UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

WESLEY SANTOS SILVA

IMPLEMENTAÇÃO E ANÁLISE DE UM BLOQUEADOR DE ANÚNCIOS E SERVIDOR DNS RECURSIVO EM UMA REDE DOMÉSTICA UTILIZANDO RASPBERRY PI

CAMPINA GRANDE - PB

2022

WESLEY SANTOS SILVA

IMPLEMENTAÇÃO E ANÁLISE DE UM BLOQUEADOR DE ANÚNCIOS E SERVIDOR DNS RECURSIVO EM UMA REDE DOMÉSTICA UTILIZANDO RASPBERRY PI

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientadora: Professora Dra. Eliane Araújo.

CAMPINA GRANDE - PB

2022

WESLEY SANTOS SILVA

IMPLEMENTAÇÃO E ANÁLISE DE UM BLOQUEADOR DE ANÚNCIOS E SERVIDOR DNS RECURSIVO EM UMA REDE DOMÉSTICA UTILIZANDO RASPBERRY PI

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr.(a.) Eliane Araújo

Orientador – UASC/CEEI/UFCG

Professora Dr.Wilkerson de Lucena Andrade

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 30 de MARÇO de 2022.

CAMPINA GRANDE - PB

ABSTRACT

In times of digital inclusion and globalization, where access to the Internet grows at an accelerated rate, online ads follow the same trend. Despite being an effective sales method for advertisers, the users of the sites are constantly bombarded with ads, obscuring the actual content that one wishes to to view. Current ad blocker solutions are usually associated with browsers and personal computers. This work presents an implementation and analysis of an ad blocker that covers an entire home network, using a Raspberry Pi. Taking advantage of the potential of the minimal sized computer, in addition to ad blocking, a local recursive DNS server was deployed to store cached domains, reducing the name resolution time for all devices connected to the network.

Implementação e Análise de um Bloqueador de Anúncios e Servidor DNS Recursivo em uma Rede Doméstica Utilizando Raspberry Pi

Wesley Santos Silva
Eliane Araújo
wesley.santos.silva@ccc.ufcg.edu.br
eliane@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

RESUMO

Em tempos de inclusão digital e globalização, onde o acesso à internet cresce de maneira acelerada, os anúncios online seguem a mesma tendência. Apesar de ser um método de venda efetivo para os anunciantes, os usuários dos sites são constantemente bombardeados com anúncios, ofuscando o real conteúdo que deseja-se visualizar. As soluções atuais de bloqueadores de anúncios são usualmente associadas a navegadores e a computadores pessoais. Este trabalho apresenta uma implementação e análise de um bloqueador de anúncios que abrange toda uma rede doméstica, utilizando um Raspberry Pi. Aproveitando do potencial do computador de dimensões reduzidas, além do bloqueio de anúncios, também foi realizada a implementação de um servidor DNS recursivo local, que visa o armazenamento de domínios em *cache*, diminuindo o tempo de resolução de nomes para todos os dispositivos conectados à rede.

PALAVRAS-CHAVE

Advertisements, Raspberry, DNS, Pi-hole, Unbound

1 INTRODUÇÃO

Nos últimos anos, apesar da pandemia do coronavírus ter afetado negativamente diversos aspectos econômicos no âmbito mundial, a indústria da propaganda e anúncios na internet continuou em pleno crescimento[1]. Os anúncios veiculados em meios digitais alcançaram certa maturidade durante a pandemia: apesar de terem crescimento desacelerado no início de 2020, 59% da receita total gasta em anúncios foi direcionada aos meios digitais, totalizando 335 bilhões de dólares[2]. No Brasil, no ano de 2020, 23,7 bilhões de reais foram investidos em publicidade digital, resultando em um aumento de 25% na quantidade pessoas que compraram produtos em meios digitais[3].

Na Figura 1 são apresentadas as redes sociais mais famosas entre os usuários brasileiros no ano de 2022 até o momento[4], com destaque à quantidade de usuários, em milhões. Uma pesquisa de julho de 2020 do site WhistleOut, plataforma que auxilia os consumidores na escolha e eventual compra de planos de celular, realizou uma análise de 8750 *posts* de redes sociais em 175 perfis, tanto pessoais quanto profissionais, e obteve os resultados apresentados na Figura 2, onde é possível observar que 1 em cada 5 publicações das redes Facebook, Instagram e LinkedIn são anúncios.

Uma pesquisa da empresa Deloitte realizada em 2017[6] revelou que 3 em cada 4 usuários de redes sociais acreditam que há anúncios

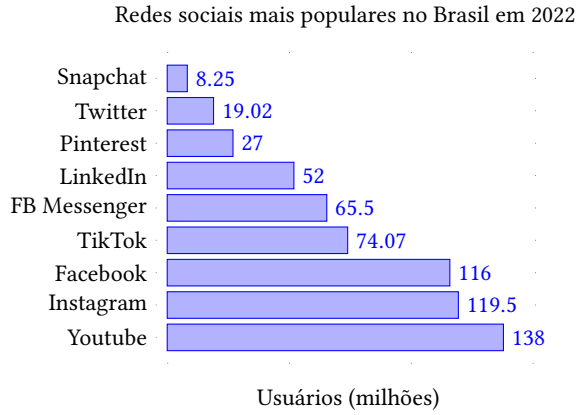

Figura 1: Redes sociais mais populares no Brasil em 2022 e suas quantidades de usuários[4]

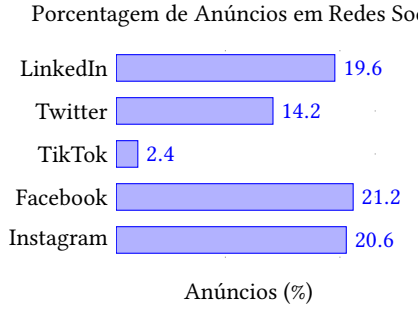

Figura 2: Porcentagem de anúncios em redes sociais em relação ao total de postagens [5]

em demasia nas redes sociais, e 44% dos usuários que participaram da pesquisa afirmaram que os anúncios eram irrelevantes.

Esse tipo de insatisfação leva os usuários a procurarem serviços de bloqueio de anúncios tanto gratuitos quanto pagos. De acordo com o site Blacklinko[8], 42,7% dos usuários de internet entre 16 e 64 anos utilizam ferramentas de bloqueio de anúncios pelo menos uma vez ao mês. Dentre esses usuários, 37% utilizam as ferramentas em um computador, 15% em *smartphones* e 10% em *tablets*.

De acordo com o site Tom's Guide[9], os AdBlockers mais famosos são extensões de navegadores, como o AdBlock Plus (disponível

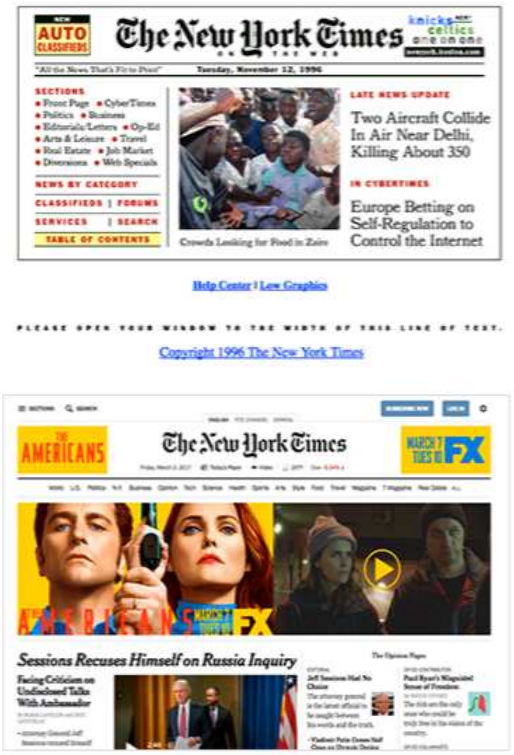

Figura 3: Comparação da primeira edição online do jornal The New York Times, em novembro de 1996 e uma edição do mesmo jornal, de março de 2017. Na segunda imagem, os anúncios ocupam ¼ da página[7]

para o Google Chrome, Firefox, Safari, Opera, Safari e Edge), Adblock (Chrome, Firefox, Safari e Edge) e o Poper Blocker (Chrome).

Deste modo, os bloqueadores de anúncio mais populares estão intimamente ligados a navegadores, e muitas vezes limitados a computadores pessoais, deixando de lado os outros dispositivos em uma rede local.

2 OBJETIVOS

O trabalho aqui apresentado tem como objetivo avaliar a implementação de um bloqueador de anúncios abrangendo **toda** uma rede local doméstica. Isso inclui todos os dispositivos conectados a ela, tanto via ethernet quanto via Wi-Fi, como *smartphones*, computadores pessoais e *streamers* de mídia. Para isso, foi utilizado um Raspberry Pi 3 B+, configurado exclusivamente para essa finalidade.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são apresentados os conceitos necessários para o entendimento do trabalho, assim como as características técnicas dos dispositivos utilizados no projeto.

3.1 Servidores DNS

Hosts (anfitriões) são normalmente identificados por endereços. Estes são perfeitamente adequados para o processamento realizado em roteadores, porém não são de fácil memorização para seres humanos. Por este motivo, um nome único é também tipicamente associado a cada *host* em uma rede.

Nomes de hosts normalmente são diferenciados de endereços de hosts de duas maneiras importantes: primeiro, eles têm comprimentos variáveis e são mnemônicos, portanto, mais fáceis de humanos os identificarem. Segundo, nomes normalmente não contêm informações úteis na localização de um host pela rede. Endereços, por outro lado, têm, algumas vezes, informações de roteamento embarcadas neles[10]. Esses endereços são os **endereços IP**.

Neste trabalho, endereços IP não serão abordados a fundo, porém, é importante que alguns conceitos sejam apresentados. Um endereço de IP consiste em **quatro** bytes, com uma rígida estrutura hierárquica. Um exemplo de endereço de IP é 142.250.218.110, onde cada ponto separa um dos bytes expressos em notação decimal de 0 a 255 (0 a $2^8 - 1$, expoente de acordo com os 8 bits de um byte). Um endereço de IP é hierárquico pois, se o analisarmos da esquerda para a direita, obteremos informações cada vez mais específicas sobre onde o host é localizado na Internet (isto é, em qual rede, na rede das redes). Similarmente, quando analisamos um endereço postal de baixo para cima, obtemos informações cada vez mais específicas sobre o endereço do remetente[11].

Primeiramente, um **espaço de nomes** (*name space*) define um conjunto de possíveis nomes e pode ser não hierárquico (nomes não são divisíveis em componentes) ou hierárquicos. Segundo, o sistema de nomes mantém uma coleção de associações de nomes a valores. O valor pode ser qualquer coisa que quisermos que o sistema de nomes retorne quando um nome é apresentado a ele: em muitos casos, um **endereço** é retornado. Finalmente, um **mecanismo de resolução** é um procedimento que, quando solicitado por um nome, retorna o valor correspondente. Um **servidor de nomes** é uma implementação de um mecanismo de resolução que está disponível em uma rede e pode ser consultado pelo envio de uma mensagem.

Nos primórdios da Internet, quando haviam apenas algumas centenas de hosts, uma autoridade central chamada de *Network Information Center* (NIC) mantinha uma tabela não hierárquica de associações nomes-endereços. Esta tabela era chamada HOSTS.TXT (cópias de alguns desses arquivos primordiais estão disponíveis em [12]). Sempre que um site era criado e desejava-se adicionar o host à internet, o administrador desse site enviava um e-mail ao NIC, informando o novo nome do host e seu respectivo endereço. Essas informações eram inseridas manualmente na tabela e a mesma era enviada internet afora para vários sites de tempos em tempos, e o administrador do sistema de cada site adicionava a tabela em cada host no site. Havia também um livro físico, como uma lista telefônica, publicada periodicamente, que listava todas as máquinas conectadas à internet e todas as pessoas que possuíam acesso à internet e um endereço de e-mail[10].

Com o crescimento da internet, as metodologias supracitadas estavam fadadas ao fracasso. Sendo assim, em meados dos anos de 1980, surge o *Domain Naming System*, DNS. O DNS emprega um espaço hierárquico, e a "tabela" de associações que implementa esse espaço de nomes é particionada em pedaços e distribuída pela

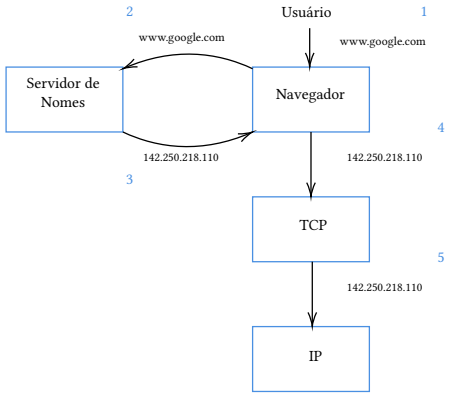

Figura 4: Nomes traduzidos em endereços, onde os números de 1 a 5 indicam a sequência de passos no processo[10] (Adaptado).

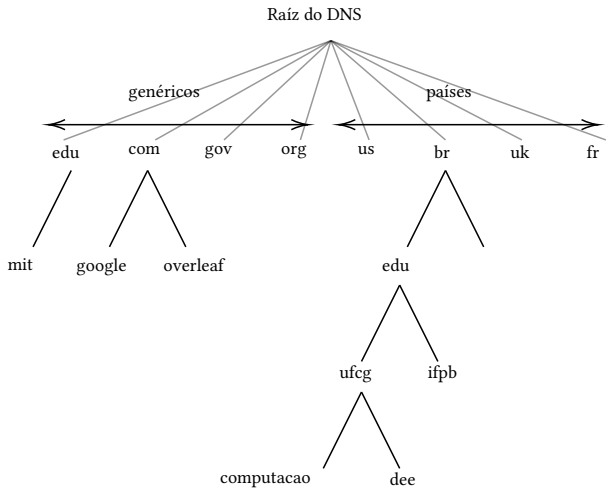

Figura 5: Estrutura hierárquica do DNS (Autoria Própria).

internet. Essas subtabelas são disponibilizadas em servidores de nomes que podem ser solicitados pela rede.

O que acontece na prática é: um usuário apresenta um nome de host a uma aplicação (possivelmente por meio de um endereço de e-mail ou URL), e essa aplicação solicita a tradução através do sistema de nomes, para que seja obtido o endereço de host correspondente, e uma conexão a esse host é aberta, por meio do fornecimento de um protocolo de transporte (como TCP, por exemplo) com o endereço de IP do host. Essa situação é ilustrada de forma extremamente simplificada na Figura 4.

3.1.1 *Hierarquia de Domínios*.

Como citado anteriormente, o DNS implementa um espaço de nomes hierárquico para objetos da internet. Esses nomes são processados da direita para a esquerda, e os pontos são usados como separadores.

A Figura 5 apresenta a estrutura hierárquica de alguns nomes de domínios, incluindo a URL para acessar o site de computação e do departamento de engenharia elétrica da UFCG.

3.1.2 *Iteratividade e Recursividade*.

Há duas maneiras de um servidor DNS obter o endereço de um site através de seu nome: por iteração ou recursão. As duas formas serão apresentadas a seguir.

Recursão e iteração são métodos distintos de resolução de problemas. Na iteração, um conjunto de instruções é repetido até a solução do problema, enquanto na recursão, um programa chama a si mesmo repetidamente até que o problema seja solucionado. A diferença muitas vezes é sutil, mas existente.

Na resolução de nomes, uma pesquisa recursiva indica que um servidor DNS realiza a recursão e continua consultando outros servidores DNS até que obtenha um endereço de IP para retornar ao cliente. Em uma pesquisa iterativa, cada consulta obtém a resposta do servidor DNS diretamente ao cliente, e, caso o endereço de IP desejado não seja retornado, o servidor responde com o endereço de outro servidor DNS, onde a consulta é realizada novamente, até que um servidor responda com o endereço de IP desejado para o domínio fornecido[12]. Um servidor DNS recursivo tem uma vantagem relevante, em termos de tempo de resposta, em relação ao servidor DNS iterativo convencional. Isso ocorre pois, após o primeiro acesso bem sucedido a um domínio, o servidor DNS recursivo salva o endereço em *cache* para, no próximo acesso, não necessitar da consulta a outros servidores para realizar a solução do nome.

3.2 Funcionamento do Pi-hole

O Pi-hole faz uso de diversas ferramentas (FTLDNS, cURL, lighttpd, PHP e AdminLTE Dashboard) para bloquear requisições DNS para domínios conhecidos de publicidade e anúncios, atuando também como um servidor DNS para redes privadas (substituindo qualquer servidor DNS pré-existente), com a capacidade de bloquear anúncios e rastrear domínios para os usuários. O Pi-hole obtém listas de anúncios e domínios de anúncios a partir de uma lista configurável de fontes pré-definidas, e compara as consultas DNS com elas. Se for encontrada uma correspondência dentro de qualquer uma das listas ou dentro de uma lista negra configurada localmente, o Pi-hole bloqueia a resolução do domínio solicitado e retorna ao dispositivo solicitante um endereço falso[13].

3.3 Raspberry Pi 3 B+

Neste trabalho, o computador dimensões reduzidas[14] Raspberry Pi 3 B+ foi utilizado. A versão B+ foi a edição mais recente da terceira geração do Raspberry Pi. O dispositivo tem as características apresentadas na Tabela 1.

3.4 Raspberry Pi OS

O Raspberry Pi OS é uma variante do Debian, originalmente chamado de Raspbian. É administrada pela Raspberry Pi Foundation desde 2013[16]. A versão utilizada no projeto fora a Raspberry Pi OS with desktop, com versão do Kernel 5.10 e versão do Debian 11. O SO foi lançado em 28 de janeiro de 2022. O sistema operacional foi o escolhido para ser instalado no Raspberry Pi.

Processador	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memória	1GB LPDDR2 SDRAM
Conectividade	• LAN sem fio de 2.4GHz e 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE; • Gigabit Ethernet por USB 2.0 (vazão máxima de 300Mbps); • 4 × portas USB 2.0
Entradas e saídas	Conector GPIO de 40 pinos
Áudio e Vídeo	• 1 x HDMI • Conector de display MIPI DSI • Conector de câmera MIPI CSI • Conector de 4 polos de saída estéreo e vídeo composto
Multimídia	Decodificador H.264, MPEG-4 (1080p30); Gráficos OpenGL ES 1.1, 2.0
Suporte a cartão SD	Soquete Micro SD para carregamento do sistema operacional e armazenamento de dados
Fonte de alimentação	5 V/2,5 A
Ambiente	Temperatura de operação: 0 – 50° C
Tempo de vida do produto	O Raspberry Pi 3 Modelo B+ será produzido até pelo menos Janeiro de 2023.

Quadro 1: Características do Rasberry Pi 3 B+[15].

Figura 6: Raspberry Pi 3 Model B+[15].

4 METODOLOGIA

Os dispositivos utilizados no desenvolvimento do projeto foram os seguintes:

- Raspberry Pi 3 B+;
- Roteador Fiberhome AN5506-04-FA;
- Cabo ethernet Cat.6;
- Computador pessoal para análise dos dados.

Para realizar o projeto foi necessário o seguinte procedimento:

(1) Instalar uma distribuição Linux compatível com o Raspberry Pi;
(2) Verificar a documentação e instalar o Pi-hole no Raspberry, software que possibilita o bloqueio de anúncios por meio de DNS;
(3) Estender a funcionalidade do Pi-hole, utilizando-o juntamente com o Unbound, solucionador DNS de código aberto – aprimorando a funcionalidade do Raspberry, tornando-o também um servidor DNS recursivo;
(4) Configurar o roteador doméstico local para utilizar o Raspberry como o único servidor DNS disponível;

Figura 7: *Dashboard* do Pi-hole acessado de um navegador através do endereço 192.168.1.244. Imagem do oitavo dia de utilização do servidor (Autoria Própria).

(5) Realizar testes durante aproximadamente 20 dias, 24h por dia, com os dispositivos conectados sendo utilizados normalmente na rede utilizando o bloqueador de anúncios e servidor DNS;
(6) Analisar os dados e discutir os resultados.

Durante a utilização do servidor DNS implementado, dados superficiais puderam ser visualizados diretamente no *dashboard* do Pi-hole, disponibilizado no endereço definido para o Raspberry durante a configuração. No caso, o endereço de IP escolhido foi o "maior" possível, sendo ele 192.168.1.244. Assim, através de um navegador, o *dashboard* é acessado pelo IP. Após o login como administrador, a tela apresentada na Figura 7 é exibida.

5 IMPLEMENTAÇÃO

Para iniciar o desenvolvimento, o Raspberry Pi OS foi instalado no dispositivo de armazenamento USB utilizado durante todo o processo. O mesmo dispositivo também foi utilizado como armazenamento dos dados do projeto. Os pacotes instalados na sequência são 100% compatíveis com o sistema operacional.

5.1 Acesso remoto ao Raspberry

A fim de acessar o Raspberry Pi para navegar nos arquivos ou instalar pacotes no mesmo sem a necessidade de conectar periféricos, o XRDP - sendo RDP Remote Desktop Protocol (Protocolo de Área de Trabalho Remota), foi instalado, juntamente com o TightVNC Server, sistema de *Virtual Network Computing*,(Computação em Rede Virtual), pacotes necessários tanto para acessar quanto controlar o servidor remotamente. Para isso, os seguintes comandos foram executados no terminal do Raspberry Pi:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install xrdp
sudo apt-get install tightvncserver
```

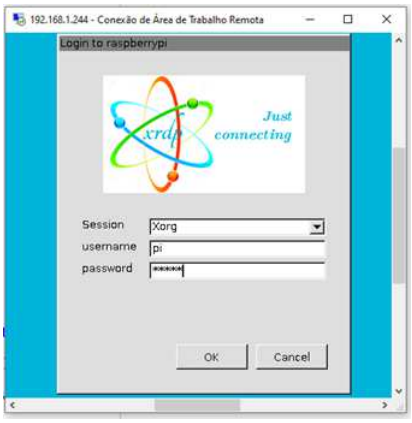

Figura 8: Aplicativo Conexão de Área de Trabalho Remota conectado ao Raspberry Pi OS (Autoria Própria).

Após o processo de instalação dos pacotes, no computador cliente, utilizando o Windows 10, o aplicativo nativo utilizado para acessar o Pi foi o "Conexão de Área de Trabalho Remota", onde o IP do Raspberry Pi, configurado anteriormente como estático e com valor 192.168.1.244 foi acessado. O usuário padrão "pi" deve ser inserido, e a senha é a mesma configurada durante a instalação do Raspberry Pi OS.

Por meio do acesso remoto, foi possível verificar ocasionais falhas causadas por quedas de energia e navegar pelos arquivos do Pi-Hole, contendo todos os dados adquiridos durante a utilização do servidor.

5.2 Pi-hole

Seguindo as instruções encontradas em [17], optou-se por realizar a instalação do Pi-hole no Raspberry através do comando simplificado

```
curl -sSL https://install.pi-hole.net | bash
```

após a atualização dos pacotes do sistema. Uma interface de usuário simplificada é exibida pelo terminal, e algumas opções são fornecidas. Durante a instalação, para este projeto, foram selecionadas as seguintes opções:

- Provedor DNS: Google
- Lista de bloqueio padrão: Steven Black's Unified Hosts List
- Bloqueio de anúncios via IPv4
- Definição de IP estático para o Raspberry como sendo 192.168.1.244
- Instalação da interface web de administrador
- Instalação do servidor web lighttpd
- Salvamento automático de todos os logs.

Após a instalação, acessando o endereço de IP do Raspberry pelo navegador, foi possível acessar a interface web do Pi-hole.

5.2.1 *Complementando o Pi-hole com o Unbound.*

Através da documentação do Pi-hole, acessada em [18], é possível encontrar o procedimento padrão para a instalação do Unbound. Após uma nova atualização dos pacotes do Debian, o comando

```
sudo apt install unbound
```

fora executado. Após a instalação, o arquivo abaixo foi aberto no editor **nano**:

`/etc/unbound/unbound.conf.d/pi-hole.conf.`

O script de configuração do Unbound, fornecido também em [18] foi colado e salvo neste arquivo. Em seguida, foi necessário desativar o DNS do Google, opção selecionada durante a instalação do Pi-hole, para que o serviço do Unbound fosse utilizado. Por padrão o Unbound encontra-se em

`localhost#5335, ou 127.0.0.1#5335,`

portanto, deve-se utilizar o redirecionador de DNS do Pi-hole para que as resoluções de nome passem a ser administradas pelo Unbound:

Figura 9: Seção de configuração de redirecionamento de servidores DNS.

A fim de realizar um teste do Pi-hole e consequentemente do Unbound no dispositivo local, as configurações do dispositivo de rede foram acessadas, e o endereço 192.168.1.244 foi inserido no campo "Servidor DNS preferencial":

Figura 10: Configurações do dispositivo de rede no computador pessoal.

Em seguida, acessando o site cnn.com foi possível observar a redução considerável dos anúncios na página, detalhe que será abordado mais a fundo durante a conclusão. O passo seguinte foi configurar o roteador local para que todos os dispositivos conectados ao mesmo tenham a resolução de nomes administrada pelo

Pi-hole+Unbound, reduzindo os anúncios na rede local como um todo.

5.3 Configuração do roteador Fiberhome

Na tela de configurações de rede do roteador Fiberhome AN5506-04-FA, foi necessário garantir que o campo "WAN IP Mode´´ se encontrava na opção DHCP, além do campo "Primary DNS Server´´ estivesse com o endereço do Raspberry Pi, definido anteriormente como 192.168.1.244, direcionando todos os dispositivos para o servidor local.

6 RESULTADOS

A rede local onde o projeto foi implementado tinha como clientes os seguintes dispositivos:
Clientes dos moradores:

- 3 smartphones;
- 3 computadores pessoais;
- 1 *streamer* de mídia;
- 1 dispositivo *smart home* (Amazon Echo Dot).

Clientes ocasionais:

- 2 smartphones.

Portanto, o Pi-hole detectou, durante o período de testes, 10 clientes. O sistema foi mantido no ar durante 20 dias, do dia 01/03/2022 até o dia 21/03/2022.

Foi realizada então uma análise do pihole-FTL.db, arquivo de banco de dados do Pi-hole contendo 11 tabelas, sendo elas:

- addinfo_by_id
- aliasclient
- client_by_id
- counters
- domain_by_id
- forward_by_id
- ftl
- message
- network
- network_addresses
- query_storage.

Das tabelas, foram obtidos alguns resultados relevantes:

Domínios mais acessados:	
Domínio	Quantidade de Requisições
firetvcaptiveportal.com	10250
wpad.lan	8310
graph.facebook.com	5715
web.diagnostic.networking.aws.dev	5289
www.google.com	5121
optimizationguide-pa.googleapis.com	4065
clients3.google.com	3566
ssl.gstatic.com	2681
api.amazonalexa.com	2524
presence.teams.microsoft.com	2508

Quadro 2: Domínios mais acessados.

Entre os domínios mais acessados estão o *captive portal* do Amazon FireTv, *streamer* de mídia conectado à rede local - possivelmente por ele estar conectado 24h por dia e fazer requisições contínuas. Outros domínios da Amazon, como api.amazonalexa.com também são acessados com mais frequência também pelo dispositivo de *smart home* realizar acessos periódicos ao domínio. Outros domínios acessados com frequência são google.com e serviços do Facebook, também relacionados ao Instagram.

Domínios mais bloqueados:	
Domínio	Quantidade de Requisições
ws.batch.com	24994
data.logentries.com	9765
device-metrics-us.amazon.com	9093
logger.foxitcloud.com	6721
api.amplitude.com	6466
www.googleadservices.com	5386
device-metrics-us-2.amazon.com	5180
s.amazon-adsystem.com	4899
ssl.google-analytics.com	3456
app-measurement.com	3427

Quadro 3: Domínios mais bloqueados.

Sobre os domínios mais bloqueados, é possível destacar o ws.batch.com, o s.amazon-adsystem.com e o www.googleadservices.com, serviços de anúncios da Amazon e da Google, principais empresas que oferecem hospedagem de anúncios.

6.1 Comparação em Sites

Nesta subseção será comparada a performance do bloqueador de anúncios e do servidor DNS implementados no projeto.

6.1.1 ***Bloqueio de Anúncios***.

De acordo com a análise do banco de dados, durante o período de tempo em que o servidor esteve ativo, 107.530 requisições DNS foram bloqueadas, de um total de 323.308, correspondendo a um bloqueio de 33,3% das requisições. Ou seja, no mínimo um terço das requisições correspondiam a anúncios. Mesmo assim, nem todos os anúncios são bloqueados, pois, após o advento dos bloqueadores de anúncios por DNS, diversos sites passaram a hospedar os *ads* em seus próprios domínios, burlando este método de bloqueio.

A seguir, encontram-se casos de sucesso do Pi-hole, assim como casos de fracasso. Os casos de sucesso correspondem a sites que não hospedam os anúncios em seus domínios, permitindo o bloqueio via Pi-hole.

Casos de sucesso:

- OLX

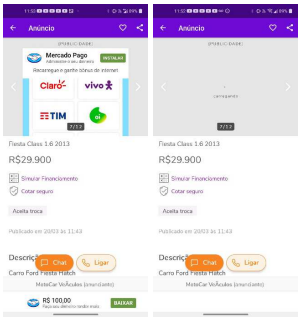

Figura 11: Remoção dos anúncios no OLX.

- CNN

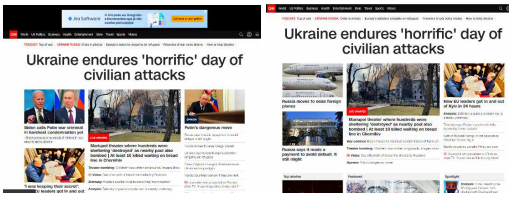

Figura 12: Remoção dos anúncios na CNN.

- MSN

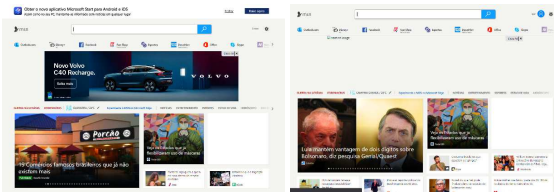

Figura 13: Remoção dos anúncios no MSN.

- Yahoo!

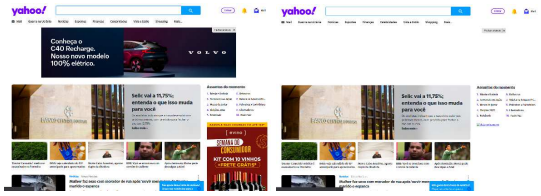

Figura 14: Remoção dos anúncios no Yahoo!

Casos de fracasso:

- Instagram:

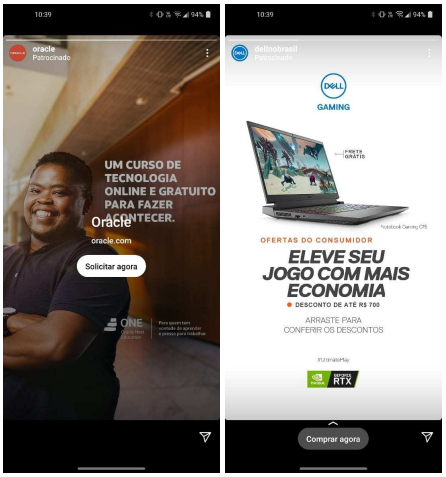

Figura 15: Anúncios no Instagram.

- canyoublockit.com

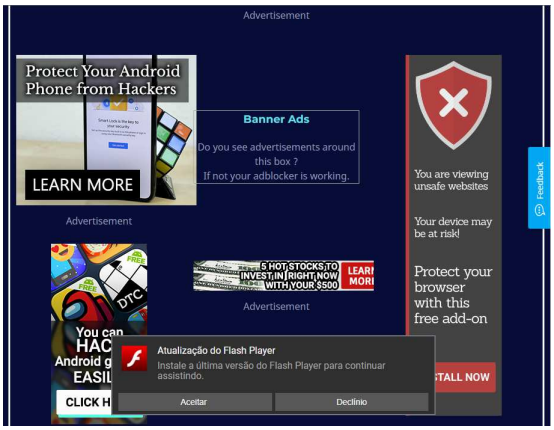

Figura 16: Anúncios no CanYouBlockIt.com.

6.1.2 *Servidor DNS*.

A seguir, serão feitas comparações nos tempos de resolução de nomes com e sem a utilização do servidor local com Pi-hole+Unbound. Para realizar os testes, o seguinte comando foi executado no *PowerShell* do Windows:

```
powershell "Measure-Command {nslookup www.site.com}".
```

Realizando análises em sites com acesso constante na rede local, temos os seguintes resultados, comparando o servidor DNS implementado neste projeto com o servidor DNS do Google (8.8.8.8):

	Tempo de resposta (ms)		
	Sem Pi-hole+ Unbound	Com Pi-hole+Unbound	
Domínio		1º acesso	Em cache
google.com	221,98	171,85	18,81
google.es	337,25	357,46	21,83
cnn.com	233,14	176,00	23,82
msn.com	222,97	299,58	28,73
yahoo.com	226,40	161,88	19,52
amazon.com	221,59	167,97	22,10
mit.edu	284,95	607,47	19,72
computacao.ufcg.edu.br	499,40	545,16	23,06
dee.ufcg.edu.br	516,92	2197,64	22,80
reddit.com	227,33	187,51	18,60
Média	299,19	487,25	21,90

Quadro 4: Comparação entre tempos de resposta do servidor DNS do Google e do servidor DNS local.

É possível verificar uma redução, em média, de mais 90% no tempo de resposta do servidor DNS local, onde os domínios, após o primeiro acesso, são salvos em *cache*, eliminando a necessidade de busca remota pela resolução de nomes. Em alguns casos específicos, como o acesso ao domínio de Computação da UFCG, o servidor DNS Google demora 0,5 s para realizar a resolução do nome, enquanto o servidor local, após o primeiro acesso, leva apenas 0,02306 s.

7 CONSIDERAÇÕES FINAIS

Apesar do bloqueador de anúncios implementado neste projeto não bloquear absolutamente todos os anúncios disponíveis nos sites acessados diariamente, um terço das requisições realizadas pelos dispositivos foram bloqueadas. Isso indica que, mesmo com muitos bloqueios, ainda há muitos outros que passam desapercebidos pelo Pi-hole. De qualquer forma, a navegação com um número consideravelmente reduzido de anúncios é mais tranquila, segura, e resulta em um fluxo menor de dados na rede. Nos computadores pessoais, quando aliados a extensões "*adblock*", os anúncios são nulos ou irrelevantes. A tendência é que, com o passar do tempo, os anúncios estejam cada vez mais preparados para burlar os bloqueadores. Em contrapartida, espera-se que os bloqueadores também sejam aprimorados.

Quanto ao servidor DNS também implementado, o resultado foi satisfatório, reduzindo consideravelmente o tempo de resolução dos sites acessados com mais frequência. Portanto, o objetivo fora alcançado.

Para projetos futuros, o intuito é utilizar ainda mais o potencial do Raspberry Pi de ser um servidor de baixo custo, adicionando armazenamento e tornando-o uma central multimídia. Se aliado a outros dispositivos como um ESP32, por exemplo, o leque de possibilidades é ainda maior. Sendo assim, o projeto aqui apresentado pode ser escalado de maneira indefinida, para diversas finalidades complementares a essa aqui apresentada.

REFERÊNCIAS

[1] Digital Commerce 360. Digital advertising keeps growing during the coronavirus pandemic, October 2020. (Accessed on 03/22/2022).
[2] Vincent Letang and Luke Stillman. *Global Advertising Forecast.* MAGNA, winter update edition, December 2020.
[3] IAB Brasil. *Digital AdSpend.* KANTAR IBOPE Media, 2020/2021 edition, October 2020.
[4] DataReportal Global Digital Insights. Digital 2022: Brazil, February 2022. (Accessed on 03/22/2022).
[5] Angelo Ilumba. New study reveals instagram shows more ads than twitter and tiktok combined, July 2020. (Accessed on 03/22/2022).
[6] The CMO Survey. *Highlights and Insights Report.* Deloitte, August 2017.
[7] Rhana Cassidy. A how to guide to ad blocking, March 2017. (Accessed on 03/22/2022).
[8] Brian Dean. Ad blockers usage and demographic statistics in 2022, March 2021. (Accessed on 03/22/2022).
[9] John Corpuz. The best ad blockers in 2022, March 2022. (Accessed on 03/22/2022).
[10] Larry Peterson. *Computer networks : A systems approach.* Morgan Kaufmann, Amsterdam Boston, 2019.
[11] James Kurose. *Computer networking : a top-down approach.* Pearson Education Limited, Harlow, 2022.
[12] Cloudflare. What is recursive dns?, September 2019. (Accessed on 03/22/2022).
[13] Pi hole Userspace. How does pi-hole work? https://discourse.pi-hole.net/t/how-does-pi-hole-work/3141, 2018. (Accessed on 03/24/2022).
[14] Cloudflare. What is a raspberry pi?, 2022. (Accessed on 03/22/2022).
[15] Raspberry Pi Foundation. *Raspberry Pi 3 Model B+ - Product Brief.* 2022.
[16] Raspberry Pi Foundation. *Raspbian Release Notes.* 2022.
[17] Adam Warner. Github - pi-hole/pi-hole: A black hole for internet advertisements, 2022.
[18] Adam Warner. Unbound - pi-hole documentation, 2022.