



Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia Informática

Dezembro, 2024

Relatório do Trabalho Prático de Engenharia de Serviços em Rede

Ricardo Miguel Queirós de Jesus, PG57898

Rui Madeira Pinto PG56010

Pedro Azevedo PG57897

Contents

1. Introdução	3
2. Arquitetura da solução	3
3. Especificação do(s) protocolo(s)	3
3.1 Formato das mensagens protocolares	4
4. Implementação	6
4.1 Servidor	6
4.2 Pontos de Presença	7
4.3 Pontos de Acesso	7
4.4 Cliente	7
5. Testes e resultados	7
5.1 Cenário normal	7
5.2 Dois clientes a pedir a mesma stream	8
6. Conclusões e trabalho futuro	8

1. Introdução

O presente relatório tem como objetivo demonstrar e explicar o percurso feito pelo grupo ao implementar a rede *Over The Top*(OTT). Uma rede OTT é uma rede que permite a distribuição de conteúdo de multimédia aos clientes por meio da internet de forma eficiente. Neste relatório, serão abordados diversos aspetos relacionados com a solução desenvolvida, incluindo a arquitetura proposta, a especificação dos protocolos adotados e os detalhes dos processos de implementação. Serão também discutidos as limitações encontradas, os testes efetuados e os respetivos resultados obtidos, culminando em conclusões relevantes e possíveis caminhos para trabalhos futuros.

2. Arquitetura da solução

Nesta secção, iremos apresentar a arquitetura proposta para o serviço de streaming de conteúdo multimédia, falando dos diferentes componentes que a constituem e as suas interações, e mostrando a topologia overlay proposta. A figura 1 ilustra a topologia de rede overlay aplicacional desenvolvida para o projeto.

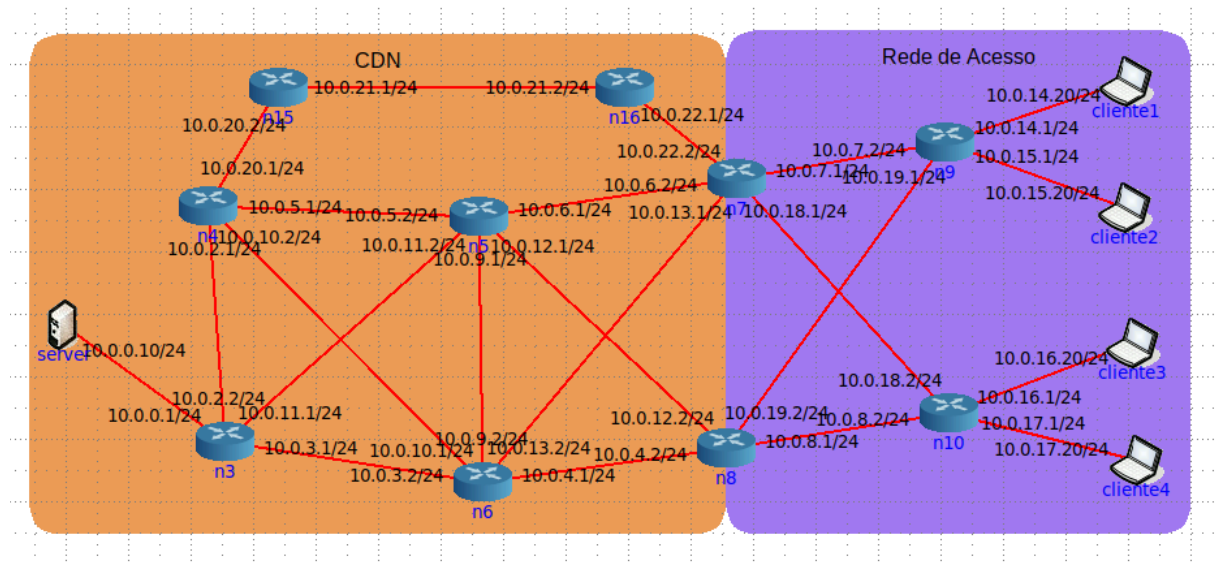


Figure 1: Topologia da Rede

- **Server:** É o responsável pela transmissão do conteúdo multimédia, vai também funcionar como o bootstraper da rede, sendo ele responsável pelo gerenciamento da rede, desde de saber quem está ou não ativo na rede como formar a árvore de transmissão, na nossa topologia vai ser o node 'server'.

- **Ponto de Acesso:** Os pontos de acesso são os nodes que fazem a conexão entre a content delivery network (CDN) e rede de acesso, na nossa topologia são os nodes 'n7' e 'n8'.

- **Ponto de Presença:** Os pontos de presença são os nodes responsáveis pelo encaminhamento da transmissão desde o server até os pontos de acesso, são todos os nodes entre estes dois elementos.

- **Cliente:** Os clientes representam o destinatário final do conteúdo multimédia transmitido na rede, estes fazem pedidos aos pontos de acesso pelas transmissões desejadas.

3. Especificação do(s) protocolo(s)

Para a implementação da rede, foi apenas utilizado o protocolo UDP, tanto na rede de acesso como na CDN. Nas mensagens trocadas entre o servidor, os diferentes nodes e os clientes da rede foram utilizadas diversos tipos de mensagem para pedidos e para o envio de *frames* do conteúdo transmitido pela rede.

3.1 Formato das mensagens protocolares

Na rede implementada, foram incorporados tipos distintos de interações possíveis:

- Mensagens de pedidos de streaming:

O cliente quando começa a execução, efetua um pedido de streaming, formando uma string com o formato **request|<stream_name>|<client_ip>**, que contém o nome da stream que o cliente quer assistir e o seu próprio ip e que será enviado, posteriormente, para o access point que tenha apresentado melhor média de *round trip time* com o cliente. Após enviar esta mensagem para o *Access Point*, o mesmo, envia diretamente uma mensagem no formato **start_stream|<stream_name>|<client_ip>** para o servidor. O servidor por sua vez, após receber este request do *Access Point*. O servidor vai então a um dicionário onde tem o melhor caminho para o *Access Point*, que obteve calculando previamente o algoritmo de *Dijkstra* para os nodes disponíveis na rede. Para cada um dos nodes presentes nesse melhor caminho, o servidor envia uma mensagem no formato **<stream_name>|<predecessor_ip>|<client_ip>** que contém o endereço de IP do nodo que o antecede, para efetuar o pedido da stream ao mesmo, o endereço de IP do cliente que pediu a *stream* e o nome da *stream* a ser transmitida na rede.

Após cada um destes nodes receber as informações do servidor, cada um deles, enviará uma mensagem no formato **stream_request|<stream_name>|<ip_client>**, para o antecessor e, de forma recursiva, os nodes irão enviar este **stream_request** para o nodo que o antecede, até este mesmo pedido chegar ao servidor. O servidor, quando recebe este **stream_request**, caso tenha a stream pedida, irá começar a transmitir os frames do video, pela rede, enviando ao nodo que lhe enviou este pedido os frames, sendo estes redirecionados de forma recursiva até chegar ao *Access Point* que recebeu o pedido do cliente, o este *Access Point* redirecionará os frames do vídeo para o cliente.

Em baixo encontram-se exemplos demonstrativos sobre o funcionamento desde o pedido de stream do cliente até à receção dos pacotes RTP com os frames no mesmo.

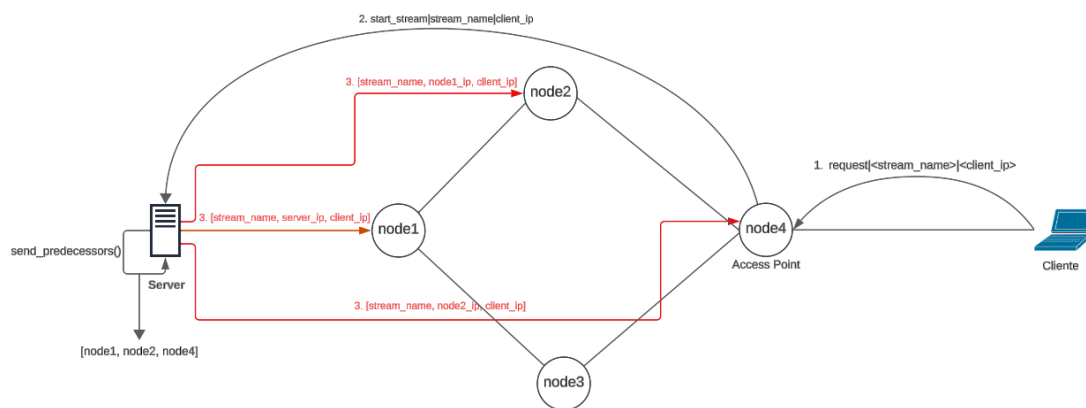


Figure 2: Pedido de stream do cliente e envio dos antecessores



Figure 3: Envio do stream request do access point até ao servidor

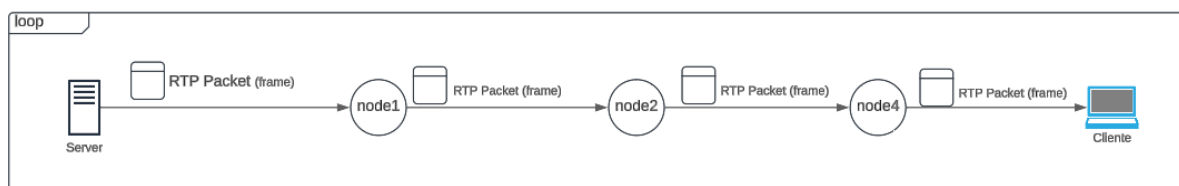


Figure 4: Envio dos pacotes RTP com os frames da stream

- Mensagens de pedido de cancelamento de streaming:

Quando um cliente quer deixar de visualizar uma stream este pode clicar para fechar a stream que vai enviar uma mensagem para o *Acess Point* que lhe estava a transmitir no formato **teardown|filename|client_ip** em que o *filename* é o nome da stream que quer deixar de ver e o *client_ip* o IP de quem quer deixar de ver. Quando o *Acess Point* recebe esta mensagem ele retira o cliente da sua lista de transmissão para a stream mencionada e verifica se essa lista está vazia, se for o caso então ele retira essa stream da lista e manda uma mensagem semelhante para o *presence point* a que estava a pedir essa stream, isto vai desencadear um efeito em cascata de forma que se chegar ao servidor que mais nenhum cliente esteja a ver a stream, então este deixa de a transmitir

- Estrutura do pacote RTP:

Os frames do video a ser transmitido são enviados no *payload* do pacote RTP, que contém no seu cabeçalho, várias informações necessárias para o bom funcionamento da nossa arquitetura. Para a estrutura do pacotes RTP foi reutilizado o conteúdo fornecido na bibliografia presente no enunciado deste trabalho prático.

Campo	Tamanho(bits)	Descrição
Version	2	Versão do protocolo RTP
Padding	1	Indica se há padding adicional no final do pacote“
Extension	1	Indica se há cabeçalho de extensão adicional
CSRC Count	4	Indica o número de identificadores CSRC incluídos no cabeçalho
Marker	1	Marcador para eventos especiais, como o início de um frame
Payload Type	7	Tipo de carga útil
Sequence Number	16	Número de sequência do pacote RTP
Timestamp	32	Timestamp do pacote RTP
SSRC	32	Identificador único da fonte de sincronização
Client IP	32	Endereço IP do cliente de destino do pacote
Source IP	32	Endereço de IP da origem do pacote
Is Movie Request	8	Indica se o frame é um pedido de stream
File Found	8	Indica se o ficheiro foi encontrado ou não
Session Number	32	Número de sessão associado ao cliente
Filename	512(64 bytes)	Nome do ficheiro solicitado
Payload	Variável	Utilizado para enviar os frames do conteúdo

4. Implementação

4.1 Servidor

Como mencionado anteriormente o servidor vai assumir o role de bootstraper, sendo o ponto central da rede. Tem um dicionário onde possui todos os nodes da topologia no CDN com todos os seus possíveis vizinhos. Sempre que um novo ponto de acesso se quer juntar à rede este têm de mandar um pedido ao servidor, que o vai adicionar à lista *nodes_network* que representa os pontos de acesso que estão de momento activos. Quando um ponto de acesso se conecta à rede o servidor vai correr um *dijkstra* nos nodes activos na rede para descobrir o caminho mais curto para o ponto de acesso, e depois vai guardar este caminho para quando o ponto de acesso requisitar uma stream o servidor informar os nodes que vão fazer parte da árvore de transmissão, após pelo menos um ponto de acesso se conectar à rede, sempre que um novo ponto de presença se conecta o servidor corre o *dijkstra* para ver se um caminho curto surgiu, caso seja o caso então ele atualiza o seu *path*.

4.2 Pontos de Presença

Os pontos de acesso ao se conectarem à rede precisam de informar o servidor da sua presença. Quando recebem um pedido de transmissão de stream de outro node, estes vão fazer verificar se por acaso já estão a pedir essa stream para apenas concatenarem o ip do node que fez o novo request, se não for o caso, vão mandar um *request* da stream ao seu predecessor que foi dado pelo servidor. Quando recebem um *RTP packet* na sua porta 9090 eles vão verificar a que stream aquele pacote pretence e vão encaminhar aos nodes que estão na sua lista de *stream_requests* para essa stream.

4.3 Pontos de Acesso

O ponto de acesso que servem como porta entre os clientes e a rede de transmissão. Ao um novo cliente pedir uma transmissão que ainda não está a ser transmitida, então ele faz o pedido ao server para este começar o processo de transmissão requisitada para esse ponto de de acesso.

4.4 Cliente

O cliente serve para os clientes se conectarem à rede e efetuarem pedido de stream e, posteriormente poderem visualizar a mesma numa janela criada para esse efeito. O cliente poderá também pedir para terminar a transmissão da *stream* clicando no botão de fechar presente no canto superior direito. Ele fazem isso conectando-se ao ponto de acesso que lhe é mais favorável, dando pings aos pontos de acesso existentes e compara a a média dos resultados de ambos.

5. Testes e resultados

5.1 Cenário normal

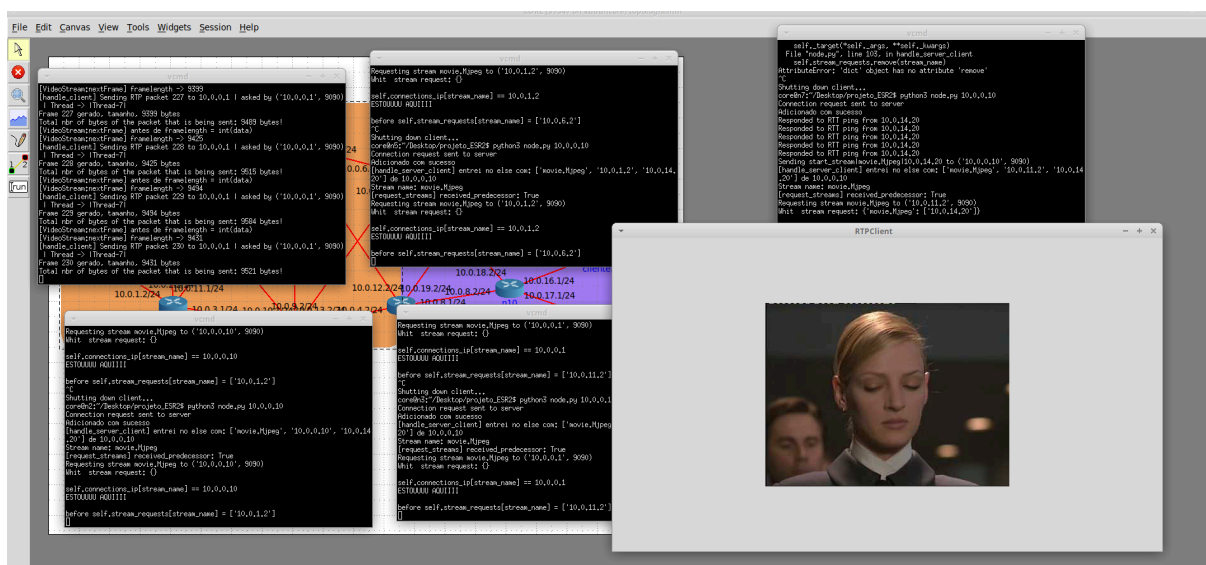


Figure 5: Cenário de teste com vídeo fornecido

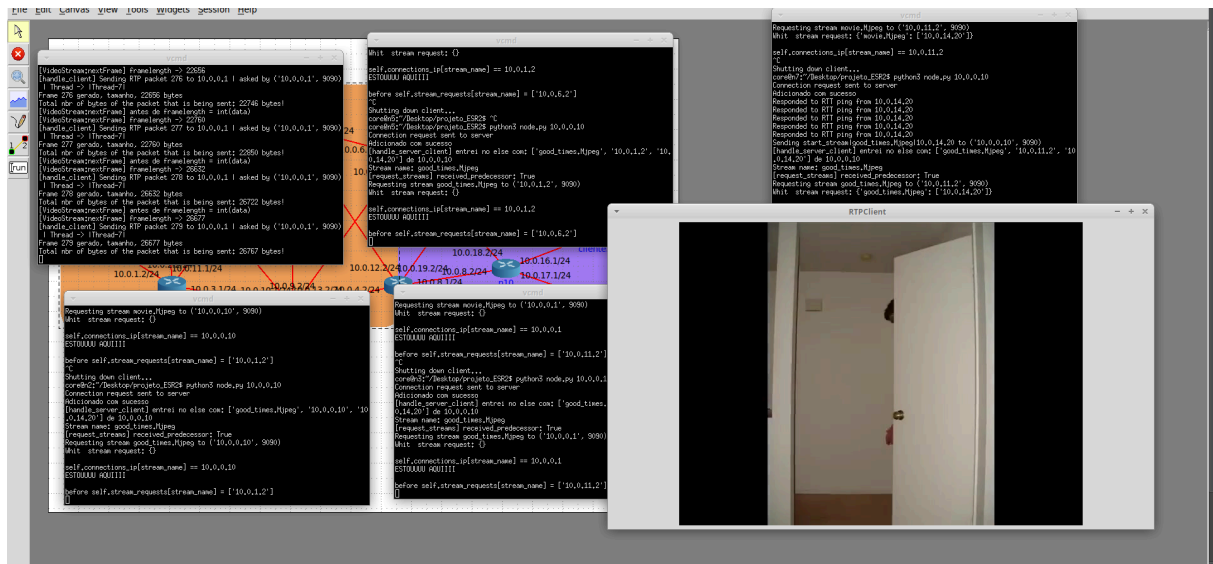


Figure 6: Cenário de teste com segundo vídeo

5.2 Dois clientes a pedir a mesma stream

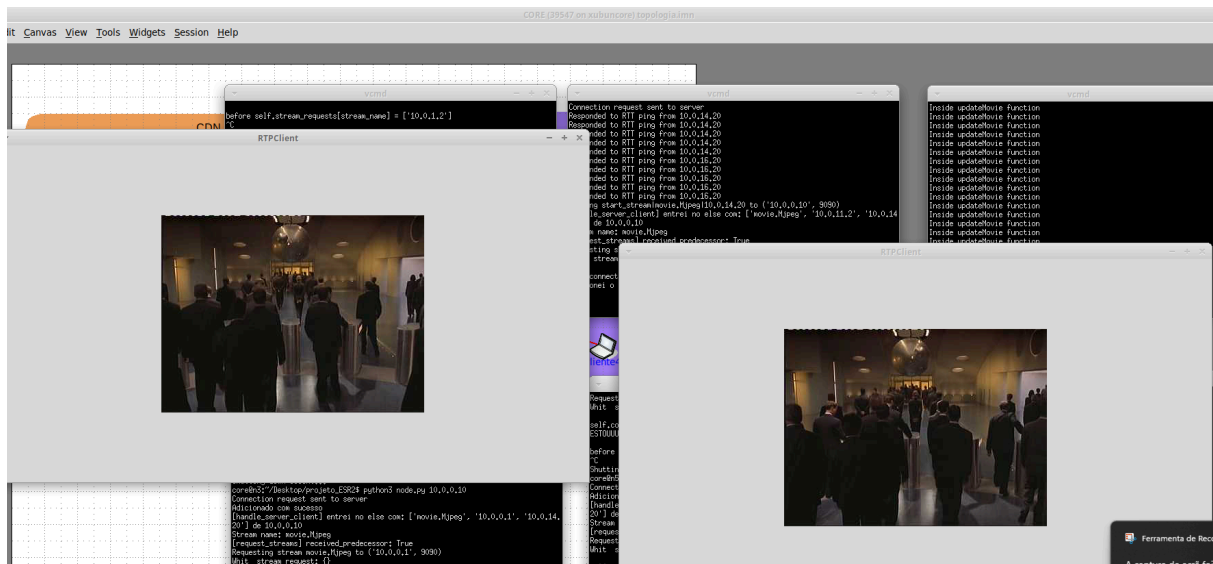


Figure 7: Cenário de teste com dois clientes a pedir a mesma stream

6. Conclusões e trabalho futuro

Neste trabalho, foi desenvolvido um sistema de transmissão de conteúdos multimédia em uma rede Over The Top (OTT), utilizando uma arquitetura distribuída e baseada no protocolo UDP. Durante a implementação, foram superados diversos desafios relacionados à comunicação eficiente entre os componentes da rede, à gestão dinâmica da topologia e ao encaminhamento correto dos pacotes de streaming.

Para trabalho futuro ficaria a implementação de monotirização de rede, onde os pontos de acesso teriam uma funcionalidade de enviar pacotes teste para os seus vizinhos e informar o servidor quais são os tempos a que recebe resposta dos mesmo, de forma a que o servidor escolha a árvore de transmissão com menos latencia.