

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-  
GRANDENSE - CAMPUS PASSO FUNDO  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

**PATRÍCIO NUNES DE MELLO**

**IMPLEMENTAÇÃO DE UMA INTERFACE DE STREAMING DE VÍDEO  
PARA APLICAÇÃO EM SISTEMAS EMBARCADOS**

**Prof. Me. Ricardo Vanni Dallsen**

**PASSO FUNDO**

**2016**

**PATRÍCIO NUNES DE MELLO**

**IMPLEMENTAÇÃO DE UMA INTERFACE DE STREAMING DE VÍDEO  
PARA APLICAÇÃO EM SISTEMAS EMBARCADOS**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Campus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador (a): Prof. Me. Ricardo Vanni  
Dallasen

**PASSO FUNDO**

**2016**

**PATRÍCIO NUNES DE MELLO**

**IMPLEMENTAÇÃO DE UMA INTERFACE DE STREAMING DE VÍDEO PARA  
APLICAÇÃO EM SISTEMAS EMBARCADOS**

Trabalho de Conclusão de Curso aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

---

Prof. Me. Ricardo Vanni Dallasen (orientador)

---

Prof. Me. Maikon Cismoski dos Santos (avaliador)

---

Prof. Vanessa Lago Machado (avaliador)

---

Prof. Me. Adilso Nunes de Souza  
Coordenador do Curso

**PASSO FUNDO**

**2016**

*À Deus, por me dar forças e iluminar  
meu caminho. Aos meus pais, pela  
compreensão e o estímulo em todos  
os momentos.*

## **AGRADECIMENTOS**

Primeira agradeço a Deus por estar sempre presente me dando forças para seguir em frente e saúde para realização deste projeto.

Aos meus pais, Jorge Luiz Alves de Mello e Rosangela Nunes de Mello, pelos ensinamentos e estímulo para sempre correr atrás dos objetivos e metas. As minhas irmãs Daniela Nunes de Mello e Gabriela Nunes de Mello, pelos pensamentos positivos e as palavras de incentivo e apoio de sempre.

Agradeço ainda ao meu orientador Ricardo Vanni Dallsen, pela paciência e ensinamentos passados.

Aos meus colegas e amigos Fagner Correa Griniuc e William Marcondes Bonadiman, que sempre me apoiaram e sempre me incentivaram a seguir em frente.

Por fim agradeço a todos os professores do curso que de uma maneira ou outra contribuíram para o meu aprendizado e que me permitiram o desenvolvimento deste trabalho.

“A tarefa não é tanto ver aquilo que  
ninguém viu, mas pensar o que ninguém  
ainda pensou sobre aquilo que todo  
mundo vê”.

Arthur Schopenhauer

## RESUMO

O presente trabalho apresenta a implementação de uma interface de *streaming* de vídeo para aplicação em sistemas embarcados. O trabalho teve como motivação a necessidade de soluções de baixo custo na área de monitoramento com Veículos Aéreos Não Tripulados (VANTs). No escopo do trabalho são descritas as tecnologias utilizadas no processo de *streaming*, a partir do levantamento bibliográfico realizado. Através deste levantamento foi definida a plataforma Raspberry Pi, um mini microcomputador, como ferramenta utilizada para o *streaming* com o auxílio do framework Userspace Video4Linux2 (UV4L) como gerenciador do processo, realizando a conversão e compactação dos vídeos enviados. Para decodificação e visualização dos vídeos no computador cliente foi utilizado o reprodutor de multimídias VideoLAN Client (VLC). Os testes foram aplicados em um modo com baixa resolução de captura e outro com alta resolução. Durante cada teste foram verificadas a resolução de imagem, uso de rede e o atraso de imagem. Constatou-se que em relação a resolução e o uso de rede não ocorreram grandes alterações de valores entre os dois cenários, porém nos resultados referentes ao atraso, o *streaming* realizado com resolução alta sofreu um atraso duas vezes maior em relação à de resolução baixa aplicada.

Palavras-chave: *Streaming* de vídeo, UV4L, Raspberry Pi.

## **ABSTRACT**

This paper presents the implementation of a streaming video interface for use in embedded systems. The work was motivated due the need for low-cost solutions in the monitoring area with Unmanned Aerial Vehicles (UAVs). The scope of the work describes the technologies used in the streaming process, from the conducted literature. Through this survey was defined Raspberry Pi platform, a mini PC as a tool used for streaming with the help of the framework Userspace Video4Linux2 (UV4L) as process manager, performing the conversion and compression of the uploaded videos. For decoding and viewing videos on the client computer was used multimedia player VideoLAN Client (VLC). The tests were applied in a mode with low resolution capture and another with high resolution. During each test were checked image resolution, network usage and image delay. It was found that with regard to resolution and network usage, there were no major changes in values between the two scenarios, however the results for the delay, streaming performed with high resolution has been delayed twice greater than the low resolution applied.

Keywords: Video streaming, UV4L, Raspberry Pi.

## LISTA DE FIGURAS

Figura 1 - Hermes 450S .....	15
Figura 2 - Flying-drones .....	15
Figura 3 - Processo streaming .....	17
Figura 4 - Esquema de rede sem fio .....	19
Figura 5 – Raspberry Pi Modelo B+ .....	21
Figura 6 – Interface gráfica do Raspbian .....	22
Figura 7 - Câmera Raspberry Pi.....	27
Figura 8 – Processo de <i>Streaming</i> .....	29
Figura 9 – Configurações arquivo interfaces.....	30
Figura 10 – Configuração de resolução em 640x480.....	31
Figura 11 – Configuração de resolução em 1080x720.....	32
Figura 12 – Streaming transmitido com resolução 640x480 .....	34
Figura 13 - Streaming transmitido com resolução 1080x720 .....	35
Figura 14 – Captura ferramenta Iftop cenário 1.....	36
Figura 15 - Captura ferramenta Iftop cenário 2 .....	36
Figura 16 – Consumo médio da banda de rede .....	37
Figura 17 – Imagem inicial e imagem final resolução 640x480.....	38
Figura 18 - Imagem inicial e imagem final resolução 1080x720.....	38

## LISTA DE TABELAS

Tabela 1 – Comparativo dos padrões 802.11 .....	19
Tabela 2 – Comparativo modelos Raspberry Pi - 2015.....	26
Tabela 3 - Detalhes da câmera Raspberry Pi.....	27

## LISTA DE ABREVIATURAS E SIGLAS

AP - Access Point  
API - Application Programming Interface  
ARM - Advanced RISC Machine  
BIOS - Basic Input/Output System  
CD - Compact Disc  
DVD - Digital Versatile Disc  
HTTP - Hypertext Transfer Protocol  
HTTPS - Hyper Text Transfer Protocol Secure  
IP - Internet Protocol  
JPEG - Joint Photographics Experts Group  
MJPEG - Motion JPEG  
MPEG - Moving Picture Experts Group  
MRTG - Multi Router Traffic Grapher  
OSH - Open-Source Hardware  
PDAs - Personal digital assistants  
SBC - Single Board Computer  
SNMP - Simple Network Management Protocol  
SSH - Secure Shell  
UAV - Unmanned Aerial Vehicle  
USB - Universal Serial Bus  
UV4L - Userspace Video4Linux2  
UVC - USB Video Class  
VCD - Video Compact Disc  
VLC – VideoLAN Client  
WAP - Wireless Application Protocol  
WLAN - Wireless Local Area Network

## SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	MOTIVAÇÃO .....	12
1.2	OBJETIVOS.....	13
1.2.1	Objetivo geral.....	13
1.2.2	Objetivos específicos .....	13
1.3	ORGANIZAÇÃO MONOGRAFIA.....	13
2.1	VANTS.....	15
2.2	<i>STREAMING</i> .....	16
2.3	REDES WI-FI.....	18
2.4	RASPBERRY PI .....	20
2.5	SISTEMAS OPERACIONAIS PARA SISTEMAS EMBARCADOS .....	21
2.6	UV4L.....	23
2.7	MONITORAMENTO DE REDES .....	24
2.8	CODIFICADORES E DECODIFICADORES.....	25
3.1	DESCRIÇÃO DO PROJETO .....	26
3.2	DESCRIÇÃO DO AMBIENTE .....	28
3.3	CONFIGURAÇÕES E FERRAMENTAS ADICIONAIS .....	29
3.4	CENÁRIOS APLICADOS.....	30
3.4.1	Streaming em resolução de 640x480 .....	31
3.4.2	Streaming em resolução de 1080x720 .....	31

3.4.3 Pontos de avaliação.....	32
4.1 RESOLUÇÃO .....	34
4.2 CONSUMO DE BANDA DE REDE .....	35
4.3 ATRASO .....	37
5 CONSIDERAÇÕES FINAIS.....	39
REFERÊNCIAS.....	41

## 1 INTRODUÇÃO

O avanço tecnológico ocorre com muita rapidez (LONGHITANO, 2010), o que faz com que novas tecnologias se tornem opção e possam ser aplicadas em diferentes áreas. Devido a estas evoluções, no que se refere ao ambiente computacional, sobretudo ligadas a sistemas embarcados, é cada vez mais fácil realizar procedimentos que necessitam de um poder computacional utilizando equipamentos de baixo custo (PEREIRA, 2014).

Da mesma forma, o aumento do processo de miniaturização de componentes eletrônicos faz com que aplicações com Veículos Aéreos Não Tripulados (VANTs) cresça constantemente, conforme ARMADA (2004, apud LONGHITANO, 2010). Seu uso em projetos científicos e civis tem sido desenvolvidos, nos quais os veículos aéreos desprovidos de tripulação podem apresentar vantagens técnicas e econômicas em relação a diferentes campos de atuação, segundo LONGHITANO (2010), como aplicação no reconhecimento ou transporte de medicamentos em áreas de difícil acesso. Modelos do tipo multi-rotor permitem facilidade no controle, assim o tornando adequado para aplicações que envolvam captura de imagem e vídeo, como em aplicações de monitoramento.

Diante deste contexto, no presente trabalho foi realizada a implementação de uma interface embarcada de *streaming* de vídeo a partir da integração de componentes, buscando assim através de técnicas de compressão de vídeo e ajustes nas configurações e componentes deste processo, realizar o envio de vídeo através de uma rede sem fio para um computador cliente.

### 1.1 MOTIVAÇÃO

O monitoramento com o uso de VANTs cresce a cada dia, segundo LONGHITANO (2010). Contudo a maioria das soluções aplicadas fazem uso de equipamentos de alto custo o que acaba dificultando a aquisição deste tipo de veículos. Soluções alternativas são desenvolvidas a fim de gerar possibilidades de aquisição e aplicação destes equipamentos em um ambiente em que o mesmo se faz necessário.

Deste modo, se torna relevante a aplicação do presente projeto, como forma implementar através do microcomputador Raspberry Pi, uma interface de monitoramento de baixo custo, para uso em VANTs.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo geral**

Implementação de um sistema embarcado de *streaming* de vídeo para monitoramento remoto utilizando VANTs.

### **1.2.2 Objetivos específicos**

- Implementar um sistema de *streaming* de vídeo a partir da integração de componentes;
- Realizar o envio das imagens capturadas via rede móvel (Wi-Fi);
- Identificar e solucionar possíveis problemas no envio das informações na utilização de redes Wi-Fi;
- Aplicar um sistema de captura/envio de vídeo que possa ser utilizado em um sistema de monitoramento

## **1.3 ORGANIZAÇÃO MONOGRAFIA**

Este trabalho é dividido em capítulos, os quais estão organizados da seguinte maneira:

- Capítulo 2: este capítulo possui a pesquisa bibliográfica, a qual serviu como base para o desenvolvimento do trabalho. Neste se encontra uma abordagem sobre VANTs, com uma parte introdutória do seu início no meio militar e posteriormente descrevendo suas aplicabilidades. Ainda serão descritas as tecnologias necessárias para a aplicação do projeto, expondo o processo de streaming, explicando seu funcionamento e métodos. Ainda é realizada uma introdução a redes sem fio, além da

plataforma Raspberry Pi, como também os sistemas operacionais que nele podem ser utilizados e por fim uma descrição do *framework* definido como ferramenta para realização do *streaming*.

- Capítulo 3: é descrita a metodologia aplicada, expondo os materiais e métodos necessários para o desenvolvimento do trabalho.
- Capítulo 4: são descritos os testes realizados, bem como os resultados obtidos após sua execução.
- Capítulo 5: são descritas as considerações finais, assim como o levantamento dos aprendizados obtidos e dificuldades encontradas, além da abordagem sobre os trabalhos futuros.

## 2 DESENVOLVIMENTO

Neste capítulo encontram-se os principais conteúdos abordados e relevantes para o entendimento e conhecimento do trabalho, constituindo-se de fontes bibliográficas da área.

### 2.1 VANTS

Os VANTS ou UAV (Unmanned Aerial Vehicle), na nomenclatura em inglês, são veículos aéreos desprovidos de tripulação, conforme visualizado na Figura 1 e Figura 2. Este tipo de veículo foi desenvolvido para aplicações militares, devido à necessidade da execução de missões aéreas que ofereciam riscos a vida humana, assim descartando a necessidade da exposição de algum soldado neste tipo de tarefa. Este tipo de veículo pode ser controlado remotamente ou operado de forma autônoma, podendo ter a capacidade de alcançar lugares hostis e de difícil acesso (LONGHITANO, 2010).

**Figura 1 - Hermes 450S**



Fonte: Fas (1999)

**Figura 2 - Flying-drones**



Fonte: Pplware (2015)

Segundo EISENBEIß (2009) VANT são veículos aéreos não tripulados que são inabitados, reutilizáveis e controlados, sendo que os mesmos podem voar de forma autônoma, semiautônoma ou manualmente guiados por um piloto em terra usando um controlo remoto.

Os VANTS se diferenciam dos aeromodelos em vários pontos relacionados aos seus sistemas de comando, estabilidade, controle e acompanhamento de voo, se são redundantes e/ou independentes. No Brasil ainda há esta diferenciação em

relação a sua utilização e aplicabilidade no ambiente. (PEDROSA, 2015) No Brasil, os VANTs são classificados e regulados conforme seu propósito de uso (ANAC, 2016). Quando o uso é destinado a lazer, esporte, hobby ou competição, este é visto como um aeromodelo, podendo ser um mini helicóptero, uma réplica de um jato ou até mesmo um helicóptero que possua quatro hélices, conhecido como "quadricóptero".

Nos casos do seu uso para outros fins, como pesquisa, experimentos ou comercial, o aparelho passa a ser entendido como um VANT. Contudo, além do fim não-recreativo, para ser um VANT, o equipamento precisa possuir uma carga útil embarcada não necessária para o equipamento voar. Exemplos dessa carga útil são as câmeras acopladas para tomadas aéreas de filmes ou entrega de correspondências, como cartas, pacotes ou até mesmo uma pizza (PEDROSA, 2015).

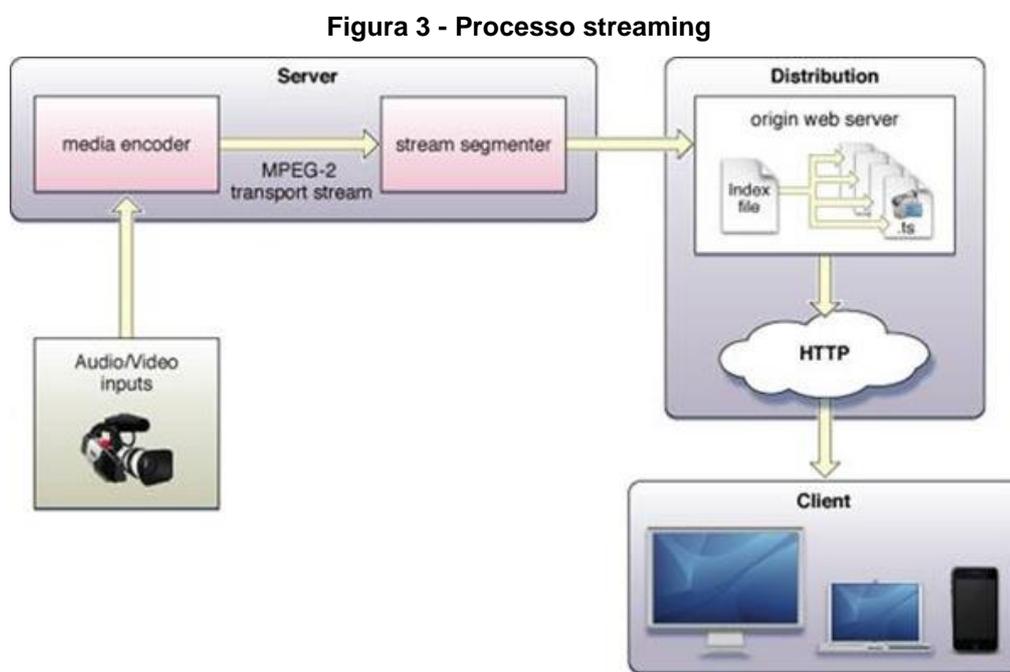
Atualmente os VANTs são utilizados em atividades além das aplicações militares, como distribuição de alimentos e medicamentos em vilarejos da África (ROBOHUB.ORG, 2016), muitos destes encontram-se inacessíveis por terra. Ainda são usados na captura de imagens aéreas em locais de difícil circulação, como imagens de reatores danificados durante o acidente de Fukushima, Japão.

Além de utilizações em mapeamentos de desmatamentos e impactos ambientais (LONGHITANO, 2010), já utilizados pelo Ibama na Floresta Amazônica. Além de utilizações na área agrícola, para fins de coleta e disponibilização de dados (CERBARO, 2016).

## **2.2 STREAMING**

O *streaming* é uma forma ou técnica de transferência de dados, onde a transmissão dos dados ocorre em fluxo contínuo assim permitindo a reprodução do conteúdo antes do término do seu envio, possibilitando a visualização das informações praticamente em tempo real (LINS, 2009). Conforme visualizado na Figura 3, o processo representado inicia pela captura de áudio ou/e vídeo, em seguida estes são codificados conforme formato definido, no caso MPEG-2 e encaminhada ao servidor de origem, o qual armazenará os arquivos e realizará a transmissão para o cliente solicitante. Isto ocorre nas transmissões de rádio e

televisão, onde é chamado de *broadcast*. O *streaming* em *broadcast* possui um único transmissor e vários receptores, porém apesar da nomenclatura diferente possuem a mesma lógica de funcionamento.



Fonte: asaweb (2013)

Convencionalmente a visualização de arquivos de mídia através da internet era realizada através do download completo do arquivo a ser visualizado, para após o término realizar a sua execução. A técnica de *streaming* torna o processo mais ágil, pois o início de visualização da mídia ocorre em curto espaço de tempo. (VIEIRA, 2014).

Ainda segundo VEIRA (2014), para que seja possível transmitir e assistir concomitantemente, o *streaming* utiliza um sistema de *buffer*, onde ele armazena o conteúdo transmitido por alguns segundos e depois realiza a reprodução para o usuário final, para sincronizar o download do conteúdo e a exibição do mesmo. A transmissão sofre um pequeno atraso, tecnicamente conceituado como *delay* (designação relacionada ao retardo de sinais em circuitos eletrônicos), para o usuário, que ocorre devido ao processo de transferência dos dados para o *buffer*, para que todo o conteúdo seja transmitido sem cortes ou falhas.

Os sistemas de *streaming* podem ser divididos em dois tipos: ao vivo (*live*) e sob demanda (*on demand*), (WILSON, 2015). *Streaming* ao vivo é a transmissão de

mídia em tempo real, simultaneamente a sua execução ou realização. Este tipo de *streaming* é realizado em quatro etapas: captura do conteúdo, conversão, transmissão e a fonte propagadora (servidor de *streaming* ou componente do processo responsável por centralizar os dados enviados).

A captura do conteúdo é o processo de obtenção do conteúdo que será transmitido. A conversão é a etapa em que ocorre a transformação do formato do conteúdo, para o formato a ser codificado, através de um programa de compactação (*encoder*). A etapa de transmissão consiste no transporte do conteúdo convertido para os servidores, como via internet. E por fim, os servidores de *streaming* realizam a distribuição do conteúdo. No *streaming* sob demanda, os dados são enviados ao servidor de armazenamento e disponibilizados para acesso conforme a solicitação, assim são transmitidos conforme a demanda.

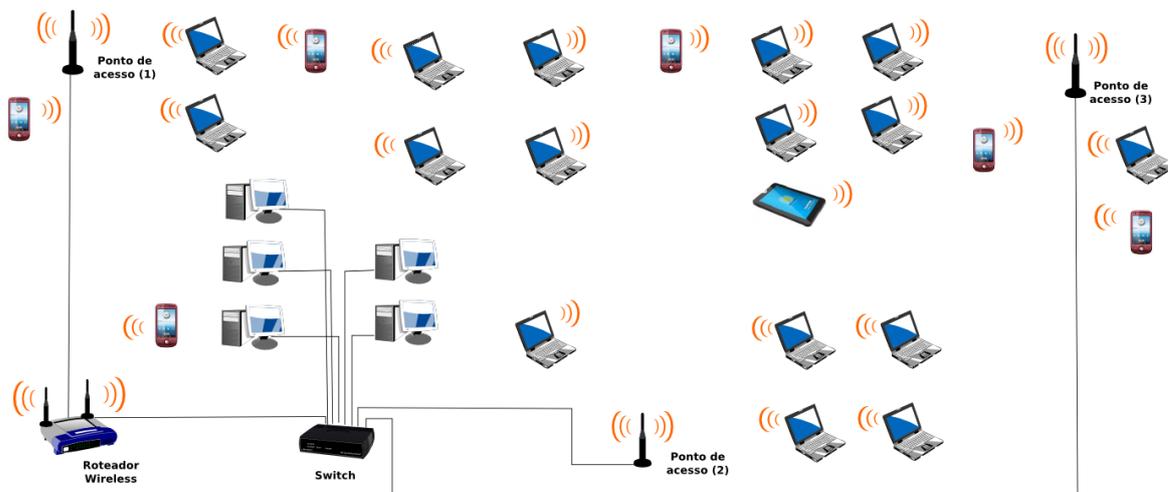
### 2.3 REDES WI-FI

A transmissão dos dados através de cabos é a forma mais rápida de emissão de informações, porém em certos ambientes este tipo de comunicação acaba se tornando impraticável ou inviável, levando em conta custos e trabalho necessário para passagem do cabeamento (MORIMOTO, 2008). As redes sem fio são uma forma de conexão onde a transmissão dos dados ocorre sem a utilização de fios ou cabos, realizada através de ondas eletromagnéticas ou radiofrequências.

Também chamadas de WLAN ou Wi-Fi, as redes sem fio permitem maior flexibilidade em relação à rede cabeada, como a facilidade da troca de máquinas para outros locais, sendo este um processo que acaba sendo "limitado" em conexões cabeadas. A disposição física neste tipo de rede é semelhante à cabeada, diferindo apenas por alguns equipamentos necessários (MORIMOTO, 2008).

Em uma rede sem fio, o *switch* é substituído pelo ponto de acesso (*access-point* em inglês, comumente abreviado como "AP" ou "WAP", de "*wireless access point*"), que tem a mesma função central que o *switch* desempenha nas redes com fios: retransmitir os pacotes de dados, de forma que todos os dispositivos da rede os recebam. Ainda, segundo MORIMOTO (2008), a topologia é semelhante à das redes de par trançado, com o *switch* central substituído pelo ponto de acesso. A diferença é que são usados transmissores e antenas em vez de cabos, conforme Figura 4.

Figura 4 - Esquema de rede sem fio



Fonte: MORIMOTO (2011)

Existem diferentes padrões de redes sem fio atualmente: 802.11a, 802.11b, 802.11g e 802.11n visualizados na Tabela 1.

Tabela 1 – Comparativo dos padrões 802.11

Padrão	802.11b	802.11a	802.11g	802.11n
Frequência	2400-2483.5GHz (2,4GHz)	5150-5250 GHz 5250-5350 GHz 5725-5825 GHz (5,7GHz)	2400-2483.5GHz (2,4GHz)	2,4GHz e 5GHz
Banda	ISM	UNII	ISM	ISM, UNII
Número de canais não sobrepostos	3	12	3	Mesmo 802.11b/a/g
Largura de banda	20Mhz	20Mhz	20Mhz	20Mhz ou 40Mhz
Taxa máxima de transmissão	11Mbps	54Mbps	54Mbps	144Mbps até 600Mbps
Throughput	< 5Mbps	< 32Mbps	< 32Mbps	< 80Mbps < 160Mbps
MAC	CSMA/CA	CSMA/CA	CSMA/CA	CSMA/CA
Tecnologia de modulação	DSSS	OFDM	OFDM DSS	OFDM/OFDMA com MIMO
Vantagens	Custo reduzido e intervalo de frequência satisfatório	Rápido, menos propenso a interferência	Rápido, intervalo de frequência satisfatório	Taxas de dados muito boas, intervalo aprimorado
Desvantagens	Lento, propenso a interferência	Custo mais elevado, intervalo de frequência mais curto	Propenso a interferência de eletrodomésticos que funcionam na faixa de 2.4GHz	

Fonte: Ti-redes (2014)

O padrão 802.11a opera numa frequência de 5GHz, apresentando velocidade superior ao padrão 802.11b, porém com alcance menor. O padrão 802.11b oferece velocidade máxima de 11 Mbps e alcance máximo operacional de 100 metros (MORIMOTO, 2011), sendo relevante a distância do ponto de acesso crucial na velocidade de acesso. Este padrão opera na faixa de frequência de 2.4 Ghz, podendo receber e exercer interferência em outros equipamentos que trabalham nesta frequência, como telefones sem fio, celulares e fornos de micro-ondas.

O padrão 802.11g pode ser considerado uma combinação das redes 802.11a e 802.11b, utilizando a frequência de 2.4 Ghz. O alcance é o mesmo que 802.11b, não sendo compatível com o padrão “a”. O padrão 802.11n possui taxa de transferência de 65 Mbps a 300 Mbps, faixa de frequência: 2,4 GHz e/ou 5 GHz.

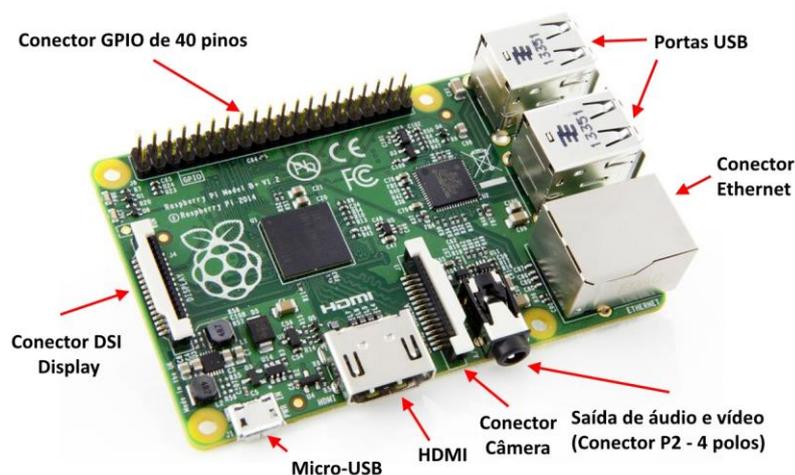
A maioria dos dispositivos disponíveis, no mercado, atualmente já permitem o acesso a este tipo de rede, como *notebooks*, *smartphones* e televisores, que já possuem um módulo Wi-Fi instalado. Porém em certos dispositivos a conexão *wireless* não é possível por padrão. Assim torna-se indispensável o uso de acessórios que permitam estes equipamentos utilizarem as redes sem fio como meio de comunicação, como uma placa Wi-Fi ou um adaptador USB Wi-Fi.

## 2.4 RASPBERRY PI

O Raspberry Pi é uma plataforma de mini microcomputador que possui o tamanho aproximado de um cartão de crédito. Foi idealizado por um grupo de universitários de Cambridge, com o objetivo de desenvolver um equipamento de baixo custo para auxiliar no processo educacional dos alunos da Universidade, reaproximando-os do computador, principalmente da experiência da programação.

O Raspberry Pi pode ser considerado um computador “completo”, podendo ser classificado como SBC (*Single-Board Computer*), pois integra em uma única placa processador, memória e interfaces de entrada e saída, como mostrado na Figura 5. O Raspberry Pi do modelo 2B utiliza um processador ARM (Cortex A7 fabricado pela Broadcom, que opera a 900 MHz).

Figura 5 – Raspberry Pi Modelo B+



Fonte: Filipeflop (2014)

Esta arquitetura de processadores também é muito utilizada em equipamentos como *smartphones*, PDAs e sistemas embarcados.

Em virtude de o microcomputador Raspberry ser baseado em um *chipset* de dispositivo móvel, ele tem requisitos de software diferentes de um computador desktop. O conjunto de instruções do processador ARM possui um conjunto de instruções diferente da arquitetura presente nos computadores pessoais (x86 e AMD64) e por isto o sistema operacional, os *drivers* de dispositivos e programas precisam ser desenvolvidos e compilados para este conjunto de instruções.

O Raspberry Pi se enquadra na categoria de equipamento denominada *open-source hardware* (OSH), ou equipamento livre. Os OSHs podem ser definidos como dispositivos elétricos ou mecânicos cujas informações sobre o projeto são disponibilizadas para todos, de forma que o público possa fazer uso, modificar, produzir ou distribuir este material (SILVA e PEREZ,2013 apud TAPR, 2007).

## 2.5 SISTEMAS OPERACIONAIS PARA SISTEMAS EMBARCADOS

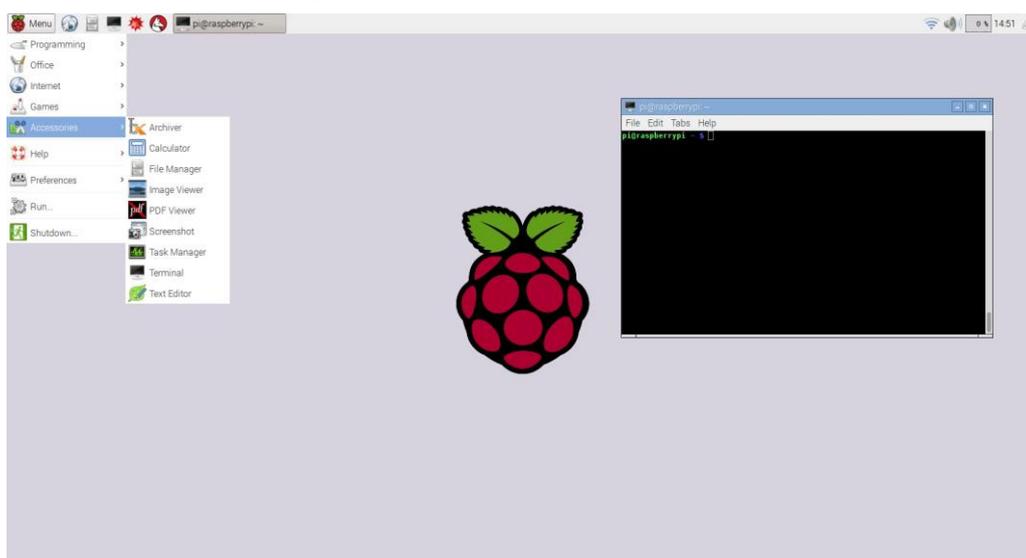
O Raspberry Pi é um microcomputador, no qual o seu sistema é instalado em um cartão de memória. Assim, para que ocorra seu funcionamento é necessária a utilização de um sistema operacional embarcado, o qual possui a função de gerenciamento da plataforma e dos recursos, conforme descreve LIMA et al. (2015).

Para que se torne utilizável o Raspberry Pi necessita da instalação de um sistema operacional. Que permite o gerenciamento dos processos a nível de *software* e também de *hardware*, pois diferente de um PC, o Raspberry não possui BIOS ou Setup. Em vez disso, todas as configurações relacionadas ao hardware e ao processo de inicialização são feitas em um arquivo de texto localizado no diretório raiz do cartão, o "config.txt". Este arquivo engloba muitas opções que em um PC estariam disponíveis no Setup, incluindo a frequência de operação do processador, (PEREIRA e JUCÁ, 2015).

Desta forma, surgiram vários sistemas operacionais, em sua grande maioria baseados em distribuições Linux (LIMA et al., 2015), específicos para utilização e gerenciamento do Raspberry Pi, como Ubuntu Mate, Snnapy Ubuntu Core, OSMC, Openelec, Pinet, Risc OS e Windows 10 IOT Core (RASPBERRY.ORG, 2015), além da Raspbian, visualizado na Figura 6, distribuição oficialmente recomendada pela fundação Raspberry (RASPBERRY.ORG, 2015).

O Raspbian é uma derivação da distribuição Linux Debian, desenvolvida especialmente para utilização na plataforma Raspberry Pi (FRPi, 2015, apud LIMA et al., 2015). Possui cerca de 35.000 pacotes *deb*, ou *softwares*, pré-compilados para utilização.

**Figura 6 – Interface gráfica do Raspbian**



**Fonte: Distrowatch (2016)**

O Raspbian também é otimizado para obter o desempenho máximo do microcomputador Raspberry. O Raspbian contém o ambiente de desktop LXDE, o gerenciador de janelas OpenBox, o navegador Midori, ferramentas para desenvolvimento de software e código fonte de exemplo de funções multimídia (LIMA et al., 2015).

## 2.6 UV4L

O UV4L (Userspace Video4Linux2) é um conjunto de módulos multiplataforma ou *framework* para dispositivos de entrada de vídeo e saída real ou virtual e outros *back-ends* (interface de interação com o usuário) ou *front-ends* (webservices e APIs de aplicação) conectáveis, e que consiste em uma série de drivers, um servidor de *streaming* e várias extensões para o servidor (LINUX-PROJECTS, 2016).

Este *framework* é composto por vários módulos, conforme mostrado na lista abaixo (LINUX-PROJECTS, 2016):

- Módulo central UV4L;
- *Servidor de streaming* com interface de interação com o usuário via Web através de HTTP / HTTPS e controle de dispositivos;
- Extensão nativa WebRTC para o servidor de fluxo contínuo (bidirecional Áudio-Vídeo- Dados! Áudio / vídeo espelhamento)
- Extensão para transmissão ao vivo de áudio e vídeo através da Web (sem necessidade de navegador), através do *software* Jitsi;
- Suporte para o *gateway* Janus WebRTC, que permite participar de salas de videoconferência na nuvem;
- *Driver* UVC, suporta câmeras compatíveis com as especificações da classe de vídeo USB;
- *Driver* XScreen, um dispositivo virtual para captura de uma determinada parte de uma tela;
- *Driver* MJPEGStream, que cria um dispositivo virtual a partir de um determinado fluxo Motion JPEG, ou seja, câmera IP;
- *Driver* Raspicam, para todos os módulos de câmera para Raspberry Pi;
- *Driver Dummy*, permite criar um dispositivo falso (útil para testar aplicações v4l2).

Para a utilização no Raspberry Pi, todas as configurações que necessitem de alteração ou de ajuste, são realizadas através do arquivo “uv4l-raspicam.conf”. Entre os parâmetros que podem ser definidos estão, resolução, formato, porta utilizada para *streaming*, além de customizações como senha de acesso para área administrativa e usuários comuns.

## 2.7 MONITORAMENTO DE REDES

O monitoramento de redes de computadores é fundamental para o conhecimento da atual situação do andamento da mesma, para que não ocorram problemas como lentidões e travamentos. Existem várias ferramentas utilizadas pelos administradores de rede para verificação da rede e principalmente o monitoramento da largura de banda, que auxiliam nesta coleta de informações.

Existem soluções como MRTG e o RRDTOOL, que utilizam o SNMP (*Simple Network Messaging Protocol*) e analisam dados com base em um período de tempo, apresentando dados estatísticos, fornecendo uma gama de informações, como tráfego da rede, largura da banda e monitoramento de ativos de rede, sendo soluções muito utilizadas (LPICERTIFICACAOLINUX, 2012).

Porém outras ferramentas se baseiam no monitoramento em tempo real, como o Iftop, que é um mecanismo de monitoramento baseado em linha de comando o qual demonstra conexões de redes ao invés de processos. Esta solução possui como principal característica a visualização do consumo da banda de rede. A qual, permite ainda visualizar as conexões de rede entre dois endereços IP's e quantos pacotes de dados atravessam o link (LUDOLF, 2013).

Ele usa a biblioteca pcap para capturar os pacotes de entrada e saída da interface de rede e, em seguida, calcular o tamanho da largura de banda total utilizada no instante, podendo gerar relatórios de um determinado dispositivo da rede, conforme filtro aplicado (FERREIRA, 2014).

A ferramenta apresenta uma visão geral de todas as conexões ativas, de todas as interfaces de rede que estejam trafegando dados e todos os protocolos utilizados neste exato momento. Do lado esquerdo, são mostrados os dispositivos de origem da interface assim como do lado direito, são mostrados os dispositivos de destino. As conexões são mostradas em ambos os sentidos determinado pelas setas =>

(origem/destino, tráfego de saída) e =< (destino/origem, tráfego de entrada). Nas 3 últimas colunas, são mostradas, para cada conexão, períodos de tempos em 2, 10 e 40 segundos. A parte inferior mostra a relação de TX (transmissão) e RX (recebimentos) assim como a média (rate) das colunas 2, 10 e 40 segundos (IPICERTIFICACAOLINUX, 2012).

## 2.8 CODIFICADORES E DECODIFICADORES

Com a popularização do *streaming*, surgiram inúmeras possibilidades em relação a este modo de transmissão de mídia, sejam elas músicas ou/e vídeos, realizados tanto pelo método sob demanda ou ao vivo. Ao mesmo tempo foram implementados vários métodos de compressão e codificação do arquivo a ser transmitido, cada um com suas vantagens e desvantagens dependendo do tipo de arquivo e transmissão realizados. Porém para que esta mídia possa ser visualizada no dispositivo solicitante é necessário que o mesmo possua um agente (*codec*) que é responsável por recuperar os dados compactados possibilitando sua exibição (HUSEMANN et al., 2006).

Atualmente, no mercado já existem várias soluções que permitem a realização deste processo, sejam elas através do próprio codec que deve instalado no computador, como aplicações que já possuem embutidas estes *codecs* para uso. O VLC (VideoLAN Client) é um reprodutor multimídia livre, de código aberto, multiplataforma, e que reproduz a maioria dos arquivos de mídia, bem como DVD, CD de áudio, VCD e vários protocolos de fluxo de rede (VIDEOLAN.ORG, 2016). Esta ferramenta já possui por padrão a maioria dos *codecs* mais utilizados, não sendo assim necessária a instalação de um pacote de codificadores. Ele reproduz arquivos, discos, *webcams*, dispositivos e envio/visualização de multimídia via internet.

### 3 DESCRIÇÃO METODOLÓGICA

Este capítulo apresenta o ambiente proposto, assim como a descrição das tecnologias e materiais aplicados no processo definido, bem como de cada etapa necessária durante o desenvolvimento.

#### 3.1 DESCRIÇÃO DO PROJETO

A seguir é descrito o projeto realizado, expondo os principais materiais utilizados para aplicação dos testes posteriormente descritos.

Para a implementação do sistema foi utilizada a plataforma Raspberry Pi 2 Modelo B, ou especificado também como 2B. Esta placa foi escolhida por ser o modelo com maior capacidade de processamento (*RASPBERRYPI.ORG*, 2015). Atualmente, este é o modelo com processador de maior frequência e também com maior memória RAM, conforme comparativo da Tabela 2.

**Tabela 2 – Comparativo modelos Raspberry Pi - 2015.**

Modelo	Processador	RAM	Ethernet	USB	Alimentação	Lançamento
A	700	256MB	Não	1	5V – 500mA	2012
B	700	512MB	Sim	2	5V – 1A	2013
B+	700	512MB	Sim	4	5V – 2A	2014
2B	900	1GB	Sim	4	5V – 2A	2015

**Fonte: Adaptado de Lima et al, 2015.**

Como sistema operacional foi utilizado o Raspbian, que é uma distribuição Linux com interface gráfica desenvolvida para uso específico neste tipo de microcomputador (*RASPBERRYPI.ORG*, 2015). Foi preparado um cartão microSD de 16GB classe 10 para gravação da imagem da distribuição Linux, com o auxílio da ferramenta Win32DiskImager, que é uma aplicação open source que permite gravar imagens para o USB ou cartão SD, criando uma unidade de disco virtual (*SOURCEFORGE*, 2016).

Apesar de serem encontradas outras distribuições, o Raspbian foi definido por ser uma distribuição otimizada para o hardware do Raspberry Pi e através da sua interface gráfica facilita a realização dos testes dos componentes utilizados em

conjunto com a plataforma, além de exigir menor espaço de armazenamento, exige um cartão microSD de 4GB (RASPBIAN.ORG, 2016), em relação a outras distribuições como o Ubuntu Mate, que exige no mínimo um cartão de 8GB (UBUNTUMATE.ORG, 2016) e ser a distribuição oficialmente recomendada pela Fundação Raspberry (RASPBERRYPI.ORG, 2015).

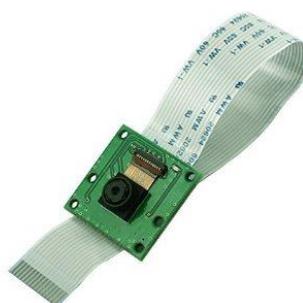
A câmera escolhida foi a Raspberry Pi Câmera, que permite capturar vídeos de até 1080p e possui peso e medidas reduzidas, conforme especificações na Tabela 3. Que facilitam a integração para aplicações móveis e situações que fatores como estes influenciam na mobilidade e aplicação do projeto. A câmera é mostrada na Figura 7.

**Tabela 3 - Detalhes da câmera Raspberry Pi.**

<b>Característica</b>	<b>Valor</b>
Dimensão	25 mm x 20 mm x 9 mm
Peso	3 g
Resolução máxima de captura de imagem	5 Megapixel (2592 x 1944)
Modelo do sensor	OmniVision OV5647
Tamanho do sensor	3.67 x 2.74 mm
Formato ótico	1/4" (Fator de corte: 9.4)
Lente	f=3.6 mm, f/2.9
Tamanho do foco	4 mm
Alguns formatos de captura de vídeo padrão	- 2592x1944 1-15fps - 1920x1080 1-30fps - 1296x972 1-42fps - 640x480 42.1-60fps

**Fonte: Adaptado de Home Automation Community (2016).**

**Figura 7 - Câmera Raspberry Pi**



**Fonte: Amazon (2016)**

Para que o microcomputador realize o envio de vídeo através da rede sem fio foi utilizada uma interface de rede sem fio USB, Railink RT5370N, da Mediatek. Este adaptador é conectado por uma interface USB 2.0. Este dispositivo pode operar em redes de padrão até 802.11n, capaz de obter taxas de transferência de até 150Mbit/s (MEDIATEK, 2016).

Para envio dos quadros capturados pela câmera do Raspberry será aplicado o método de streaming de vídeo do tipo live, que permitirá um acompanhamento em tempo real dos dados recebidos no computador cliente. Já o microcomputador será responsável pelo processamento, codificação e compactação dos dados para envio via interface de rede sem fio, realizando o papel de servidor de streaming no processo.

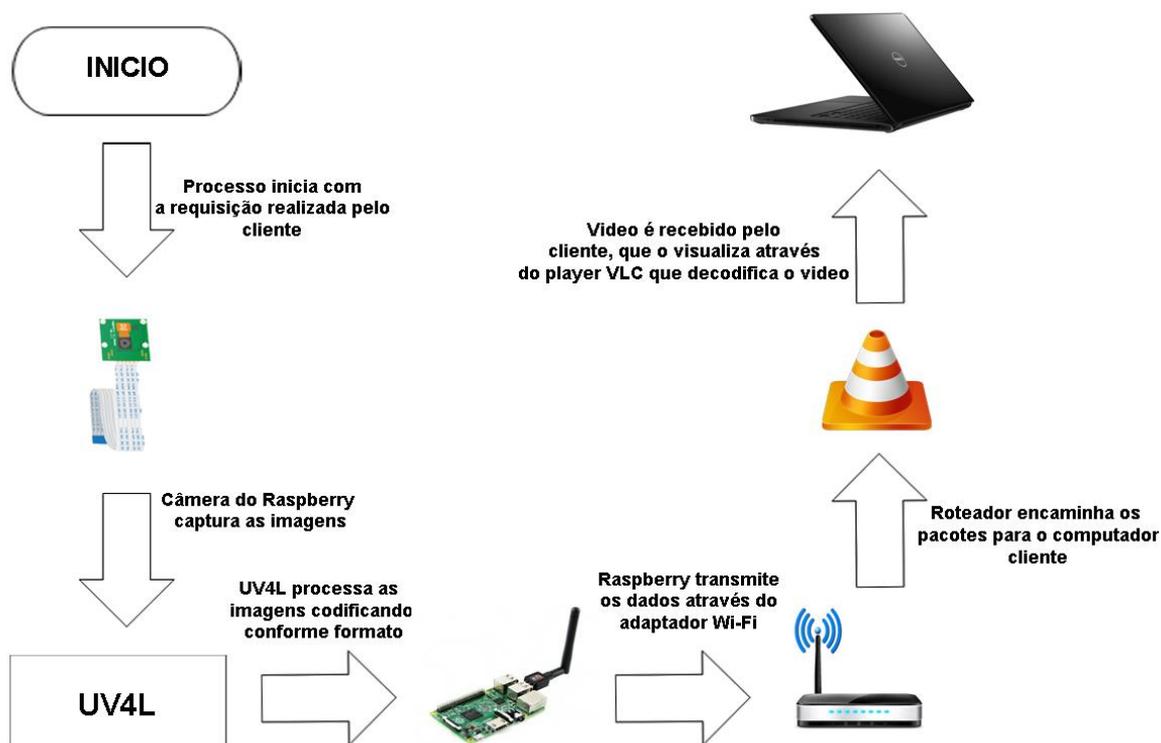
Para o processo de codificação e compactação será utilizado o *framework* UV4L, que já possui por padrão os drivers necessários para captura através do módulo da câmera Raspberry Pi, além de um servidor de *streaming* funcional, concentrando todas as configurações que se tornem necessárias no arquivo `uv4l-raspicam.conf`.

### 3.2 DESCRIÇÃO DO AMBIENTE

Para a realização dos testes foi utilizado um roteador TP-LINK AC 750 Dual Band, Archer C2 para criar uma rede sem fio dedicada para a aplicação. Como computador cliente Windows, com o software VLC instalado para decodificação/reprodução do *streaming*.

O funcionamento da interface de *streaming* se realizará da seguinte maneira: os quadros capturados pela câmera são processados pelo Raspberry Pi através do UV4L, compactados e codificados, em seguida através do adaptador Wi-Fi os dados são enviados para um computador cliente por meio de uma rede sem fio, sendo decodificados para visualização nesta máquina.

Na Figura 8 são mostrados os processos realizados no streaming de vídeo, apresentando desde a captura dos quadros de imagens até a visualização pelo computador cliente.

Figura 8 – Processo de *Streaming*

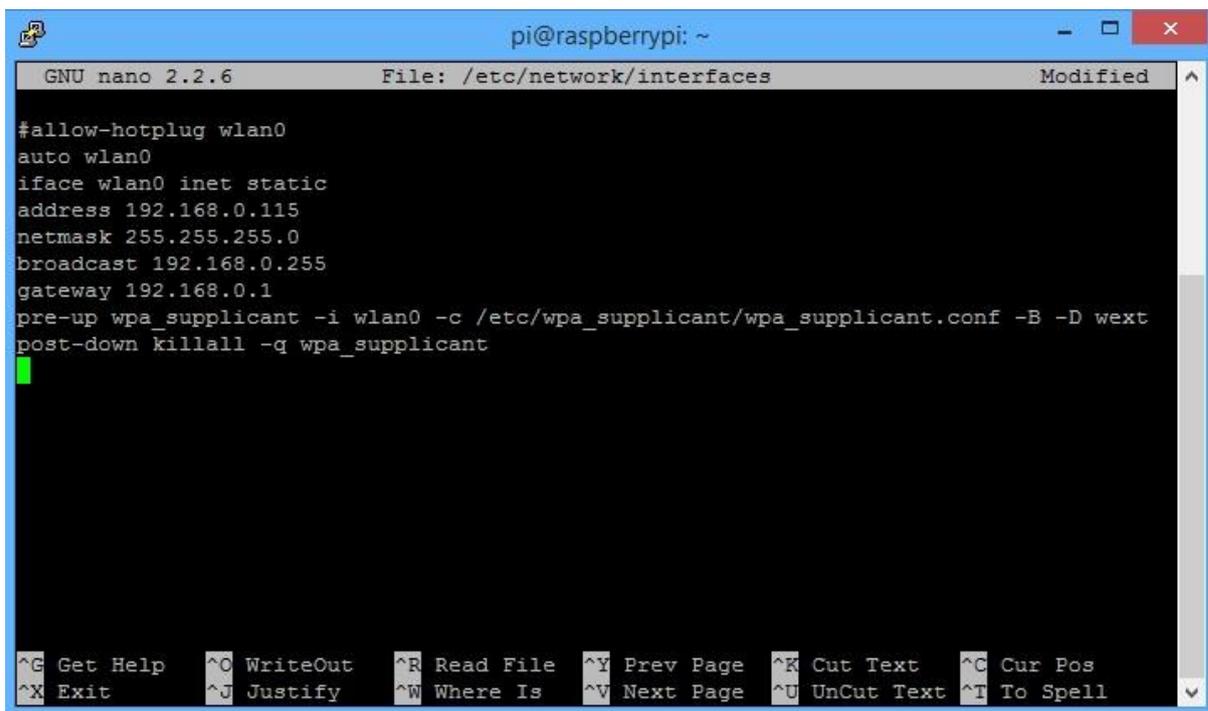
Fonte: Do autor

### 3.3 CONFIGURAÇÕES E FERRAMENTAS ADICIONAIS

Para auxiliar na execução dos testes e tornar os dispositivos funcionais foram utilizadas algumas ferramentas adicionais para ajuste dos mesmos.

Para reconhecimento automático do adaptador de rede USB, foi utilizado o utilitário `wpa_supplicant`, que em distribuições Linux é o responsável pelo gerenciamento de redes sem fio (MORIMOTO, 2008). Basicamente os procedimentos realizados neste caso, foram a geração do arquivo `wpa_supplicant.conf` (comando: `# wpa_passphrase minharede minhapassphrase > /etc/wpa_supplicant.conf`) e as definições de rede (IP, máscara, local do arquivo de configuração do `wpa_supplicant`) no arquivo `/etc/network/interfaces`, para reconhecimento deste componente no boot (inicialização do sistema Raspbian) sem a necessidade de ativá-lo manualmente sempre, conforme verificado na Figura 9, que apresenta o arquivo `interfaces` mencionado.

Figura 9 – Configurações arquivo interfaces



```
pi@raspberrypi: ~  
GNU nano 2.2.6 File: /etc/network/interfaces Modified  
#allow-hotplug wlan0  
auto wlan0  
iface wlan0 inet static  
address 192.168.0.115  
netmask 255.255.255.0  
broadcast 192.168.0.255  
gateway 192.168.0.1  
pre-up wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf -B -D wext  
post-down killall -q wpa_supplicant  
  
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos  
^X Exit         ^J Justify     ^W Where Is    ^V Next Page    ^U UnCut Text  ^T To Spell
```

Fonte: Do autor

Com o objetivo de reduzir o número de dispositivos conectados ao microcomputador e dispensáveis ao projeto, como teclado, *mouse*, foi utilizado acesso remoto. O PuTTY é um programa cliente para protocolos de rede SSH, Telnet e Rlogin, que permite o acesso a máquinas Linux através de computadores Windows, possibilitando a configuração e manipulação do mesmo em um console de modo texto. Para acesso no Raspberry Pi foi necessário apenas a ativação do protocolo SSH.

O TightVNC é uma outra opção de ferramenta de controle remoto. Com TightVNC, é possível ver o ambiente gráfico de uma máquina remota e controlá-lo com o mouse e teclado locais. Assim foi realizada a instalação do pacote correspondente no microcomputador (*tightvnc-server*) e executado o *software* cliente na máquina Windows (*VNC viewer*).

### 3.4 CENÁRIOS APLICADOS

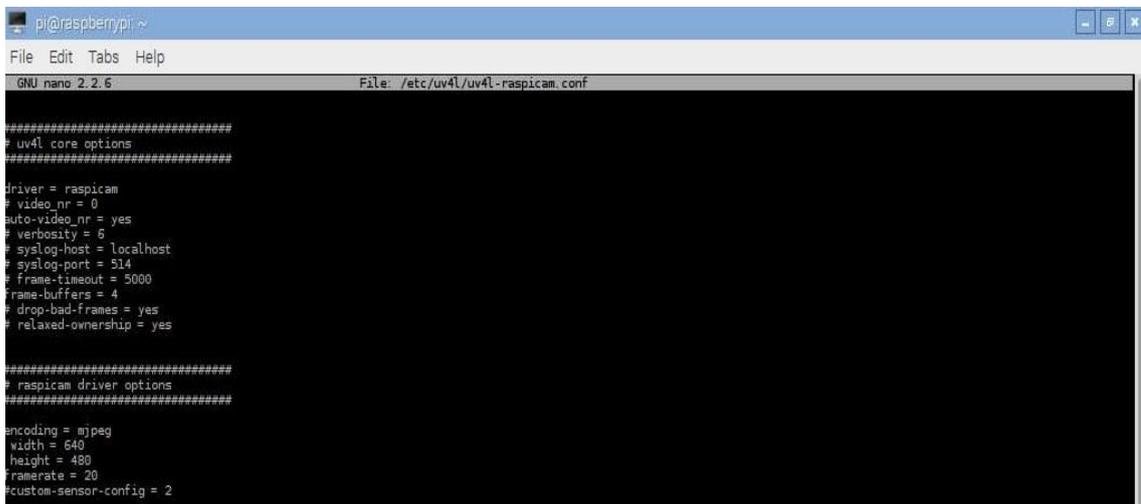
Com o objetivo de realizar a comparação das configurações disponíveis no UV4L para utilização com a Raspicam, câmera do Raspberry Pi, foram definidos dois

cenários para análise baseados em pontos como formato do arquivo transmitido e resolução da câmera.

### 3.4.1 Streaming em resolução de 640x480

Para aplicação deste cenário foi definido o parâmetro *width* (largura) para 640 e *height* (altura) para 480, definido o campo *encoding* (codificação) como mjpeg e mantido o campo de *framerate* (taxa de quadros), no seu valor padrão, 20, conforme Figura 10.

Figura 10 – Configuração de resolução em 640x480



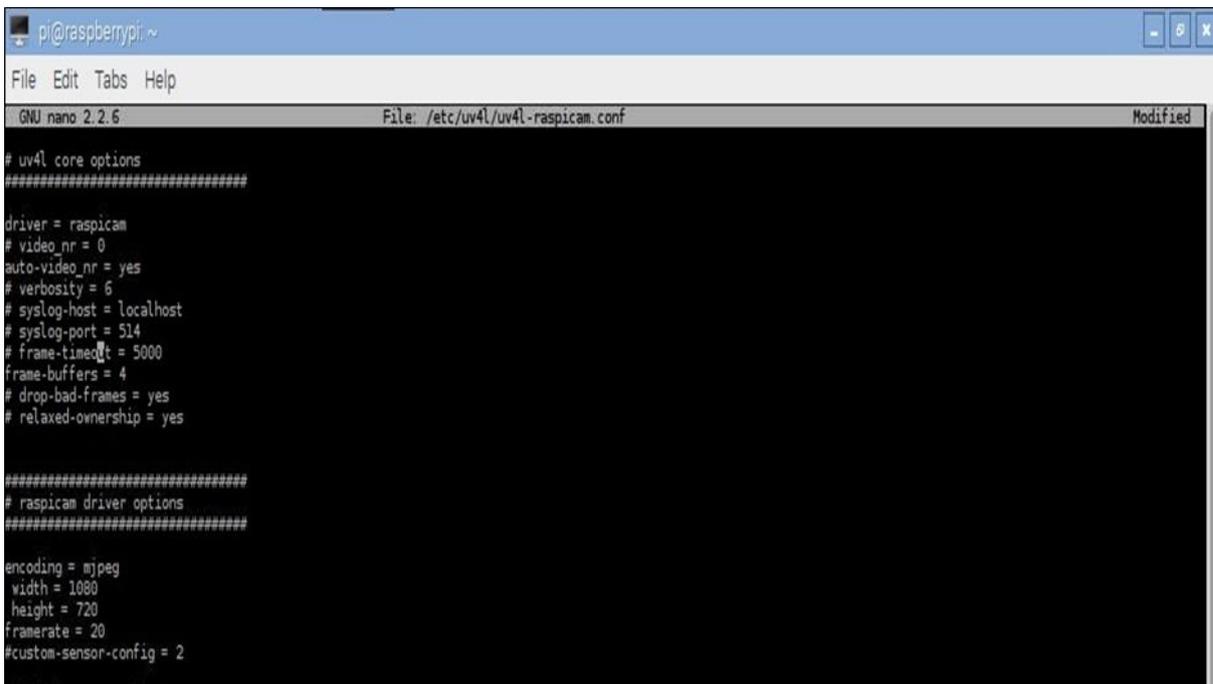
```
pi@raspberrypi ~  
File Edit Tabs Help  
GNU nano 2.2.6 File: /etc/uv4l/uv4l-raspicam.conf  
#####  
# uv4l core options  
#####  
driver = raspicam  
# video_nr = 0  
# auto-video_nr = yes  
# verbosity = 6  
# syslog-host = localhost  
# syslog-port = 514  
# frame-timeout = 5000  
# frame-buffers = 4  
# drop-bad-frames = yes  
# relaxed-ownership = yes  
  
#####  
# raspicam driver options  
#####  
encoding = mjpeg  
width = 640  
height = 480  
framerate = 20  
#custom-sensor-config = 2
```

Fonte: Do autor.

### 3.4.2 Streaming em resolução de 1080x720

Para aplicação do cenário 2 foi definido o parâmetro *width* (largura) com valor 1080 e *height* (altura) como 720. Como método de *encoding* (codificação) foi mantido o formato mjpeg, assim como mantido o campo de *framerate* (taxa de quadros) visualizado na Figura 11.

Figura 11 – Configuração de resolução em 1080x720



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/uv4l/uv4l-raspicam.conf Modified
# uv4l core options
#####
driver = raspicam
# video_nr = 0
auto-video_nr = yes
# verbosity = 6
# syslog-host = localhost
# syslog-port = 514
# frame-timeout = 5000
frame-buffers = 4
# drop-bad-frames = yes
# relaxed-ownership = yes

#####
# raspicam driver options
#####
encoding = mjpeg
width = 1080
height = 720
framerate = 20
#custom-sensor-config = 2
```

Fonte: Do autor

### 3.4.3 Pontos de avaliação

Como pontos de avaliação foram verificados três quesitos: resolução, qualidade da imagem capturada; consumo de banda da rede, utilização da rede durante o *streaming* e atraso, período necessário para que uma imagem capturada pela câmera possa ser visualizada pelo computador cliente, definidos com o objetivo de verificar entre os cenários aplicados qual obteve o melhor comportamento em uma rede sem fio.

Para a qualidade de imagem, foi verificada a visibilidade com que os objetos capturados pela câmera eram visualizados no computador cliente, comparando os dois processos de streaming realizados.

Para o consumo de banda da rede foi utilizada a ferramenta Iftop, utilitário para monitoramento da rede que demonstra estas conexões através de estatísticas em tempo real. Foram capturados os índices de uso da rede durante o processo de *streaming* em cada cenário durante o período de três minutos. Foram verificados os dados da ferramenta nos campos de TX (transmissão), períodos de tempos de 2, 10 e 40 segundos, além da média (*rate*) destas três colunas.

Para a verificação do atraso no vídeo, foi cronometrado o tempo necessário para que um determinado objeto retirado do campo de visão da câmera também não fosse mais visualizado pelo dispositivo cliente.

## 4 RESULTADOS

Após a aplicação dos testes nos cenários definidos, foram observados os resultados abaixo, conforme os critérios pré-definidos.

### 4.1 RESOLUÇÃO

De acordo com os testes realizados foi verificado que em ambos os casos a nitidez de imagem recebida no computador cliente foi equivalente a resolução definida e aproximada da visibilidade que a câmera oferece, conforme é observado na Figura 12 da transmissão em 640x480.

**Figura 12 – Streaming transmitido com resolução 640x480**



Fonte: Do autor

Assim, como no segundo cenário, também foram obtidas imagens nítidas no dispositivo cliente, com os vídeos transmitidos na resolução de 1080x720, conforme visualizado na Figura 13.

**Figura 13 - Streaming transmitido com resolução 1080x720**

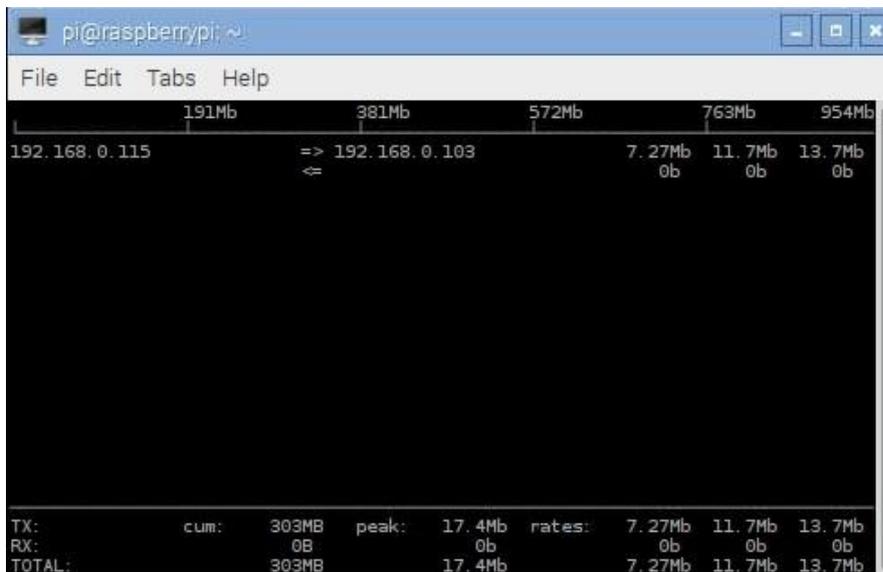


Fonte: Do autor

## 4.2 CONSUMO DE BANDA DE REDE

Através da ferramenta Iftop (LPICERTIFICACAOLINUX, 2012), foram capturados os valores informados pela ferramenta durante o processo de *streaming*. No cenário 1, conforme Figura 14, após o tempo estabelecido foi verificado o valor de 303MB transmitidos, além de possuir nos períodos de 2, 10 e 40 segundos as médias de 7.27Mb, 11.7Mb e 13.7Mb, respectivamente, as quais de mantiveram durante o processo de *streaming*.

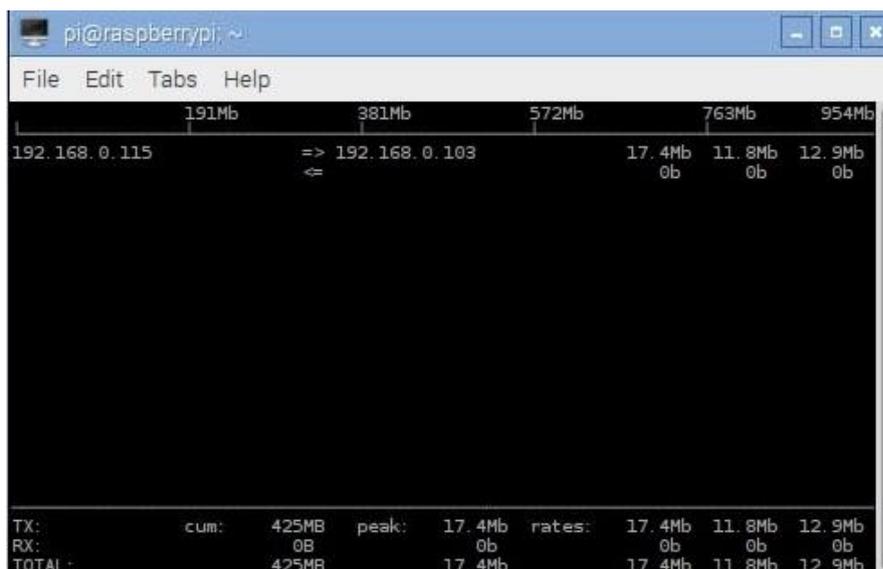
Figura 14 – Captura ferramenta lftop cenário 1



Fonte: Do autor

Na verificação realizada no cenário 2 se constatou a taxa de 425MB transmitidos no período, como também na coluna de 2 segundos, 17.4Mb, na coluna 10 segundos, 11.8Mb e na terceira de 40 segundos, 12.9Mb. Assim como suas médias se mantiveram nestes valores, conforme verificado na Figura 15.

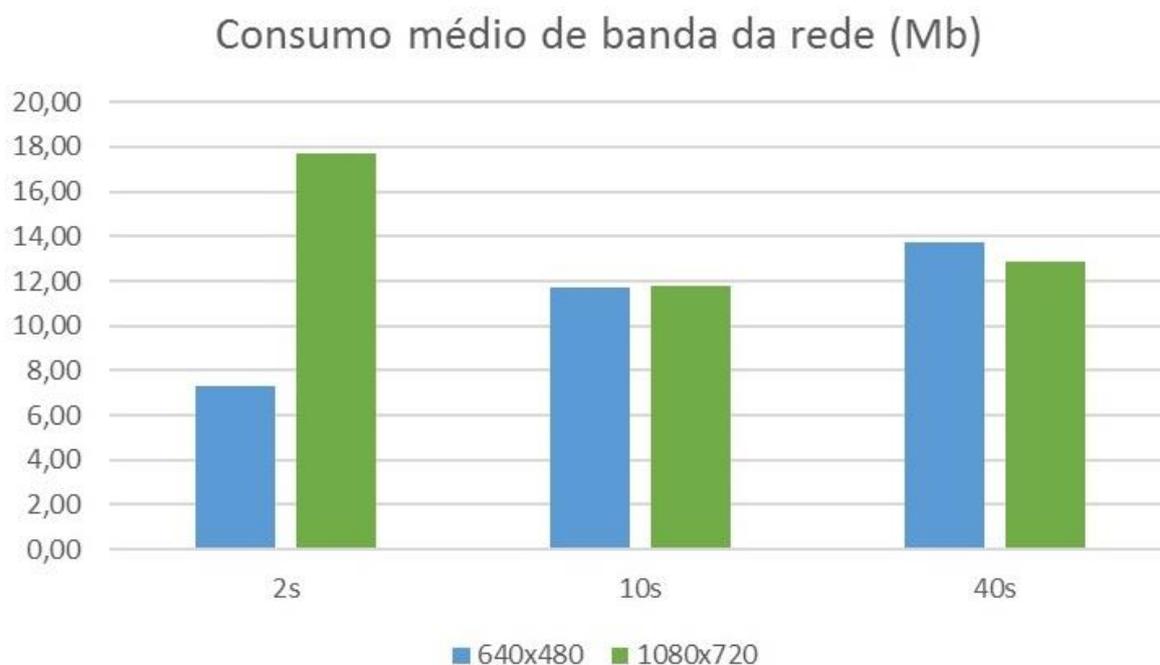
Figura 15 - Captura ferramenta lftop cenário 2



Fonte: Do autor

Na Figura 16 é visualizado um comparativo dos dois resultados, percebendo-se uma discrepância dos dados, principalmente em relação à média na coluna de 2 segundos.

**Figura 16 – Consumo médio da banda de rede**



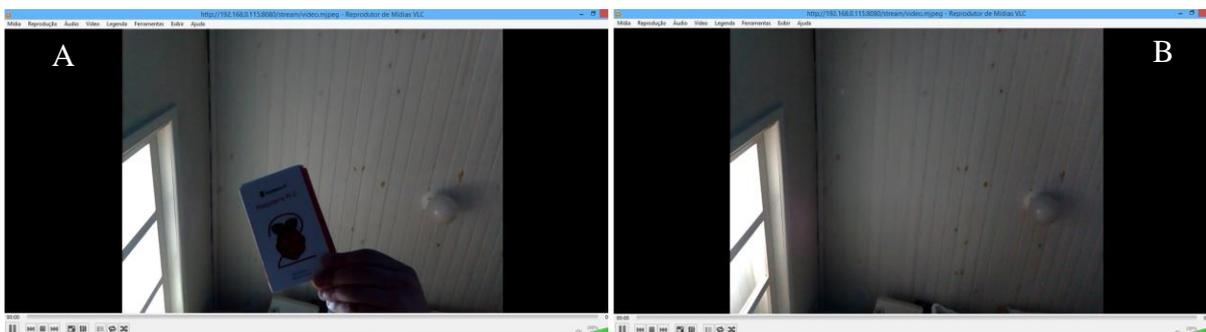
Fonte: Do autor

### 4.3 ATRASO

No cenário 1 foi inserido um objeto em frente a câmera e em seguida realizada a remoção do objetivo da área de captura. Desta maneira foi cronometrado o tempo necessário para a captura ser visualizada pelo computador cliente. Foi constatado o tempo médio aproximado de 2,4 segundos para a transição da Imagem A (esquerda) para a Imagem B (direita), conforme mostrado na Figura 17.

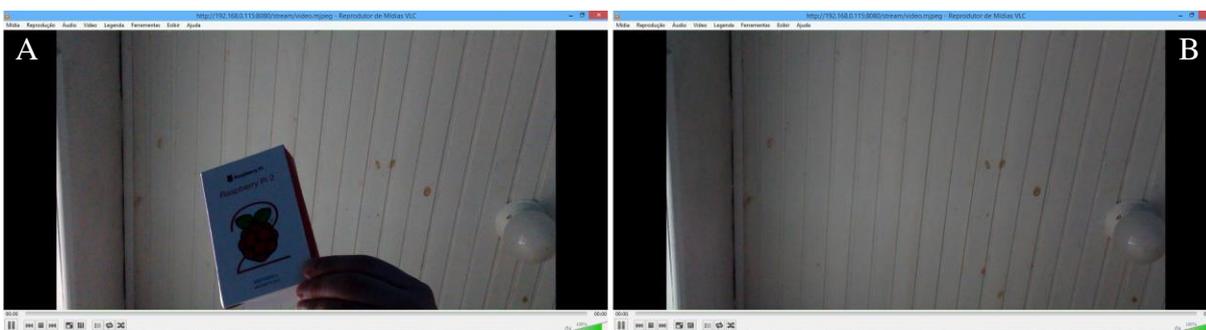
Da mesma forma, no cenário 2, foi realizado o mesmo procedimento com o objeto na frente da câmera e da também cronometrado o tempo que foi necessário para que objeto fosse visualizado pelo destinatário. Desta forma, foi constatado o período aproximado de 5,2 segundos para que ocorresse a transição da imagem A (esquerda) para a imagem B (direita), conforme visualizado na Figura 18.

Figura 17 – Imagem inicial e imagem final resolução 640x480



Fonte: Do autor

Figura 18 - Imagem inicial e imagem final resolução 1080x720



Fonte: Do autor

Assim, foi obtido um atraso menor na resolução de 640x480, aplicada no primeiro cenário, em relação ao segundo cenário com aplicação de uma resolução de 1080x720, que apresentou praticamente o dobro do atraso, em relação a resolução menor.

## 5 CONSIDERAÇÕES FINAIS

O avanço tecnológico ocorre com muita rapidez, o que faz com que novas tecnologias se tornem opção e possam ser aplicadas em diferentes áreas. Aplicações de monitoramento com VANTs crescem a cada dia e procura-se buscar soluções de baixo custo para integração a estes dispositivos, a fim de tornar a sua utilização possível em ambientes do dia a dia.

Deste modo, com a implementação da interface de streaming de vídeo, obteve-se o retorno de vídeo nos dois cenários definidos, com resolução de 640x480 e no segundo com resolução de 1080x720, podendo realizar uma análise e verificação dos pontos testados.

Através dos testes aplicados foi observado que em relação a qualidade de imagem recebida pelo computador cliente, esta foi visualizada de maneira nítida em ambos os cenários. Na verificação de consumo de banda de rede, os dois cenários apresentaram resultados semelhantes na taxa média transmitida, sendo mais elevada conseqüentemente na resolução maior, principalmente em relação a média de 2 segundos e ao total transmitido. Contudo, na verificação de atraso (*delay*) do vídeo, com a aplicação de uma resolução menor se obteve melhores resultados, praticamente reduzindo o tempo de visualização do cliente pela metade.

Assim pode-se dizer que os objetivos foram alcançados, já que a interface de *streaming* se mostrou funcional, rodando normalmente com o gerenciamento do framework UV4L, e com pequeno atraso, conseqüente do processo de *streaming* e de seu modo de funcionamento, com as definições de resolução baixa de vídeo retornando melhor resultado, em relação ao tempo de resposta na máquina cliente.

Como pontos em que há a possibilidade de aprimoramentos e melhorias neste trabalho, através de trabalhos futuros, pode-se citar:

- Aplicação do processo de *streaming* de vídeo em uma rede maior, a fim de verificar a sua estabilidade e alterações perante pontos abordados neste projeto;
- Utilização do formato H264, para fins de comparação, ampliando os testes executados;
- Utilização de outras plataformas embarcadas, com o objetivo de realizar um comparativo com o Raspberry Pi.

- Aplicação do projeto com a utilização de duas câmeras, a fim de testar a funcionalidade para tal funcionamento, disponível no *framework* utilizado.
- Realização de novos testes a partir da integração da interface de *streaming* com o VANT.
- Expansão da interface a partir da possibilidade do armazenamento das imagens capturadas pela máquina cliente.

## REFERÊNCIAS

AMAZON. Disponível em: <<http://ecx.images-amazon.com>>. Acesso em: 15 maio 2016.

ANAC. Disponível em: <<http://www2.anac.gov.br/rpas/>>. Acesso em: 02 jun 2016.

ASAWEB. Disponível em: <<http://www.asaweb.com.br/site/streaming-em-http>>. Acesso em: 17 nov 2015.

CERBARO. Vinícius Andrei. Crop drone: uma solução para coleta, armazenamento e disponibilização de dados agrícolas. Dissertação - Universidade de Passo Fundo - Instituto de Ciências Exatas e Geociências, 2016.

CROTTI, Y. Raspberry Pi e Experimentação Remota. ICBL2013 – International Conference on Interactive Computer aided Blended Learning. 2013. p 143-145.

DISTROWATCH. Disponível em: <<http://distrowatch.com>>. Acesso em: 15 maio 2016.

EISENBEIß, Henri. VANT Photogrammetry. Zurich, University of Technology Dresden. Doctor of Sciences, 2009, p 237.

FAS. Disponível em: <<http://fas.org/man/dod-101/sys/ac/row/hermes-450.htm>>. Acesso em: 15 set 2015.

FERREIRA. Ricardo. 10 Ferramentas para monitorar a largura de banda no Linux. Disponível em: <<http://www.linuxdescomplicado.com.br/2014/12/10-ferramentas-para-monitorar-largura.html>>. Acesso em: 05 maio 2016.

FILIPEFLOP. Disponível em: <<http://blog.filipeflop.com/embarcados/tutorial-raspberry-pi-linux.html>>. Acesso em: 15 set 2015.

HOMEAUTOMATIONCOMMUNITY. Disponível em: <<http://www.home-automation-community.com/surveillance-with-raspberry-pi-noir-camera-howto/>>. Acesso em: 12 mar 2016.

HUSEMANN. Ronaldo. et al. Análise da implementação de algoritmos de codificação e decodificação de vídeo MPEG-2 HD escalável em hardware. Campo Grande, 2006.

LIMA, Victor Gutemberg Santos. et al. Análises de Sistemas Operacionais Linux usando Plataforma Embarcada. Sergipe, 2015.

LINS, Xerxes. Disponível em: <<https://www.vivaolinux.com.br/dica/O-que-e-tecnologia-streaming>>. Acesso em: 09 jun 2016.

LINUX-PROJECTS. Disponível em: <<http://www.linux-projects.org>>. Acesso em: 17 abr 2016.

LONGHITANO, G. A. Vants para sensoriamento remoto: aplicabilidade na avaliação e monitoramento de impactos ambientais causados por acidentes com cargas perigosas. São Paulo, p. 163, set. 2010.

LPICERTIFICACAOLINUX. Monitorando o tráfego de sua rede com iftop. Disponível em: <<http://www.lpicertificacaolinux.com.br/monitorando-o-trafego-de-sua-rede-com-iftop/>>. Acesso em: 05 maio 2016.

LUDOLF. Deividson. Desbravando o comando iftop. Disponível em: <<http://sejalivre.org/desbravando-o-comando-iftop/>>. Acesso em: 05 maio 2016.

MEDIATEK. RT5370. Disponível em: <<http://www.mediatek.com/en/products/connectivity/wifi/broadband/usb/rt5370/>>. Acesso em: 05 maio 2016.

MORIMOTO. Carlos E. Configurando redes wireless no Linux, parte 1. Disponível em: <<http://www.hardware.com.br/tutoriais/rede-wireless-linux/pagina3.html>>. Acesso em: 13 mar. 2015.

MORIMOTO. Carlos E. Redes, Guia Prático 2ª Ed. Brasil: Sul Editores, 2011.

MORIMOTO. Carlos E. Redes, Guia Prático 2ª Ed. Disponível em: <<http://www.hardware.com.br/livros/redes/redes-wireless.html>>. Acesso em: 25 set 2015.

MORIMOTO. Carlos E. Expandindo a rede Wi-Fi com pontos de acesso adicionais. Disponível em: <<http://www.hardware.com.br/tutoriais/expandindo-wifi/>>. Acesso em: 13 out. 2015.

PEDROSA, Leyberson. Drones, Vants ou RPAs? Entenda mais sobre essas aeronaves não tripuladas. Disponível em: <<http://www.ebc.com.br/tecnologia/2015/02/drones-vants-ou-rpas-entenda-mais-sobre-essas-aeronaves-nao-tripuladas#1>>, 2015, Acesso em: 05 de out. 2015.

PEREIRA, Renata I. S.; JUCÁ, Sandro C. S. Sistema Embarcado Linux baseado em Raspberry Pi para gravação online de Microcontroladores PIC. Ceará, 2015.

PPLWARE. Disponível em: <<http://pplware.sapo.pt/informacao/california-ira-proibir-os-drones-voar-sobre-propriedade-privada/>>. Acesso em: 20 ago 2015.

RASPBIAN.ORG. Disponível em: <<http://www.raspbian.org/RaspbianInstaller>>. Acesso em: 19 maio 2016.

RASPBERRYPI.ORG. Faqs. Disponível em: <<https://www.raspberrypi.org>>. Acesso em: 25 set 2015.

RICHARDSON, Matt; SHAWN, Wallace. Primeiros Passos com o Raspberry Pi. Novatec: São Paulo, p. 36, mar. 2013.

ROBOHUB.ORG. Disponível em: <<http://robohub.org/flying-donkey-challenge-announces-33-entrants-for-first-subchallenge/>>. Acesso em: 02 jun 2016.

SILVA, Eduardo Germano; PEREZ, Anderson Luiz Fernandes. Aplicação de hardware de baixo custo na automação residencial. Universidade Federal de Santa Catarina. 2013.

SOURCEFORGE. Disponível em: <<https://sourceforge.net/projects/win32diskimager/>>. Acesso em: 26 maio 2016.

TI-REDES. Disponível em: <<http://ti-redes.webnode.com.br/wireless/wlan-intro/>>. Acesso em: 25 set 2015.

UBUNTUMATE. Disponível em: <<https://ubuntu-mate.org/raspberry-pi/>>. Acesso em: 02 jun 2016.

VIDEOLAN.ORG. Disponível em: <<http://www.videolan.org/vlc/>>. Acesso em: 01 jul 2016.

VIEIRA, Felipe. O que é Streaming. Disponível em: <<http://www.upx.com.br/historia-streaming/>>. Acesso em: 02 out 2015.

WILSON, Tracy. Servidores de streaming. Disponível em: <<http://tecnologia.hsw.uol.com.br/streaming-video-e-audio3.htm>>. Acesso em: 12 dez 2015.