

Giacomo Fedrigo Luigi Dorigatti
A.Y. 2020/2021

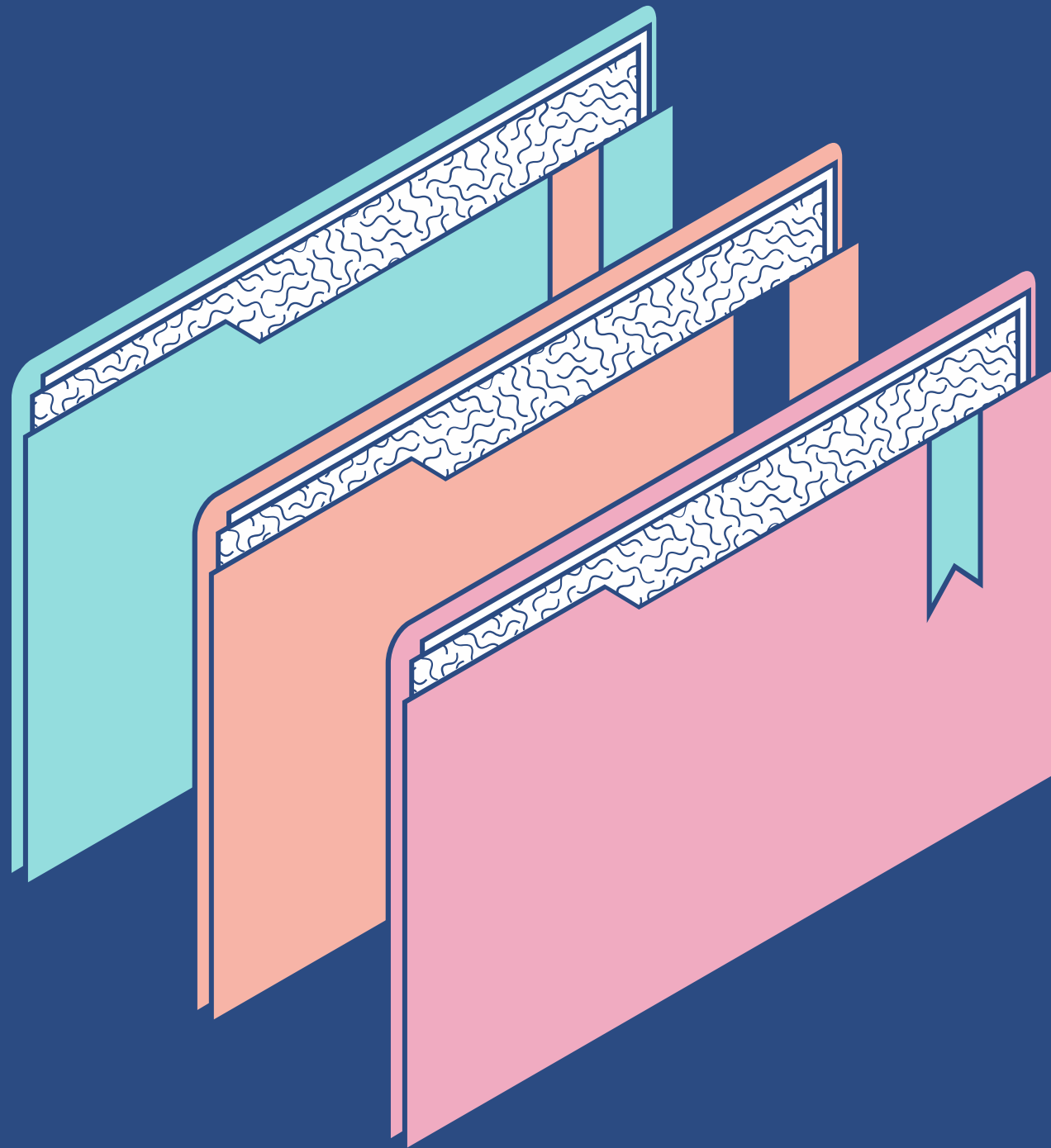
HTTP/3 + QUIC

Design of networks and communications
systems

DISI-University of Trento

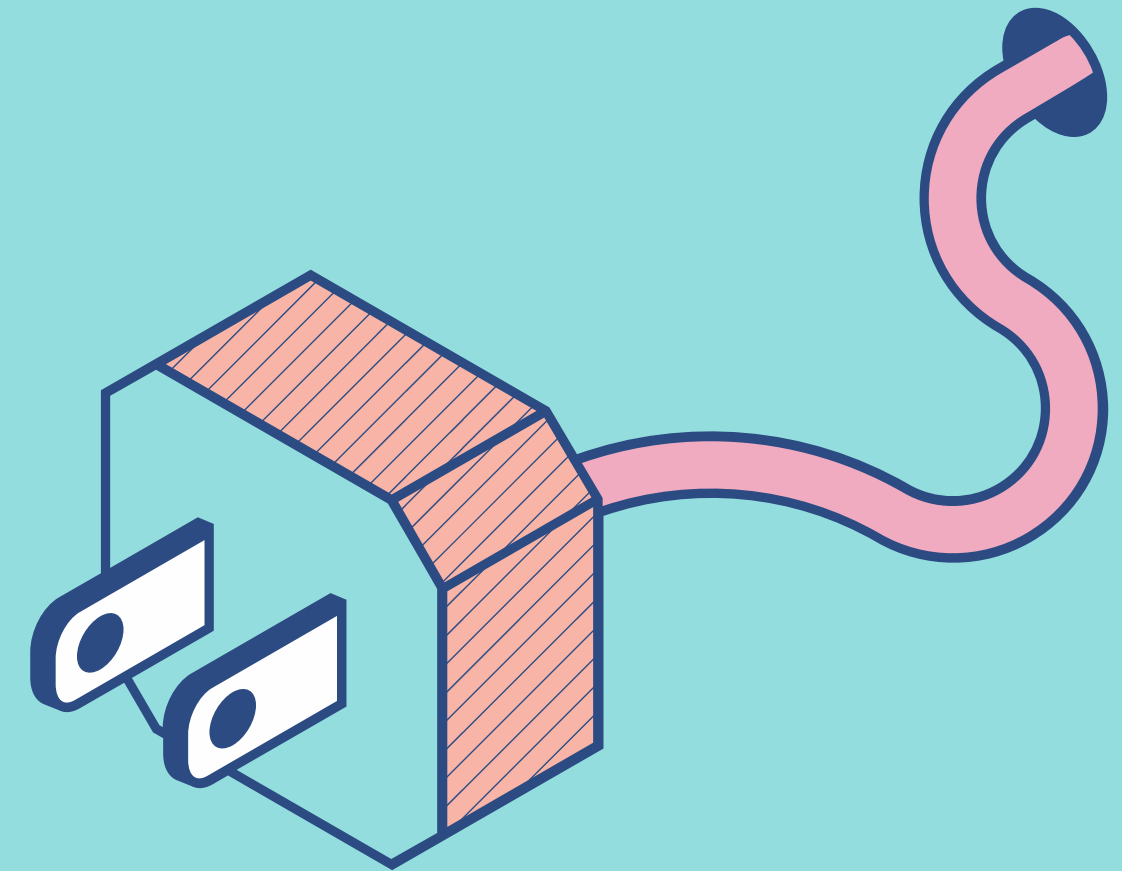
Obbiettivo

L'obiettivo del progetto è di costruire un framework virtualizzato per analizzare le prestazioni di HTTP/3 + QUIC, rispetto ad HTTP/2 o TCP.



Laboratorio

- **Uso di Vagrant software**
Per replicare un network realistico
- **Uso di Docker containers**
- **Use di tutti e tre i protocolli**
HTTP/1.1, HTTP/2 e HTTP/3 +
QUIC comparati
- **Tre container statici**
Pagina web statica confrontata tra i protocolli
- **Tre HLS stream container**
Utilizzo dei flussi video a confronto
tra i protocolli

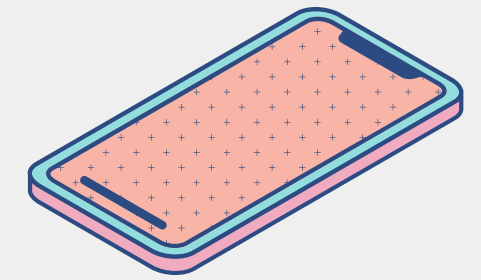
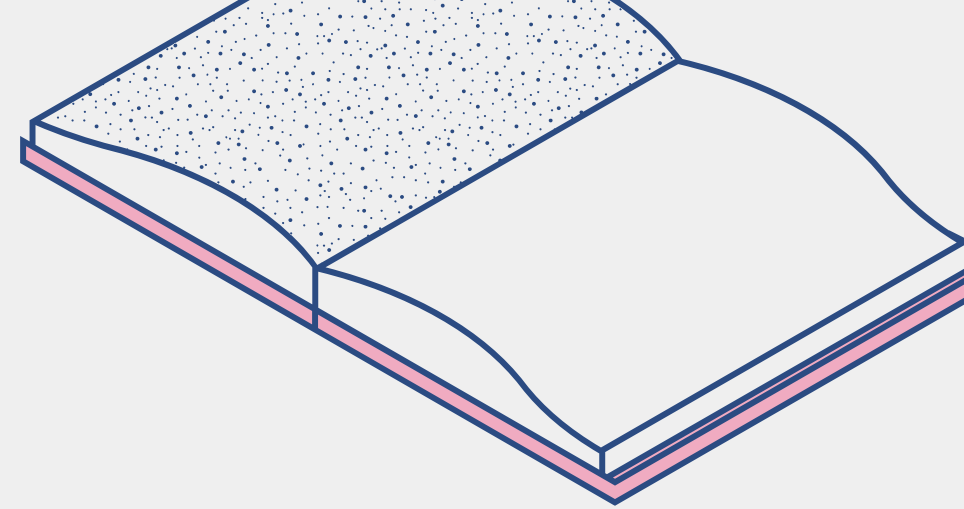
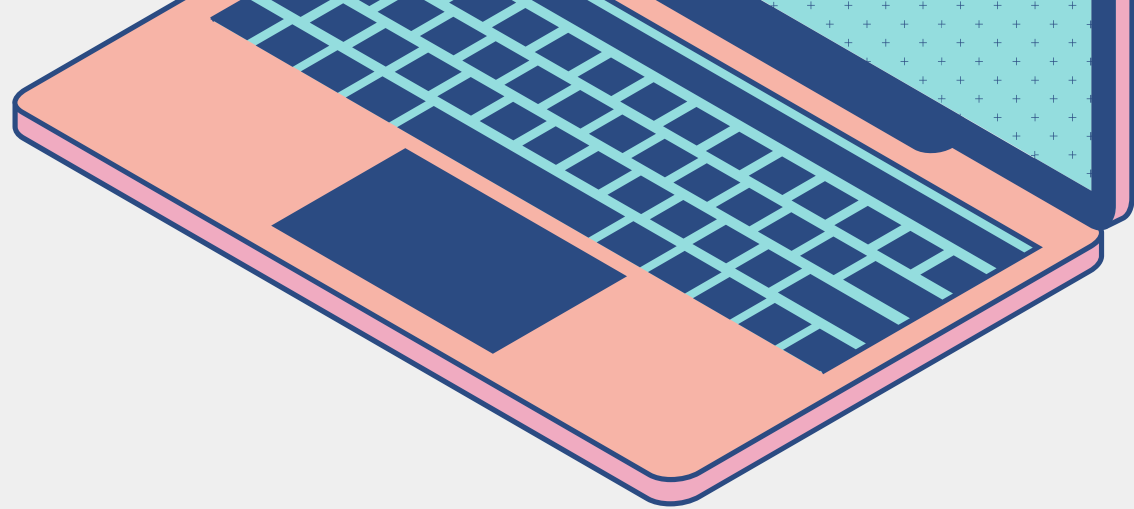




Per svolgere il progetto useremo il web server NGINX.

E' un web server open source che, a cominciare dal suo successo iniziale come server, è ora utilizzato anche come proxy inverso, cache HTTP e bilanciatore di carico.

Nginx è stato progettato per garantire un basso consumo di memoria e un'elevata concorrenza. Aniché creare nuovi processi per ogni richiesta web, Nginx utilizza un approccio asincrono, basato sugli eventi, in cui le richieste vengono gestite in un singolo thread.



RAM upgrade

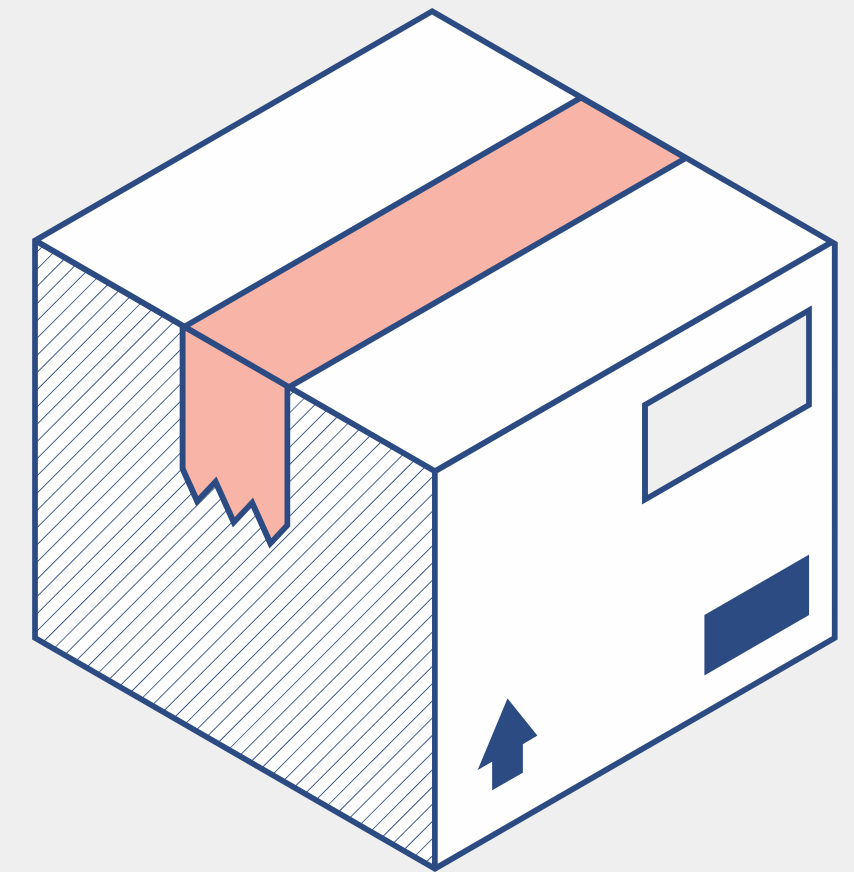
La RAM della VM del client e del server deve essere aumentata per poter eseguire tutto il software richiesto

X11 server

La configurazione a Vagrantfile per ottenere l'inoltro X11, usato per eseguire il browser nel client

Docker Hub

Utilizzo di immagini scaricate da Docker Hub per risparmiare tempo



Docker hub

luigidoricats / **quiche-web**
Updated a few seconds ago

 Not Scanned  0  0  Public

luigidoricats / **quiche-video**
Updated 2 minutes ago

 Not Scanned  0  4  Public

L'approccio scelto è stato quello di costruire 2 diverse immagini Docker, distribuendo 6 contenitori: la prima serve allo scopo di eseguire un web-server, mentre l'ultima è utilizzata per lo streaming video HLS.

L'immagine per lo streaming video e quella per il web-server e sono entrambe basate sul server NGINX, più precisamente NGINX 1.16.1 (questa particolare versione è necessaria per eseguire la quiche-patch).

Infatti entrambe le immagini sono preconfigurate come HTTP/3, ma saranno limitate nel file di configurazione nginx.conf per funzionare su TCP, HTTP/2 e HTTP/3+QUIC come richiesto (al fine di completare la valutazione delle prestazioni).



SSL

USO DI LET'S ENCRYPT PER
FIRMARE UN CERTIFICATO
SSL DA CA PER UN
SOTTODOMINIO CHE
PUNTA A UN LOCAL IP
COME WEB.DORICATS.DEV
CHE PUNTA AD 192.168.2.2

Configurazione

1

2

3

4

5

STEP

STEP

STEP

STEP

STEP

Il sistema viene
avviato con il
comando
vagrant up

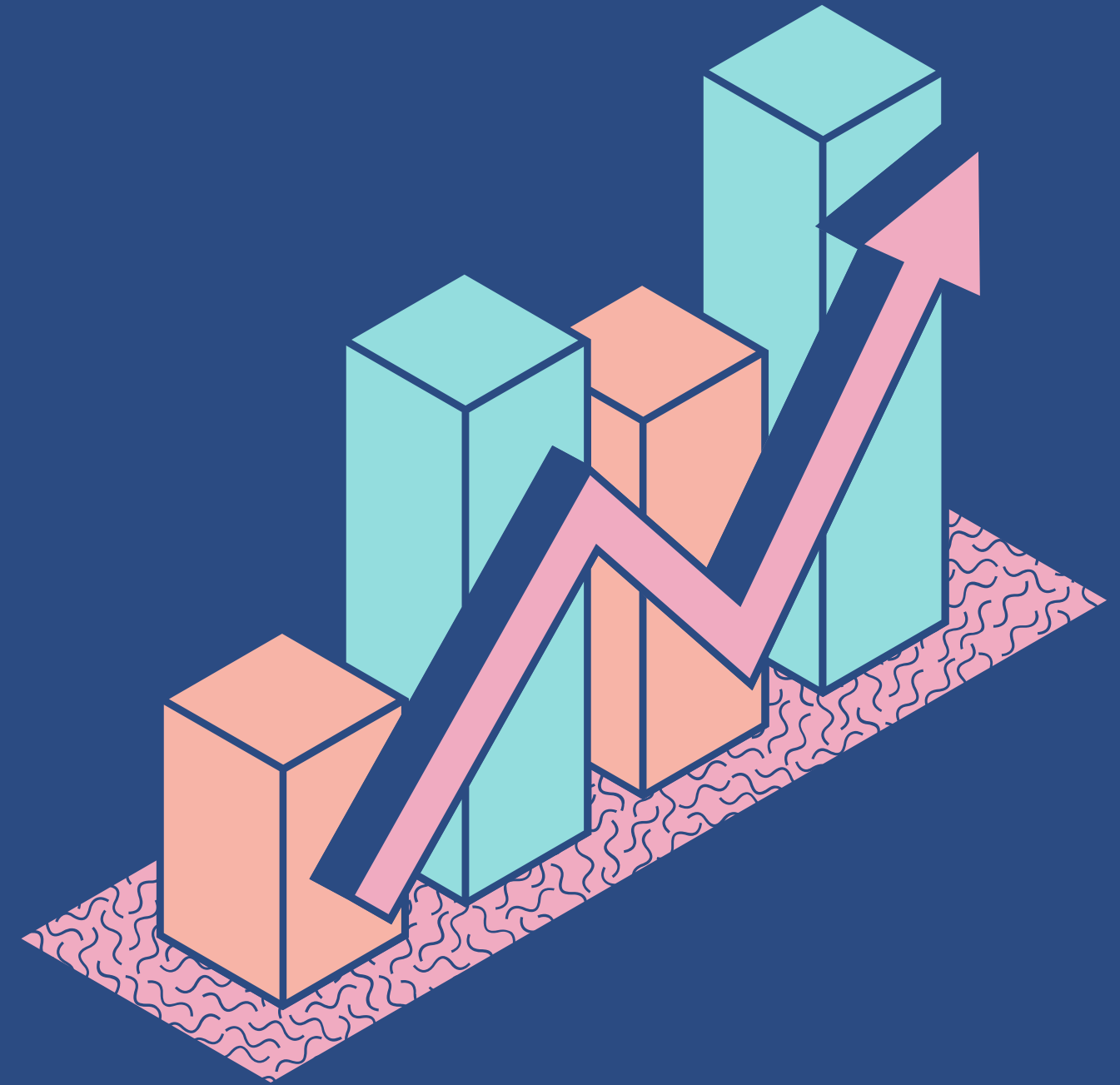
docker pull
delle immagini
da Docker Hub

docker_run.sh per la
parametrizzazione

docker_run.sh per
eseguire i sei
contenitori sui
due server

I server NGINX
sono avviati con
le configurazioni
e i certificati
corretti

Analisi delle performance



Tool Usati



Google Chrome DevTools

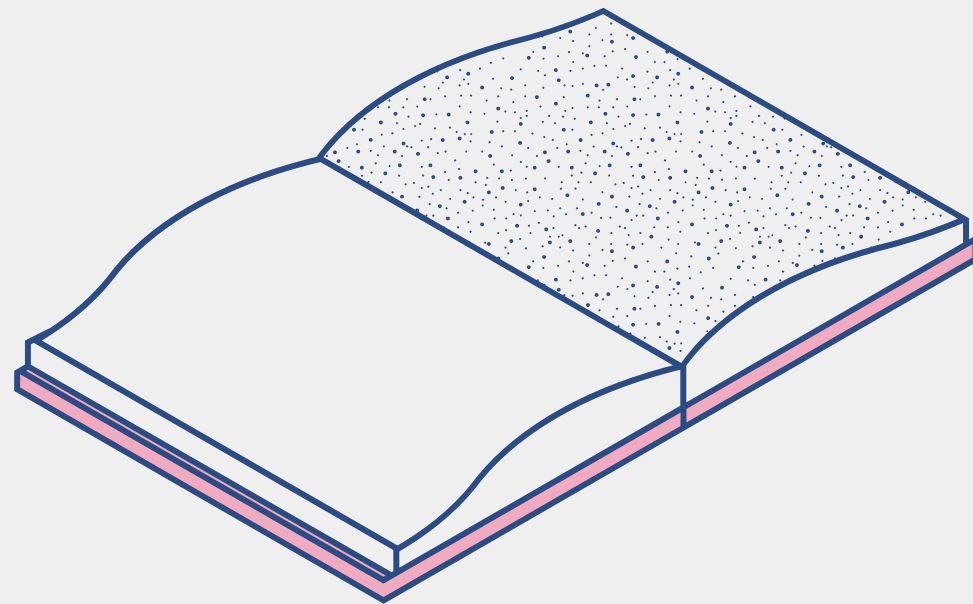
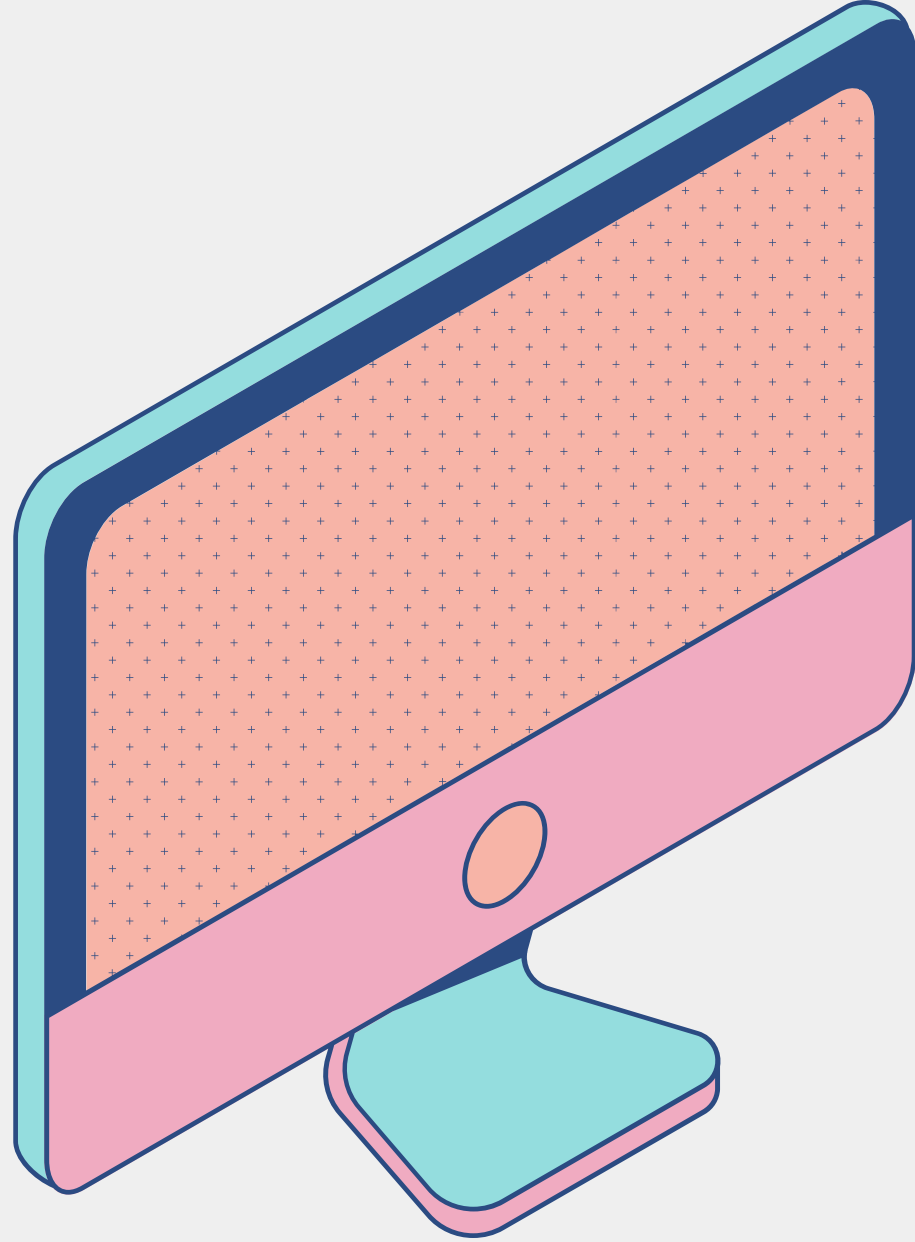
Per il contenuto statico della pagina web è sufficiente usare le statistiche di Chrome DevTools

httpstat

Questo strumento a riga di comando permette di ottenere statistiche con facilità d'uso e una bella visualizzazione

Web HLS players

Lettori web HLS come TheoPlayer e Hls.js per ottenere metriche (ad esempio il tempo di avvio)



Come valutare

L'intera valutazione delle prestazioni è stata fatta utilizzando l'ultima versione stabile di Google Chrome (v 88.0.4324.150) dal client, con l'aiuto di httpstat per facilità d'uso e visualizzazione. Per abilitare HTTP/3 e QUIC l'applicazione deve essere lanciata con il seguente comando:

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

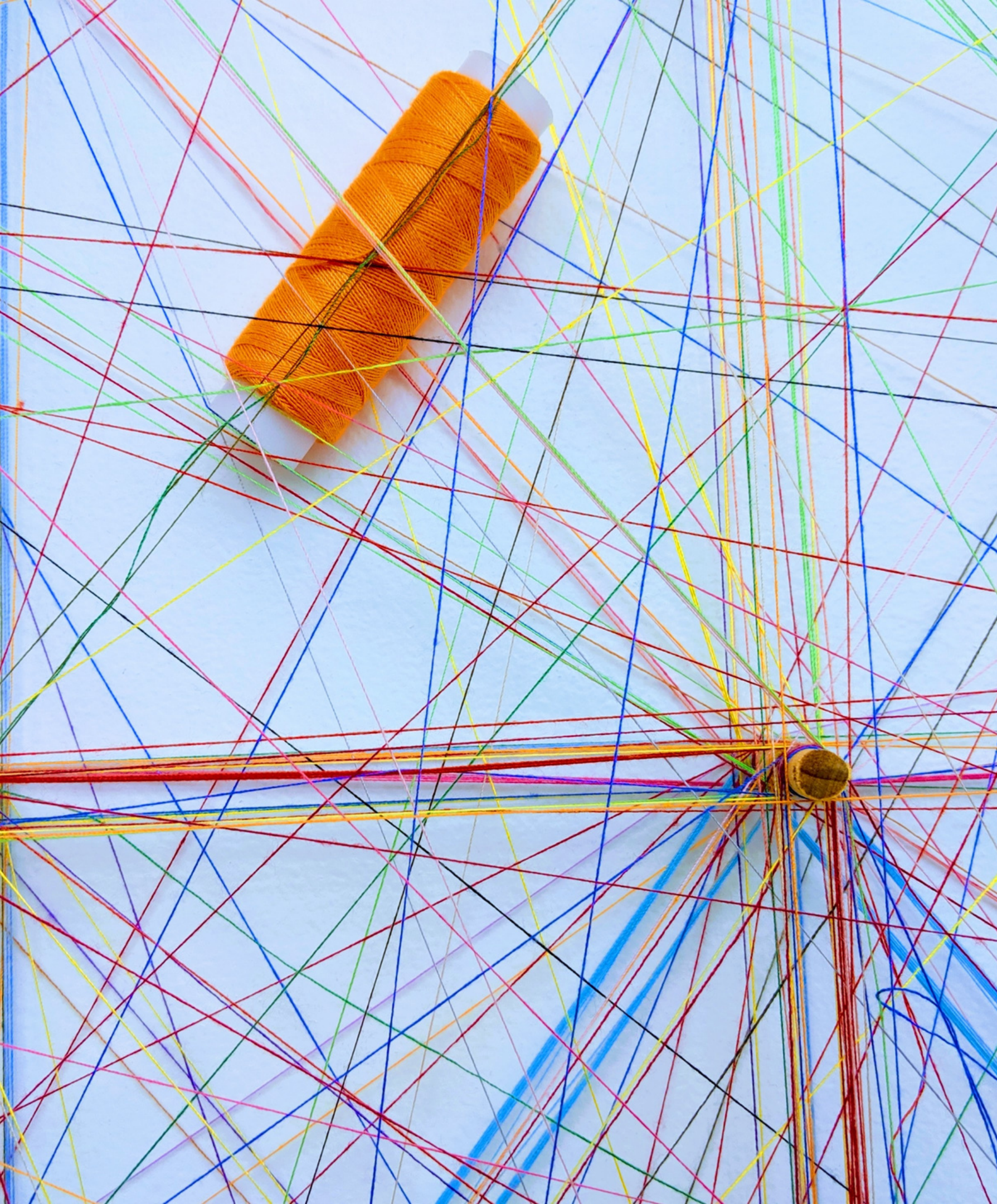
Dati di comparazione

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 :

- Startup time
- Latency
- Bitrate
- Dropped frames



Risultati

Oracle VM VirtualBox Gestore

File Macchina Aiuto



Strumenti



DNCS-HTTP3-main_router_1627318235791_25002

In esecuzione



DNCS-HTTP3-main_switch_1627318307875_66347

In esecuzione



DNCS-HTTP3-main_web-server_1627318427140_40080

In esecuzione



DNCS-HTTP3-main_video-server_1627318644434_71985

In esecuzione



DNCS-HTTP3-main_client_1627318910164_7046

In esecuzione

DNCS-HTTP3-main_client_1627318910164_7046 [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

ubuntu-bionic login: vagrant

L'opzione **Acquisizione automatica della tastiera** è attiva. La macchina virtuale **acquisirà** automaticamente la tastiera ogni volta

Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-147-generic x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

System information as of Mon Jul 26 17:04:39 UTC 2021

System load:	1.11	Processes:	111
Usage of /:	3.9% of 38.71GB	Users logged in:	0
Memory usage:	11%	IP address for enp0s3:	10.0.2.15
Swap usage:	0%	IP address for enp0s8:	192.168.1.2

27 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@client:~\$ httpstat https://web.doricats.dev:443_

Controller: ICH AC97

Rete

Scheda 1: Intel PRO/1000 MT Desktop (NAT)

Scheda 2: Intel PRO/1000 MT Desktop (Rete interna, 'broadcast_router-client')

USB

Controller USB: OHCI

Filtri dispositivi: 0 (0 attivo)

Cartelle condivise

Cartelle condivise: 1

Descrizione

Nessuna

HTTP/3 + QUIC web page:

```
vagrant@client:~$ httpstat https://web.doricats.dev:443
```

Connected to 192.168.2.2:443 from 192.168.1.2:60864

HTTP/2 200

```
server: nginx/1.16.1
```

```
content-type: text/html
```

```
content-length: 84200
```

```
etag: "60251560-19ed8"
```

```
alt-svc: h3-27=":443"; ma=86400
```

```
accept-ranges: bytes
```

Body stored in: /tmp/tmp1G1tZP


```
vagrant@client:~$ httpstat https://web.dorigatti.dev:451
Connected to 192.168.2.2:451 from 192.168.1.2:43398
```

```
HTTP/2 200
server: nginx/1.16.1
content-type: text/html
content-length: 84200
etag: "60251560-19ed8"
accept-ranges: bytes
```

Body stored in: /tmp/tmp7A0ND6

DNS Lookup	TCP Connection	TLS Handshake	Server Processing	Content Transfer
4ms	1ms	14ms	3ms	6ms
total:28ms				

HTTP/2 + SSL web page:

```
vagrant@client:~$ httpstat https://web.doricats.dev:451
```

Connected to 192.168.2.2:451 from 192.168.1.2:43398

HTTP/2 200

```
server: nginx/1.16.1
```

```
content-type: text/html
```

```
content-length: 84200
```

```
etag: "60251560-19ed8"
```

```
accept-ranges: bytes
```

Body stored in: /tmp/tmp7A0ND6

TCP + SSL web page:

```
vagrant@client:~$ httpstat https://web.doricats.dev:452
Connected to 192.168.2.2:452 from 192.168.1.2:32988
```

```
HTTP/1.1 200 OK
Server: nginx/1.16.1
Content-Type: text/html
Content-Length: 84200
Connection: keep-alive
ETag: "60251560-19ed8"
Accept-Ranges: bytes
```

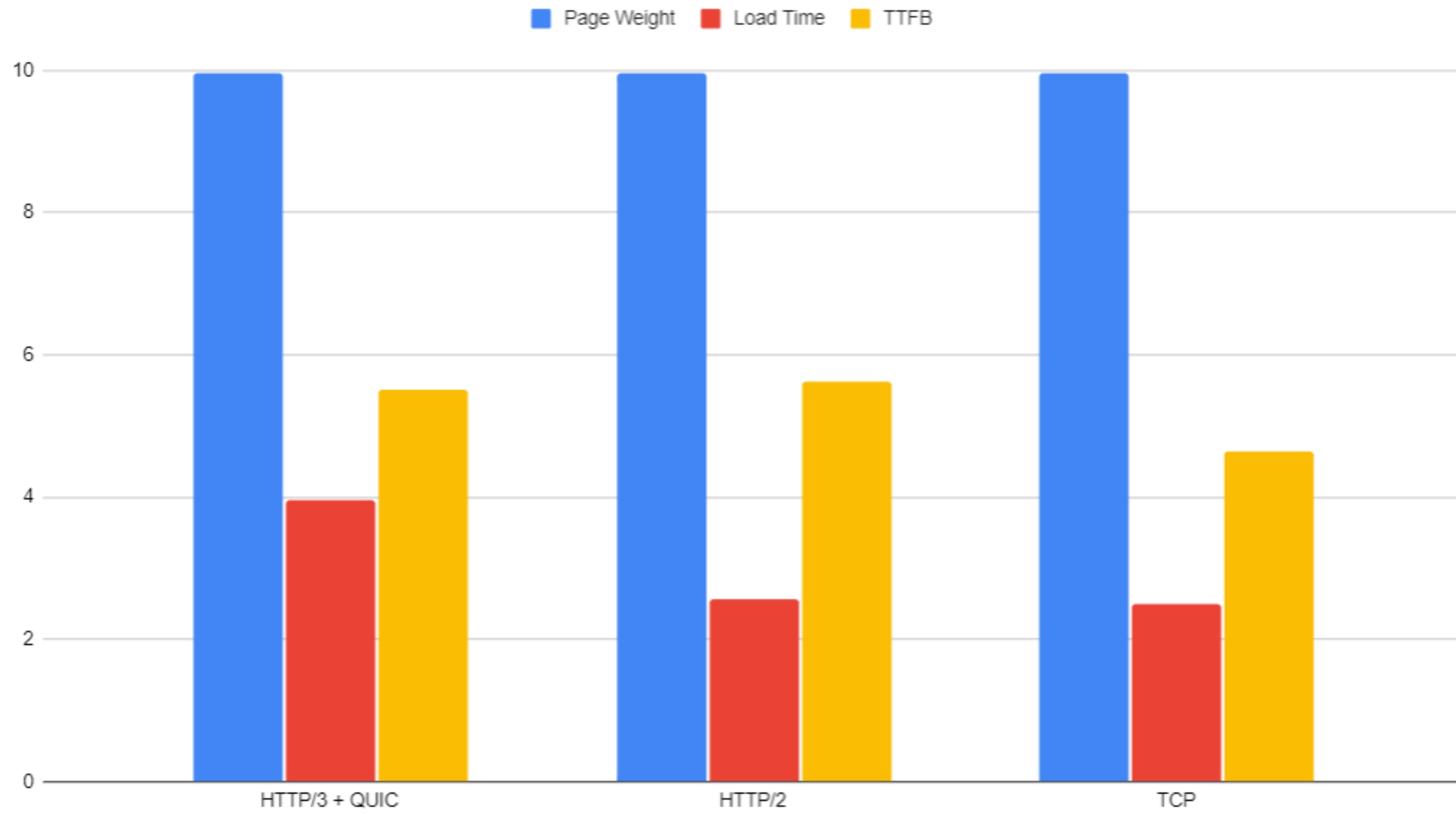
```
Body stored in: /tmp/tmpH9UJe
```

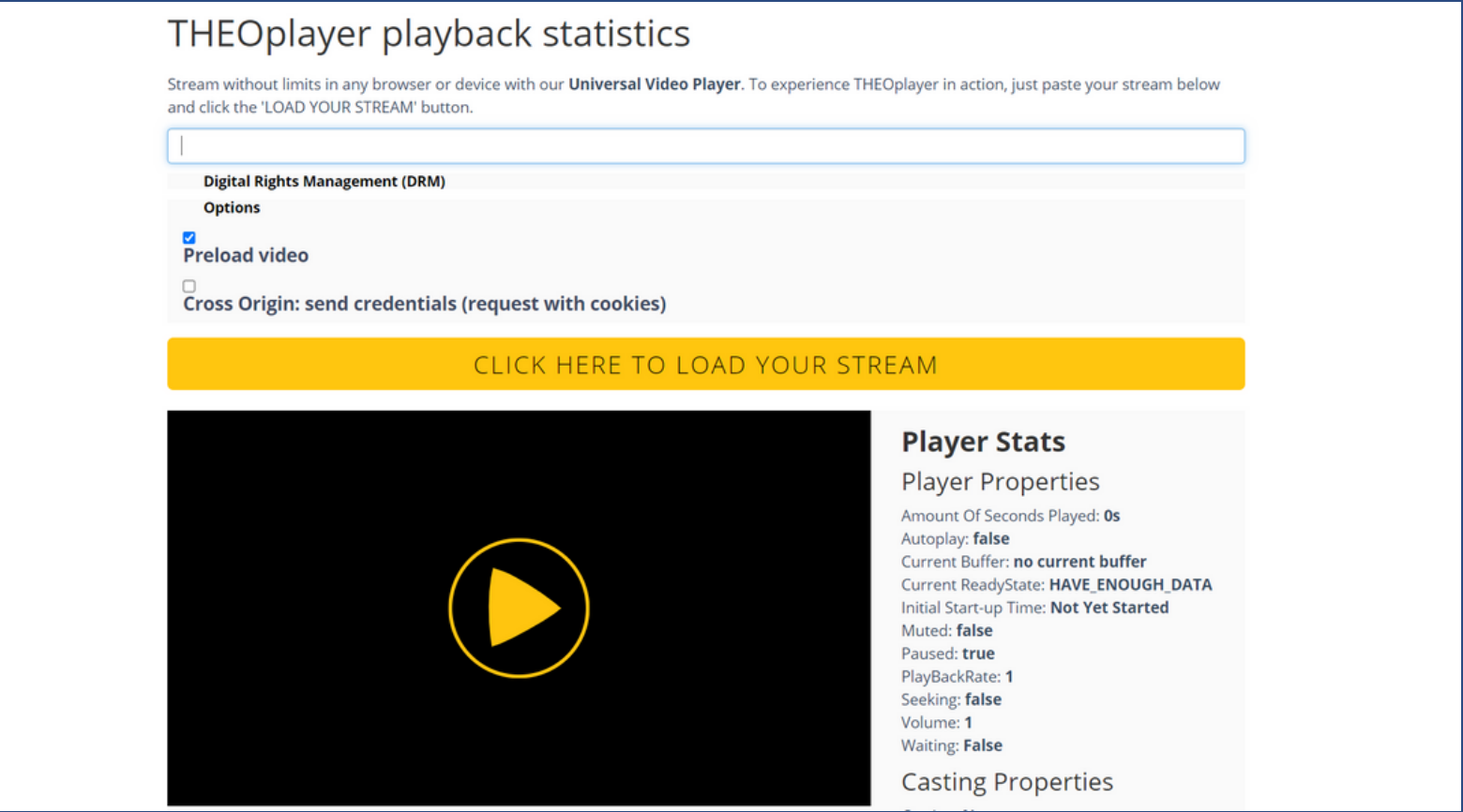
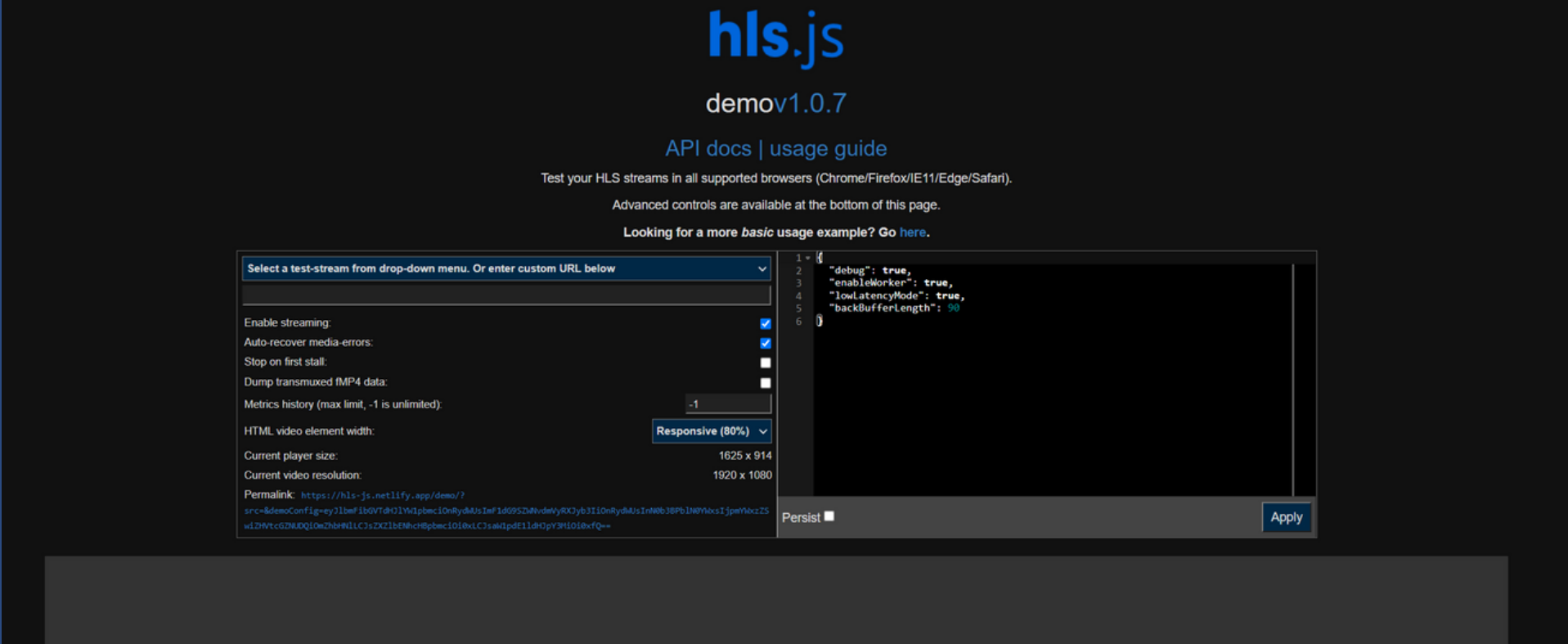
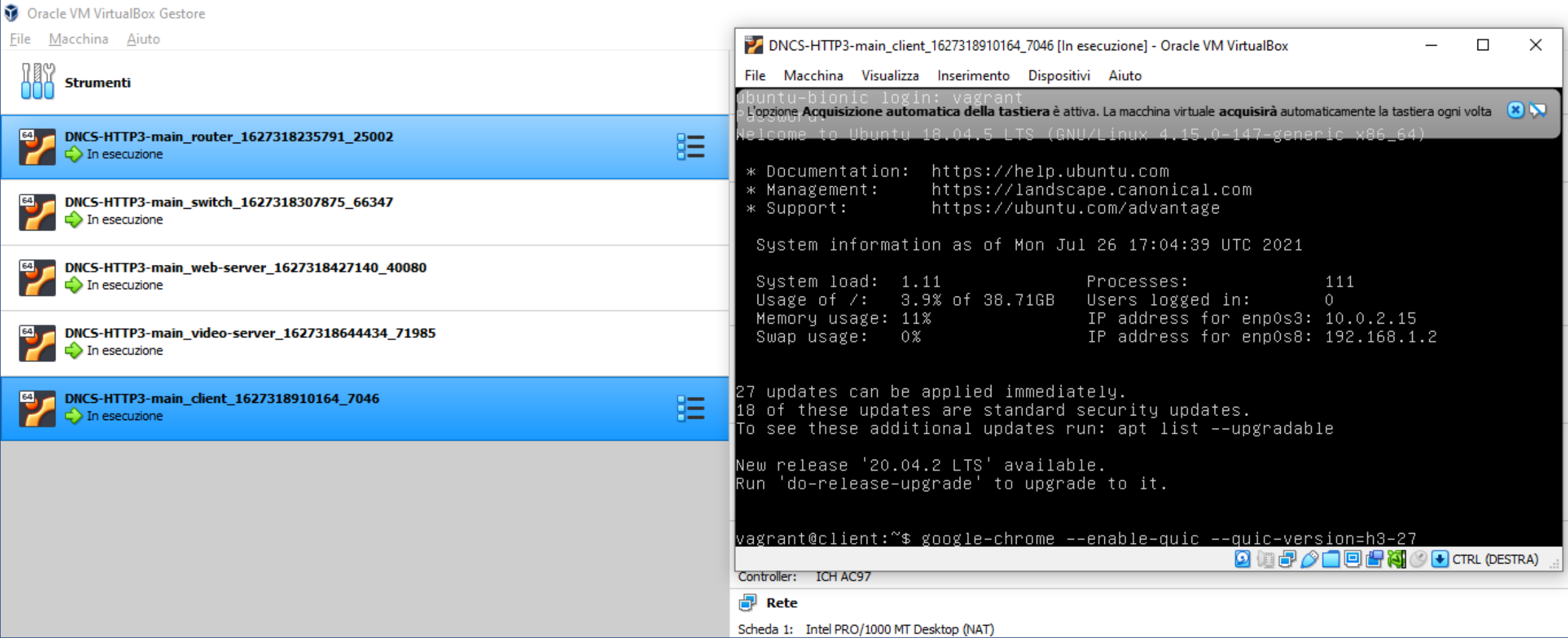


Web page

Protocol	Page weight	TTFB	Load time	# requests
HTTP/3 + QUIC	10.1 MB	5.51 msec	3.96 sec	140
HTTP/2	10.1 MB	5.63 msec	2.57 sec	129
TCP	10.1 MB	4.64 msec	2.49 sec	125

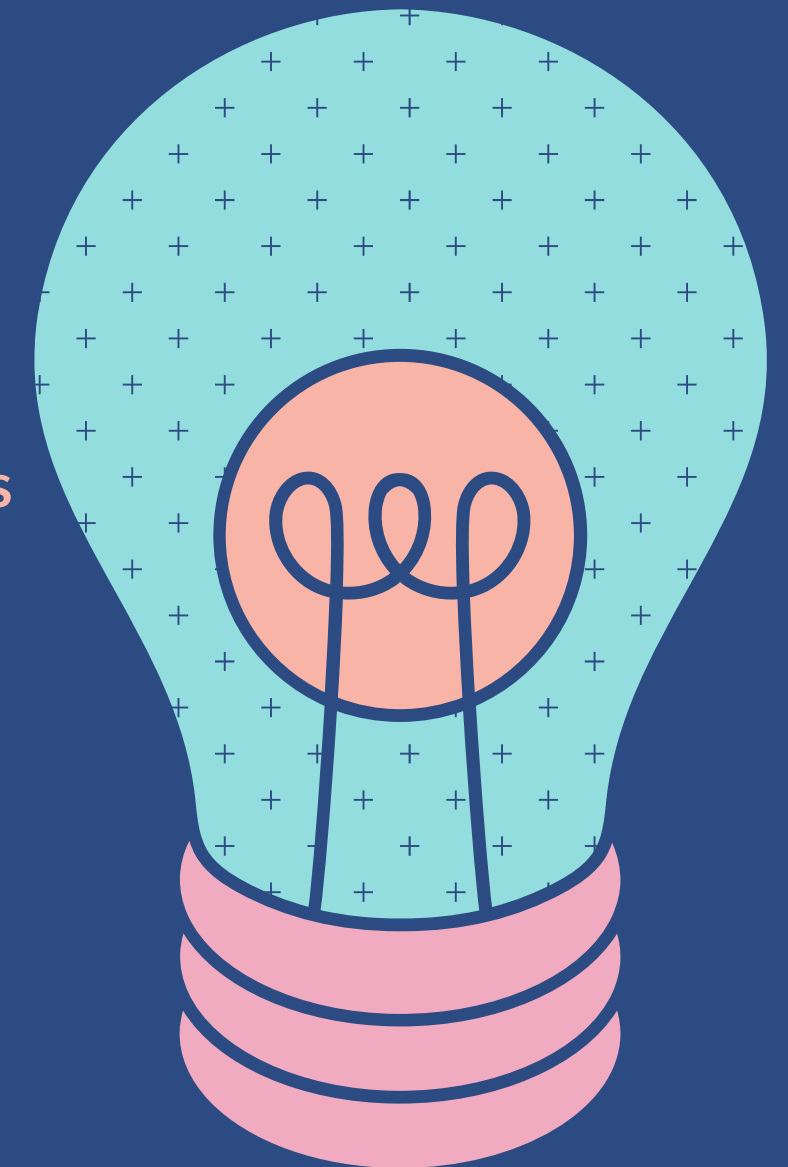
web streaming performance





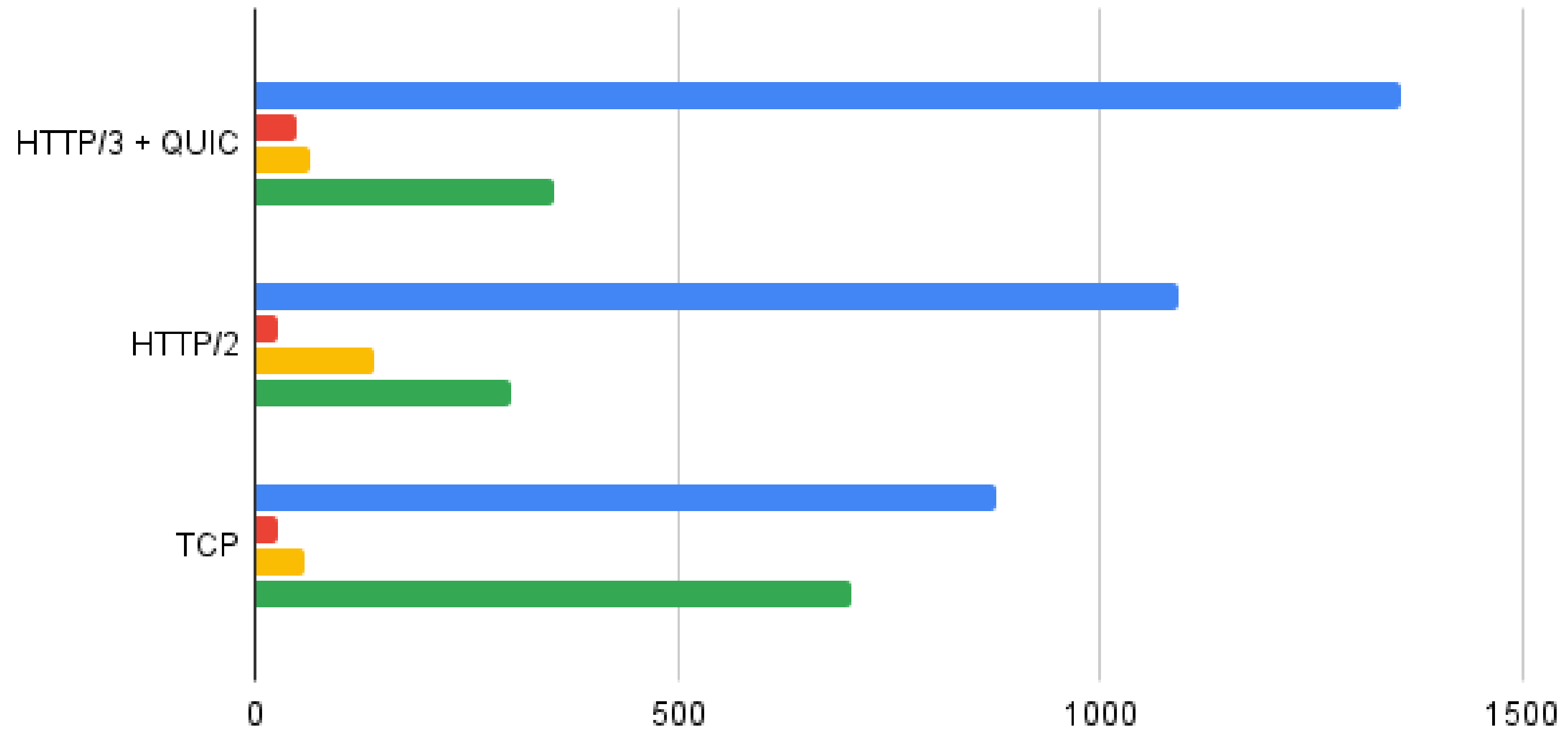
Video stream

Protocol	Startup time	Avg latency	Avg bitrate	Dropped frames
HTTP/3 + QUIC	1356 msec	48,35msec	64,17 MB/s	352
HTTP/2	1094msec	28,32 msec	139,57 MB/s	302
TCP	876 msec	25,78 msec	57,44 MB/s	705

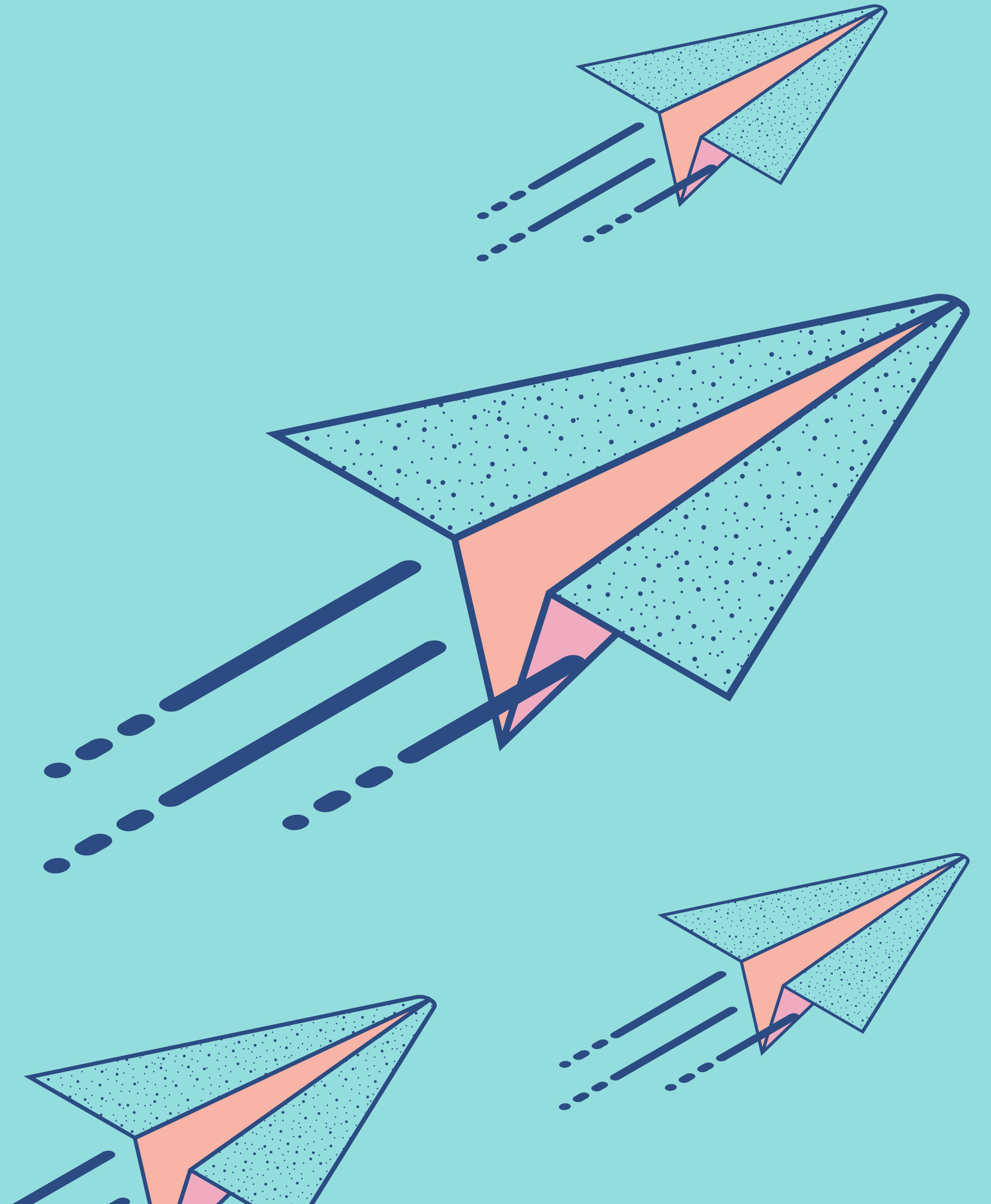


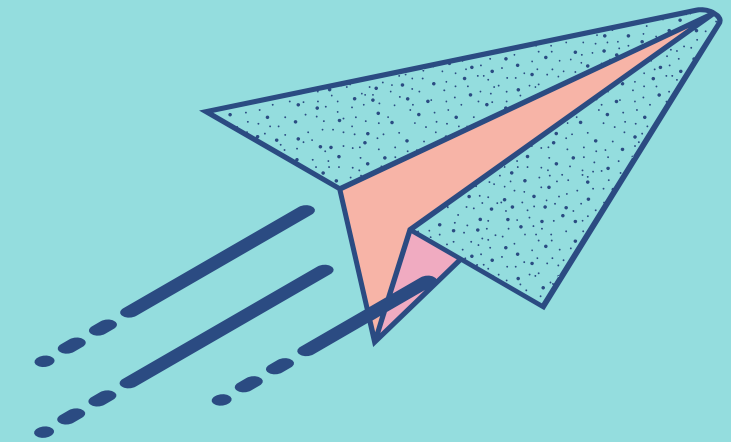
video streaming performance

■ startup time (msec) ■ avg latency (msec) ■ avg bitrate (msec) ■ dropped frames (MB/s)



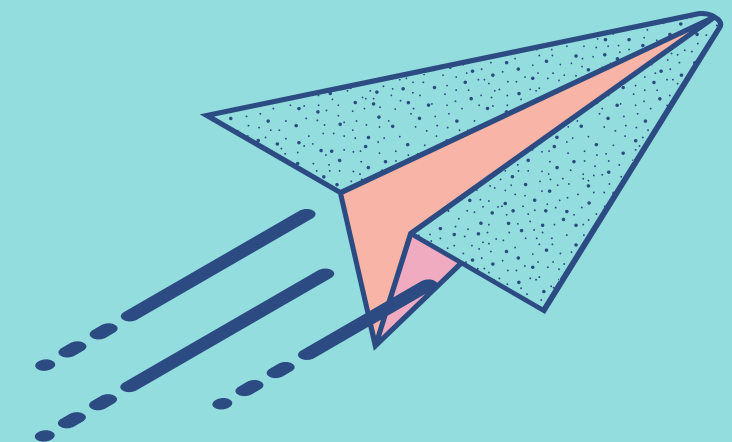
Conclusioni





Osservando i risultati si evince che il protocollo `http3+quic` è più lento di `http2`. Sicuramente si tratta di una nuova frontiera con grande potenzialità.

Infatti Quic Assegna a ogni connessione un ID univoco, quindi ricorda l'ultimo indirizzo da cui hai ricevuto un pacchetto per quella connessione, idealmente nel kernel. Ciò rende gli indirizzi IP completamente trasparenti alle applicazioni, proprio come lo sono gli indirizzi MAC



La Topologia

