

Serviço Over the Top para entrega de multimédia

Bernardo Saraiva^[pg50259], Daniel Azevedo^[pg50311], and José Gonçalves^[pg50519]

Universidade do Minho, R. da Universidade, 4710-057 Braga, Portugal
{pg50259,pg50311,pg50519}@alunos.uminho.pt

Resumo O presente trabalho prático da unidade curricular de Engenharia de Serviços em Rede, foi realizado com o intuito de criar um serviço para entrega de vídeo e áudio em tempo real a partir de um servidor para diversos clientes. Inicialmente, foi criada a topologia *overlay* baseada na rede *underlay*, seguidamente foram estabelecidas conexões entre todos os nós, por fim são estabelecidas rotas para a transmissão de dados. Depois de concluídas estas tarefas é possível realizar stream entre o(s) servidor(es) e os seus clientes.

Keywords: Rede overlay · RTP · Streaming · Server-Client

1 Introdução

Ao longo do último meio século de vida da Internet, observou-se uma mudança de paradigma. A comunicação extremo-a-extremo deixou de ser tão utilizada, dando lugar ao consumo de conteúdos a todo o instante e a qualquer hora, e muitas vezes em tempo real. Este novo padrão de uso coloca grandes desafios à infraestrutura IP. Apesar desta não ter sido originalmente desenhada com esse requisito, tem sido possível resolver a entrega de conteúdos usando redes sofisticadas de entrega de conteúdos (CDNs) e serviços desenhados sobre a camada aplicacional, os Over the Top (OTT).

Um CDN permite a transferência rápida de recursos necessários para carregar conteúdos da Internet, incluindo páginas HTML, imagens e vídeos. Hoje a maior parte do tráfego da Web é atendida por meio de CDNs, incluindo o tráfego de sites conhecidos como Facebook, Netflix e Amazon.

Os serviços de multimédia OTT, usam uma rede overlay aplicacional para contornar os problemas de congestão e limitação.

Neste trabalho, pretende-se conceber e prototipar um destes serviços, que promova a eficiência e a otimização de recursos para existir uma melhor experiência do utilizador.

2 Arquitetura da solução

O principal objetivo deste trabalho passa pela construção da rede overlay, a partir da qual um servidor pode realizar stream de vídeos para vários clientes. Para tal, foi usado *python* como linguagem de programação para o desenvolvimento da solução.

Visto que a implementação da solução requer streaming de vídeo em tempo real, considerou-se conveniente usar UDP como protocolo de transporte para realizar a troca de pacotes de vídeo dada a sua rapidez no envio de pacotes relativamente ao TCP, mesmo sabendo que a confiabilidade não é a mesma, mas que se torna a melhor opção, uma vez que se está a trabalhar com streaming em tempo real e a perda de pacotes não é tão preocupante, quando ocorre um trade-off significativo com o desempenho da transmissão. Relativamente à troca de mensagens de controlo utilizou-se TCP, dado que este protocolo de transporte é orientado à conexão, garantindo a entrega e a integridade dos dados pacotes.

No serviço desenvolvido existem três entidades principais, tendo cada uma delas tarefas importantes para o funcionamento de toda a solução:

- **Servidor** - Responsável por alojar vídeos e efetuar streaming.
- **Nodos** - Entidades intermédias que têm como função encaminhar mensagens do servidor para o cliente ou do cliente para o servidor.
- **Cliente** - Consumidores do serviço de streaming.

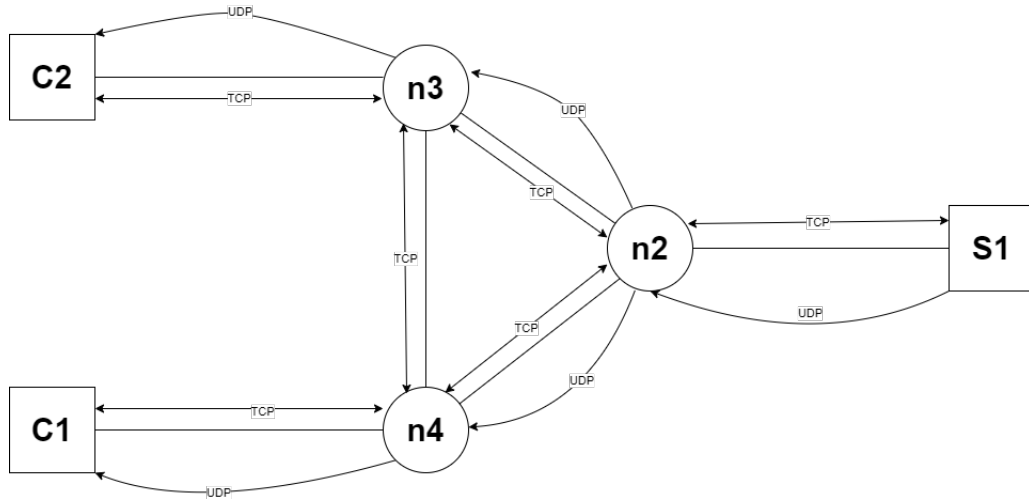


Figura 1. Arquitetura da aplicação

3 Especificação de protocolos

Com a finalidade de transmitir conteúdos de streaming foi usado RTP (Real-time Transport Protocol), um protocolo de transporte vocacionado especificamente para streaming e transporte de conteúdo em tempo real. No seu funcionamento, este usa UDP como protocolo da camada de transporte, sendo tolerável a perda de pacotes, no entanto, é sensível aos atrasos na entrega de pacotes afetando a sequência com que são dispostos e não garante a qualidade de serviço (QoS).

O protocolo RTP é usado em conjunto com o protocolo RTSP (Real Time Streaming Protocol). Este usa RTP para transporte de dados, sendo RTSP usado para controlar a transmissão de multimédia.

3.1 Formato das mensagens protocolares

Para que o serviço funcione corretamente, é necessário trocar algumas mensagens entre as diferentes entidades, existindo os seguintes tipos de mensagem:

- **REQ_NEIGHBOURS** - Mensagens usada para pedir a lista de vizinhos.
- **STATUS** - Mensagem enviada para monitorizar a rede overlay.
- **DATA** - Mensagens responsáveis pelo envio de dados. No seu interior contém os dados relativo aos pacotes RTP, estes são reencaminhados pelos nós intermédios até chegar aos clientes.
- **Stream_Request** - Mensagens enviada para pedir streaming de um vídeo.
- **SEND_NEIGHBOURS** - Mensagem usada para enviar a lista de vizinhos de um nodo.
- **NAK** - Mensagem usada quando o vizinho não consegue encontrar a stream pedida.

4 Implementação

Para a implementação do serviço OTT de entrega de multimédia foram idealizadas 4 etapas distintas.

Ao longo deste capítulo, serão apresentados diagramas tendo como objetivo auxiliar na interpretação da solução adotada.

4.1 Construção da Topologia Overlay

No relatório são utilizados os termos overlay e underlay, sendo estes termos referentes à topologia com que se trabalha. Uma rede underlay é uma infraestrutura física sobre a qual as overlays são construídas e permitem a comunicação entre os equipamentos e os servidores.

A rede overlay corresponde a uma virtualização construída sobre a underlay, desenhada para aumentar a escalabilidade e a eficiência na comunicação e transporte. Estes nodos são instanciados diretamente em alguns dos nodos da rede underlay, pois na verdade são aplicações em execução em determinados sistemas físicos previamente selecionados. De salientar que a rede overlay terá muito menos nodos que a underlay, evitando um mapeamento desnecessário de um para um.

Para um nó da rede overlay começar a sua execução, é necessário ter um nó conhecido (bootstrapper), com o objetivo de se dar a conhecer e de obter conhecimento dos nós com quem se pode ligar.

O ponto de partida para a construção da rede overlay é a leitura de um ficheiro JSON, este contém informação acerca dos nodos da topologia assim como os seus vizinhos de cada nó (neighbours) e das interfaces associadas (names). Em seguida, é efetuada conexão com todos nodos, sendo que cada um deles irá pedir, ao bootstrapper, a lista dos seus vizinhos.

O serviço apenas irá avançar para a próxima etapa quando todos os nós da rede overlay estiverem conectados.

A Figura 3 representa um diagrama explicativo das interações que ocorrem no processo em que um nodo se conecta ao servidor com o objetivo de obter as informações relativas à sua vizinhança.

```

1 {
2     "s1": {
3         "neighbours": ["10.0.2.1"],
4         "names": ["10.0.2.10"]
5     },
6     "n1": {
7         "neighbours": ["10.0.2.10s"],
8         "names": ["10.0.2.1", "10.0.1.1", "10.0.0.1"]
9     }
10 }
```

Listing 1.1. Exemplo de um ficheiro de configuração

4.2 Monitorização da Rede Overlay

Na estratégia adotada para obter conhecimento das condições de entrega na rede overlay, o servidor começa por enviar mensagens de controlo (métricas) incluindo a sua identificação, instante de envio, número de saltos efetuados na rede e a lista de nós visitados. Esta lista de nós visitados é importante para evitar repetições em ciclo. Estas mensagens de controlo são enviadas de 15 em 15 segundos, sendo este o intervalo de tempo que se considerou razoável para o efeito.

Ao receber as mensagens vindas do servidor, cada nó é responsável por fazer flooding controlado para todos os seus vizinhos. Ao reencaminhar a mensagem, os nós devem atualizar as métricas necessárias para posteriormente ser possível escolher as melhores rotas. Foram usadas duas métricas para a escolha de rotas, sendo elas o menor atraso (delay) e o número de saltos, sendo que esta última só é usada quando o atraso entre duas rotas é semelhante, ou seja a variação temporal é inferior a 10%.

Na Figura 4 é possível observar um diagrama explicativo do que foi apresentado nesta subsecção. De reforçar que esta monitorização ocorre periodicamente (a cada 15 segundos) e permite que os nodos estejam atualizados face às melhores métricas.

4.3 Construção de Rotas para a Entrega de Dados

Depois de criada toda a rede overlay, é possível avançar para a construção das diversas rotas, necessárias para a entrega de dados aos consumidores da stream.

A construção de rotas para a entrega de dados começa no momento em que um cliente pretende visualizar uma stream. O cliente realizará o pedido ao nodo vizinho, sendo que este irá verificar se tem a stream, e em caso afirmativo irá transmitir ao cliente.

No caso em que um nodo não tem a stream, este irá verificar a existência da mesma em todos os seus vizinhos, que por sua vez irão verificar se tem a stream e, no caso de não terem, irão também verificar a sua vizinhança e assim sucessivamente. Quando é encontrado um nodo que se encontra previamente a receber os pacotes associados à stream, irão ser atualizadas as métricas correspondentes a essa stream e difundidas aos vizinhos interessados (este processo ocorre também em recursão), de modo a que seja possível utilizar a melhor rota no processo de obtenção dos pacotes de vídeo. Caso não existam nodos, na vizinhança, previamente a processar pacotes da stream requisitada esse pedido será remetido recursivamente através dos nodos com melhores métricas de acesso ao servidor.

No final da construção de rotas todos os nodos intervenientes no processo sabem exatamente a que nodo devem encaminhar o pedido para transmissão da stream para que esta seja difundida com o menor delay possível, e de forma a diminuir a quantidade de fluxos.

4.4 Serviço de Streaming

Para a realização do serviço de streaming, o processo é composto por toda a interação desde o cliente até ao servidor, para além dos nodos intermédios. Deste modo, o servidor encontra-se à escuta de pedidos de streaming por parte do cliente, quando lhe é efetuado um pedido, este envia o vídeo frame a frame através de pacotes RTP para um o nodo vizinho.

Uma das funcionalidades presentes no serviço de streaming desenvolvido foi a possibilidade de, simultaneamente, ter mais que um servidor com a stream. Assim sendo, esta funcionalidade aumenta a viabilidade da aplicação, no sentido em que caso seja detetada uma falha em um dos servidores (com ajuda das métricas), é possível assegurar a transmissão aos nodos por outro servidor existente.

O diagrama presente na Figura 5 ilustra o que é descrito nas secções 4.3 e 4.4, começando pelo pedido de um cliente, seguindo-se a escolha dos elementos constituintes para a transmissão e, finalmente, a receção da stream por parte do mesmo.

5 Testes e resultados

Nesta secção serão apresentados resultados obtidos em testes efetuados numa topologia do core. Os testes apresentados têm como objetivo verificar a correção e performance das diversas funcionalidades implementadas. Na figura 2, encontra-se a topologia underlay que será usada para realizar todos os testes. Nesta topologia consideramos dois servidores, cinco clientes e cinco routers que irão pertencer à topologia overlay (O2, O3, O5, O6, O7).

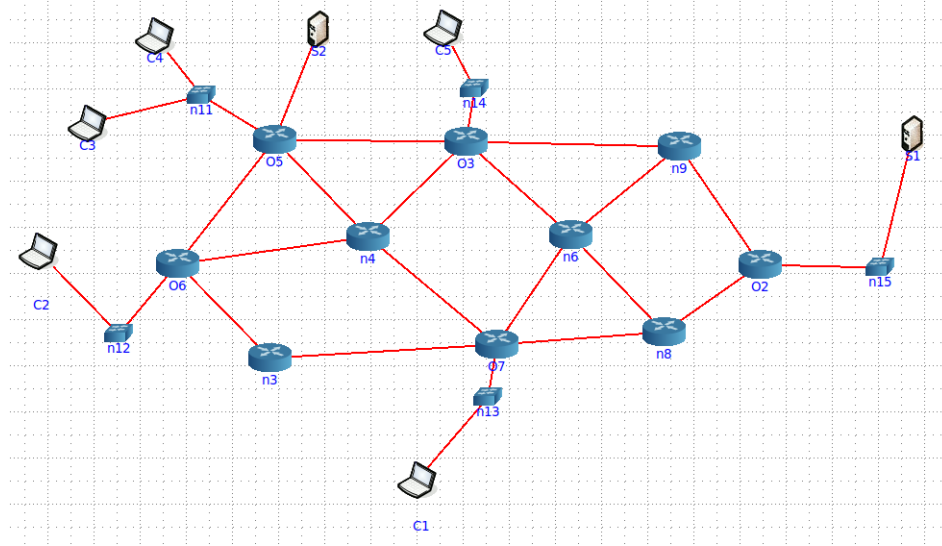


Figura 2. Rede underlay de teste

5.1 Cenários de teste

Primeiramente foram realizados testes tendo apenas o servidor 1 ativo.

Cliente C5 pede a stream Sendo este o cenário o mais simples, ocorrendo quando existe só um cliente a pedir a stream, neste caso será o C5. A stream será enviada, percorrendo os seguintes nodos da rede overlay: O2 e O3. Isto acontece pois esta é a melhor rota, ou seja, a que contém menos delay.

Cliente C4 e C2 pedem a stream Depois de todos os nodos da rede overlay estarem conectados, o cliente C4 faz um pedido de stream, este irá receber através da seguinte rota: O2, O3 e O5. Posteriormente o cliente C2 realiza um pedido de stream, sendo que o nodo da rede overlay, mais próximo que contenha fluxo de dados é o O5, este pedido será reencaminhado até ele, que irá enviar RTP packets para o cliente 2.

Cliente C1 e C2 pedem a stream, existindo delay na ligação Neste cenário de teste foi acrescentado delay na ligação do router n3-O7 e também na ligação o7-n4. Inicialmente o cliente 1 realiza o pedido de stream, esta stream vai passar por os seguintes nós da rede overlay: O2, O7. Posteriormente o cliente 2 faz o mesmo pedido, no entanto ele recebe a stream pela seguinte rota: O2, O3, O5 e O6. Isto deve-se ao facto de a rota escolhida é aquela que contém o menor atraso na entrega de pacotes.

Cliente C5 pede a stream, existindo delay na ligação Para a realização deste teste, foi adicionado 10 ms de delay entre a conexão do router O3-n9 e na ligação O3-n6 e também foi inicializado o servidor 2. Ao ser adicionado este delay o cliente 5 irá receber a stream do servidor 2 visto que a métrica principal para essa escolha é o tempo de atraso.

6 Conclusões e trabalho futuro

Dado por concluída a realização do presente trabalho prático, reconhece-se a sua importância para aprofundar os conhecimentos relativamente ao funcionamento de protocolos de streaming de vídeo, comunicação entre diferentes clientes e servidores e também transmissão de dados usando RTP.

Realça-se como aspeto positivo o facto de terem sido implementados todos os requisitos do serviço, atingindo os principais objetivos traçados pela equipa docente.

Futuramente, poderiam ser implementadas funcionalidades extras tais o desenvolvimento de um mecanismo de autenticação, por parte dos cliente, permitindo assim existir maior segurança dentro da rede.

7 Anexos

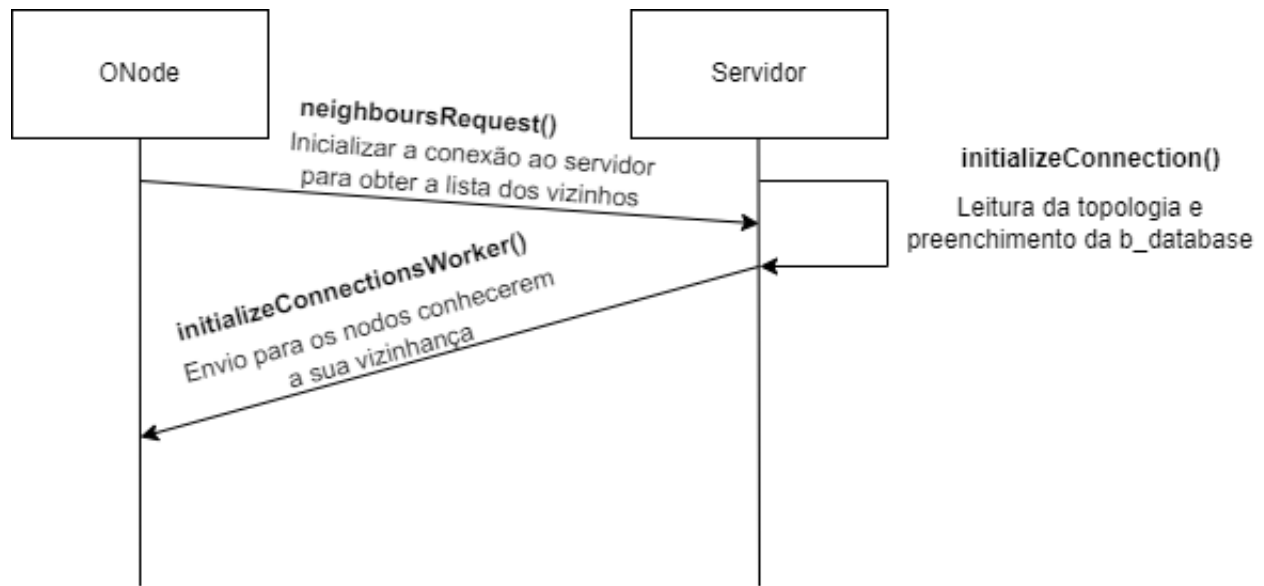


Figura 3. Diagrama explicativo da conexão entre nodo e servidor, em que o nodo fica a conhecer a vizinhança

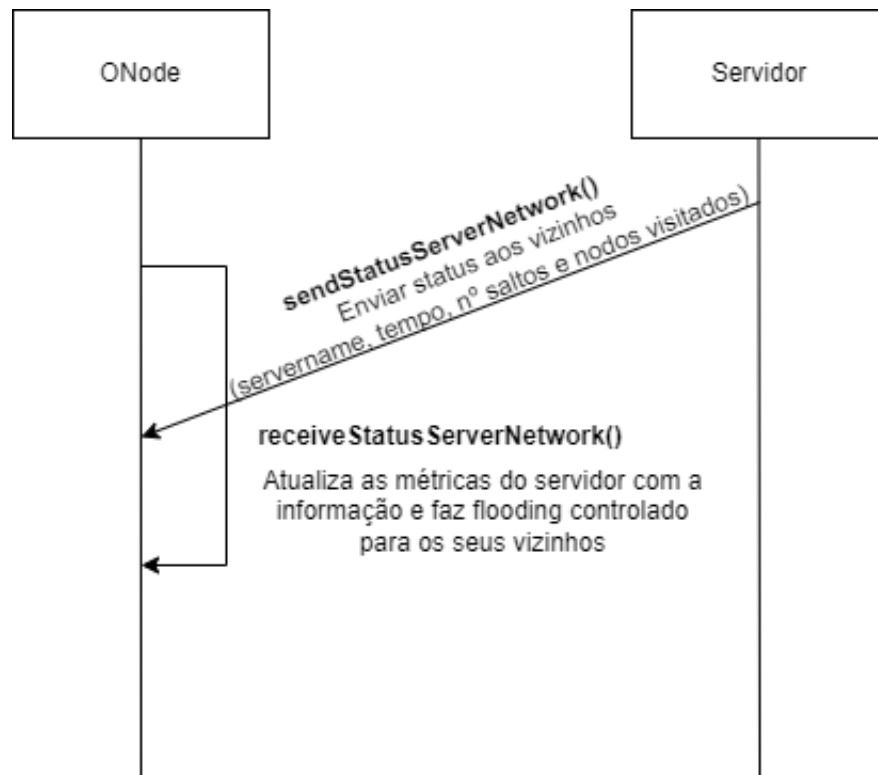


Figura 4. Diagrama explicativo da atualização de métricas pelos nodos

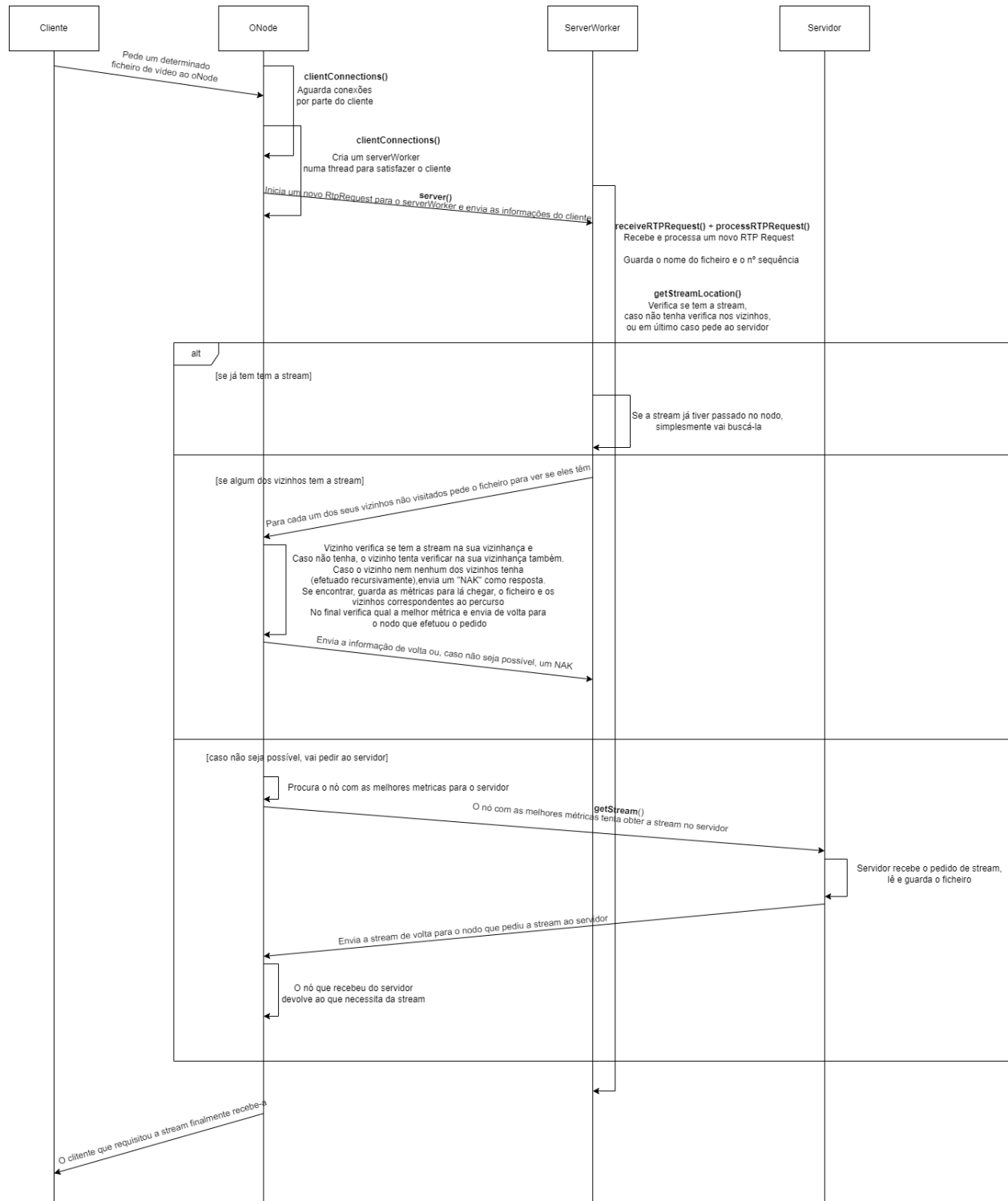


Figura 5. Diagrama explicativo do processo desde o pedido de stream por parte de um cliente até à recepção da mesma