

# **Streaming vídeo com RTSP e RTP**

Redes de Alto Desempenho (RAD)

SSC-144

ICMC/USP

Prof. Edson Moreira

PAE: Roberto Sadao

# O que veremos no demo:

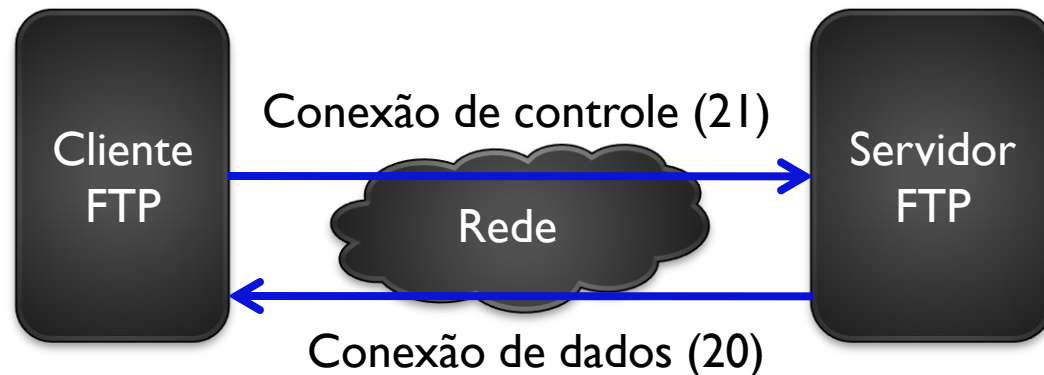
- Implementação de um servidor e um cliente de vídeo de fluxo contínuo. O cliente usará um protocolo de fluxo contínuo em tempo real (RTSP) para controlar as ações do servidor. O servidor usará o protocolo de tempo real (RTP) para empacotar o vídeo.

# Como funciona:

- O cliente envia comandos RTSP SETUP, PLAY, PAUSE e TEARDOWN e o servidor responde aos comandos.
- Quando o servidor estiver no estado de reprodução, ele pega periodicamente um quadro JPEG armazenado, empacota o quadro com RTP e envia o pacote RTP para um socket UDP.
- O cliente recebe os pacotes RTP, extrai os quadros JPEG, descomprime os quadros e os apresenta no seu monitor.

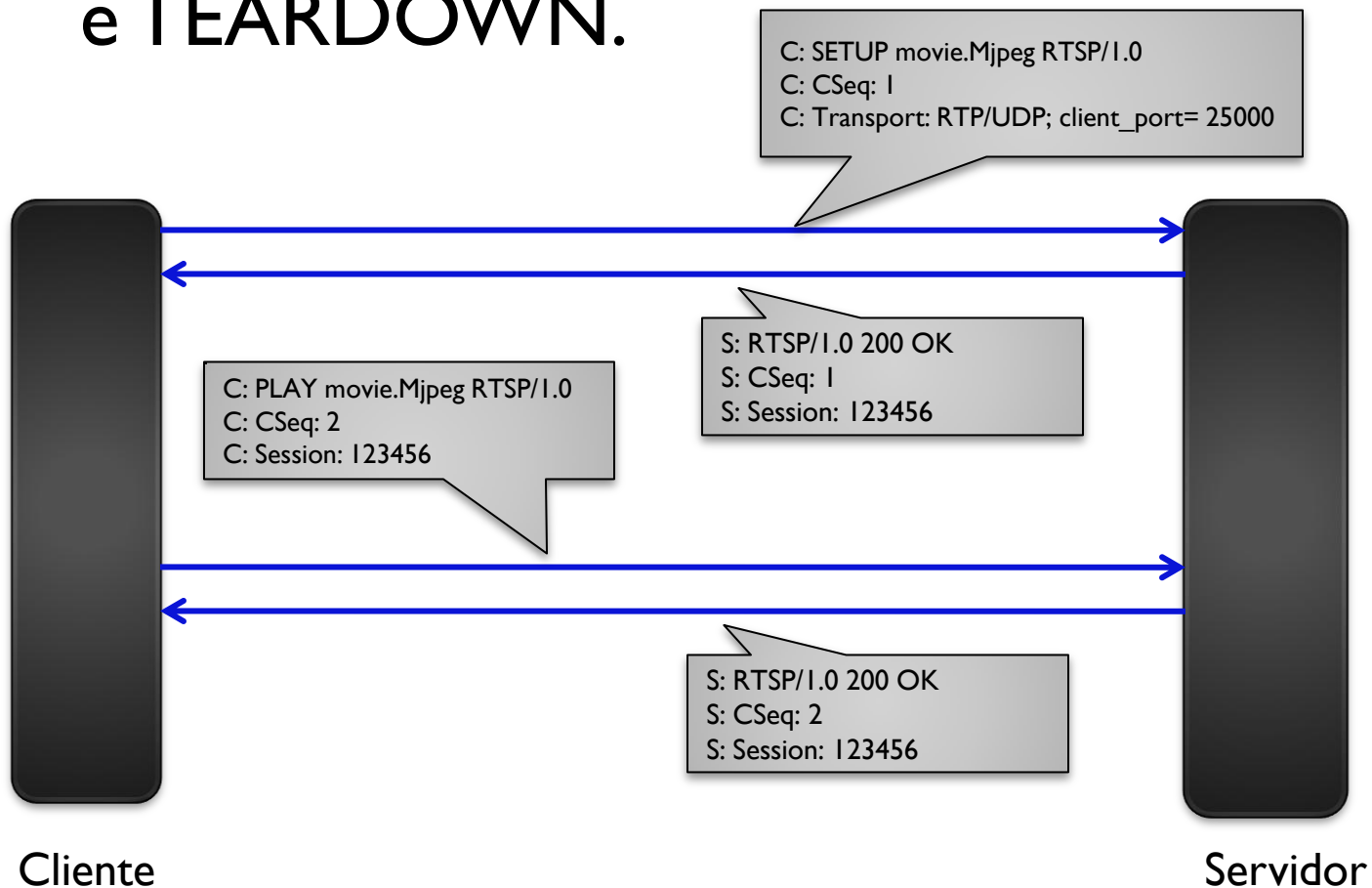
# RTSP (Real Time Streaming Protocol)

- Permite que um transdutor controle a transmissão de uma corrente de mídia.
- É um protocolo “fora da banda”.
  - FTP é um exemplo de protocolo fora da banda:

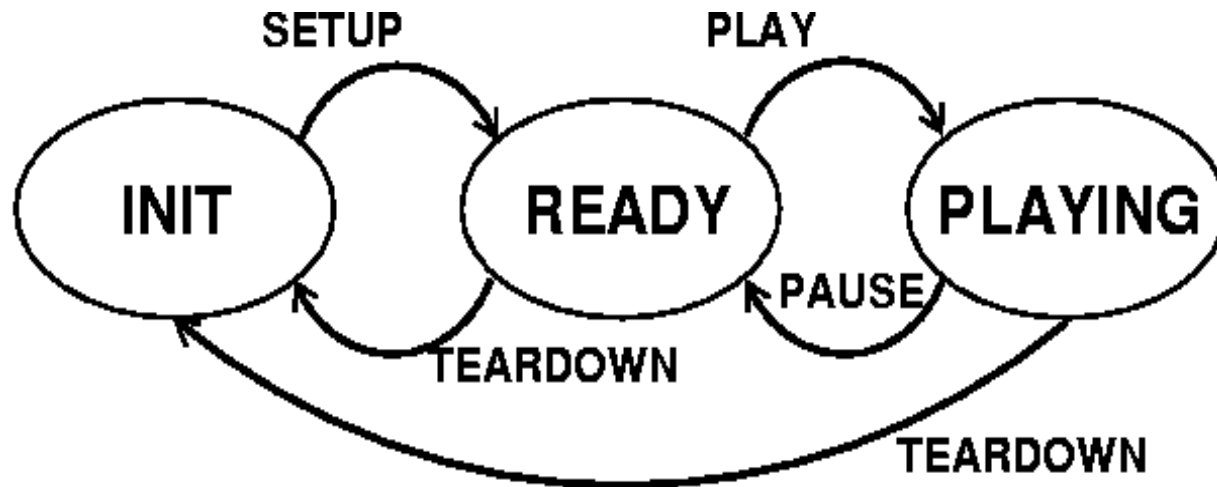


# Exemplo de mensagens RTSP

- Tipos de Mensagens: SETUP, PLAY, PAUSE e TEARDOWN.



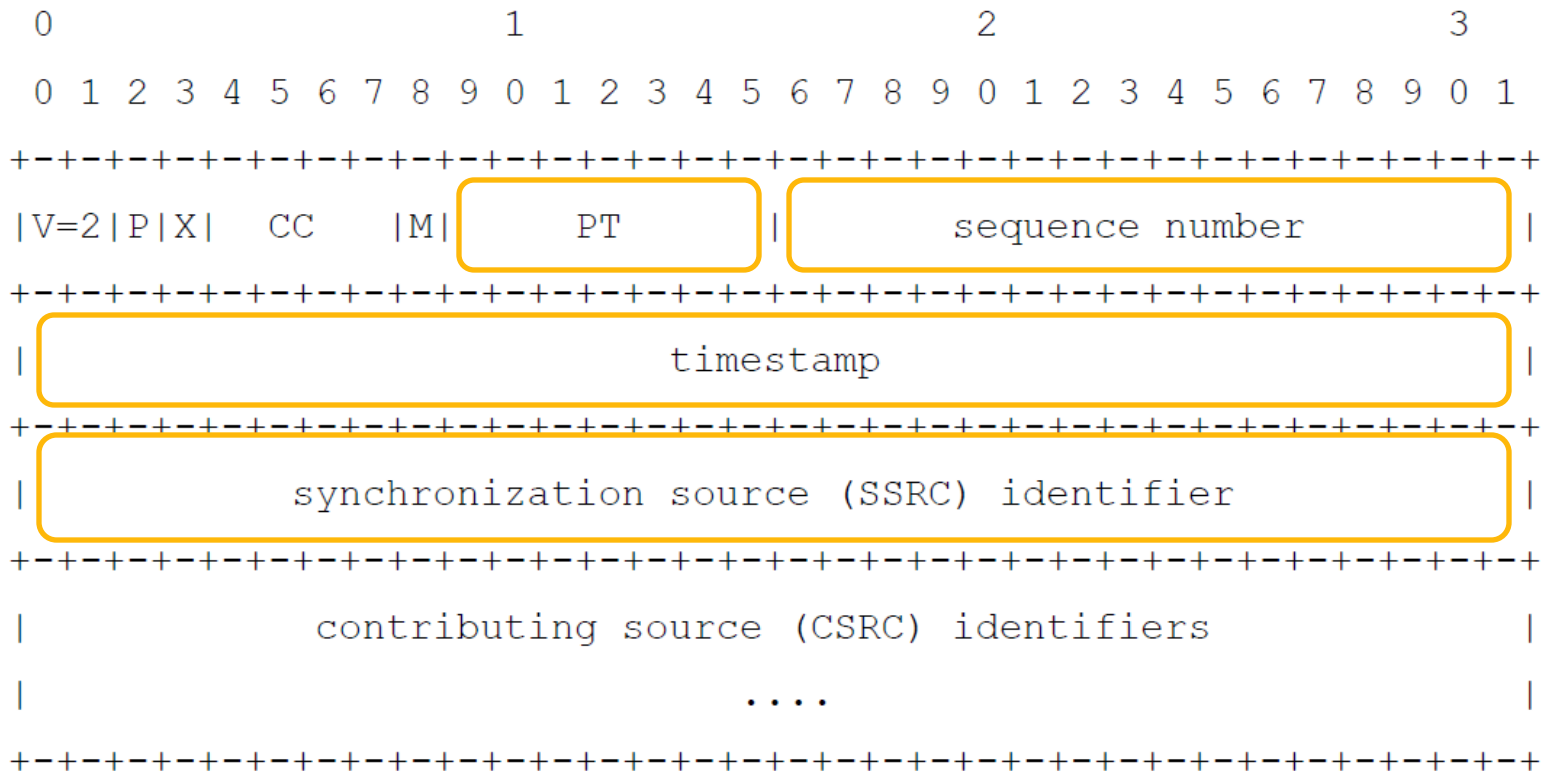
# Estados do Cliente e do Servidor



# RTP Básico

- RTP (Real Time Protocol), definido na RFC-3550.
  - Características, contém campos no cabeçalho de:
    - Número da sequencia.
    - Marcas de tempo.
  - Normalmente roda sobre UDP

# Cabeçalho do RTP





# Cabeçalho do RTP

- ***Payload Type***: Indica o tipo de carga útil

Número do Tipo de carga útil	Formato de vídeo
26	Motion JPEG
31	H.261
32	Vídeo MPEG1
33	Vídeo MPEG2

- ***Sequence Number***: Número sequencial que é incrementado a cada pacote RTP. Utilizado para ocultar os dados perdidos.

# Cabeçalho do RTP

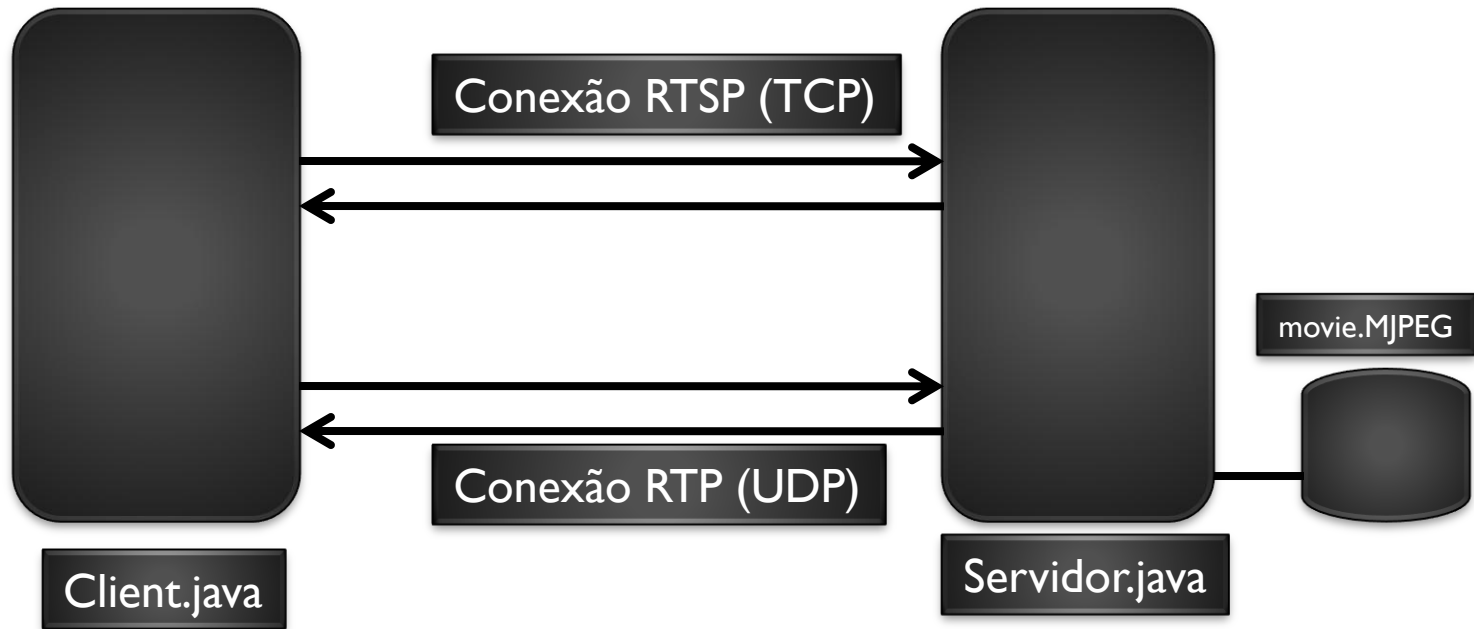
- ***Timestamp***: Marca o tempo da amostragem do primeiro byte no pacote RTP. Utilizado para eliminar a variação do atraso.
- ***SSRC***: Identifica a fonte do corrente (*stream*) RTP, número atribuído pela fonte.

# Tarefa de programação

- Arquivos:
  - *Server.java*
  - *Client.java*
  - *RTPpacket.java*
  - *VideoStream.java*
  - *movie.MJPEG*
- Pegar os arquivos na [wiki.icmc.usp.br](http://wiki.icmc.usp.br)
- Os códigos implementam o protocolo RTSP no servidor e o desempacotamento RTP no cliente, também cuida da apresentação do vídeo transmitido. Serão necessários a implementação do RTSP no cliente e do RTP no servidor.

# Visão Geral:

- Interação Cliente-Servidor



# Exercício:

- A. Calcule as estatísticas sobre a sessão. Você precisará calcular a taxa de perda de pacotes RTP, a taxa de dados de vídeo (em bits ou bytes por segundo) e qualquer outra estatística interessante que você conseguir encontrar.
  
- B. A interface de usuário no cliente possui 4 botões para as 4 ações. Se você compará-la a um transdutor padrão, tal como RealPlayer ou transdutor Windows, você verá que eles possuem apenas 3 botões para as mesmas ações, chamadas, PLAY, PAUSE e STOP (correspondendo exatamente ao TEARDOWN). Não há nenhum botão de SETUP disponível para o usuário. Dado que o SETUP é mandatório numa interação RTSP, como você implementaria isso? É apropriado enviar TEARDOWN quando o usuário clica no botão *stop*?
  
- C. Até aqui, o cliente e o servidor implementam apenas o mínimo necessário de interações RTSP e PAUSE. Implemente o método DESCRIBE, que é usado para passar informações sobre o *media stream*. Quando o servidor recebe uma requisição DESCRIBE, ele envia de volta um arquivo de descrição de sessão que diz ao cliente que tipos de streams estão na sessão e quais codificações estão sendo utilizadas.