

Projeto Server RTP/RTSP

Jhonatas Santana dos Anjos, Luiz Sacramento²

Departamento de Ciências Exatas e da Terra
Universidade do Estado da Bahia (UNEB) – Salvador, BA – Brasil

jhsantana11@hotmail.com

sacramento.tecnico@gmail.com

Abstract. *Fast of all, this project is about construction of an RTP/RSTP with the function of creating video frames' server. Then, there new protocols as UDP and also a security technology called Quality of Service. Like that, it comes about a most complex project than a simple HTTP Server.*

Resumo. *O atual objetivo do trabalho é a construção de um servidor RTP/RSTP com a função de criar quadros de vídeos e armazená-los. Atrélado a isso, um novo protocolo foi adicionado à esse estudo como o UDP e também uma tecnologia de segurança denominada QoS.*

1. Informações gerais

“The Real-Time Streaming Protocol (RTSP) is a tried-and-true video technology. It’s used to control audio/video transmission between two endpoints and facilitate the transportation of low-latency streaming content across the internet.

Along with the Real-Time Messaging Protocol (RTMP), RTSP once dominated the streaming landscape. That’s no longer the case today. Yet, RTSP remains standard in many surveillance and closed-circuit television (CCTV) architectures. The reason for this is simple: it’s still the protocol of choice for IP cameras.”(Ruether, 2021) A partir desse protocolo, supõe-se que há o controle do transporte do conteúdo de baixa latência pela internet e junto a esse protocolo muitos outros também participam do processo de troca de pacotes, como o TCP, IP e etc... Esses trabalham para que toda essa estrutura funcione. “Real-time Transport Protocol (RTP) is a network standard designed for transmitting audio or video data that is optimized for consistent delivery of live data. It is used in internet telephony, Voice over IP and video telecommunication. It can be used for one-on-one calls (unicast) or in one-to-many conferences (multicast)”(Wright, 2021) O RTP diferente do RTSP, por sua vez está mais voltado para a telefonia IP e videoconferência. Assim, cada tipo de tecnologia tem sua importância para determinada solução específica.

2. Primeira página

Esse sistema é muito utilizado em IP-Camera, pois os serviços de vídeo Streaming não é uma solução apenas para entretenimento(Netflix, Amazon, Youtube), mas também para a segurança, como mencionado. Além disso, há a possibilidade de fazer tal implementação com linguagens de programação modernas a exemplo do Java e Python. Outra inovação pouco citada é a telefonia interativa, via internet(Telefonia IP). Entretanto, há problemas com atrasos e perda de dados pela rede. Isso, não necessariamente é uma má implementação, mas sim uma limitação tecnológica.

3. Segurança QoS

“Quality of Service (QoS) is a set of technologies that work on a network to guarantee its ability to dependably run high-priority applications and traffic under limited network capacity. QoS technologies accomplish this by providing differentiated handling and capacity allocation to specific flows in network traffic. This enables the network administrator to assign the order in which packets are handled, and the amount of bandwidth afforded to that application or traffic flow.”(Palo Alto Networks, 2021)

É uma tecnologia que tenta priorizar hierarquicamente o tráfego e aplicações, assim como gerenciar a rede de modo geral. Um artifício que facilita muito a troca de pacotes. Algo que a primeira vista parece muito técnico, mas é essencial para garantir a segurança e a integridade dos dados, ou seja, que será entregue o que se espera e da melhor forma possível, evitando atrasos e perdas é um gerenciamento inteligente de fluxo de dados/informações, por assim dizer. Por isso, que muito se refere a essa tecnologia com a denominação qualidade de serviço.

4. TCP/UDP/IP

São protocolos da camada de transporte, sem eles a maneira que se conhece a internet hoje, seria impossível de acontecer. Para que toda essa cadeia de eventos ganhe forma precisa-se seguir um modelo, o qual há um conjunto de regras pré-estabelecido.

UDP (User Datagram Protocol), não garante a entrega de pacotes, por isso, é julgado como não confiável para aplicações que demandam uma estrutura segura, a pesar de sua alta velocidade. Já o *TCP(Transmission Control Protocol)*, é o contrário do *UDP*, pois possui o *handshake(aperto de mãos)*, sendo a verificação se os pacotes estão fazendo o seu curso natural. E finalmente o *IP(Internet Protocol)*, diferente dos outros está na camada de Rede, tenho como uma de suas funções identificar um dispositivo, uma rede, fazer o roteamento(escolher um caminho sobre o qual os dados serão enviados), cuida também da confiabilidade dos pacotes, dita como o Gateway e os Hosts processam os pacotes e até mesmo o comportamento das mensagens de erro.

5 – TIMESTAMPING

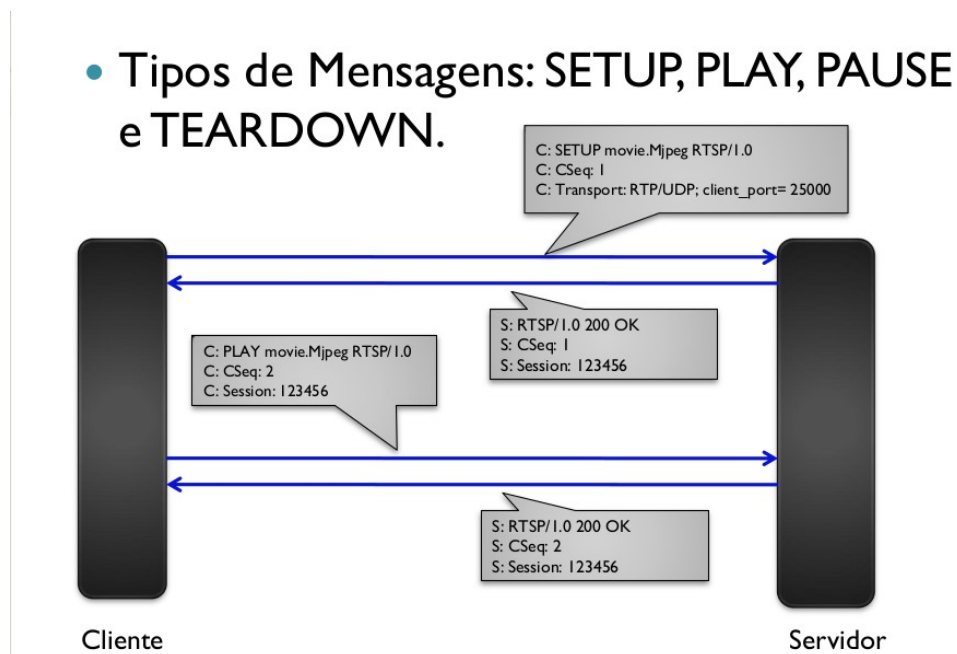
“a Internet é uma rede de transmissão de pacotes que são encaminhados independentemente através de redes compartilhadas, não garantindo a chegada ao destino na mesma ordem que foram enviados. Porém, aplicações multimídia requerem apropriada temporização na transmissão e exibição dos dados. O RTP fornece registro do tempo (timestamping), numeração seqüencial e outros mecanismos para tratar dessas dificuldade” (TSCHÖKE, 2001)

Ou seja, o timestamping é uma forma de que os possíveis atrasos nas transmissões online não seja perceptíveis aos humanos, não é uma tarefa tão fácil, já que os dados não são enviados de maneira organizada, mas sim difusa, mesmo em uma transmissão ao vivo, é dessa maneira que ocorrem os serviços via streaming.

6.1.1 Formato de mensagem RTSP

Figura 1. Exemplo de mensagens RTSP

O dia diagrama ilustra a troca de mensagem cliente/servidor e os diferentes tipos de mensagens.



Fonte: Criação Edson Moreira; Roberto Sadao, 2021

Figura 2. Cabeçalho RTP

Demonstração do datagrama RTP, com informações técnicas e úteis para entender as limitações e capacidades da tecnologia durante a troca de pacotes.

7

Cabeçalho RTP

Tipo de carga útil	Número de seqüência	Marca de tempo	Identificador de sincronização da fonte	Campos variados
--------------------	---------------------	----------------	---	-----------------

Cabeçalho RTP

Tipo de Carga (7 bits): Usado para indicar o tipo de codificação que está sendo usado no momento. Se um transmissor muda o tipo de codificação durante uma conferência, o transmissor informa o receptor através deste campo de tipo de carga.

- Tipo de carga 0: PCM mu-law, 64 kbps
- Tipo de carga 3, GSM, 13 kbps
- Tipo de carga 7, LPC, 2.4 kbps
- Tipo de carga 26, Motion JPEG
- Tipo de carga 31. H.261
- Tipo de carga 33, MPEG2 vídeo

Número de seqüência (16 bits): O número de seqüência é incrementado de um a cada pacote RTP enviado; pode ser usado para detectar perdas de pacotes e para recuperar a seqüência de pacotes.

© 2005 by Pearson Education

7 - 49

PEARSON
Addison
Wesley

Fonte: Criação Kurose; Ross, 2005

Figura 3. Código fonte IP Camera

O código fonte mostra um script em Python para renderizar vídeos de uma camera em uma página HTML, através do protocolo RTSP.

```
from flask import Flask, render_template, Response
import cv2

app = Flask(__name__)

camera = cv2.VideoCapture('rtsp://freja.hiof.no:1935/rtplive/_definst/_hessdalen03.stream') # use 0 for web camera
# for cctv camera use rtsp://username:password@ip_address:554/user=username_password='password'_channel=channel_number_stream=0.sdp' instead of camera
# for local webcam use cv2.VideoCapture(0)

def gen_frames(): # generate frame by frame from camera
    while True:
        # Capture frame-by-frame
        success, frame = camera.read() # read the camera frame
        if not success:
            break
        else:
            ret, buffer = cv2.imencode('.jpg', frame)
            frame = buffer.tobytes()
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n') # concat frame one by one and show result

@app.route('/video_feed')
def video_feed():
    #Video streaming route. Put this in the src attribute of an img tag
    return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/')
def index():
    """Video streaming home page."""
    return render_template('index.html')

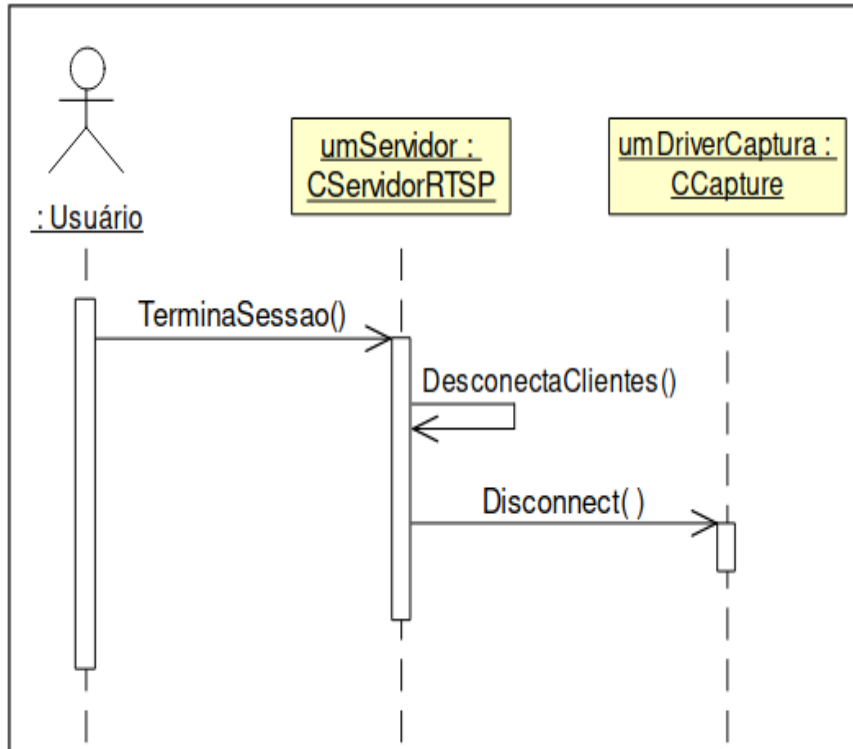
if __name__ == '__main__':
    app.run(debug=True)
```

Fonte: Criação Nakul Lakhotia, 2020

Figura 4. Diagrama de classes

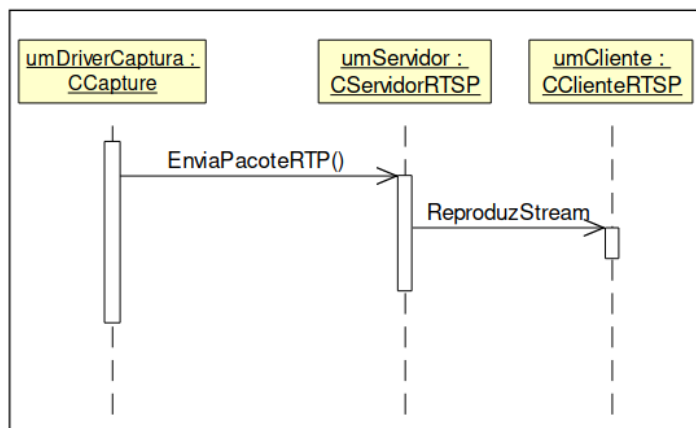
Nessa representação, será possível ver a classe, atributos método e outros elementos que compõem os algoritmos para a construção desse servidor de streaming de vídeo.

5.2 Termina sessão



Fonte: Criação Clodoaldo Tschöke, 2001

5.3 Reproduz Stream



Fonte: Criação Clodoaldo Tschöke, 2001

6. Funcionamento

Então, o funcionamento se dá da seguinte forma, o cliente enviará alguns comandos como SETUP, RTSP, PLAY e outros. Já o servidor por sua vez responderá à esses comandos. Inicialmente o sistemas de requisição e resposta lembra o trabalho anterior, que envolvia servidores HTTP, porém com um pouco mais de complexidade. Sendo assim, no momento em que o servidor estiver no estado de reprodução, ele pega periodicamente um quadro armazenado, por conseguinte empacota o quadro com RTP e envia o pacote RTP para um socket UDP. O cliente recebe os pacotes RTP, extrai os quadros e os descomprime para apresentá-los no monitor.

11. Referências

EDSON, Moreira; ROBERTO Sadao. **Streaming vídeo com RTSP e RTP**. Disponível em: <<http://wiki.icmc.usp.br/images/4/43/RTP-RTSP-slides.pdf>> Acesso em 29/09/2021

FORTINET. **What is Quality of Service (QoS) in Networking?** Disponível em: <<https://www.fortinet.com/resources/cyberglossary/qos-quality-of-service>> Acesso em 30/09/2021

RUETHER, Traci. **RTSP: The Real-Time Streaming Protocol Explained (Update)**. Disponível em: <<https://www.wowza.com/blog/rtsp-the-real-time-streaming-protocol-explained>> Acesso em 30/09/2021

TSCHÖKE, Clodoaldo. **CRIAÇÃO DE STREAMING DE VÍDEO PARA TRANSMISSÃO DE SINAIS DE VÍDEO EM TEMPO REAL PELA INTERNET**. Disponível em: <<http://www.inf.furb.br/~pericas/orientacoes/Streaming2001.pdf>> Acesso em 05/09/2021

KUROSE James; Ross Keith. **Redes Multimídia**. Disponível em: <<https://www.ic.unicamp.br/~nfonseca/arquivos/3ed/cap07.pdf>> acesso em 07/10/2021

LAKHOTIA Nakul. **Video Streaming in Web Browsers with OpenCV & Flask**. Disponível: <<https://towardsdatascience.com/video-streaming-in-web-browsers-with-opencv-flask-93a38846fe00>> Acesso em 07/10/2021

LAKHOTIA, Nakul. **Live-Streaming-using-OpenCV-Flask**. Disponível: <<https://github.com/NakulLakhotia/Live-Streaming-using-OpenCV-Flask/blob/main/app.py>> Acesso em 07/10/2021

ORETES, Yan . **O que é TCP, UDP e quais as diferenças?** Disponível em: <<https://www.alura.com.br/artigos/quais-as-diferencas-entre-o-tcp-e-o-udp>> acesso em 07/10/2021

UFRGS. **Protocolo IP**. Disponível em: <<http://penta2.ufrgs.br/Esmilda/ip.html>> acesso em 07/10/2021

WRIGHT, Gavin. **Real-time Transport Protocol (RTP)**. Disponível em: <<https://www.techtarget.com/searchnetworking/definition/Real-Time-Transport-Protocol>> Acesso em 30/09/2021