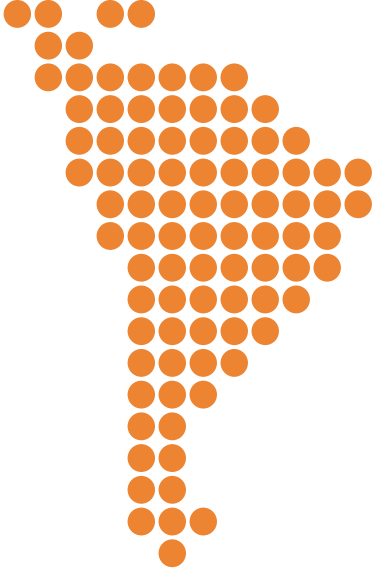

TELECORPO

mesa de corte de vídeo para redes de computadores

Pedro Sousa Lacerda
Grupo de Pesquisa Poéticas Tecnológicas

Abstract

TeleCorpo is a video transmission tool for computer networks, it was created during research on the body and digital culture in Network Arts. It was tested and executed live in internationally synchronized Telematics Dance events, demonstrating itself to be a straightforward tool, simple to install, with easy architecture and implementation, as well as proving very capable for the tasks at hand. This article discusses TeleCorpo, considering its motivation, subjacent technologies, codification, limitations and orientations for use by other researchers and technicians of Network Arts.

Keywords:

Telematics Dance; Network Art; Video transmission; Computer programming; TeleCorpo

Lacerda, Pedro. TeleCorpo: mesa de corte de vídeo para redes de computadores. Revista Eletrônica MAPA D2 - Mapa e Programa de Artes em Dança (e performance) Digital, Ivani Santana (Org) Salvador: PPGAC, Nov. 2015; 2(2): 457-470.

This text contains Hiperlinks to facilitate reader access to references and relevant literature. A simplified version of this article may be found at the project website.

Introduction

TeleCorpo is a video transmission tool for academic internets or local networks. It was developed through scientific research carried out by the Grupo de Pesquisa Poéticas Tecnológicas: corpoaudiovisual (GP Poética) at Universidade Federal da Bahia, which uses digital technology in art, in the relation to studies of the body and digital culture as articulated in several areas.

TeleCorpo was created with the main objective of improving the quality of the telematics events of GP Poética and was tested and run live in internationally synchronized Telematics Dance events, through a high throughput academic network. It proved itself to be a quick, straightforward tool, simple to install, with easy architecture and implementation, as well as being very capable for the task of synchronizing the transmissions.

This article discusses the motivation behind the development of the tool, and details the architecture and implementation, possible uses, their advantages, limitations and errors, besides some of the factors that affect the quality of video transmissions.

TeleCorpo

TeleCorpo stands out by having high tolerances to packet loss, and compatibility with artistic programs such as Pure Data and Max, as well as for transmitting multi-camera events on Youtube. It can be thought of as a type of video switcher, where each monitor can alternate between cameras spread throughout the network. It is the product of research conducted after the telematics dance event called *Embodied in Varios Darmstadt 58*, by the Grupo de Pesquisa Poéticas Tecnológicas at the Universidade Federal da Bahia. The product was developed for Personare, a similar performance, presented live simultaneously in Brazil, Chile and Portugal on September 27 and 28, 2014. The tool is an initiative with the intentions of contributing to the excellence of the Group's telematics events.

Other video transmission tools are: Arthron (Vieira *et al.*, 2012), LoLa (Drioli, 2013), Open Broadcaster Software, Scenic, Snowmix, UltraGrid (Gharai *et al.*, 2006), etc. TeleCorpo is unable to transmit audio, except for on Youtube, therefore tools such as JackTrip, NetJack, or the above mentioned ones may be used for that purpose.

Video Transmission

> *Representing video material in a digital form requires a large number of bits. The volume of data generated by digitising a video signal is too large for most storage and transmission systems (despite the continual increase in capacity and transmission 'bandwith'). This means that compression is essential for most digital video applications.* (Richardson, 2002: 27)

In order to transmit video through computer networks, as any other type of information, it is necessary to encode it into bits. This transformation is handled by a complex algorithm that, usually, besides just encode the data, also compresses it to reduce the amount of information transmitted. The receptor on the other end uses a compatible algorithm to decode and decompress the bits into images. This pair of algorithms, encoder/decoder is called a "codec."

The choice of the codec and the parameters that configure it greatly influence not only the transmission delay (being more, or less, efficient), but also the quality of the image (compressed to the point of losing large amounts of data), or the quantity of information to be transmitted (greater or lesser compression ratio). Since Richardson (2002), the bandwidth has grown significantly, reflected by the creation of applications such as LoLa and UltraGrid, which don't use (or have the option not to use) codecs in order to reduce computational power needed, resulting in less latency between capture, transmission and exhibition (Drioli et al., 2013; Gharai et al., 2006).

The power of the computers has an impact on the transmission latency (delay) because it takes time to capture, encode, decode, and exhibit the images. More powerful computers can achieve these tasks more quickly. Compressing (encoding) the frames uses more processing power than decompressing (decoding), and as a result exhibition is "lighter" than capture.

A third influencing factor related to latency is the implementation quality of the of the tool, and the algorithms used by it, being that it is impossible to disassociate the conceptual (architecture, algorithms, etc.) from the concrete (implementation, tools, etc.).

Naturally, you also expect that some of the network's own characteristics will influence the quality of the transmission, such as described in the following table:

Network Characteristic	Impact on Transmission	
Delay	Delay in transmission	To send a data packet from one locale to another, it needs to travel certain distances and pass through network equipments.
Packet loss	Image degradation	Losses above 1% may threaten transmission, making it difficult for the decoder to reconstruct images. So, the codec's parameters are configured to reduce the compressor's efficiency, augmenting redundancy and making losses less relevant.
Jitter	Small delays in transmission	Due to the characteristics of computer networks, packets may arrive out of order, some later than others. However, a cache/buffer waits for late packets to avoid erroneous image construction, but it rejects very late packets, deeming them lost.

These characteristics are normally out of the artist's control and depend on the state and infrastructure of the network, such as deterioration of equipment, and if there is sufficient bandwidth, for example. While some tools may help in the diagnostics, such as: ping (superficial measurement of latency and losses); iperf (measurement of losses, jitter, and available bandwidth); traceroute (identification of route, or network equipment between two places); and mtr (route identification and superficial measurement of latency and losses for equipment along the route).

Architecture, implementation and technical decisions

The architectural design of TeleCorpo's current version (v0.92) consists of three essential modules for the program to function properly as well as a fourth used in transmitting events to audiences not physically present at the performance, as the following table shows:

Module	Description
tc.producer	Captures one or more cameras, making them available as RTSP streams.
tc.viewer	Video editing bay that alternates between the available streams.
tc.server	Manages active streams.
tc.youtube	Transmits video to the mass public.

The protocol chosen was the RTSP, which is similar to HTTP, but it transmits audiovisual content rather than hypertext, usually through the subjacent protocols RTP and UDP (Schulzrinne, 1998). However, the main advantage is that the RTSP makes the content (streams) available via an URL and offers great compatibility with many multimedia players, making it a more accessible tool to users who are not tech-savvy.

The multimedia frameworks chosen were GStreamer1.0 and the gst-rtsp-server, both written in the "C" programming language. Due to the slow pace in which applications are developed in that language, TeleCorpo was developed in Python3, uusing such frameworks through bindings automatically created by the middleware GObject Introspection.

The codec chosen was H.264 for no reason other than its wide popularity, compatibility, and relative modernity. More modern codecs are capable of better image compression, reducing bandwidth consumption, typically at the cost of processing. When the encoder sends a key-frame (IDR picture), that is, a complete frame equal to a photograph, the following frames will be incomplete, but complementary, so that they update the image with following movements (Itu, 2014). When packets es are lost, the reference image is lost, making it necessary to receive another IDR picture to properly reconstruct the image.

In order to increase the H.264's resilience to losses, the largest temporal distance between two key-frames was reduced. Now, when losses occur, the decoder can quickly recover by receiving a new complete image. In GStreamer, this behavior is controlled by the option key-int-max. In the x264, the library used by GStreamer, it is controlled by the option -keyint.

According to the Hunt & Thomas (1999), TeleCorpo's code wouldn't be considered spaghetti, since changes in one module don't interfere in the others because there are no explicit dependencies between them. Loosely coupled, therefore. But there are dependencies during execution that happen in the system dynamics, demanding a specific initialization sequence to start the components.

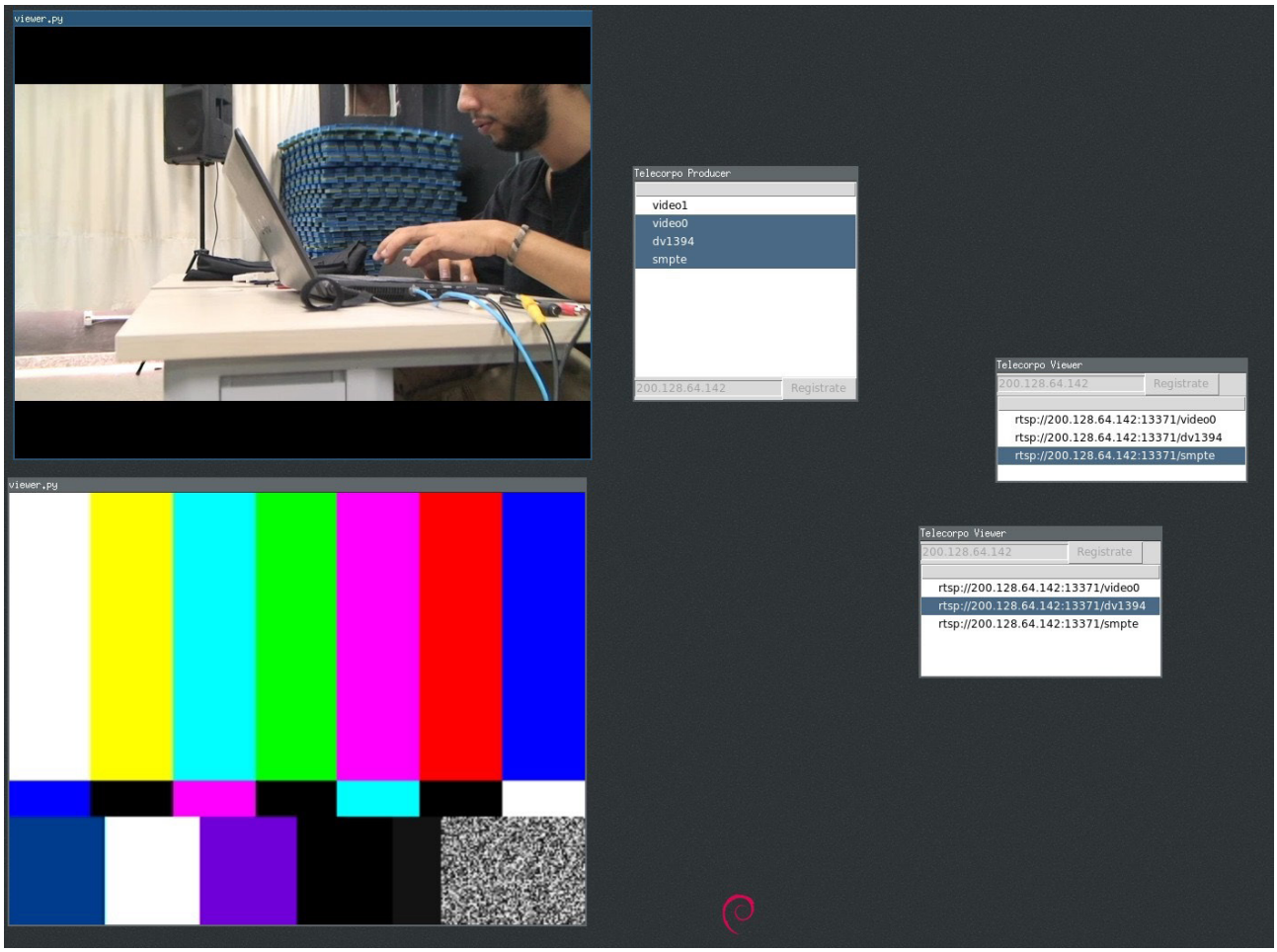

Figure 1. TeleCorpo in action with a producer and two viewers.

While the server is running, the producer registers the produced streams (captured cameras). The server then periodically consults each URL's stream to verify if it is still active, and disconnects it if verification is failed. The viewer is simpler, only inquiring periodically on the remaining active URLs on the server, which have yet to be closed. **Fig. 2** and **Fig. 3** are examples of two distinct systems demonstrating the exchange of messages.

Whereas the youtube module, used to transmit videos live to Youtube Live, is completely independent from the other, being able to be used to transmit content that was not produced by TeleCorpo. The module was based on scripts developed by Maersk-Moller (2013) for the project Snowmix. The choice to transmit via a content delivery networks rather than use own resources was to avoid network saturation in the case of many on-line spectators.

User Manual

The system needs only one server instance to work properly. The producers and viewers can be initialized, as many as are needed. If there are two cameras in two different countries, for example, it will be necessary to initialize two producers, one in each country. Similarly if there are three video projectors in three different countries, it will be necessary to initialize three viewers, one in each country.

Due to limitations in the implementation and lack of research, it is not possible to use TeleCorpo (specifically the producer and server) on machines protected by firewalls.

Besides requiring a specific startup order for the modules due to the distributed systems own characteristics, there are implementation defects that reaffirm the need for care during system initialization. The server module is always the first to be executed, since producers and viewers are registered on it. Thus, theoretically, when they are needed, producers and viewers may be initialized in any order, but due to discrepancies between the architecture and the implementation, first the producers and then the viewers should be run.

In Fig. 4 one sees the example of three nodes (or niches), each one possessing a different amount of cameras and projectors. This configuration would require four viewers, one for each projector. But only three producers would be needed, one for each niche, since each one can capture multiple cameras.

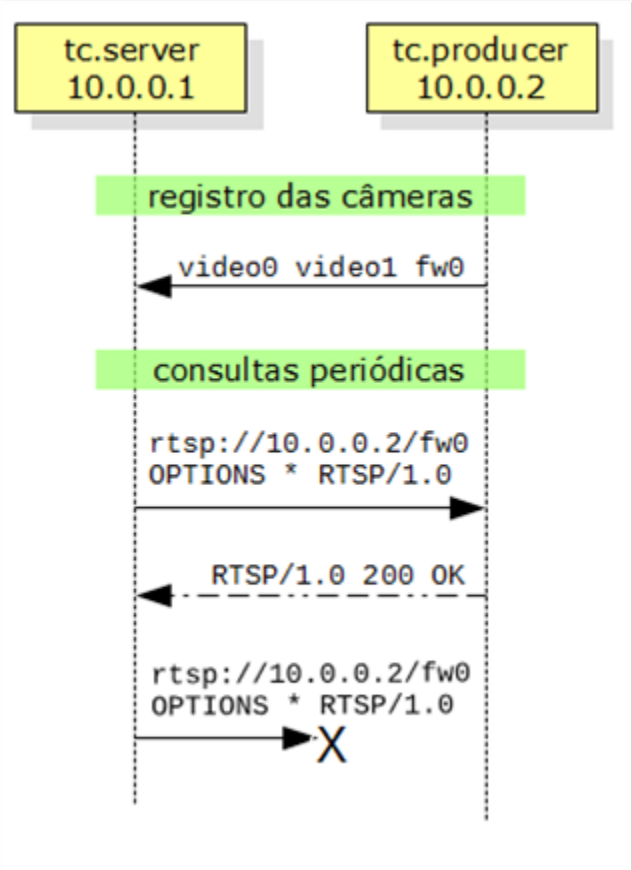

Figure 2. Register of a producer with three cameras, then message exchanging regarding one of the cameras until the producer is closed, when the consultation fails and the producer is unregistered.

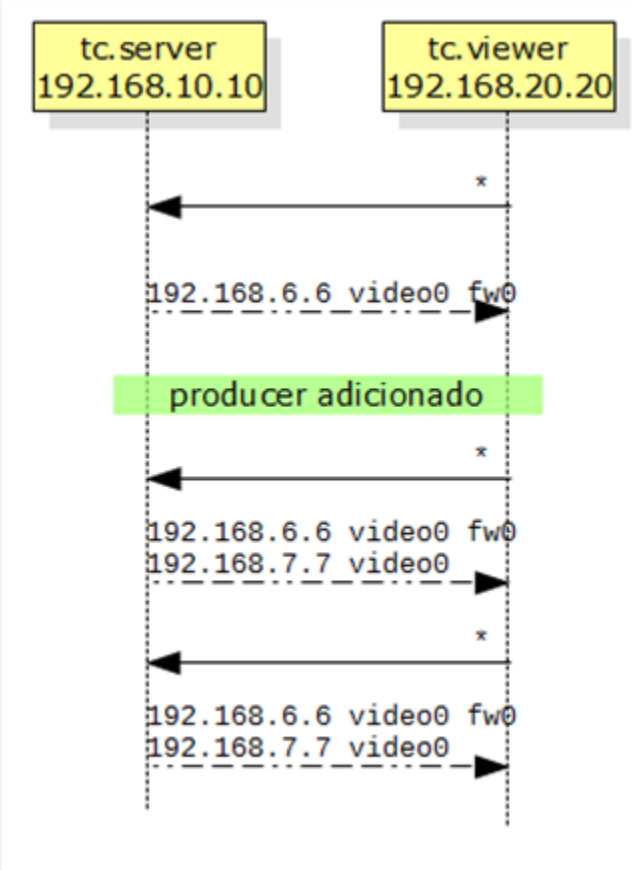

Figure 3. a viewer inquiring about active producers, then as second producer is registered on the server.

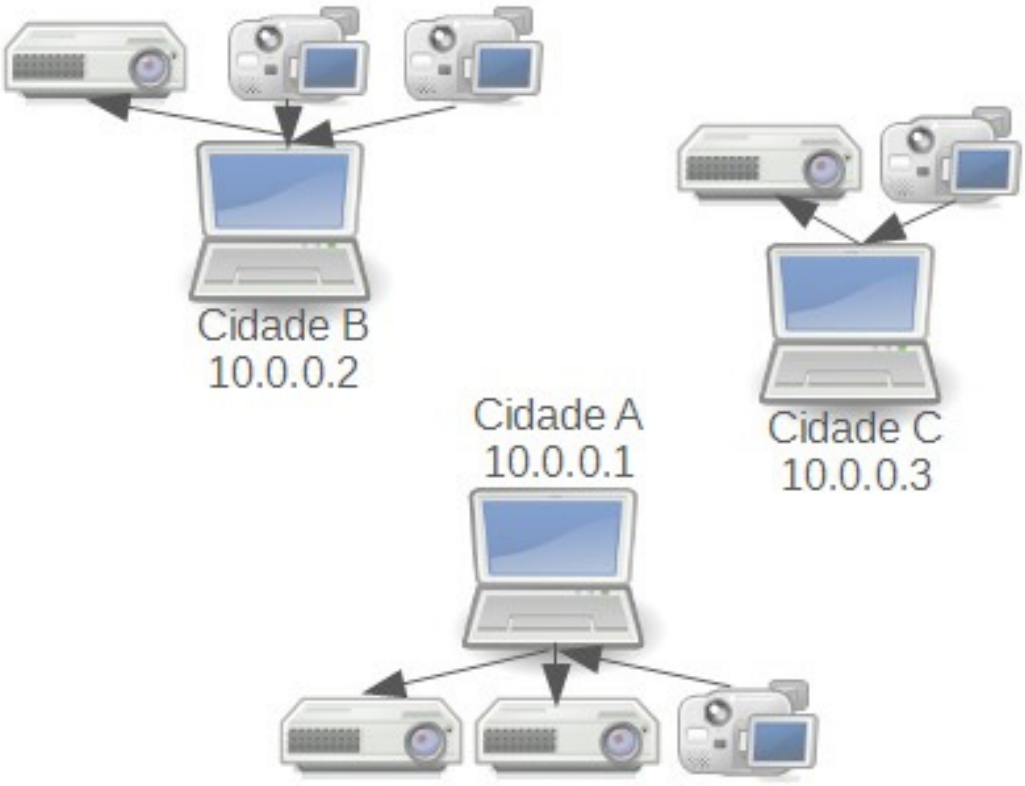

Figure 4. Example of a possible configuration.

The use of the tc.youtube module demands the creation of a live event on Youtube Live, and, it's execution configuring it with the values obtained at the event's configuration page:

```
$ python3 -m tc.youtube -r 360p -t my_name.
a1b2-c3d4-e5f6-g7h8 \

        rtsp://10.0.0.1:13371/video0
```

If the parameter -b is used, the event will also be transmitted to a backup server as well to Youtube's main server. This module requires a running instance of the server JACK Audio Connection Kit on the same machine. It is also possible to transmit events independently from TeleCorpo adjusting the URL; the first webcam of a computer, for example, is usually available through the URL: v4l2:///dev/video0.

Video reception by external applications

The main reason for choosing RTSP protocol was its wide compatibility with different multimedia programs. Applications based on QuickTime, for example, inherit the capacity to digest this type of streams, such as Max, where it is possible to configure the object *[jit.qt.movie]* or *[jit.movie]* with a message like [read rtsp://10.0.0.1:13371/fw0(.

Pure Data is not naturally able to accept these streams, but it is possible to use the v4l2loopback as a bridge between Pure Data and other video applications if they are in the same machine. The solution is to create virtual video devices (dev) with v4I2loopback, receive the RTSP stream with certain softwares, redirect it to the device created, and receive it on Pure Date with the Gem's object [pix_film] or [pix_video] using a message similar to [device 10(:

```
$ sudo modprobe v4l2loopback video_nr=10

$ gst-launch-1.0 rtspsrc
location=rtsp://10.0.0.1:13371/fw0 latency=30 !
decodebin \

      ! queue ! v4l2sink sync=false device=/
dev/video10
```

In this way it is possible to faithfully recreate the video switcher functionality, on Max as well as Pure Data, getting rid of the tc.viewer module. The latency property of the previous command determines the size of the cache/buffer in milliseconds for use in reception, where it should be notably larger than the network jitter, in order to deal with turbulence.

Final Considerations

Having been specifically designed for the the performance of Personare, TeleCorpo exceeds many of the requirements for video transmission of telematics dance as developed in the work of the Grupo de Pesquisa Poéticas Tecnológicas: corpoaudiovisual. Thanks to the frameworks GStreamer and gst-stsp-server and the codification technology used, it proved to be extremely efficient, resilient to packet losses, with a very low latency compared to other transmission programs. Although no quantitative study was undertaken, the transmission speed was on par with other relevant programs in this area. However, it should be expected that other applications developed specifically with the objective of low latency would manage an even better performance, such as UltraGrid and LoLa.

One advantage that TeleCorpo had over other programs is its small size and simplified architecture, allowing for quick understanding by junior or senior developers, and that it also accommodate modifications easily.

Future prospects include video transmission using multicast technology, avoiding redundant traffic, and thus reducing bandwidth consumed. This modification will permit simplifications of the architecture, making the tc.server module unnecessary. Adoption of more modern codecs is not desirable due to the low compatibility with external applications.

Notes

1 Spaghetti Code: an interweaving, unstructured code, with control jumping from one part to the next.

References

DRIOLI, C. ; ALLOCCHIO, C. ; BUSO. *Networked performances and natural interaction via LOLA: Low latency high quality A/V streaming system. Information Technologies for Performing Arts, Media Access, and Entertainment*. Springer Berlin Heidelberg, p. 240-250. Disponível em <http://www.internetsociety.org/sites/default/files/pdf/accepted/32_LOLA.pdf>Acesso em: 21 ago. 2015.

Gharai, Ladan, et al. "Experiences with high definition interactive video conferencing". *Multimedia and Expo, 2006 IEEE International Conference on. IEEE*, 2006.

HUNT, A. ; THOMAS, D. *O Programador Pragmático: de aprendiz a mestre*. Bookman, 2010.

ITU, Telecommunication Standartization Sector. H.264: *Advanced video coding for generic audiovisual services*. 2014. Disponível em <https://www.itu.int/rec/T-REC-H.264-201402-I> Acesso em: 21 ago. 2015.

MAERSK-MOLLER, P. *Snowmix and CDNs*. 2013. Disponível em: <http://sourceforge.net/p/snowmix/wiki/Snowmix%20and%20CDNs/>. Acesso em: 21 ago. 2015.

RICHARDSON, I. E. G. *Video Codec Design: Developing Image and Video Compression Systems*. Chichester: John Wiley & Sons, 2002.

SCHULZRINNE, H. ; RAO, A. ; LANPHIER, R. "RFC 2326: Real Time Streaming Protocol (RTSP)". *The Internet Society*. 1998. Disponível em <https://tools.ietf.org/html/rfc2326>. Acesso em: 21 ago. 2015.

VIEIRA, E. ; PASSOS, M. ; SANTOS, B. A. ; OLIVEIRA, S. S. ; MELO, E. A. ; MOTTA, G. H. M. B. ; TAVARES, T. A. ; SOUZA FILHO, Guido Lemos de. "Uma Ferramenta para Gerenciamento e Transmissão de Fluxos de Vídeo em Alta Definição para Telemedicina". *In: Salão de Ferramentas do Simpósio Brasileiro de Redes de Computadores 2012*, 2012, Ouro Preto. Anais do Salão de Ferramentas do Simpósio Brasileiro de Redes de Computadores 2012. v. 1, 2012. Disponível em <http://gtavcs.lavid.ufpb.br/wp-content/uploads/2011/11/ARTHRON_2-O_salao-ferramentas-SBRC-2012_final.pdf>. Acesso em: 21 ago. 2015.

About the author

Pedro lacerda é programador e usuário de computadores. É bacharel em Ciência & Tecnologia pela Universidade Federal da Bahia. Possui experiência em linguagens de programação na etnomatemática, desenvolvimento de sistemas, modelagem e simulação de sistemas para biofisicoquímica farmacêutica, e transmissão e codificação de vídeo para dança digital. Atualmente investiga inteligência artificial e suas aplicações. No seu tempo livre se diverte com interfaces entre diferentes linguagens de programação.