



Universidade do Minho
Escola de Engenharia

Engenharia de Serviços em Rede

Streaming de áudio e vídeo a pedido e em tempo real

Relatório Trabalho Prático 2

Grupo **PL43**

MEI - 1º Ano - 1º Semestre

Trabalho realizado por:

PG53601 - Afonso Xavier Cardoso Marques

PG55093 - Pedro Alexandre da Silva Oliveira

PG55097 - Vasco Rafael Barroso Gonçalves Rito

Braga, 26 de novembro de 2024

Índice

1. Introdução	4
2. Arquitetura da Solução	4
2.1. Script	6
2.2. Servidores de Streaming	6
2.3. Points of Present (POP)	7
2.4. Nodo intermédio	7
2.5. Cliente	7
3. Especificação dos protocolos	8
4. Etapa 1 - Construção da Topologia Overlay e Underlay	8
4.1.	8
5. Etapa 2 - Streaming adaptativo sobre HTTP (MPEG-DASH)	9
6. Etapa 3 - Streaming RTP/RTCP unicast sobre UDP e multicast com anúncios SAP	10
6.1. Perguntas e Respostas	10
7. Conclusão	10

Lista de Figuras

Figura 1: Topologia escolhida	4
Figura 2: Arquitetura do Sistema	5
Figura 3: Script para correr Topologia	6

1. Introdução

Neste trabalho prático foi-nos proposto o desenvolvimento de um serviço Over the Top para a entrega de conteúdos de multimédia. Tendo como objetivo desenvolver e prototipar um serviço de entrega de vídeo em tempo real, recorrendo à utilização de *Points of Presence* (POP) e uma rede de *overlay*. NO FINAL DO TRABALHO COMPLETAR !!!!!

A topologia escolhida para testar este trabalho possui um servidor, três nodos de fronteira (POP), três nodos intermédios e quatro clientes.

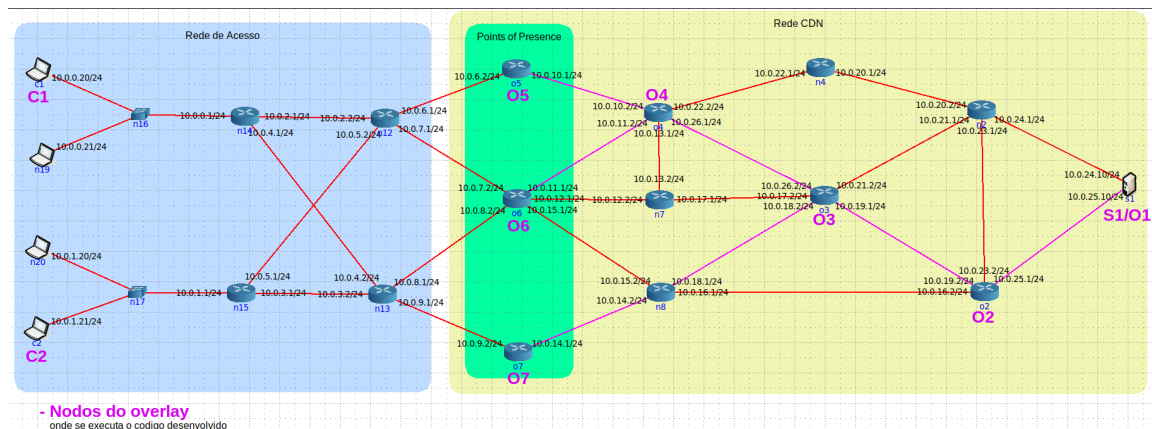


Figura 1: Topologia escolhida

2. Arquitetura da Solução

Neste trabalho é pretendido que se crie um sistema de entrega de multimédia em tempo real, partindo de um servidor de *streaming* para um conjunto de clientes. Para isso os nodos *Points of Presence*, ao qual nos iremos referir como POPs, irão estar na fronteira da rede de distribuição de conteúdo, *Content Distribution Network* ou CDN, e a rede de acesso, fazendo de ponto de contacto dos clientes com a CDN. Os POPs recebem o conteúdo via multicast e depois enviam-no via unicast para cada um dos clientes que pretendem aceder àquele conteúdo. A seguinte imagem mostra a arquitetura geral de um sistema destes.

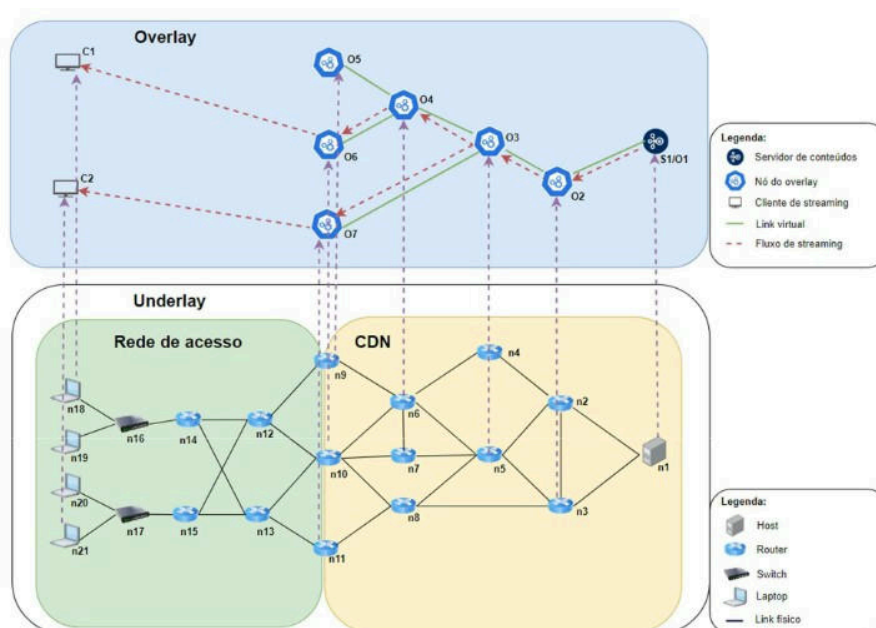
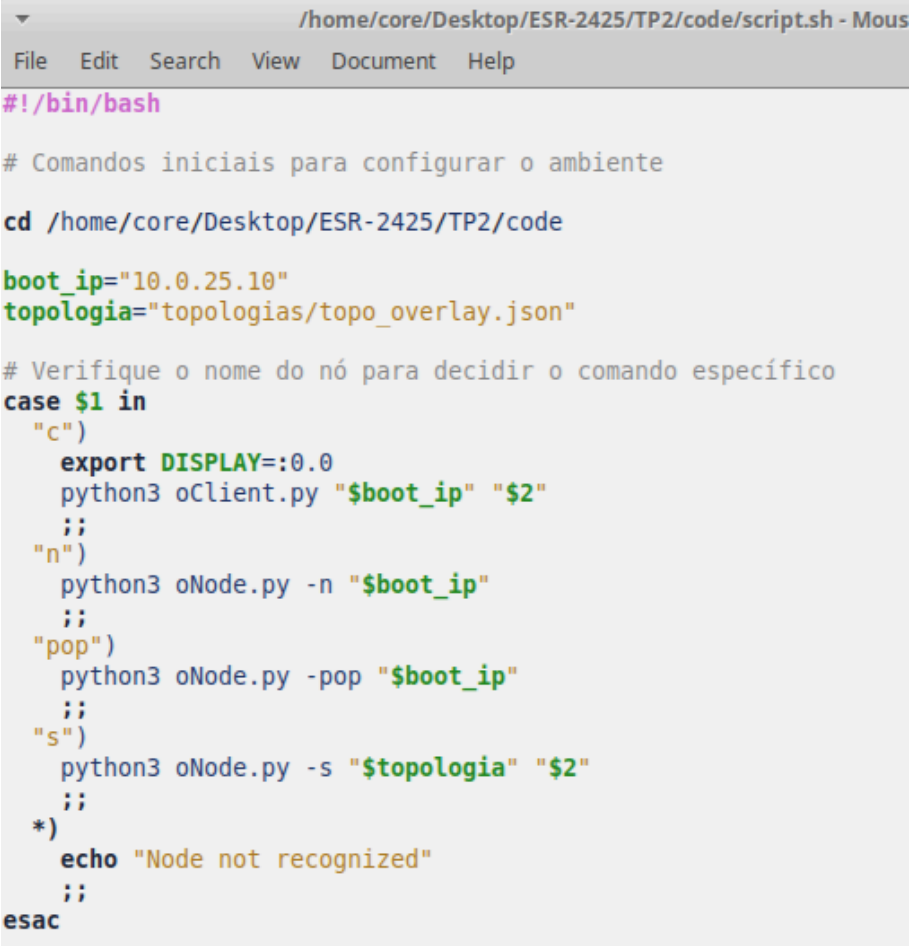


Figura 2: Arquitetura do Sistema

Para a implementação do trabalho prático utilizamos a linguagem de programação Python pois, além de possuir bibliotecas que facilitam o uso de sockets e threads, também é a linguagem com que o grupo se sente mais à vontade. Para a construção da rede de *overlay* decidimos utilizar um ficheiro de configuração em formato JSON, onde está presente a informação que cada nodo irá utilizar. Segue em baixo a explicação dos argumentos necessários e a estrutura dos ficheiros de configuração.

2.1. Script

De forma a tornar mais rápida e simples a execução da topologia decidimos desenvolver um pequeno *script* que será aplicado apenas nos nodos que atuam no serviço de *streaming*, isto é, os servidores, nodos intermédios, POPs e os clientes.



```
#!/bin/bash

# Comandos iniciais para configurar o ambiente

cd /home/core/Desktop/ESR-2425/TP2/code

boot_ip="10.0.25.10"
topologia="topologias/topo_overlay.json"

# Verifique o nome do nó para decidir o comando específico
case $1 in
    "c")
        export DISPLAY=:0.0
        python3 oClient.py "$boot_ip" "$2"
        ;;
    "n")
        python3 oNode.py -n "$boot_ip"
        ;;
    "pop")
        python3 oNode.py -pop "$boot_ip"
        ;;
    "s")
        python3 oNode.py -s "$topologia" "$2"
        ;;
    *)
        echo "Node not recognized"
        ;;
esac
```

Figura 3: Script para correr Topologia

2.2. Servidores de Streaming

Os servidores de *streaming* são responsáveis por enviar o conteúdo pela rede CDN até aos POPs. Estes servidores, de forma a serem inicializados, devem saber qual é o ficheiro JSON que contem a configuração da topologia e uma *flag* que define se o servidor será ou não o *bootstrapper*. O *bootstrapper* é o único nó que coordena a descoberta da rede e deve ser sempre iniciado em primeiro lugar. Em todas as topologias, deve apenas existir um único servidor *bootstrapper*. O seguinte comando permite iniciar o servidor:

```
python3 oNode.py -s "topologia.json" 1
```

- A *flag* ‘-s’ indica que o nó é do tipo servidor.
- O número ‘1’, no final, especifica que o servidor será inicializado como *bootstrapper*.
- Se for ‘0’, o servidor será inicializado como não *bootstrapper*, ficando apenas disponível para distribuir conteúdo.

2.3. Points of Present (POP)

Os POPs são responsáveis por retransmitir o conteúdo recebido pelos servidores para os clientes. Atuam na fronteira entre as duas redes, sendo os únicos nodos que mantêm ligação contínua com o cliente durante a transmissão do vídeo. A sua implementação permite escalabilidade e eficiência na transmissão, sendo assim capazes de lidar com vários clientes ligados ao mesmo tempo.

Para serem inicializados, os POPs necessitam de saber o endereço IP do servidor que atua como *bootstrapper*. Falando com o servidor neste endereço, os POPs conseguem descobrir e conectar-se aos seus vizinhos na topologia. Para inicializar um POP, utiliza-se o comando:

```
python3 oNode.py -pop "ip_bootstrapper"
```

- A *flag* ‘-pop’ indica que o nodo é um POP.
- “ip_bootstrapper” deve ser substituído pelo endereço IP do *bootstrapper*.

2.4. Nodo intermédio

Este tipo de nodo apenas trata de receber e encaminhar os pacotes relativos às diversas *streams* desde o servidor até aos POPs. Possui uma base de dados onde guarda todos os seus vizinhos, quem vai receber pacotes de uma determinada *stream* e toda a lógica de escolha de caminhos para construir o fluxo de transmissão. De forma a ser inicializado, apenas é necessário indicar a *flag* ‘-n’ e o endereço do *bootstrapper*.

```
python3 oNode.py -n "ip_bootstrapper"
```

2.5. Cliente

Os clientes são os nodos finais da rede que solicitam e consomem o conteúdo transmitido. Cada cliente conecta-se à rede falando primeiro com o servidor que atua como *bootstrapper* de forma a conseguir descobrir todos os servidores e POPs responsáveis por fornecer o conteúdo solicitado. Antes de executar o cliente, é necessário configurar a variável de ambiente `DISPLAY`, que define o ecrã onde será exibido o conteúdo. Para inicializar um cliente, utilizam-se os seguintes comandos:

```
export DISPLAY=:0.0  
python3 oClient.py "ip_bootstrapper" "movie_name"
```

- “ip_bootstrapper” deve ser substituído pelo IP do servidor *bootstrapper*.
- “movie_name” representa o nome do conteúdo que será transmitido.

3. Especificação dos protocolos

O protocolo aplicacional para o streaming foi o Real-time Transport Protocol (RTP), usámos como base a implementação que foi fornecida pela equipa docente. Oferecendo simplicidade e baixa latência, que são características fundamentais para transmissões em tempo real. A transmissão dos

4. Etapa 1 - Construção da Topologia Overlay e Underlay

4.1.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura.

5. Etapa 2 - Streaming adaptativo sobre HTTP (MPEG-DASH)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura.

6. Etapa 3 - Streaming RTP/RTCP unicast sobre UDP e multicast com anúncios SAP

6.1. Perguntas e Respostas

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura.

7. Conclusão

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura.