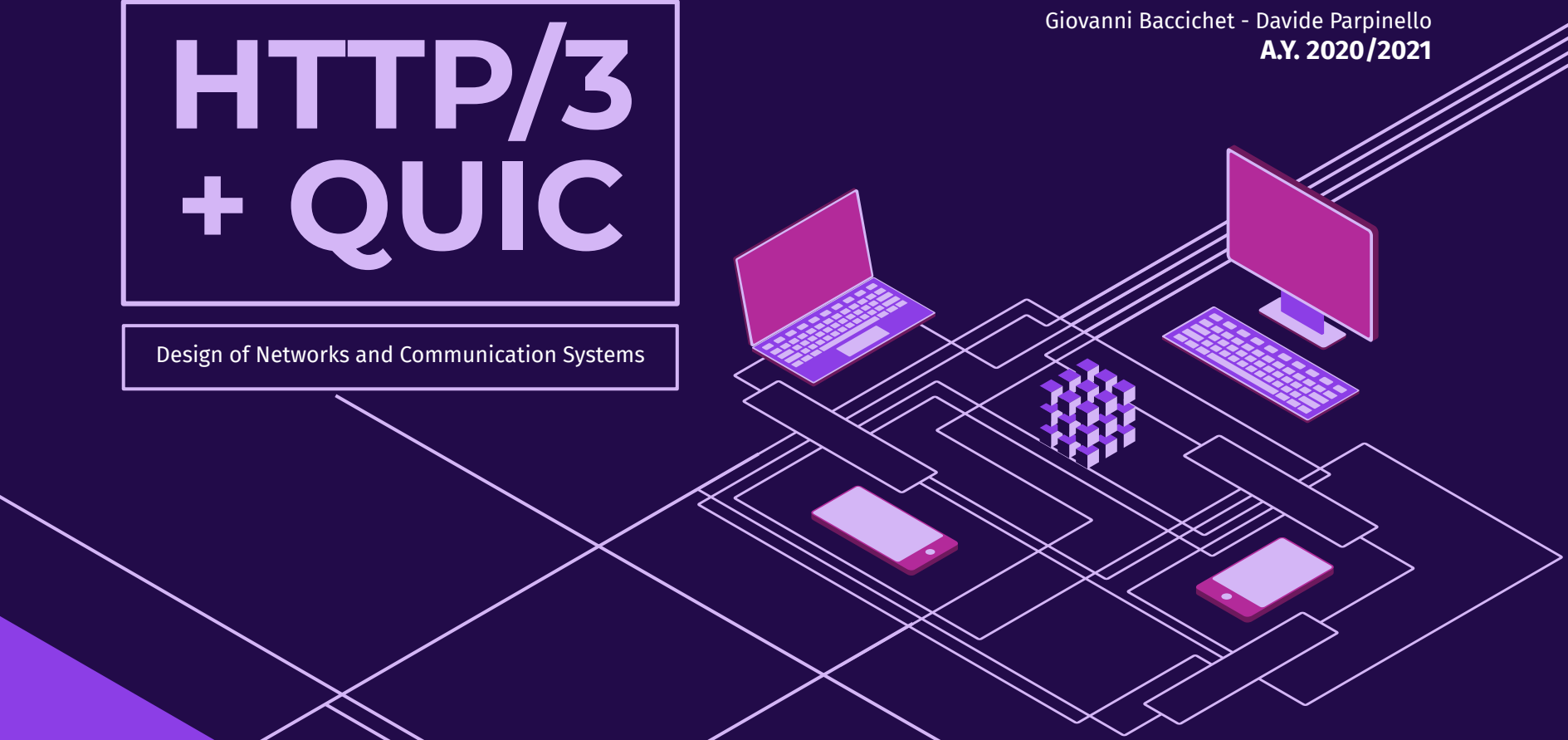
HTTP/3 + QUIC

Design of Networks and Communication Systems

Giovanni Baccichet - Davide Parpinello

A.Y. 2020/2021



DISI - University of Trento

PROJECT GOALS

The goal of the project is to build a **virtualized framework** for analyzing the performance of **HTTP/3 + QUIC**, with respect to HTTP/2 or TCP.

LAB ENVIRONMENT

01

Virtualized Lab

Unbiased and replicable to everyone

02

Use of Vagrant software

To replicate a realistic network scenario

03

Use of Docker containers

For convenience (two images for six containers)

04

Use of all three protocols

HTTP/1.1, HTTP/2 and HTTP/3 + QUIC comparison

05

Three static content containers

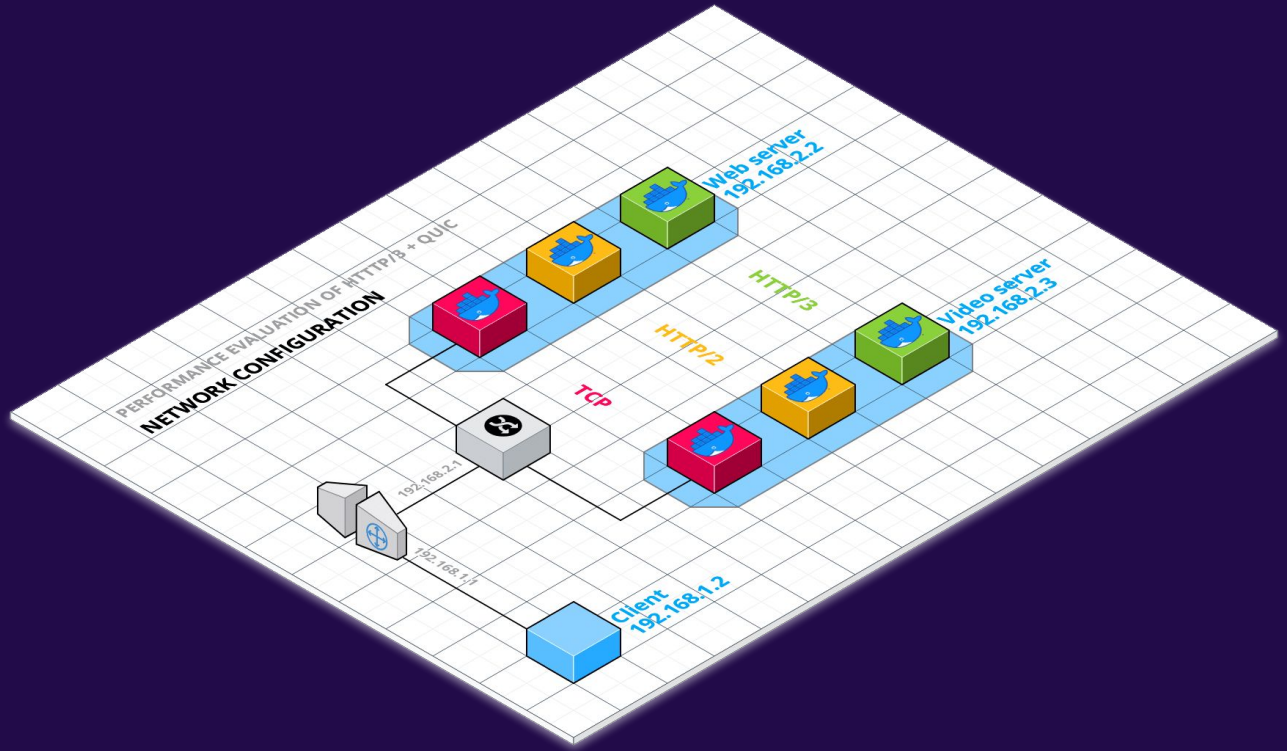
Static web page compared between protocols

06

Three HLS stream containers

Video streams usage compared between protocols

NETWORK TOPOLOGY



VAGRANT CONFIGURATION

X11 Forwarding

Adding the configuration to Vagrantfile to achieve X11 Forwarding, used to run browser in the client

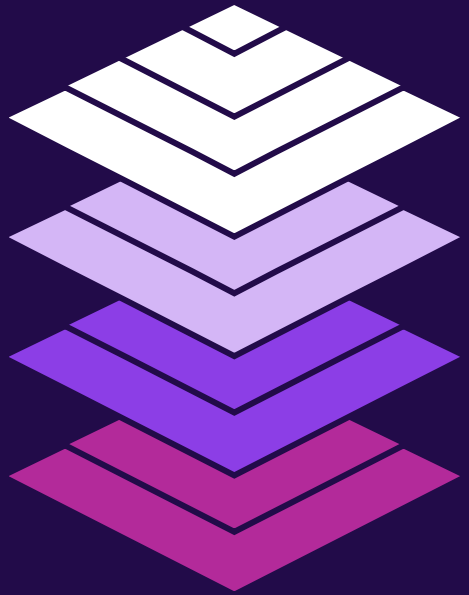
RAM upgrade

The client and servers VM's RAM needs to be upgraded in order to run all the software required

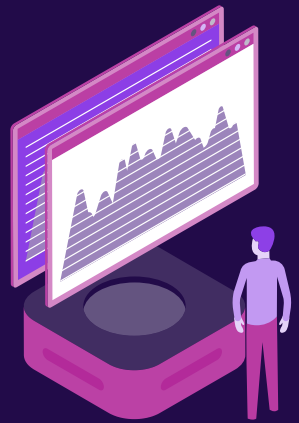
Docker Hub

Use of images downloaded from Docker Hub in order to save time

SYSTEM CONFIGURATION



- **HTML or HLS content**
  Content used to analyze performance
- **NGINX 1.16.1**
  Web server patched for HTTP/3 support
- **Docker container**
  Multiple containers to run different protocols
- **Vagrant**
  Virtual machines management



PROBLEM

In order for QUIC to work properly a valid SSL certificate from a trusted CA is needed

SOLUTION

Use of **Let's Encrypt** to sign an SSL certificate for a subdomain pointing to a local IP (e.g. `web.bacci.dev` pointing to `192.168.2.2`)

NOTE

HTTP/3 needs to run on port `443` in order to work correctly. For this specific reason the containers are divided in 2 different hosts (one for static content and one for video)

WEB PAGE IMAGE

Subsystem

Ubuntu Linux distro

NGINX

NGINX web server
patched with Quiche

Alt-svc Header

Add Alt-Svc header to
negotiate HTTP/3.

```
add_header alt-svc
'h3-29=":443"; ma=86400';
```

Different conf files

Based on the NGINX conf
file the server will run the
three different protocols

HLS STREAM IMAGE

Subsystem

Ubuntu Linux distro

NGINX

NGINX web server patched with Quiche

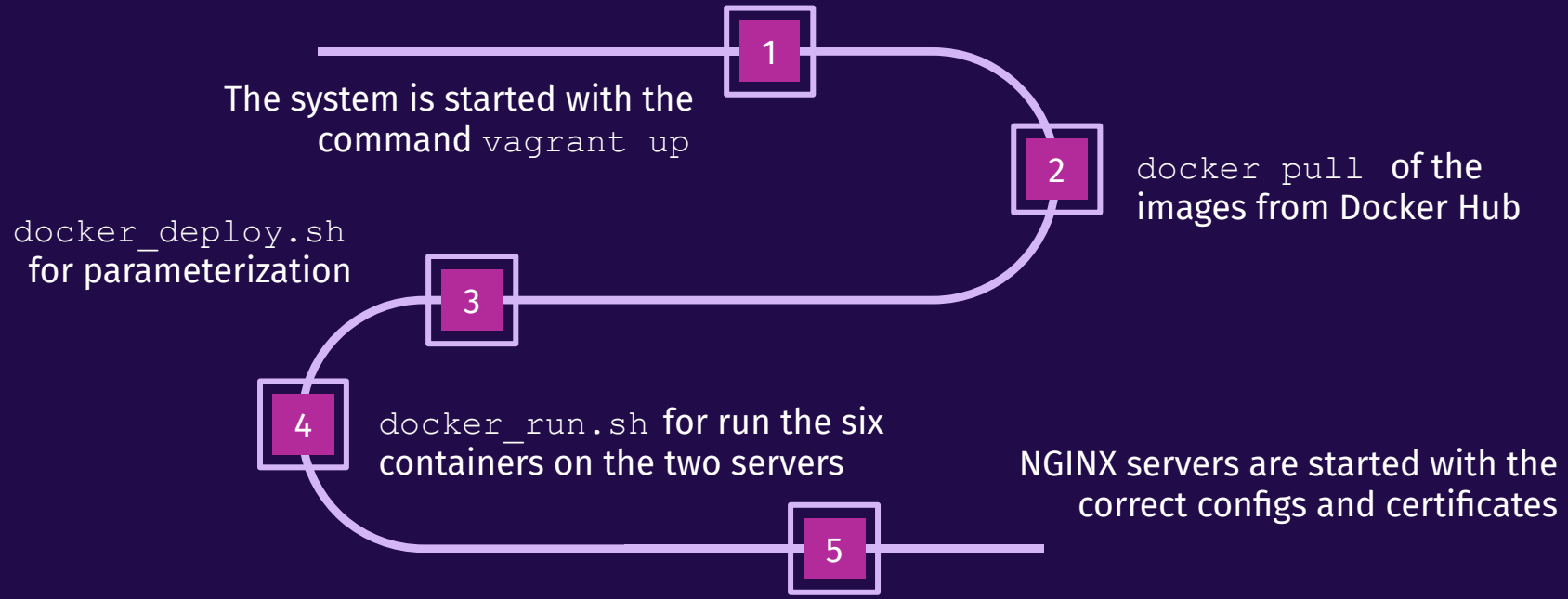
RTMP Module

Use of RTMP module in NGINX to provide the HLS stream

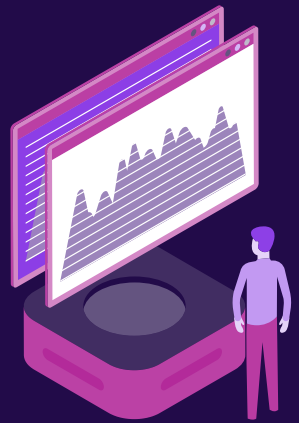
FFMPEG

Use of `ffmpeg` to send a video stream created from a file to NGINX

DEPLOYMENT


1
The system is started with the command vagrant up
2
docker pull of the images from Docker Hub
docker_deploy.sh for parameterization
3
4
docker_run.sh for run the six containers on the two servers
NGINX servers are started with the correct configs and certificates
5


PERFORMANCE
ANALYSIS



PROBLEM

To test the system, it's necessary a browser that support HTTP/3 + QUIC, Chrome doesn't do that natively

SOLUTION

Chrome has to be launched with the following commands

```
--enable-quic
--quic-version=h3-27
```

TOOLS USED

Google Chrome DevTools

For the web page static content it’s enough to use Chrome DevTools statistics (e.g. load time)

httpstat

This command line tool allows to get statistics with ease of use and a nice visualization

Web HLS players

Web HLS players like TheoPlayer and Hls.js to get metrics (e.g. startup time)

EVALUATION CRITERIA

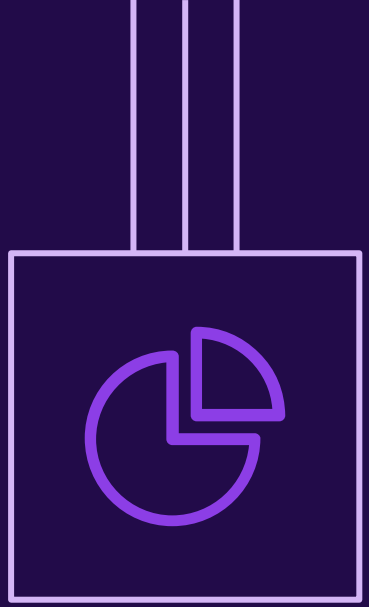
Two different evaluation methodologies were applied to the web-static content and the video streaming

WEB-STATIC CONTENT:

- **TTFB** (Time to first byte)
- **Page weight**
- **Load time**
- **Number of requests**
- **TCP connection time**
- **TLS handshake time**
- **Server processing**
- **Content transfer time**

VIDEO STREAMING (test duration 90 seconds)**:**

- **Startup time**
- **Latency**
- **Bitrate**
- **Dropped frames**



RESULTS

PAGE SIZE: 3.5 MB

	TCP	HTTP/2	HTTP/3
Page weight	3.5 MB	3.5 MB	3.5 MB
TTFB	2.16 ms	3.21 ms	2.83 ms
Load Time	899 ms	961 ms	1.08 s
# requests	30	30	30
# TCP connections	6	1	0

PAGE SIZE: 7.3 MB

	TCP	HTTP/2	HTTP/3
Page weight	7.3 MB	7.3 MB	7.3 MB
TTFB	4.64 ms	5.63 ms	5.51 ms
Load Time	2.92 sec	2.99 sec	3.33 sec
# requests	120	123	134
# TCP connections	6	1	0

VIDEO HLS STREAM

	TCP	HTTP/2	HTTP/3
Startup time	799 ms	985 ms	1242 ms
Avg latency	25.78 ms	27.86 ms	46.39 ms
Avg bitrate	57.44 MB/s	139.57 MB/s	64.17 MB/s
Dropped frames	603	256	328

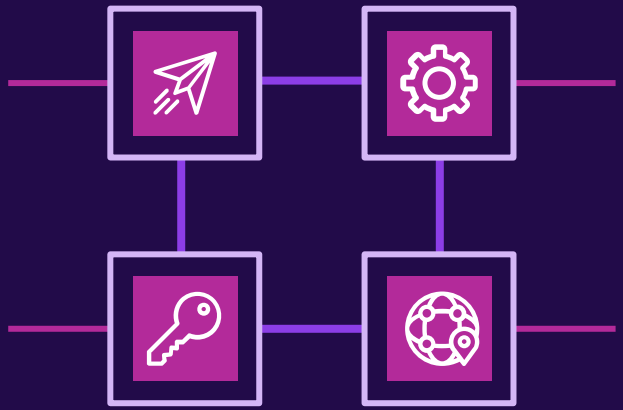
CONCLUSIONS

HTTP/3 is slower than HTTP/2

Based on the tests using pages of different sizes

Security

HTTP/3 uses SSL encryption by default



Protocol is still a beta

Improvements should come with a stable release

More reliable

Solves HTTP/2 OSI-model layering violations: useful for mobile networking

THANKS!

GitHub

https://github.com/GiovanniBaccichet/DNCS-HTTP3

Authors:
Giovanni Baccichet
Davide Parpinello