

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

BRUNA PUGGINA LIMA

**Análise do crescimento do protocolo IPv6
em dispositivos móveis**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Valter Roesler

Porto Alegre
2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^a. Patricia Helena Lucas Pranke

Pró-Reitora de Ensino (Graduação e Pós-Graduação): Prof^a. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Diretora da Escola de Engenharia: Prof^a. Carla Schwengber Ten Caten

Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Bibliotecária-chefe da Escola de Engenharia: Rosane Beatriz Allegretti Borges

“Ao infinito e além” — BUZZ LIGHTYEAR

AGRADECIMENTOS

Agradeço a minha família pelo apoio para poder concluir o ensino superior, em especial a minha mãe Enilda, meu companheiro Igor e minha irmã Carolina.

A Universidade Federal do Rio Grande do Sul, pela infraestrutura cedida e o ensino de boa qualidade, a todos os meus professores, em especial ao meu coordenador Valter que sem o seu apoio esse trabalho não seria possível.

Por fim, pela oportunidade de ser bolsista do Departamento de Engenharia de Redes do CPD e de participar na prática de alguns projetos na área.

RESUMO

Neste trabalho vou apresentar as camadas de protocolos para o funcionamento da Internet, as principais características do protocolo IPv6, além do esgotamento do IPv4, técnicas de transição e motivações do uso do novo Protocolo Internet. Além de mais endereços disponíveis, ele trouxe outras mudanças em questões segurança, qualidade de serviços e vários outros fatores. Com a utilização do banco de dados de uma grande operadora de telefonia do Brasil, contendo logs de um ano de funcionamento (de março de 2019 a fevereiro de 2020), foi feita a análise de alguns aspectos relevantes à existência de endereços IPv6 em dispositivos móveis. Os dados cedidos pela operadora consistem em 24 arquivos no formato csv, sendo 12 referentes às conexões TCP e 12 referentes às conexões UDP, sendo cada arquivo referente ao log de cada mês do período analisado. Foram desenvolvidos trechos de código em Python, usando a biblioteca Pandas, a fim de ler e analisar os dados cedidos. Este desenvolvimento está descrito no capítulo de metodologia. Das análises feitas foram gerados gráficos e tabelas no capítulo de resultados.

Palavras-chave: Internet. IPv6. Endereçamento IP. Python.

Analysis of the existence of IPv6 addresses on mobile devices

ABSTRACT

In this paper I will present the layers of protocols for the functioning of the Internet, the main characteristics of the IPv6 protocol, in addition to the exhaustion of IPv4, transition techniques and motivations for the use of the new Internet Protocol. In addition to more addresses available, it brought other changes in issues of safety, quality of service and several other factors. Using the database of a large Brazilian telephone operator, containing logs of one year of operation (from March 2019 to February 2020), an analysis was made of some aspects relevant to the existence of IPv6 addresses on mobile devices . The data provided by the operator consist of 24 files in the csv format, 12 of which refer to TCP connections and 12 to UDP connections, with each file referring to the log of each month of the analyzed period. Python code snippets were developed, using the Pandas library, in order to read and analyze the data provided. This development is described in the methodology chapter. Graphs and tables were generated from the analyzes made in the results chapter.

Keywords: Internet, IPv6, IP Addressing, Python.

LISTA DE ABREVIATURAS E SIGLAS

CIDR	Classless Inter-Domain Routing
DC	Dual Carrier
DiffSer	Differentiated Services
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Disk Operating System
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HSPA	High Speed Packet Access
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IntServ	Integrated Services
IoT	Internet of Things
IP	Internet Protocol
IPSec	IP Security Protocol
ISP	Internet Service Provider
LTE	Long Term Evolution
NAT	Network Address Translation
NICK	Núcleo de Informação e Coordenação
NIR	Registros Nacionais de Internet
P2P	Peer-to-Peer
QoS	Quality of Service

RA	Router Advertisement
RIR	Registros Regionais de Internet
RTT	Round Trip Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMST	Universal Mobile Telecommunication System
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network

LISTA DE FIGURAS

Figura 2.1	Funcionamento do TCP/IP	16
Figura 2.2	Registros Regionais de Internet	20
Figura 2.3	Impacto das medidas paliativas	25
Figura 2.4	Número de endereços IPv4 alocados por mês	26
Figura 2.5	Distribuição dos blocos IPv4 alocados correspondente à fase 3	27
Figura 2.6	Previsão inicial da adoção do IPv6	30
Figura 2.7	Previsão atual da adoção do IPv6	31
Figura 3.1	Processo de Metodologia	33
Figura 3.2	Google Colab	34
Figura 3.3	Google Colab Pro	35
Figura 3.4	Restrição do padrão de número de linhas	36
Figura 4.1	Existência de endereços IPv6 nos dados analisados	42
Figura 4.2	Disposição Total de Conexões Analisadas	44
Figura 4.3	Disposição Total de Conexões com Ausência de Endereços IPv6	44

LISTA DE TABELAS

Tabela 4.1	Existência de endereços IPv6 em conexões TCP	43
Tabela 4.2	Existência de endereços IPv6 em conexões UDP	43
Tabela 4.3	Comparação de conexões analisadas nos dois protocolos.....	44
Tabela 4.4	Comparação de tecnologias - TCP	45
Tabela 4.5	Comparação de tecnologias - UDP.....	45
Tabela 4.6	Comparação de tecnologias - OUTROS.....	46
Tabela 4.7	Android versões 100%.....	47
Tabela 4.8	Android versão 10.0	48
Tabela 4.9	Comparação de versões de Android mais usadas - TCP	48
Tabela 4.10	Principais marcas TCP	50
Tabela 4.11	Marcas com 100% de erro.....	51

SUMÁRIO

1 INTRODUÇÃO	13
2 CONCEITOS BÁSICOS	15
2.1 ISP: <i>Internet Service Provider</i>	15
2.2 Arquitetura TCP/IP	15
2.2.1 Camada de transporte.....	16
2.2.1.1 TCP: <i>Transmission Control Protocol</i>	16
2.2.1.2 UPD : <i>User Datagram Protocol</i>	17
2.2.2 Camada de Internet	18
2.2.2.1 IP: <i>Internet Protocol</i>	18
2.2.2.2 Endereçamento do IPv4	18
2.2.2.3 Endereçamento do IPv6	19
2.3 Órgãos Reguladores de endereços IP	19
2.4 Motivações do uso do IPv6	20
2.5 Características do IPv6	20
2.5.1 Segurança na transmissão de dados	21
2.5.2 Qualidade de Serviço (QoS)	22
2.5.2.1 Diffserv	22
2.5.2.2 Intserv	22
2.6 Esgotamento dos endereços IPv4	23
2.6.1 Soluções paliativas	23
2.6.2 O esgotamento dos endereços IPv4 na atualidade	25
2.6.3 Fases do Esgotamento do IPv4 pela LACNIC	25
2.7 Métodos de transição para o IPv6	27
2.7.1 Método de Transição de Pilha Dupla	28
2.7.2 Método de Tunelamento ou Encapsulamento	29
2.7.3 Método de Tradução	29
2.8 Como está hoje a adoção do IPv6	30
2.9 Tecnologias de Conexão com a Internet	31
2.9.1 3G.....	31
2.9.2 4G.....	32
2.10 Android e o DHCPv6	32
3 METODOLOGIA	33
3.1 Google Colab	33
3.1.1 Google Colab Pro.....	34
3.2 Leitura dos dados	35
3.3 Análise dos dados	36
3.3.1 Visualização dos dados	36
3.3.2 Análise dos <i>DataFrames</i>	37
3.3.2.1 Informações dos dados.....	37
3.3.2.2 Análise de colunas em um <i>DataFrame</i>	38
3.3.2.3 Dados faltantes.....	38
3.4 Codificação para a análise dos dados disponibilizados	39
3.4.1 Obtenção dos dados para a análise da existência de endereços IPv6 em dispositivos	39
3.4.2 Obtenção dos dados para a análise de endereços IPv6 não existentes por determinados fatores	40
3.5 Organização dos dados e Geração de gráficos e tabelas	41

4 RESULTADOS	42
4.1 Verificação da existência de Endereços IPv6 nas conexões em <i>smartphones</i>	42
4.2 Verificação da não existência de endereços IPv6 por tecnologias de conexão ...	43
4.3 Verificação da não existência de endereços IPv6 por versão do Android	47
4.4 Verificação da não existência de endereços IPv6 por marcas	49
5 CONCLUSÃO	52
REFERÊNCIAS	54
ANEXO A - CÓDIGO TCP DA SESSÃO 3.2.1	56
ANEXO B - CÓDIGO UDP DA SESSÃO 3.2.1	58
ANEXO C - CÓDIGO TCP DA SESSÃO 3.2.2	60
ANEXO D - CÓDIGO UDP DA SESSÃO 3.2.2	64

1 INTRODUÇÃO

Com o número de usuários da Internet crescendo ao longo dos anos, cada vez mais as redes residenciais tornaram-se comuns, e a população dispõe de mais dispositivos que se conectam na Internet.

O número necessário de endereços IP aumentou de forma significativa, principalmente com a Internet das Coisas (IoT) e com o crescimento do número de dispositivos móveis. Dessa forma, é necessário que as empresas, e principalmente os provedores de Internet, consigam disponibilizar conectividade IPv6 aos seus clientes. “É essencial que todos os provedores de Internet implantem IPv6 em suas redes”, segundo (MOREIRAS; PATARA, 2018, p.10)

A mudança para o novo protocolo da Internet é necessária e está se tornando inevitável. Uma vez que os endereços IPv4 disponíveis estão perto da extinção, a Internet caminha para ser baseada apenas em IPv6.

Atualmente, a adoção do IPv6 é feita “de forma paralela ao uso do IPv4, ou seja, os usuários e dispositivos devem utilizar ambos os endereços” (MOREIRAS; PATARA, 2018, p.11), pois ainda hoje existem serviços e dispositivos na Internet que trabalham somente com IPv4.

Em um primeiro momento, a disponibilização de ambos os protocolos pelos provedores não resolve o problema da falta de endereços IPv4 disponíveis, mas é o primeiro passo para que a internet possa ser um dia baseada em IPv6.

Mesmo com a disponibilização de endereços IPv6 pelos Provedores de Internet aos seus clientes, nem sempre eles estão disponíveis no dispositivo durante as conexões realizadas. Nessa pesquisa, irei especificar estatisticamente através da análise de uma grande quantidade de dados, os fatores que podem influenciar nesse quesito.

Este trabalho tem por objetivo analisar a presença de endereços IPv6 em dispositivos móveis, considerando alguns fatores com base em dados reais do uso de conexões TCP e UDP no período de um ano (março de 2019 a fevereiro de 2020), sendo eles: versão do Android; principais marcas de celulares; e tecnologias de conexão.

Primeiramente foi estabelecida uma base teórica do protocolo, da sua implementação e de como está sendo a transição entre o IPv4 e o IPv6.

Outro objetivo intermediário foi determinar o modo como seria feito o processamento dos dados, a leitura dos arquivos e verificar, dentre tantas informações contidas nos arquivos originais, quais seriam as mais relevantes à análise.

Este capítulo tem como objetivo apresentar o contexto da pesquisa e o que será feito no desenvolvimento do trabalho.

No capítulo 2, chamado de Conceitos básicos, irei introduzir os conceitos relevantes ao Protocolo Internet (IP), à motivação para a implementação do IPv6, os métodos de transição e os conceitos relevantes ao uso do protocolo em dispositivos móveis.

No capítulo 3, chamado Metodologia, será mostrado como foi o processo para o desenvolvimento da análise dos arquivos CSV.

No capítulo 4, chamado Resultados, os gráficos e tabelas obtidos pelos dados gerados no processo de metodologia e a análise dos mesmos.

No capítulo 5 estão as conclusões e sugestões de possíveis seguimentos para trabalhos futuros.

E ao final, as referências bibliográficas e os trechos de códigos desenvolvidos nos anexos.

2 CONCEITOS BÁSICOS

Neste capítulo serão introduzidos os conceitos básicos e os principais protocolos para o funcionamento da internet.

Serão abordadas as características do IPv6, os motivos pelos quais recomenda-se a sua adoção e como está ocorrendo a transição para o novo protocolo Internet.

E, adicionalmente, alguns tópicos relevantes ao uso do IPv6 em *smartphones* .

2.1 ISP: *Internet Service Provider*

Um provedor de serviços de Internet (ISP) é uma empresa que fornece aos clientes acesso à Internet. Os dados podem ser transmitidos usando várias tecnologias, incluindo dial-up, DSL, modem a cabo, sem fio ou interconexões dedicadas de alta velocidade, segundo (TECHOPEDIA, c2021).

Um Provedor de Serviços de Internet é responsável por estabelecer conexões locais, regionais ou mesmo nacionais para facilitar a transmissão de dados on-line entre usuários, outros ISPs e servidores de internet, segundo (SPEEDCHECK, 2021)

Segundo (SPEEDCHECK, 2021) a maioria dos provedores oferecem os mesmos tipos de serviços de Internet aos clientes, incluindo:

- acesso à internet;
- acesso de e-mail;
- registro de nome de domínio;
- serviços de data center.

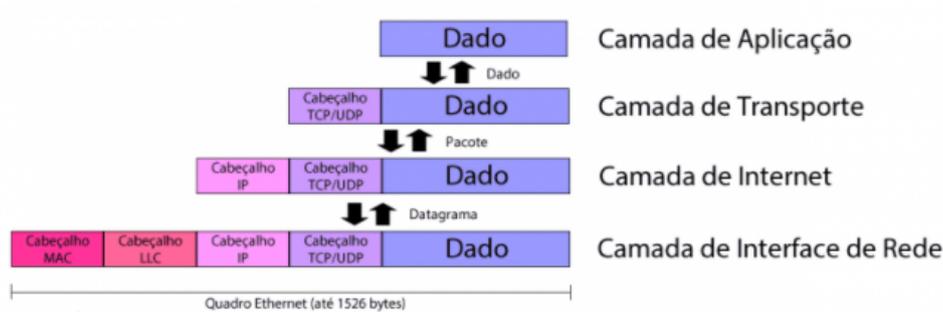
2.2 Arquitetura TCP/IP

A pilha TCP/IP é um conjunto de protocolos de comunicação que permite que diversos equipamentos consigam se comunicar entre si. Ela é estruturada por camadas nas quais cada uma utiliza e presta serviço às camadas adjacentes. Sendo assim, cada camada trata das informações referentes à sua função.

Cada camada recebe os dados da camada superior, encapsulando informações referentes à sua função em forma de cabeçalhos e os envia para a camada inferior, até que

o quadro¹ chegue ao meio físico.

Figura 2.1: Funcionamento do TCP/IP



Fonte: www.citisystems.com.br/protocolo-tcp-ip/

2.2.1 Camada de transporte

“Esta camada reúne os protocolos com funções de estabelecer um transporte de dados fim-a-fim” (LOPES, 2003), ou seja, considera-se apenas a origem e o destino dos dados transmitidos.

Nessa camada existem dois principais protocolos: UDP e TCP.

Os protocolos desta camada podem ou não oferecer confiabilidade, garantia de entrega, controle de fluxo, entre outros. Apesar de existirem vantagens em usar UDP para alguns tipos de fluxo e TCP para outros, várias aplicações usam ambos para funções distintas.

2.2.1.1 TCP: Transmission Control Protocol

O TCP faz um serviço de transferência de dados orientado à conexão, proporcionando entrega na sequência correta e verificação de erros dos pacotes de dados.

Sob o TCP são rodadas a maioria das aplicações da Internet como o SSH, FTP e HTTP.

Algumas das principais características do protocolo segundo (WIKIPEDIA, 2020c) são:

- **Orientado à conexão:** Precisa estabelecer uma conexão para fazer o envio de dados.
- **Ponto a ponto:** Uma conexão é estabelecida ponto a ponto.

¹informação que é traduzida e transmitida entre dois ou mais pontos da rede

- **Full-duplex:** Permite que as duas máquinas envolvidas transmitam e recebam ao mesmo tempo.
- **Confiabilidade:** O TCP proporciona entrega confiável de pacotes de dados, permitindo a recuperação de pacotes perdidos, a eliminação de pacotes duplicados e a recuperação de dados corrompidos. Pode recuperar a conexão antes estabelecida em caso de perda de conectividade.
- **Controle de fluxo:** O TCP usa controle de fluxo baseado na troca de mensagens. O receptor, à medida que recebe os dados, envia uma mensagem ACK (*Acknowledgement*) confirmando a recepção de um segmento.
- **Controle de congestionamento:** Baseado no número de mensagens ACK recebidos pelo remetente por unidade de tempo calculada com os dados do tempo de ida e volta, o RTT (*Round Trip Travel*). O protocolo prediz o quanto a rede está congestionada e diminui sua taxa de transmissão, de modo que o núcleo da rede não fique sobrecarregado.

2.2.1.2 UDP : User Datagram Protocol

O UDP faz um serviço de transferência de dados não orientado à conexão, sem a garantia de que o pacote chegue corretamente. As entregas podem chegar fora de ordem e os datagramas podem ser perdidos. Sendo assim, não há confiabilidade ou garantia de entrega.

Segundo (WIKIPEDIA, 2020b) é adequado para fluxos de dados em tempo real, em especial os que admitem perda de parte do conteúdo como videoconferências ou chamadas por voz (VoIP). Ideal também às aplicações sensíveis a atrasos na rede, mas pouco sensíveis à perdas de pacotes, como jogos de computadores.

Ainda segundo (WIKIPEDIA, 2020b) o UDP suporta *broadcasting* e *multicasting*. Quando esses recursos são necessários ele deve ser utilizado.

Poucas aplicações usam exclusivamente o protocolo UDP. Um serviço de *streaming*, por exemplo, troca informações de controle usando o TCP e libera o envio das informações por UDP.

2.2.2 Camada de Internet

É responsável pelas conexões entre as redes locais, estabelece interconexões e adiciona ao pacote o endereço IP¹ de origem e destino.

2.2.2.1 IP: Internet Protocol

O Protocolo Internet (IP) é um protocolo de comunicação usado para o encaminhamento de dados entre diferentes dispositivos na rede.

O IP possui duas funções básicas:

- **Fragmentação:** Permite o envio de pacotes maiores que o limite de tráfego estabelecido de um enlace, dividindo-os em pacotes menores.
- **Endereçamento:** Permite identificar o destino e a origem dos pacotes a partir do endereço armazenado no seu cabeçalho. Este número, denominado endereço IP, identifica cada dispositivo na Internet de forma única, permitindo que os pacotes de dados sejam enviados corretamente de uma rede para outra, desde o dispositivo de origem até o seu destino final.

Há duas versões de IP utilizadas hoje na Internet: a versão mais antiga utilizada desde 1983 e ainda a mais usada globalmente, o IPv4, e a versão mais atual cuja utilização vem aumentando na Internet, o IPv6. (MOREIRAS; PATARA, 2018)

2.2.2.2 Endereçamento do IPv4

Os endereços IPv4 são números de 32 bits (4 bytes). São representados por uma notação decimal formada por 4 conjuntos de 8 bits separados por ".".

O endereçamento de 32 bits significa que é capaz de endereçar mais de 4 bilhões de dispositivos na rede. Atualmente, quase todos os endereços IPv4 já foram distribuídos para alguma empresa ou instituição. Na prática está praticamente esgotado em nível global. (MOREIRAS; PATARA, 2018)

Há três tipos de endereçamento IPv4:

- **Unicast:** o envio de um pacote é feito de um *host* para outro *host*, usado em comunicação de "um pra um";
- **Broadcast:** o envio de um pacote é feito de um *host* para todos os *hosts* da rede,

¹Endereço lógico de um pacote ou datagrama

usado em comunicação de "um para muitos"

- **Multicast:** o envio de um pacote é feito de um *host* para um grupo de *hosts* selecionados, usado em comunicação de "um para muitos".

2.2.2.3 Endereçamento do IPv6

Os endereços IPv6 são números de 128 bits (16 bytes). São representados em notação hexadecimal formadas por 8 palavras de 16 bits separadas por ":".

No IPv6, com o aumento da quantidade de bytes de endereçamento, os endereços IP deixam de ser um recurso escasso.

Com o uso do IPv6 crescendo na Internet global, ele se torna um dos fatores que permitirá a continuidade do crescimento da Internet, da inclusão digital, da Internet das coisas (IoT), entre outras inovações. (MOREIRAS; PATARA, 2018)

- **Unicast:** funciona da mesma forma que o *unicast* do endereçamento IPv4;
- **Anycast:** identifica um conjunto de interfaces. Um pacote encaminhado a um endereço *Anycast* é entregue à interface pertencente ao conjunto mais próximo da origem. É utilizado em comunicações de "um para um de muitos", segundo (IPV6.BR, 2012b);
- **Multicast:** funciona da mesma forma que o *multicast* do endereçamento IPv4.

No IPv6 não existe endereço *broadcast*. Esta função foi atribuída a tipos específicos de endereços *multicast*, segundo (IPV6.BR, 2012b).

2.3 Órgãos Reguladores de endereços IP

A IANA (*Internet Assigned Numbers Authority*) é responsável pelo controle dos endereços IP (IPv4 e IPv6).

Parte dos endereços IP são delegados pela IANA para cada um dos Registros Regionais de Internet (RIR), segundo (CANNO, 2013). Estes os gerenciam e distribuem às suas respectivas regiões, conforme a imagem abaixo:

Figura 2.2: Registros Regionais de Internet



Fonte: www.iana.org/numbers

Na nossa região o RIR responsável pelo gerenciamento e distribuição de blocos de endereços IPv4 e IPv6 é o LACNIC.

Segundo (MOREIRAS; PATARA, 2018) o LACNIC possui dois Registros Nacionais de Internet (NIR): o NIC.BR (Brasil) e o NIC.MX (México).

2.4 Motivações do uso do IPv6

Segundo (IPV6.BR, 2012a), apesar do IPv4 ter se mostrado robusto, de fácil implantação e interoperabilidade, seu projeto original não previu alguns aspectos como:

- O crescimento das redes e um possível esgotamento dos endereços IPv4.
- O aumento da tabela de roteamento.
- Problemas relacionados à segurança dos dados transmitidos.
- Prioridade na entrega de determinados tipos de pacotes.

(IPV6.BR, 2012a)

Fatores, nos quais, serão vistos com mais detalhes nas sessões 2.5 e 2.6.

2.5 Características do IPv6

O protocolo IPv6 foi criado com o objetivo de substituir o IPv4 de forma gradual devido ao esgotamento dos endereços IPv4.

O IPv6 não é apenas um *upgrade* do IPv4, mas um novo protocolo com endereçamento diferente, com cabeçalhos mais especializados e flexíveis, com mais opções de

controle de fluxo e de segurança e com suporte a novas funcionalidades de roteamento.

As principais mudanças em relação ao IPv4 foram:

- Maior capacidade de endereçamento: no IPv6 o espaço para endereçamento aumentou de 32 bits para 128 bits, permitindo: níveis mais específicos de agregação de endereços; identificar uma quantidade muito maior de dispositivos na rede; e implementar mecanismos de autoconfiguração. A escalabilidade do roteamento *multicast* também foi melhorada através da adição do campo "escopo" no endereço *multicast*. E um novo tipo de endereço, o anycast, foi definido;
- Simplicidade no formato do cabeçalho: alguns campos do cabeçalho IPv4 foram removidos ou tornaram-se opcionais, com o intuito de reduzir o custo do processamento dos pacotes nos roteadores;
- Suporte a cabeçalhos de extensão: as opções não fazem mais parte do cabeçalho base, permitindo um roteamento mais eficaz, limites menos rigorosos em relação ao tamanho e a quantidade de opções, e uma maior flexibilidade para a introdução de novas opções no futuro;
- Capacidade de Identificar novos fluxos de dados: foi adicionado um novo recurso que permite identificar de pacotes que pertençam a determinados tráfegos de fluxos, para os quais podem ser requeridos tratamentos especiais;
- Suporte a autenticação e privacidade: foram especificados cabeçalhos de extensão capazes de fornecer mecanismos de autenticação e garantir a integridade e a confidencialidade dos dados transmitidos.

(SANTOS et al, 2010, p. 17)

Além disso, por não necessitar de NAT, garante a comunicação fim-a-fim da Internet.

2.5.1 Segurança na transmissão de dados

A segurança no IPv6 foi uma preocupação desde o início e várias ferramentas foram implementadas no protocolo. Porém, incidentes de segurança vêm sendo reportados e as vulnerabilidades encontradas devem crescer com o aumento do seu uso.

A maior parte das vulnerabilidades torna o sistema sujeito a ataques DoS e a maioria afeta os equipamentos de rede. (SANTOS et al, 2010)

Várias ferramentas de segurança foram implementadas para o IPv6, mas a principal diferença foi que o suporte ao IPSec, que era opcional pro IPv4, passou a ser obrigatório no IPv6.

O IPSec Implementa criptografia e autenticação de pacotes na camada de rede, fornecendo segurança fim-a-fim, garantindo integridade, confiabilidade e autenticidade dos dados, segundo (SANTOS et al, 2010).

2.5.2 Qualidade de Serviço (QoS)

“Algumas aplicações necessitam que seus pacotes sejam transportados com a garantia de que haja o mínimo de atraso, latência ou perda de pacotes.” (SANTOS et al, 2010, p.118), como o VoIP, videoconferências, jogos on-line, entre outras aplicações.

O QoS (*Quality of Service*) é empregado em protocolos que visam garantir a transmissão de determinados tipos de tráfego. As principais arquiteturas são o DiffServ (*Differentiated Services*), e o IntServ (*Integrated Services*), ambas utilizam políticas de tráfego e podem ser combinadas para LANs (Redes Locais) ou WANs (Redes de longa distância). (SANTOS et al, 2010)

2.5.2.1 Diffserv

Segundo (SANTOS et al, 2010) o DiffServ trabalha com classes, agregando e priorizando pacotes com requisitos de QoS similares. Não exige identificação ou gerência dos fluxos. É muito usado pela facilidade de implementação.

No cabeçalho do pacote IPv4 é representado pelo campo "Tipo de Serviço (ToS)" e no IPv6 pelo campo "Classe de Tráfego".

A prioridade pode ser definida tanto pela origem quanto pelos roteadores, podendo ser redefinida por roteadores intermediários. (IPV6.BR, 2012c)

2.5.2.2 Intserv

Segundo (SANTOS et al, 2010) o IntServ utiliza o conceito de reserva de recursos por fluxo e sua utilização geralmente está associada ao protocolo RSVP.

O RSVP utiliza alguns elementos do protocolo IPv6, como o campo Identificador de Fluxo e o cabeçalho de extensão *Hop-by-Hop*. Pacotes RSVP são enviados com o mesmo valor no campo Identificador de Fluxo, junto com o cabeçalho de Extensão *Hop-By-Hop*, usado para transportar uma mensagem *Router Alert*, indicando para cada roteador no caminho do tráfego QoS, que o pacote IP deverá ser processado. (SANTOS et al, 2010, p. 121)

“Pacotes de um mesmo fluxo devem possuir o mesmo endereço de origem e destino e o mesmo valor no campo Identificador de Fluxo.” (SANTOS et al, 2010, p. 120)

2.6 Esgotamento dos endereços IPv4

Quando o IPv4 foi implementado em 1983 as conexões de Internet eram feitas exclusivamente para computadores.

Com o avanço da tecnologia e o uso de outros dispositivos conectados à Internet houve uma grande demanda de endereços IP.

Os principais fatores agravantes para o esgotamento, segundo (WIKIPEDIA, 2020a), foram:

- **Uso Ineficiente dos endereços:** A divisão por classes na década de 80 fez com que empresas pedissem blocos maiores do que necessitavam.
- **Acessibilidade:** O aumento das redes domésticas tornou a Internet mais acessível à população.
- **Dispositivos Móveis:** Com a evolução da tecnologia os *smartphones* ganharam mais espaço e causaram uma grande demanda por rede.
- **Conexões sempre on-line:** Com a banda larga passaram a existir conexões que não se encerram.

2.6.1 Soluções paliativas

Diante do cenário de escassez do IPv4, foram tomadas algumas medidas para retardar ao máximo o fim dos seus endereços.

Entre elas, as principais medidas paliativas que ainda são utilizadas:

CIDR: Definido em 1993, o CIDR (*Classless Inter-Domain Routing*) aumentou a

eficiência da alocação de endereços para resolver o problema do desperdício do sistema de classes, permitindo a divisão de faixas de endereços contínuas em faixas menores.

Essa divisão padronizada dos blocos de endereços por classes resultava num grande desperdício, uma vez que era designada uma faixa de endereços inteira para o solicitante mesmo que este não fosse utilizá-la toda.

“Se são necessários apenas 1000 endereços, por exemplo, poderia ser usada uma máscara /22 (que permite o uso de 1022 endereços), em vez de uma faixa de classe B inteira, como seria necessário antigamente.” (MANQUEIRO, 2010).

DHCP: Definido em 1997, o DHCP (*Dynamic Host Configuration Protocol*) faz alocação dos endereços IP dinamicamente em uma rede. Funciona da seguinte maneira: Um endereço IP é alocado ao *host* quando este é ligado à rede e ao se desconectar o endereço IP é liberado ficando disponível para outro usuário.

“Através do DHCP um host é capaz de obter um endereço IP automaticamente e adquirir informações adicionais como máscara de sub-rede, endereço do roteador padrão e o endereço do servidor DNS local.” (SANTOS et al, 2010, p. 12)

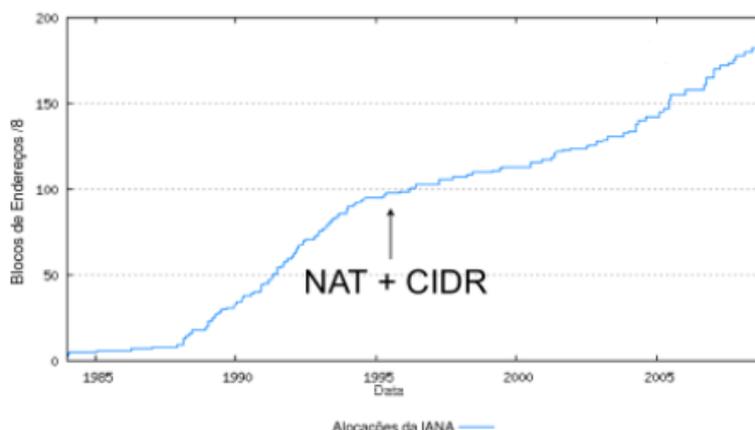
“O DHCP tem sido muito utilizado pelos ISPs por permitir a atribuição de endereços IP temporários a seus clientes conectados” (SANTOS et al, 2010, p. 11)

NAT: Definido em 1999, o NAT (*Network Address Translation*) traduz endereços de IP de uma rede interna usado para o restante da Internet, de forma que todos os *hosts* da rede tenham um IP privado (não exclusivo globalmente) usado para o roteamento na rede interna, porém apenas um IP público (exclusivo globalmente) válido para o roteamento na Internet.

Apesar de ter sido um método muito usado por empresas e ter ajudado a retardar o esgotamento do IPv4, segundo (IPV6.BR, 2012a) ele viola o princípio fim-a-fim da Internet, não permitindo o contato direto entre dois *hosts*, o que dificulta o uso de diversas aplicações como P2P, VoIP e VPNs.

Por alterar alguns valores do cabeçalho dos pacotes, interfere no funcionamento de alguns protocolos de segurança como o IPsec.

Figura 2.3: Impacto das medidas paliativas



Fonte: <http://ipv6.br/post/introducao/>

“A adoção dessas técnicas reduziu em apenas 14% a quantidade de blocos de endereços solicitados à IANA e a curva de crescimento da Internet continuava apresentando um aumento exponencial.” (SANTOS et al, 2010, p. 13)

2.6.2 O esgotamento dos endereços IPv4 na atualidade

Segundo (NICK.BR, 2011) em fevereiro de 2011 a IANA distribuiu os últimos cinco blocos de endereços IPv4 às cinco RIRs (Registros Regionais da Internet).

O RIPE NCC¹ anunciou a locação de seu último /22 em novembro de 2019, segundo (IPV6.br, 2019); o LACNIC² esgotou suas reservas de endereçamento oficialmente em agosto de 2020, segundo (LACNIC, 2020). Os RIRs restantes estão se encaminhando para o esgotamento.

2.6.3 Fases do Esgotamento do IPv4 pela LACNIC

A LACNIC separou o término do IPv4 em sua região (América Latina e algumas ilhas do Caribe) em quatro fases distintas, descritas mais detalhadamente em (LACNIC, 2020), são elas:

Fase 0: Iniciou-se em outubro de 2013 com a alocação de blocos IPv4 até o último /9¹ disponível.

¹RIR responsável pela distribuição de endereços IP na Europa, o Oriente Médio e partes da Ásia Central

²RIR responsável pela distribuição de endereços IP na América Latina e algumas ilhas do Caribe

¹O CIDR usa a notação / (barra) para indicar a quantidade de bits que identifica a rede, segundo

Nesta fase as solicitações para um bloco /15 ou maior eram avaliadas em conjunto pelos NIR e LACNIC.

Fase 1: Iniciou-se em 19 de Maio de 2014 com a alocação de blocos IPv4 até esgotar o /10 reservado para a fase de esgotamento gradativo.

Nessa fase nem o LACNIC nem os NIR podiam garantir que as atribuições fossem agregadas, por exemplo, se a empresa solicitasse um /14 poderia receber mais de um prefixo, cujo total combinado equivale a um /14.

Fase 2: Iniciou-se em 10 de junho de 2014 com a alocação de blocos IPv4 até esgotar o /10 reservado para a fase 2.

Nesta fase apenas atribuições equivalentes a um /24 até um /22 podiam ser feitas. As organizações solicitantes podiam solicitar recursos adicionais a cada 6 meses.

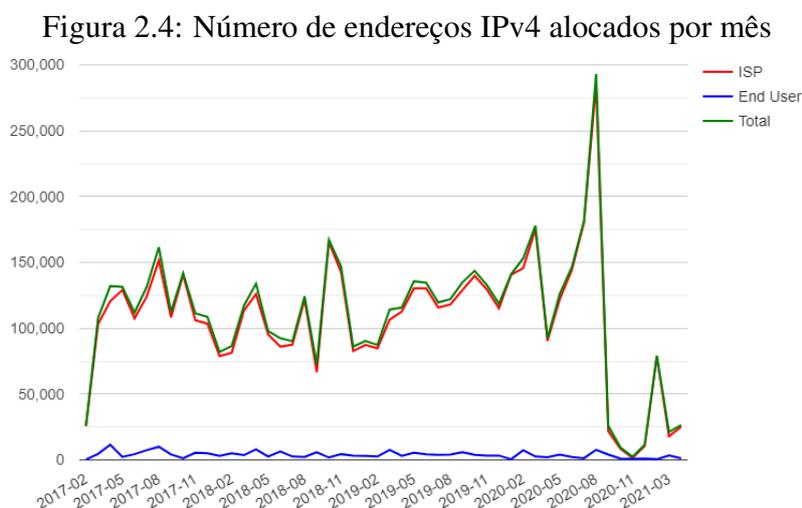
Fase 3: A partir de fevereiro de 2017.

“Esta reserva é o último espaço disponível do LACNIC. É composto por blocos IPv4 pós-esgotamento designados pela IANA, junto com blocos recuperados e devolvidos.” (LACNIC, 2020)

Somente podem ser feitas alocações iniciais limitadas a um /22, não podendo serem solicitados recursos adicionais.

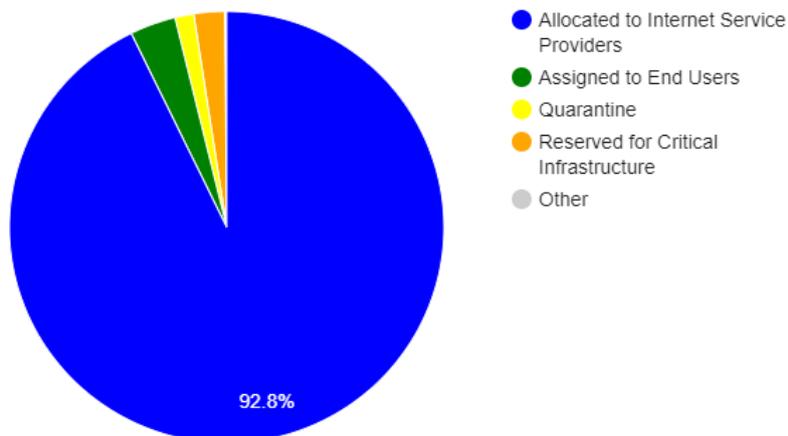
“Em 19 de agosto de 2020, o LACNIC esgotou seu pool de endereços IPv4 contando apenas com recursos recuperados e devolvidos, e uma reserva destinada exclusivamente a infraestrutura crítica.” (LACNIC, 2020)

Os gráficos abaixo mostram detalhes das alocações feitas durante a fase 3.



Fonte: <http://www.lacnic.net/1077/3/lacnic/fases-de-esgotamento-do-ipv4>

Figura 2.5: Distribuição dos blocos IPv4 alocados correspondente à fase 3



Fonte: <http://www.lacnic.net/1077/3/lacnic/fases-de-esgotamento-do-ipv4>

2.7 Métodos de transição para o IPv6

As técnicas de transição podem ser classificadas segundo sua funcionalidade:

- **Pilha dupla:** Consiste na convivência do IPv4 e do IPv6 nos mesmos equipamentos, de forma nativa, simultaneamente. Essa técnica é a técnica padrão escolhida para a transição para IPv6 na Internet e deve ser usada sempre que possível.
- **Túneis:** Permitem que diferentes redes IPv4 comuniquem-se através de uma rede IPv6, ou vice-versa.
- **Tradução:** Permitem que equipamentos usando IPv6 comuniquem-se com outros que usam IPv4, por meio da conversão dos pacotes.

(IPV6.BR, 2012c)

Isso porque “o IPv4 e o IPv6 não são diretamente compatíveis, já que o IPv6 não foi projetado para ser uma extensão ou complemento do IPv4, mas sim, um substituto que resolve o problema do esgotamento de endereços” (IPV6.BR, 2012c). Ainda hoje há necessidade de conectividade IPv4 tendo em vista que alguns serviços e dispositivos na Internet ainda trabalham somente com IPv4.

Segundo (IPV6.BR, 2012c), as técnicas de transição podem ser classificadas em *Stateful* e *Stateless*. As técnicas *Stateful* necessitam manter tabelas de estado com informações de pacotes ou endereços. Nas *Stateless* cada pacote é tratado de forma independente e não é preciso manter as tabelas de estados.

As técnicas de Túneis e Tradução são as que causam maiores impactos na segurança da rede.

2.7.1 Método de Transição de Pilha Dupla

“A utilização deste método permite que dispositivos e roteadores estejam equipados com pilhas para ambos os protocolos, tendo a capacidade de enviar e receber os dois tipos de pacotes, IPv4 e IPv6.” (IPV6.BR, 2012c)

“Cada nó¹ IPv6/IPv4 é configurado com ambos endereços, utilizando mecanismos IPv4 (exemplo: DHCP) para adquirir seu endereço IPv4 e mecanismos IPv6 (exemplo: configuração manual, autoconfiguração stateless e/ou DHCPv6) para adquirir seu endereço IPv6.” (IPV6.BR, 2012c)

Este método facilita o gerenciamento da implantação IPv6 por permitir que seja feito de modo gradual em uma rede. Além de que, caso o IPv4 não seja mais usado no futuro, basta desabilitar a pilha IPv4 de cada nó. (SANTOS et al, 2010)

Segundo (SANTOS et al, 2010) ao implementar a técnica de pilha dupla alguns detalhes devem ser considerados: a necessidade de mudanças na infraestrutura da rede; a estruturação do serviço de DNS, que deve estar habilitado para resolver nomes e endereços de ambos os protocolos (IPv4 e IPv6); a configuração de *firewalls*; protocolos de roteamento interno; e possíveis mudanças no gerenciamento da rede.

A configuração do roteamento IPv4 geralmente é independente da configuração do IPv6. Por exemplo, em uma rede que usa OSPFv2 para o roteamento interno do IPv4 deverá ser configurado o OSPFv3 para o roteamento do IPv6. (IPV6.BR, 2012d)

Pode ser incorporado em redes corporativas que usam o NAT, pois o IPv6 nativo pode ser usado em conjunto com o IPv4 compartilhado, segundo (IPV6.BR, 2012c).

Ao receber endereços IPv4 e IPv6, inicialmente dava-se preferência ao IPv6, e em caso de erro usava-se o IPV4. Essa implementação apresentou um grave problema, pois caso a conexão IPv6 tenha problemas, o usuário com IPv6 sofrerá um grande *delay* para acessar o site, o que não ocorreria usando-se apenas o IPv4.

Para resolver esse problema, foi desenvolvida uma técnica chamada *Happy Eyeballs*². “Seu funcionamento consiste em tentar se conectar às duas conexões simultanea-

¹Dispositivo eletrônico ligado a uma rede capaz de enviar, receber ou transmitir informações através de um canal de comunicação. (LINKFANG, 2020)

²RFC 6555 (Acesso em abril de 2021). <https://tools.ietf.org/html/rfc6555>

mente e utilizar aquela que é estabelecida mais rapidamente, dando uma leve preferência para a conexão IPv6.” (IPV6.BR, 2012d)

2.7.2 Método de Tunelamento ou Encapsulamento

Algumas das principais técnicas de tunelamento são:

- Túneis *6over4*: É estabelecido um túnel manual, chamado de *6over4*, entre dois nós IPv4 para enviar tráfego IPv6. O tráfego IPv6 carregado pelo túnel *6over4* usa o encapsulamento *6in4*, no qual pacotes IPv6 são encapsulados em pacotes IPv4. (IPV6.BR 2012c)

“Esse tipo de túnel pode ser utilizado para contornar um equipamento ou enlace sem suporte ao IPv6 numa rede, ou para criar túneis estáticos entre duas redes IPv6 através do IPv4.” (IPV6.BR, 2012c)

- TEREDO: Segundo (IPV6.BR, 2012c), este método de tunelamento automático permite que nós localizados atrás de NAT obtenham conectividade IPv6 encapsulando pacotes IPv6 em pacotes UDP.

“Esta técnica não é muito eficiente devido ao *overhead* e à complexidade de seu funcionamento, entretanto, quando o *host* está atrás de NAT, ela é uma das únicas opções” (SANTOS et al, 2010).

Segundo (SANTOS et al, 2010), as técnicas de tunelamento são mais suscetíveis a ataques DoS, falsificação de pacotes e de endereços de roteadores. Como forma de proteção, deve-se permitir a entrada de tráfego apenas em túneis autorizados e dar preferência a túneis estáticos ao invés de automáticos.

2.7.3 Método de Tradução

Os métodos de tradução “possibilitam um roteamento transparente na comunicação entre nós IPv6 com nós IPv4 e vice-versa.” (SANTOS et al, 2010, p 203). Podem atuar em camadas e formas distintas. (SANTOS et al, 2010)

Técnicas de tradução podem implicar em problemas relacionados à incompatibilidade com mecanismos de segurança existentes, similar ao que acontece com o NAT no IPv4.

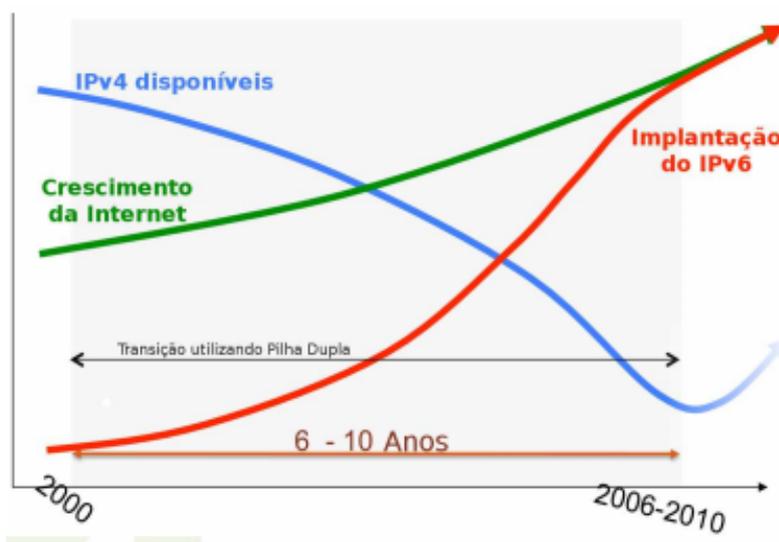
2.8 Como está hoje a adoção do IPv6

A cada dia que passa mais dispositivos se conectam na Internet. No entanto, para que isso seja escalável a infraestrutura da Internet precisa crescer.

O protocolo IPv6 veio para substituir gradativamente o IPv4, porém a adoção do novo protocolo está crescendo de forma mais lenta do que se previa inicialmente.

Em 1993 quando o protocolo surgiu, acreditava-se que a adoção se daria de forma gradual e contínua, e que em dez anos seria uma realidade para a maioria das empresas e usuários finais, acompanhando o ritmo do crescimento dos usuários da Internet, segundo (CANNO, 2013) conforme o gráfico abaixo :

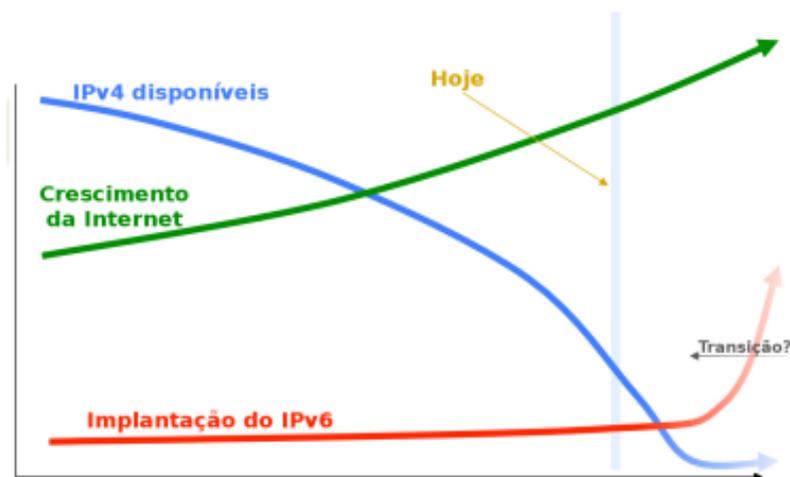
Figura 2.6: Previsão inicial da adoção do IPv6



Fonte: <http://www.ipv6.br/media/arquivo/ipv6/file/48/IPv6-apostila.pdf>

Na atualidade, estamos longe do previsto inicialmente. A maioria das empresas fez a migração pro IPv6 somente quando o IPv4 estava mais próximo do seu inevitável esgotamento, ou fará apenas quando houver necessidade.

Figura 2.7: Previsão atual da adoção do IPv6



Fonte: <http://www.ipv6.br/media/arquivo/ipv6/file/48/IPv6-apostila.pdf>

Em uma medição da APNIC num período de 30 dias (de 11/10/2020 a 09/11/2020), a taxa de compatibilidade com IPv6 do Brasil foi de 34,53% e do mundo de 27,21%, segundo (APNIC, 2020).

As medições feitas pelo Google da adoção do IPv6 entre usuários do Google mostram um crescimento considerável apenas a partir de 2012. (Google, 2020)

2.9 Tecnologias de Conexão com a Internet

Tecnologia de conexão se refere à tecnologia que será usada para que o acesso à Internet seja feito.

Para *smartphones*, além da banda larga acessada pelo wi-fi, existem várias outras formas de acesso fornecidas pelas operadoras.

2.9.1 3G

O 3G (Tecnologia de 3ª Geração) foi criado para levar Internet rápida para celulares.

"O 3G foi (e ainda é) muito importante para democratizar o acesso à Internet no Brasil. Por se tratar de uma rede móvel, o modem 3G foi a salvação para pessoas que moram em locais onde não há sinal de banda larga fixa."(BRAGA, 2018)

As principais tecnologias 3G são:

- HSPA (*High Speed Packet Access*)
- HSPA+ (*Evolved HSPA*)
- HSDPA (*High Speed Downlink Packet Access*)
- HSUPA (*High Speed Uplink Packet Access*)
- UMST (*Universal Mobile Telecommunication System*)

2.9.2 4G

O 4G (Tecnologia de 4ª Geração) também é chamado de LTE (*Long Term Evolution*).

Além de ser mais rápida do que a tecnologia 3G, permite que mais gente se conecte sem perder a qualidade do sinal, segundo (CRUZ, 2018).

“Outra diferença é que ela prioriza o tráfego de dados (áudio, texto, vídeo, foto) na Internet, e não mais o tráfego de voz.” (CRUZ, 2018)

2.10 Android e o DHCPv6

Em 2012 um proprietário de um Samsung Galaxy Nexus com Android 4.0.2 abriu um caso no Rastreador de problemas do Google solicitando suporte para protocolo DHCPv6. Em 2018 essa solicitação foi classificada como "Sem correção (Comportamento Esperado)"¹ pelo engenheiro do Google Lorenzo Colitti, segundo (SANDERS, 2018).

Foram propostas várias maneiras de configuração de redes IPv6, como o RDNSS² (Servidores DNS recursivos), para que seja compatível também ao Sistema Operacional Android, segundo (SANDERS, 2018).

O RFC 7934³, do qual o autor principal é Lorenzo Colitti, foi considerado a melhor prática atual. (SANDERS, 2018)

¹Originalmente: "*Won't Fix (Intended Behavior)*"

²RFC 8106 (Acesso em abril de 2021). <https://tools.ietf.org/html/rfc8106>

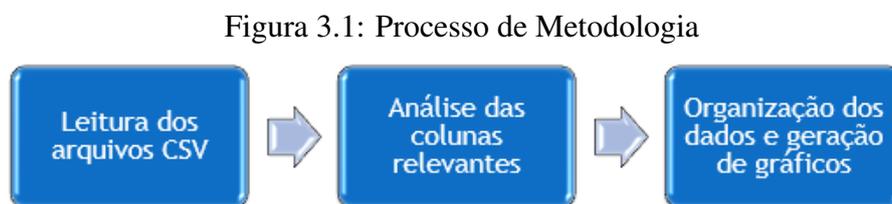
³RFC 7934 (Acesso em abril de 2021). <https://tools.ietf.org/html/rfc7934>

3 METODOLOGIA

Foram utilizados logs de um ano de funcionamento (de março de 2019 a fevereiro de 2020) do banco de dados de uma grande operadora de telefonia do Brasil, com a finalidade de analisar alguns aspectos relevantes à obtenção de endereços IPv6 em dispositivos móveis.

Os dados cedidos pela operadora consistem em 24 arquivos no formato csv, sendo 12 referentes às conexões TCP e 12 referentes às conexões UDP, sendo cada arquivo referente ao log de cada mês do período analisado.

O processo de metodologia, foi dividido como na imagem abaixo:



Fonte: A autora

Primeiramente era necessário fazer a leitura dos arquivos, em seguida a sua análise e por último a quantificação dos dados.

Para a análise e leitura dos arquivos .csv, foram desenvolvido trechos de código na linguagem Python, usando a biblioteca Pandas. Trata-se de uma biblioteca muito utilizada para análise e manipulação de dados, permitindo trabalhar de forma rápida e eficiente com arquivos do tipo csv, excel, txt, entre outros.

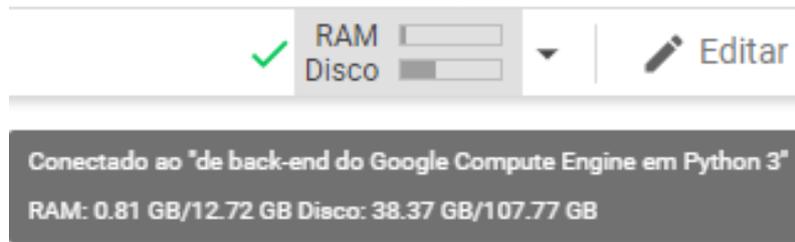
Para executar os códigos, optei por usar o Google Colaboratory, também chamado de Google Colab.

3.1 Google Colab

Similar ao Jupyter Notebook, no Colab temos uma lista de células contendo textos explicativos ou códigos executáveis com suas respectivas saídas.

É um ambiente de Notebooks Jupyter, executado na nuvem, que usa os recursos de processamento e memória RAM de um servidor remoto, conforme a imagem abaixo.

Figura 3.2: Google Colab



Fonte: A autora

As principais vantagens do uso do Colab Notebook são:

- No ambiente online da ferramenta você pode salvar o notebook no Google Drive¹, podendo ser acessado de qualquer computador.
- Por ser executado na nuvem, não precisa ter um computador com grande quantidade de memória RAM e nem uma boa capacidade de processamento.
- Ótima capacidade de processamento: “O Google Colab permite que o programador efetue a aceleração de GPU, que é a responsável pelo processamento das informações adicionadas no notebook.” (ZUCHER, 2020)
- Grande espaço de memória RAM: disponibiliza 13GB de memória RAM;
- Bibliotecas pré-instaladas: O Google Colab possui as principais bibliotecas da linguagem Python instaladas em seu ambiente como Numpy, Pandas e Matplotlib, segundo (ZUCHER, 2020).
- Sem configurações : “O Colaboraty, além de rodar na nuvem e não precisar instalar nada na máquina, ainda é pré configurado. Basta acessar e começar a usar.” (ZUCHER, 2020)
- É gratuito.

3.1.1 Google Colab Pro

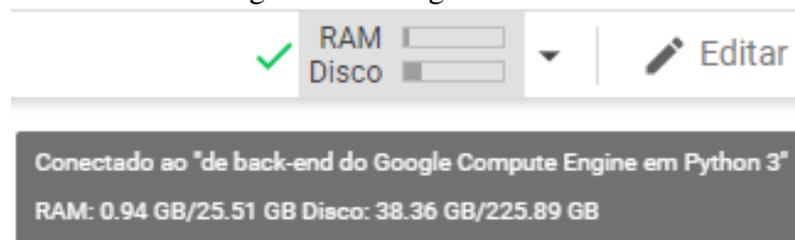
Pode ser necessária uma grande quantidade de memória RAM para carregar grandes quantidades de dados.

Existe também uma versão paga de forma mensal do Google Colab, na qual são disponibilizados mais memória RAM e processamento.

Como é possível observar na imagem abaixo, os recursos disponíveis aumentam significativamente na versão Pro.

¹ espaço de armazenamento em nuvem do Google

Figura 3.3: Google Colab Pro



Fonte: A autora

3.2 Leitura dos dados

Tanto na leitura quanto na análise dos dados será usada a biblioteca Pandas. Ela deve ser declarada para que os comandos de leitura e análise dos dados possam funcionar da seguinte maneira:

```
import pandas as pd
```

“O comando padrão da biblioteca Pandas para ler um arquivo .csv é o `read_csv`, onde o único parâmetro obrigatório é o caminho do arquivo. Existe uma quantidade enorme de parâmetros que podem ser utilizados de acordo com a necessidade.” (NETO, 2019)

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
Netmetric_03_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
osversion', 'brand'])
```

Alguns parâmetros que podem ser necessários para a leitura do arquivos:

- *low_memory = False*: O pandas tenta determinar qual o tipo de configuração deve ser definido analisando os dados de cada parte, exigindo para isso muita memória e demanda. Setar esse parâmetro como *False* diminui a quantidade de memória gasta para a leitura de um arquivo.
- *error_bad_lines = False*: Linhas com muitos campos (por exemplo, uma linha csv com muitas vírgulas), por padrão, causam o surgimento de uma exceção e nenhum *DataFrame* será retornado. Se for *False*, essas "bad_lines" serão eliminadas do *DataFrame* que é retornado. (PANDAS, c2008-2021)
- *engine = 'python'*: *Parser* a ser usado, o mecanismo C é mais rápido, enquanto o mecanismo python é atualmente o mais completo. (PANDAS, c2008-2021)

- *quoting = 3*: O comportamento do campo de controle *quoting* será através das constantes *csv.QUOTE_**. Para isto use um destes: *QUOTE_MINIMAL* (0), *QUOTE_ALL* (1), *QUOTE_NONNUMERIC* (2) ou *QUOTE_NONE* (3). (PANDAS, c2008-2021)
No exemplo foi usado "*QUOTE_NONE* (3)".
- *usecols*: Carrega no *DataFrame* apenas as colunas especificadas pelo comando, ao invés do arquivo inteiro.

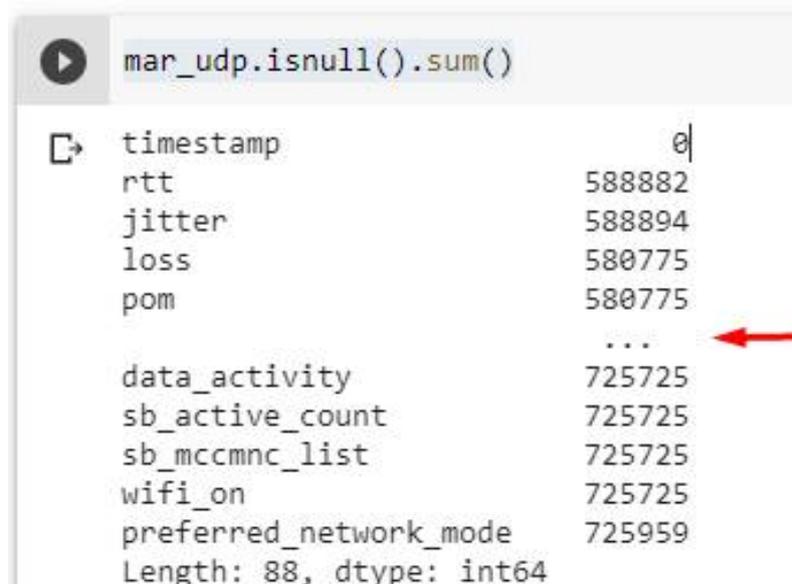
3.3 Análise dos dados

A análise dos dados tem a finalidade de visualizar e analisar as informações dos arquivos .csv lidos pelo comando "read_csv".

3.3.1 Visualização dos dados

Como existe uma restrição do número de linhas que aparecem na visualização quando se utiliza o Pandas no Google Colab, deve-se utilizar outros meios, sendo um deles o comando *set_option*.

Figura 3.4: Restrição do padrão de número de linhas



```

mar_udp.isnull().sum()
timestamp      0
rtt            588882
jitter        588894
loss          580775
pom           580775
...
data_activity  725725
sb_active_count 725725
sb_mccmnc_list 725725
wifi_on       725725
preferred_network_mode 725959
Length: 88, dtype: int64

```

Fonte: A autora

```
pd.set_option('display.max_rows', None)
```

```
pd.set_option('display.max_rows', 30)
4
display(df)
```

Usando o parâmetro "None" carregará mais linhas, porém demorará mais o processamento, não sendo interessante o seu uso para carregar as linhas de grandes arquivos, mas é muito útil para carregar informações de colunas, como na figura 3.4.

Restringindo o número de linhas, como no exemplo acima com 30 linhas, a execução será mais rápida e já teremos uma noção melhor da estrutura da tabela.

O *display* é o comando para visualização dos dados do *DataFrame* declarado na leitura (df).

3.3.2 Análise dos *DataFrames*

Nesta sessão serão tratadas as principais funções para a análise de dados usadas na codificação.

3.3.2.1 Informações dos dados

```
1 df.info()
3 display(df.columns.values)
5 df.isnull().sum()
```

O comando `.info()` mostrará informações do *DataFrame*, como o número de linhas e de colunas. Além de informações sobre cada coluna:

- *Column* : O nome da coluna.
- *Non-Null Count* : A quantidade de linhas não nulas por colunas.
- *Dtype* : O tipo de dados de cada coluna.

O comando `display(df.columns.values)` mostra um *Array* com todas as colunas do *DataFrame*.

O `isnull().sum()` lista a quantidade de linhas nulas por coluna.

3.3.2.2 Análise de colunas em um DataFrame

```

1 df.groupby(by='coluna').size()
3 len(df[df['coluna'] == 'item'])
5 df1 = df.groupby(by=['coluna1'])['coluna2'].apply(tuple)
  display(df1)
7 df.groupby(['coluna']).size().plot(kind="bar")

```

A função `.size()` do `groupby` seleciona uma coluna mostrando seus resultados de acordo com o número de ocorrências. Em alguns casos, pode ser necessário usar o comando *"set.options"*, descrito na sessão 3.3.1.

A função `len`, na linha 3, vai resultar no número de itens (item) contidos na coluna (coluna) do *DataFrame* (`df`).

A função da linha 5 cria um *DataFrame* com a junção de duas colunas na tabela. Pode ser útil para analisar como duas colunas se relacionam entre si.

O *display* mostrará as linhas do novo *DataFrame*.

A função `plot` do `groupby` na linha 8 mostra as informações em forma de gráfico, sendo útil para a visualização dos itens presentes em uma coluna de modo quantitativo.

3.3.2.3 Dados faltantes

Em tabelas com grandes quantidades de dados, muitas informações aparecem como "NaN", ou seja, um dado ausente.

Em algumas análises pode ser útil gerar um *DataFrame* apenas com as linhas da tabela nas quais uma determinada coluna aparece como nula.

Para outras análises, pode ser útil gerar um *DataFrame* apenas com as linhas que não estiverem como nulas.

```

1 dfnotnull = df.query('coluna.notnull()', engine='python')
2 dfisnull = df.query('coluna.isnull()', engine='python')

```

Na linha 1 declaramos um *DataFrame* (`dfnotnull`) como um conjunto das linhas do *DataFrame* original (`df`), onde as informações de uma coluna específica (`coluna`) não aparecem como "NaN", ou seja, tem um resultado definido.

Na linha 2 declaramos um *DataFrame* "dfisnull" como um conjunto de linhas onde as informações de uma coluna aparecem como "NaN".

3.4 Codificação para a análise dos dados disponibilizados

Os trechos de códigos usados para a leitura e a análise dos arquivos estão descritos nos ANEXOS A, B, C e D.

Para a execução dos trechos de códigos foi utilizado o Google Colab Pro.

Foram utilizados os dados referentes a quatro colunas dos arquivos CSV:

- *lastip6*: Endereço IPv6 obtido, de modo que quando essa informação está vazia, o aparelho não possui um endereço IPv6 disponível durante a conexão;
- *conn_tech*: Tecnologia de Conexão utilizada;
- *osversion*: Versão do Android utilizada;
- *brand*: Marca do *smartphone*.

Cada linha do arquivo se refere a uma conexão feita usando o protocolo especificado, UDP ou TCP.

3.4.1 Obtenção dos dados para a análise da existência de endereços IPv6 em dispositivos

Os trechos de código referentes à esta análise estão especificados nos anexos A e B. Sendo o ANEXO A referente aos arquivos TCP, e o ANEXO B referente aos arquivos UDP.

O objetivo deste trecho de código é contabilizar o número de conexões realizadas no mês, e comparar com as conexões feitas onde o *smartphone* tinha um endereço IPv6 disponível durante a conexão.

```
1 import pandas as pd
3 pd.set_option('display.max_rows', 4)
5 df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
   Netmetric_01_2020_UDP.csv', usecols=['lastip6'])
7 dfnv6 = df.query('lastip6.notnull()', engine='python')
```

```

9 display ( df )
11 display ( dfnv6 )

```

No trecho de código acima, foi carregado em um *DataFrame* a coluna "lastip6", e num segundo *DataFrame*, as linhas não nulas do primeiro *DataFrame*.

O comando `set_options` foi utilizado para que o comando `display` mostrasse apenas quatro linhas de cada *DataFrame*, bem como o número de linhas totais dos *DataFrames*.

3.4.2 Obtenção dos dados para a análise de endereços IPv6 não existentes por determinados fatores

Os trechos de código referentes à esta análise estão especificados nos anexos C e D. Sendo o ANEXO C referente aos arquivos TCP, e o ANEXO D referente aos arquivos UDP.

Foram carregadas em um *DataFrame* as colunas "lastip6", "conn_tech", "osversion" e "brand" e num segundo *DataFrame*, as linhas não nulas do primeiro *DataFrame*.

Foi usada a função `groupby.size()` para coletar informações dos fatores analisados ("brand", "osversion" e "con_tech"). O retorno da função consiste nos elementos das colunas com suas respectivas quantidades.

Para aparecerem todos os elementos das colunas, foi utilizado o `set_options` da linha 2.

```

1 import pandas as pd
  pd.set_option('display.max_rows', None)
3 df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
  Netmetric_01_2020_TCP.csv', usecols=['lastip6', 'conn_tech',
  osversion', 'brand'])
dfv6 = dfv6.query('lastip6.isnull()', engine='python')
5
df.groupby(by='brand').size()
7
dfv6.groupby(by='brand').size()
9
df.groupby(by='osversion').size()
11
dfv6.groupby(by='osversion').size()

```

```
13 df.groupby(by='conn_tech').size()  
15 dfv6.groupby(by='conn_tech').size()
```

Devido a existir uma grande quantidade de marcas diferentes (coluna "brand"), fiz a análise de todas somente nos arquivos TCP.

Para os arquivos UDP analisei apenas as que tiveram um percentual de 100% no protocolo TCP.

Para isso, carreguei os arquivos UDP e usei o comando abaixo:

```
len ( nov_udp [ nov_udp [ 'brand' ] == 'Masscom' ] )
```

Sendo no exemplo, "nov_udp" o nome dado ao *DataFrame* e "Masscom" a marca.

3.5 Organização dos dados e Geração de gráficos e tabelas

Para cada fator analisado foi criada uma tabela no Microsoft Excel, com o retorno dos dados gerados pelos trechos de código descritos na sessão acima (3.4).

As tabelas criadas foram usadas para estabelecer os dados e os gráficos da sessão de resultados.

Os gráficos foram gerados usando o recurso de "Análise Rápida".

4 RESULTADOS

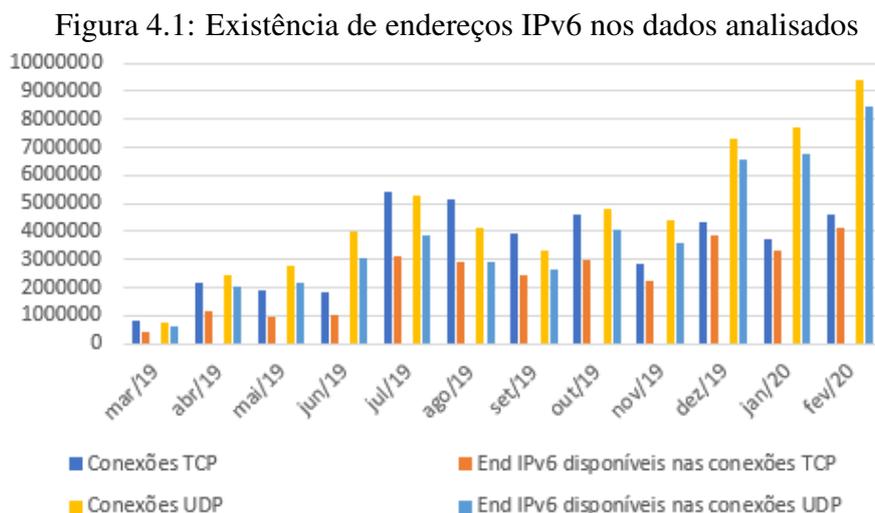
Os resultados obtidos através do processo de metodologia são baseados na presença de endereços IPv6 em dispositivos móveis durante a conexão, o que não significa que o IPv6 foi utilizado na mesma.

A telefônica que me cedeu os arquivos usa a técnica de Pilha Dupla, ou seja, disponibiliza aos seus clientes endereços IPv4 e IPv6.

4.1 Verificação da existência de Endereços IPv6 nas conexões em *smartphones*

Essa análise foi feita somando o número de conexões totais que foram realizadas em dispositivos móveis nos protocolos TCP e UDP, conforme os arquivos disponibilizados, informando quantas tinham um endereço IPv6 disponível durante a conexão.

O gráfico abaixo mostra uma visualização geral da pesquisa:



Fonte: A autora

Abaixo a tabela referente às conexões feitas com o protocolo TCP:

Como pode-se observar, houve uma nítida evolução de endereços IPv6 disponíveis no decorrer dos meses do período analisado.

Abaixo a tabela referente às conexões feitas com o protocolo UDP:

Nas conexões feitas usando o protocolo UDP houve um resultado mais constante. Foram analisadas 97.692.897 conexões entre os protocolos TCP e UDP.

Nas sessões posteriores foram verificadas as conexões onde, por alguns fatores, não haviam endereços IPv6 disponíveis.

Tabela 4.1: Existência de endereços IPv6 em conexões TCP

<i>Mês/Ano</i>	<i>Total de Conexões TCP</i>	<i>Conexões com Endereços IPv6</i>	<i>%</i>
mar/19	799723	437474	54%
abr/19	2160053	1142980	52,91%
mai/19	1891078	972174	51,41%
jun/19	1835307	1015063	55,31%
jul/19	5411113	3097756	57,25%
ago/19	5162735	2942187	56,99%
set/19	3963946	2468657	62,28%
out/19	4610464	2986521	64,78%
nov/19	2820999	2247764	79,68%
dez/19	4362132	3865669	88,62%
jan/20	3725816	3306857	88,76%
fev/20	4586049	4116223	89,76%

Fonte: A Autora

Tabela 4.2: Existência de endereços IPv6 em conexões UDP

<i>Mês/Ano</i>	<i>Total de Conexões UDP</i>	<i>Conexões com Endereços IPv6</i>	<i>%</i>
mar/19	725959	595815	82,07 %
abr/19	2469389	2010161	81,40%
mai/19	2803311	2162425	77,14%
jun/19	4028408	3036478	75,38%
jul/19	5262125	3864325	73,44%
ago/19	4121583	2936552	71,25%
set/19	3322147	2674269	80,50%
out/19	4783763	4067602	85,03%
nov/19	4404300	3623386	82,27%
dez/19	7328910	6539436	89,23%
jan/20	7730226	6801755	87,99%
fev/20	9383361	8439791	89,94%

Fonte: A Autora

4.2 Verificação da não existência de endereços IPv6 por tecnologias de conexão

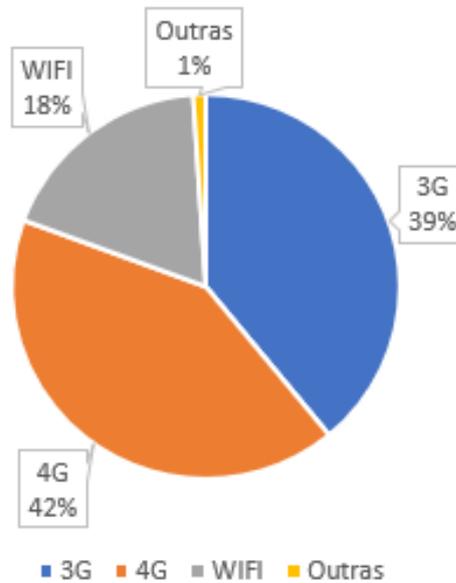
Essa análise inicial foi feita com a soma das conexões TCP e UDP, usando diferentes tecnologias de conexão. Como verificado nos gráficos abaixo, o tipo de conexão mais problemática foi a wi-fi.

Tabela 4.3: Comparação de conexões analisadas nos dois protocolos

<i>Protocolo</i>	<i>Soma de Conexões Analisadas</i>	<i>Soma de Conexões com Endereços IPv6</i>	<i>%</i>
TCP	41329415	28599325	69,20 %
UDP	56363482	46751995	82,95%

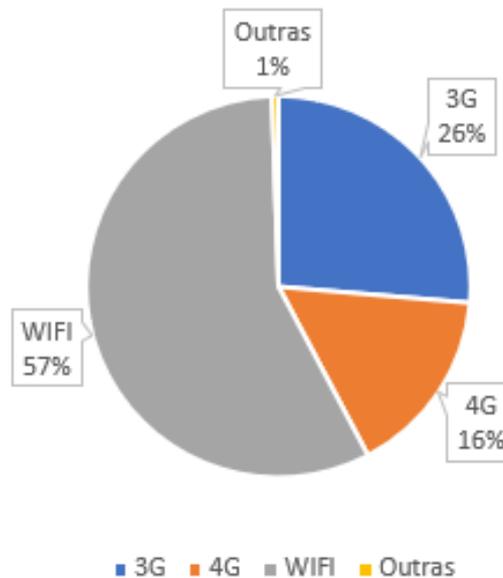
Fonte: A Autora

Figura 4.2: Disposição Total de Conexões Analisadas



Fonte: A autora

Figura 4.3: Disposição Total de Conexões com Ausência de Endereços IPv6



Fonte: A autora

Para os gráficos foram agrupadas algumas tecnologias de conexão, sendo elas:

- **3G:** HSPA, HSPA+, HSUPA, HSDPA e UMTS.
- **4G:** LTE.
- **Wi-fi:** WIFI.
- **Outras:** GPRS (2G), EDGE (2G), GMS(2G), IWLAN, UNKNOWN (não definida nos arquivos), LTE_CA, 4G, TD_SCDMA, DC_HSPAP e DC.

Tanto no protocolo UDP quanto no TCP as tecnologias de conexão mantiveram taxas percentuais parecidas, como pode ser verificado nas tabelas 4.4 e 4.5, que se apresentam por ordem de maior número de conexão por tecnologia:

Tabela 4.4: Comparação de tecnologias - TCP

<i>Tecnologia</i>	<i>Soma Conexões Analisadas</i>	<i>Soma Conexões sem Ends IPv6</i>	<i>%</i>
LTE	15259255	1291776	8,47 %
WIFI	13405227	9564748	71,35%
HSPA+	8720345	1321271	15,15%
HSPA	2647449	417967	15,79%
HSUPA	887919	62964	7,09 %
EDGE	170951	13935	8,15%
UMTS	135027	25986	19,25%
UNKNOWN	33829	6206	18,35%
GPRS	32156	5290	16,45%
HSDPA	24120	9769	40,50%
IWLAN	322	14	4,35%
OUTROS	187	68	36,36%

Fonte: A Autora

Tabela 4.5: Comparação de tecnologias - UDP

<i>Tecnologia</i>	<i>Soma Conexões Analisadas</i>	<i>Soma Conexões sem Ends IPv6</i>	<i>%</i>
LTE	25327077	2263637	8,94 %
HSPA+	17805851	2882765	16,19%
HSPA	5286497	857187	16,21%
WIFI	4519017	3191597	70,63%
HSUPA	2152840	213549	9,92 %
EDGE	488626	43407	8,88%
UMTS	411812	67701	16,44%
UNKNOWN	183527	33676	18,35%
GPRS	108528	18087	16,67%
HSDPA	57046	21960	38,50%
IWLAN	730	60	8,22%
OUTROS	290	90	31,03%

Fonte: A Autora

Nessa análise a maioria das tecnologias se mantiveram com menos de 30% de

endereços IPv6 não existentes, exceto as de "WIFI" com 71,17%, "HSPDA" com 39,09% e "OUTROS" com 33,26%.

O item "OUTROS" consiste em um conjunto de tecnologias que estão especificadas na tabela 4.6.

O termo "UNKNOWN" se refere às conexões nas quais o tipo não foi definido.

No TCP tivemos mais que o dobro de conexões utilizando a "WIFI" do que nas conexões UDP, o que provavelmente tem relação com o percentual menor de endereços IPv6 não existentes analisadas na sessão 4.1.

Para quantificar as tecnologias com poucas conexões, que chamei nas tabelas anteriores de "OUTROS", foram somadas as conexões dos protocolos TCP e UDP, conforme a tabela 4.6.

Tabela 4.6: Comparação de tecnologias - OUTROS

<i>Tecnologia</i>	<i>Soma Conexões Analisadas</i>	<i>Soma Conexões sem Ends IPv6</i>	<i>%</i>
4G	116	98	84,48 %
DC_HSPAP	115	4	3,84%
LTE_CA	113	21	18,58%
TD_SCDMA	73	8	10,96%
DC	43	15	34,88 %
GSM	17	12	70,59%

Fonte: A Autora

O "TD_SCDMA" trata-se de um padrão chinês de 3G e o termo "DC" significa "Dual Carrier".

A tecnologia denominada como "4G" teve uma diferença muito grande de comportamento se comparada com o "LTE", por isso, apesar do nome, manteve como tecnologias diferentes.

Analisando as conexões 2G "GPRS" e "EDGE" nas tabelas 4.4. e 4.5 e a "GMS" na 4.6, foi verificado que esta última foi a única rede 2G que apresentou uma alta incompatibilidade.

A conexão "GMS" também foi a menos utilizada das três, não tendo ocorrências suficientes para estabelecer um padrão de comportamento.

Houveram duas tecnologias nas quais ocorreram menos de 3 conexões, portanto não apareceram oficialmente nas tabelas.

4.3 Verificação da não existência de endereços IPv6 por versão do Android

Na tabela 4.9 foram listados o número de conexões por versão de Android com mais de 300 acessos, somando as conexões de TCP e UDP por ordem de versões.

A maioria das versões tiveram ocorrências durante todo o período analisado, exceto as versões 9.0, que teve ocorrências a partir de julho de 2019, e a 9.1, com ocorrências a partir de agosto de 2019.

Entre as versões mais utilizadas, as que tiveram um alto percentual de endereços IPv6 inexistentes foram as anteriores à versão 4.4, à versão 4.4.3, as versões 6.1, 7.1 e 8.1, e novamente, a partir da versão 9.0.

Através da análise dos dados surgiu uma preocupação com as versões mais novas, visto que a partir da versão 9.0 elas estão com uma taxa percentual muito alta de endereços IPv6 inexistentes.

A tabela 4.7 se refere às versões que tiveram 100% de ausência de endereços IPv6 com as somas de conexões TCP e UDP. Devido às poucas conexões fica difícil saber se tem uma relação direta com a versão do Android.

A tabela 4.8 se refere às conexões com o Android 10.0 que só constaram nos últimos dois meses analisados, ou seja, janeiro e fevereiro de 2020.

Provavelmente exista um grande crescimento do uso da versão 10.0 em períodos posteriores à minha análise, mas como não teve um número de conexões suficiente para identificar incompatibilidade, acredito que seja melhor ser tratado futuramente.

Foram excluídas desta análise as versões nas quais a soma de ocorrências nos protocolos UDP e TCP foram menores que 10, pois não há como tirar conclusões relevantes com uma taxa de ocorrências tão baixa.

Tabela 4.7: Android versões 100%

<i>Versão</i>	<i>Soma Conexões Analisadas</i>	<i>Soma Conexões sem Ends IPv6</i>	<i>%</i>
2.0.6	12	12	100%
5.0.5	14	14	100%
5.5.5	19	19	100%
Kanux 7.1.2	16	16	100%
9.2.0	11	11	100%
10.0	78	78	100%

Fonte: A Autora

Tabela 4.8: Android versão 10.0

<i>Protocolo</i>	<i>jan 2020</i>	<i>fev 2020</i>	<i>Soma das conexões</i>	<i>% das Conexões sem Ends IPv6</i>
UDP	5	55	60	100%
TCP	1	17	18	100%

Fonte: A Autora

Tabela 4.9: Comparação de versões de Android mais usadas - TCP

<i>Versão</i>	<i>Soma TCP</i>	<i>Sem end IPv6</i>	<i>%</i>	<i>Soma UDP</i>	<i>Sem end IPv6</i>	<i>%</i>
4.2.1	179	173	96,65%	335	330	98,51%
4.2.2	92279	61922	67,10%	127220	80955	63,63%
4.3	41409	22686	54,79%	67385	33365	49,51%
4.4	187	101	54,01%	150	50	33,33%
4.4.1	129	30	23,26%	435	20	4,60%
4.4.2	139586	70150	50,26%	204181	71556	35,05%
4.4.3	3109	1870	60,15%	5135	2710	52,78%
4.4.4	233221	99295	42,58%	341046	88824	26,04%
5.0	87796	37119	42,28%	125615	40590	32,31%
5.0.1	86359	28904	33,47%	126575	23995	18,96%
5.0.2	652796	222447	34,08%	1031796	212280	20,57%
5.1	641025	341112	53,21%	866629	405485	46,79%
5.1.1	3305166	932648	28,22%	5766143	727865	12,62%
6.0	3368217	1123641	33,36%	4337472	1036968	23,91%
6.0.1	13071759	3610003	27,62%	16455362	1807954	10,99%
6.1	1236	1125	91,02%	1665	1645	98,80%
7.0	2745500	834752	30,40%	3411175	577643	16,93%
7.1	205	181	88,29%	220	205	93,18%
7.1.1	1886241	638721	33,86%	2909531	630020	21,65%
7.1.2	800226	199373	24,91%	1040594	148096	14,23%
8.0	2302	735	31,93%	2295	145	6,32%
8.0.0	4561017	1305724	28,63%	5942936	850736	14,32%
8.1	546	313	57,33%	725	535	73,79%
8.1.0	7769274	2585617	33,28%	10396858	2157662	20,75%
9	1837233	609269	33,16%	3198659	708858	22,16%
9.0	277	258	93,14%	255	230	90,20%
9.1	1465	1378	94,06%	2215	2075	93,68%

Fonte: A Autora

4.4 Verificação da não existência de endereços IPv6 por marcas

Foram excluídas desta análise as marcas que tiveram menos de 40 conexões com o protocolo TCP, o que resultou em 182 marcas diferentes. Destas, mostrarei as com mais de 1230 registros na tabela 4.10, e as que tiveram um percentual de incompatibilidade de 100% na tabela 4.11.

Algumas marcas apareceram escritas de modo diferente, como "motorola" e "Motorola". Optei por manter as duas formas na tabela como se fossem marcas diferentes para não modificar os dados originais. Outras apareceram como o modelo do celular ao invés da marca, como S10 e J7 Prime da tabela 4.11, também mantive estes como apareceram originalmente.

A maioria das marcas mais populares tiveram um percentual baixo de endereços IPv6 inexistentes.

Muitas das marcas menos conhecidas tiveram uma taxa percentual muito alta, isto ocorre devido à tecnologia de conexão utilizada, hardware do dispositivo, ou versão específica de Android com taxa mais alta de incompatibilidade, como é o caso do S10, da tabela 4.11.

Através da análise dos dados, constatei que as marcas iguais com nomenclatura diferente tiveram resultados conflitantes, como por exemplo "motorola" teve 39,99% enquanto "Motorola" 87,41% . "Positivo" teve 34,44% enquanto "positivo" 78,26%.

A tabela 4.8 se refere às versões que tiveram 100% de ausência de endereços IPv6 tanto no protocolo TCP quanto no UDP, sendo que coloquei somente as que tiveram mais de 40 registros no TCP.

Tabela 4.10: Principais marcas TCP

<i>Marca</i>	<i>Soma Conexões Analisadas</i>	<i>Soma Conexões sem Ends IPv6</i>	<i>%</i>
samsung	24494706	6710924	27,40%
motorola	9024983	3609142	39,99%
LGE	4647737	1244473	26,78%
TCL	664248	168931	25,43%
Xiaomi	515449	199838	38,77%
LENOVO	464888	142098	30,57%
LG Electronics	366660	82713	22,56%
asus	316284	174221	55,08%
Sony	153516	64413	41,96%
Quantum	121292	45704	37,68%
Multilaser	115506	29279	25,35%
Positivo	95300	32817	34,44%
BullittGroupLimited	66616	31132	46,73%
BLU	50062	27142	54,22%
positivo	48360	37846	78,26%
alps	16475	15138	91,88%
HUAWEI	14711	5185	35,25%
ZTE	13675	10763	78,71%
Mirage	9806	3486	35,55%
ZUK	7042	6155	87,40%
TP-LINK	6119	1662	27,16%
DOOGEE	5662	4753	83,95%
OnePlus	5529	4588	82,98%
Lenovo	5334	1840	34,50%
UMIDIGI	4709	4051	86,03%
X23	4161	3739	89,86%
Motorola	4147	3625	87,41%
nubia	3456	2956	85,53%
OUKITEL	3410	2867	84,08%
vernee	2959	2393	80,87%
HTC	2824	2296	81,30%
LEAGOO	2704	2397	88,65%
SKY	2323	2110	90,83%
TCT	2132	1285	60,27%
CUBOT	2126	1787	84,05%
Blackview	2050	1789	87,27%
Android	2022	1850	91,49%
HOMTOM	1952	1693	86,73%
PHONEMAX	1801	1734	96,28%
Ulefone	1765	1535	86,97%
Qtouch	1735	324	18,67%
FLIX	1648	1611	97,75%
Huawei	1603	348	21,71%
ulefone	1490	1327	89,06%
GOME	1391	1229	88,35%
ORRO	1289	1156	89,68%
Meizu	1267	1032	81,45%
lenovo	1241	510	41,10%

Tabela 4.11: Marcas com 100% de erro

<i>Marcas</i>	<i>SomaTCP</i>	<i>Sem End IPv6</i>	<i>%</i>	<i>SomaUDP</i>	<i>Sem End IPv6</i>	<i>%</i>
SKY DEVICES	279	279	100%	230	230	100%
J7 Prime	197	197	100%	160	160	100%
UMI TOUCH	149	149	100%	390	390	100%
S10	126	126	100%	95	95	100%
J8 Pro	121	121	100%	220	220	100%
P30 Pro	98	98	100%	100	100	100%
MALATA	89	89	100%	170	170	100%
HDT-7427G P	87	87	100%	515	515	100%
Reliance	84	84	100%	275	275	100%
World	71	71	100%	45	45	100%
welcome	70	70	100%	35	35	100%
Mobistel	61	61	100%	80	80	100%
YEZZ	57	57	100%	0	0	NaN
Energizer	56	56	100%	5	5	100%
JYT	53	53	100%	25	25	100%
RTS712	48	48	100%	90	90	100%
Masscom	47	47	100%	10	10	100%

Fonte: A Autora

5 CONCLUSÃO

O IPv6 veio pra substituir gradativamente o IPv4, porém esse crescimento não foi gradual conforme esperado, mas foi crescendo de acordo com a necessidade de mais endereçamentos. A adoção do IPv6 é feita de modo paralelo ao IPv4, ou seja, os usuários e dispositivos devem utilizar ambos os endereçamentos.

O objetivo deste trabalho é analisar a existência de endereços IPv6 em dispositivos móveis, considerando alguns fatores com base em dados reais do uso de conexões TCP e UDP ao longo de um ano, sendo eles versão do Android, principais marcas de celulares e tecnologias de conexão.

O principal impedimento da geração de endereços IPv6 nos *smartphones* analisados foi a conexão via wi-fi.

As conexões do tipo wi-fi analisadas não são necessariamente disponibilizadas pela Operadora de Telefonia que me disponibilizou seus dados, mas de qualquer banda larga na qual os celulares analisados se conectaram. Esse alto percentual reflete que ainda estamos longe de ter uma Internet como um todo funcionando apenas em IPv6, e que muitos ISPs não tinham ainda disponibilizado o IPv6 aos seus clientes na época da minha análise.

Evidenciou-se um aumento considerável de endereçamentos IPv6 disponíveis no decorrer dos meses nas conexões em que usou-se o protocolo TCP (protocolo no qual 32,44% das conexões foram através da wi-fi), o que reflete em uma adoção acelerada do IPv6 na medida em que foram-se esgotando os endereços IPv4 disponibilizados pela LACNIC.

Minha pesquisa foi focada na presença de endereços IPv6 em dispositivos móveis, o que não significa que o IPv6 será utilizado na conexão sempre que disponível no *host* final. De modo que, mesmo que o dispositivo suporte a obtenção de endereços e rotas, pode não ter melhor desempenho do que o IPv4 e este ser escolhido pela técnica *Happy Eyeballs*, ou não ser utilizado por algum outro motivo.

Como possível seguimento, eu proponho uma verificação prática do uso do IPv6 em uma rede de pilha dupla, detalhando os fatores relevantes à escolha de qual versão do protocolo IP será usada em cada situação.

Outro possível seguimento seria uma pesquisa de melhores configurações de rede para que um ISP, uma universidade ou uma empresa possam disponibilizar IPv6 também para o sistema operacional Android, tendo em vista a sua incompatibilidade com

o DHCPv6 e levando em conta o peso dos anúncios RA (*Router Advertisement*) para a bateria dos *smartphones*.

REFERÊNCIAS

SANTOS, R. R. et al **Curso IPv6 básico**. São Paulo: Nick.Br, 2010. Disponível em: : <http://www.ipv6.br/media/arquivo/ipv6/file/48/IPv6-apostila.pdf>. Acesso em: 8 out 2020 e 11 nov 2020.

MOREIRAS, A.; PATARA, R. **Fascículos sobre a Infraestrutura da Internet: Endereços IP e ASNs - Alocação para Provedores Internet**. 2018. E-book. Disponível em: <http://www.nic.br/publicacao/fasciculos-sobre-a-infraestrutura-da-internet-endereco-s-ip-e-asns-alocacao-para-provedores-internet/>. Acesso em: 9 out 2020.

IPV6.BR. **Introdução**. 2012a. Disponível em: <http://ipv6.br/post/introducao/>. Acesso em: 6 nov 2020.

IPV6.BR. **Endereçamento**. 2012b. Disponível em: <https://ipv6.br/post/enderecamento/>. Acesso em: 6 nov 2020.

IPV6.BR. **Transição**. 2012c. Disponível em: <http://ipv6.br/post/transicao/>. Acesso em: 7 nov 2020.

IPV6.BR. **Happy Eyeballs**. 2012d. Disponível em: <http://ipv6.br/post/happy-eyeballs/>. Acesso em: 6 abr 2021.

IPV6.BR. **RIPE NCC anuncia fim dos endereços IPv4 na Europa**. 2019. Disponível em: <https://ipv6.br/post/fim-ipv4-ripe/>: :text=Seguindo%20o%20movimento%20mundial%20de,fim%20de%20seu%20estoque%20IPv4. Acesso em 18 nov 2020.

NICK.BR. **IANA distribui últimos endereços IPv4**. 2011. Disponível em: <https://nic.br/noticia/na-midia/iana-distribui-ultimos-enderecos-ipv4/>. Acesso em 18 nov 2020.

APNIC. **IPv6 Capable Rate**. 2020. Disponível em: <https://stats.labs.apnic.net/ipv6>. Acesso em: 10 nov 2020.

BRAGA, L **Entenda a diferença entre 3G, LTE, 4G, 4G+, 4,5G e 5G**. 2018. Disponível em: <https://tecnoblog.net/236506/diferenca-internet-celular-lte-4g-4gmais-5g/> Acesso em: 10 out 2020.

CANNO, R. M. **Redes IP I: IPv4**. 2013. Disponível em: https://www.teleco.com.br/tutoriais/tutorialredeip1/pagina_2.asp. Acesso em: 2 out 2020 e 10 nov 2020.

MANQUEIRO, J. C. B. **Entendendo o CIDR (máscaras de tamanho variável)**. 2010. Disponível em: <https://www.hardware.com.br/dicas/entendendo-cidr.html>. Acesso em: 10 out 2020.

LOPES, N. G. **Introdução à Arquitetura TCP/IP da Internet**. 2013. Disponível em: https://www.gta.ufrj.br/grad/03_1/ip-security/paginas/introducao.html. Acesso em:

10 dez 2019.

WIKIPEDIA. **Exaustão do IPv4**. 2020a. Disponível em: https://pt.wikipedia.org/wiki/Exaust%C3%A3o_do_IPv4. Acesso em: 11 out 2020.

WIKIPEDIA. **User Datagram Protocol**. 2020b. Disponível em: https://pt.wikipedia.org/wiki/User_Datagram_Protocol. Acesso em: 16 out 2020.

WIKIPEDIA. **Transmission Control Protocol**. 2020c. Disponível em: https://pt.wikipedia.org/wiki/Transmission_Control_Protocol. Acesso em: 16 out 2020.

Google. **A adoção do IPv6**. 2020. Disponível em: www.google.com/intl/pt-BR/ipv6/statistics.html . Acesso em: 18 nov 2020.

LACNIC. **Fases de Esgotamento do IPv4**. 2020. Disponível em: <http://www.lacnic.net/1077/3/lacnic/fases-de-esgotamento-do-ipv4>. Acesso em: 2 nov 2020.

SANDERS, J. **Android's lack of DHCPv6 support frustrates enterprise network admins**. 2018. Disponível em: <https://www.techrepublic.com/article/androids-lack-of-dhcpv6-support-frustrates-enterprise-network-admins/> . Acesso em: 28 de abr 2021.

NETO, M. G. **Otimizando o armazenamento da memória de um Dataframe Pandas**. 2019. Disponível em: <https://miltongamaneto.medium.com/otimizando-o-armazenamento-da-mem%C3%B3ria-de-um-dataframe-pandas-ca8c4eb45699> . Acesso em: 20 de abr 2021.

ZUCHER, V. **Google Colab: O que é e o que tem a ver com Ciência de Dados?**. 2020. Disponível em: <https://www.lewagon.com/pt-BR/blog/google-collab-o-que-e-ciencia-de-dados/> . Acesso Em: 2 de abr 2021.

CRUZ, B. S. **3G, 4G e 5G: entenda a tecnologia por trás da conexão do seu celular**. 2018. Disponível em: <https://www.uol.com.br/tilt/noticias/redacao/2018/10/18/entenda-a-tecnologia-por-tras-do-3g-4g-e-5g.html> . Acesso Em: 2 de mai 2021.

PANDAS. **pandas.read_csv**. c2008-2021, Disponível em: https://pandas.pydata.org/pandas-docs/dev/reference/api/pandas.read_csv.html . Acesso em: 25 de Fev 2021.

CANTU, E. **CIDR**. 2014. Disponível em: <http://wiki.foz.ifpr.edu.br/wiki/index.php/CIDR>. Acesso em: 05 de Abr 2021.

TECHOPEDIA **Internet Service Provider (ISP)**. c2021. Disponível em: <https://www.techopedia.com/definition/2510/internet-service-provider-isp>. Acesso em 09 Mai 2021.

LINKFANG **Nó (redes de comunicação)**. 2020. Disponível em: <https://pt.linkfang.org/wiki/Nodo> . Acesso em 09 Mai 2021.

ANEXO A - CÓDIGO TCP DA SESSÃO 3.2.1

```
1 import pandas as pd
pd.set_option('display.max_rows', 4)
3
#leitura dos dados
5 mar_tcp_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_03_2019_TCP.csv', usecols=['lastip6'])
abr_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_04_2019_TCP.csv', usecols=['lastip6'])
7 mai_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_05_2019_TCP.csv', usecols=['lastip6'])
jun_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_06_2019_TCP.csv', usecols=['lastip6'])
9 jul_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_07_2019_TCP.csv', usecols=['lastip6'])
ago_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_08_2019_TCP.csv', usecols=['lastip6'])
11 set_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_09_2019_TCP.csv', usecols=['lastip6'])
out_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_10_2019_TCP.csv', usecols=['lastip6'])
13 nov_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_11_2019_TCP.csv', usecols=['lastip6'])
dez_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_12_2019_TCP.csv', quoting = 3, usecols=['lastip6'])
15 jan_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_01_2020_TCP.csv', usecols=['lastip6'])
fev_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_02_2020_TCP.csv', usecols=['lastip6'])
17
martcpv6 = mar_tcp.query('lastip6.notnull()', engine='python')
19 abrtcpv6 = abr_tcp.query('lastip6.notnull()', engine='python')
maitcpv6 = tcp_tcp.query('lastip6.notnull()', engine='python')
21 juntcpv6 = jun_tcp.query('lastip6.notnull()', engine='python')
jultcpv6 = jul_tcp.query('lastip6.notnull()', engine='python')
23 agotcpv6 = ago_tcp.query('lastip6.notnull()', engine='python')
settcpv6 = set_tcp.query('lastip6.notnull()', engine='python')
25 outtcpv6 = out_tcp.query('lastip6.notnull()', engine='python')
novtcpv6 = nov_tcp.query('lastip6.notnull()', engine='python')
27 deztcpv6 = dez_tcp.query('lastip6.notnull()', engine='python')
```

```
jantcpv6 = jan_tcp.query('lastip6.notnull()', engine='python')
29 fevtcpv6 = fev_tcp.query('lastip6.notnull()', engine='python')

31 #analise dos dados
   display(mar_tcp)
33 display(martcpv6)
   display(abr_tcp)
35 display(abrtcpv6)
   display(mai_tcp)
37 display(maitcpv6)
   display(jun_tcp)
39 display(juntcpv6)
   display(jul_tcp)
41 display(jultcpv6)
   display(ago_tcp)
43 display(agotcpv6)
   display(set_tcp)
45 display(settcpv6)
   display(out_tcp)
47 display(outtcpv6)
   display(nov_tcp)
49 display(novtcpv6)
   display(dez_tcp)
51 display(deztcpv6)
   display(jan_tcp)
53 display(jantcpv6)
   display(fev_tcp)
55 display(fevtcpv6)
```

ANEXO B - CÓDIGO UDP DA SESSÃO 3.2.1

```
1 import pandas as pd
pd.set_option('display.max_rows', 4)
3
#leitura dos dados
5 mar_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_03_2019_UDP.csv', usecols=['lastip6'])
abr_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_04_2019_UDP.csv', usecols=['lastip6'])
7 mai_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_05_2019_UDP.csv', usecols=['lastip6'])
jun_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_06_2019_UDP.csv', usecols=['lastip6'])
9 jul_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_07_2019_UDP.csv', usecols=['lastip6'])
ago_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_08_2019_UDP.csv', usecols=['lastip6'])
11 set_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_09_2019_UDP.csv', usecols=['lastip6'])
out_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_10_2019_UDP.csv', usecols=['lastip6'])
13 nov_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_11_2019_UDP.csv', usecols=['lastip6'])
dez_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_12_2019_UDP.csv', usecols=['lastip6'])
15 jan_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_01_2020_UDP.csv', usecols=['lastip6'])
fev_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_02_2020_UDP.csv', usecols=['lastip6'])
17
marudpv6 = mar_udp.query('lastip6.notnull()', engine='python')
19 abrudpv6 = abr_udp.query('lastip6.notnull()', engine='python')
maiidpv6 = udp_udp.query('lastip6.notnull()', engine='python')
21 junudpv6 = jun_udp.query('lastip6.notnull()', engine='python')
juludpv6 = jul_udp.query('lastip6.notnull()', engine='python')
23 agoudpv6 = ago_udp.query('lastip6.notnull()', engine='python')
setudpv6 = set_udp.query('lastip6.notnull()', engine='python')
25 outudpv6 = out_udp.query('lastip6.notnull()', engine='python')
novudpv6 = nov_udp.query('lastip6.notnull()', engine='python')
27 dezudpv6 = dez_udp.query('lastip6.notnull()', engine='python')
```

```
janudpv6 = jan_udp.query('lastip6.notnull()', engine='python')
29 fevudpv6 = fev_udp.query('lastip6.notnull()', engine='python')

31 #analise dos dados
   display(mar_udp)
33 display(marudpv6)
   display(abr_udp)
35 display(abrudpv6)
   display(mai_udp)
37 display(maiudpv6)
   display(jun_udp)
39 display(junudpv6)
   display(jul_udp)
41 display(juludpv6)
   display(ago_udp)
43 display(agoudpv6)
   display(set_udp)
45 display(setudpv6)
   display(out_udp)
47 display(outudpv6)
   display(nov_udp)
49 display(novudpv6)
   display(dez_udp)
51 display(dezudpv6)
   display(jan_udp)
53 display(janudpv6)
   display(fev_udp)
55 display(fevudpv6)
```

ANEXO C - CÓDIGO TCP DA SESSÃO 3.2.2

```
1 import pandas as pd
pd.set_option('display.max_rows', None)
3
#leitura dos dados
5 mar_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_03_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand', 'end_tech'])
abr_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_04_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
7 mai_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_05_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
jun_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_06_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
9 jul_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_07_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
ago_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_08_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
11 set_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_09_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
out_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_10_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
13 nov_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_11_2019_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
dez_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_12_2019_TCP.csv', quoting = 3, usecols=['lastip6', '
    conn_tech', 'osversion', 'brand'])
15 jan_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_01_2020_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
fev_tcp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_02_2020_TCP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
```

```
    'osversion', 'brand' ])
```

17

```
nmarutcpv6 = mar_tcp.query('lastip6.isnull()', engine='python')
```

19

```
nabrtcpv6 = abr_tcp.query('lastip6.isnull()', engine='python')
```

```
nmaitecpv6 = tcp_tcp.query('lastip6.isnull()', engine='python')
```

21

```
njuntcpv6 = jun_tcp.query('lastip6.isnull()', engine='python')
```

```
njultcpv6 = jul_tcp.query('lastip6.isnull()', engine='python')
```

23

```
naotcpv6 = ago_tcp.query('lastip6.isnull()', engine='python')
```

```
nsettcpv6 = set_tcp.query('lastip6.isnull()', engine='python')
```

25

```
nouttcpv6 = out_tcp.query('lastip6.isnull()', engine='python')
```

```
nnovtcpv6 = nov_tcp.query('lastip6.isnull()', engine='python')
```

27

```
ndeztcpv6 = dez_tcp.query('lastip6.isnull()', engine='python')
```

```
njantcpv6 = jan_tcp.query('lastip6.isnull()', engine='python')
```

29

```
nfevtcpv6 = fev_tcp.query('lastip6.isnull()', engine='python')
```

31

```
#analise dos dados
```

33

```
#OS VERSION
```

```
mar_tcp.groupby(by='osversion').size()
```

35

```
nmartcpv6.groupby(by='osversion').size()
```

```
abr_tcp.groupby(by='osversion').size()
```

37

```
nabrtcpv6.groupby(by='osversion').size()
```

```
mai_tcp.groupby(by='osversion').size()
```

39

```
nmaitecpv6.groupby(by='osversion').size()
```

```
jun_tcp.groupby(by='osversion').size()
```

41

```
njuntcpv6.groupby(by='osversion').size()
```

```
njultcpv6.groupby(by='osversion').size()
```

43

```
naotcpv6.groupby(by='osversion').size()
```

```
ago_tcp.groupby(by='osversion').size()
```

45

```
naotcpv6.groupby(by='osversion').size()
```

```
set_tcp.groupby(by='osversion').size()
```

47

```
nsettcpv6.groupby(by='osversion').size()
```

```
out_tcp.groupby(by='osversion').size()
```

49

```
nouttcpv6.groupby(by='osversion').size()
```

```
nov_tcp.groupby(by='osversion').size()
```

51

```
nnovtcpv6.groupby(by='osversion').size()
```

```
dez_tcp.groupby(by='osversion').size()
```

53

```
ndeztcpv6.groupby(by='osversion').size()
```

```
jan_tcp.groupby(by='osversion').size()
```

55

```
njantcpv6.groupby(by='osversion').size()
```

```
nfevtcpv6.groupby(by='osversion').size()
```

```
57 nfevtcpv6.groupby(by='osversion').size()
#CONN_TECH
59 mar_tcp.groupby(by='conn_tech').size()
nmartcpv6.groupby(by='conn_tech').size()
61 abr_tcp.groupby(by='conn_tech').size()
nabrtcpv6.groupby(by='conn_tech').size()
63 mai_tcp.groupby(by='conn_tech').size()
nmaitcpv6.groupby(by='conn_tech').size()
65 jun_tcp.groupby(by='conn_tech').size()
njuntcpv6.groupby(by='conn_tech').size()
67 jul_tcp.groupby(by='conn_tech').size()
njultcpv6.groupby(by='conn_tech').size()
69 ago_tcp.groupby(by='conn_tech').size()
nagotcpv6.groupby(by='conn_tech').size()
71 set_tcp.groupby(by='conn_tech').size()
nsettcpv6.groupby(by='conn_tech').size()
73 out_tcp.groupby(by='conn_tech').size()
nouttcpv6.groupby(by='conn_tech').size()
75 nov_tcp.groupby(by='conn_tech').size()
nnovtcpv6.groupby(by='conn_tech').size()
77 dez_tcp.groupby(by='conn_tech').size()
ndeztcpv6.groupby(by='conn_tech').size()
79 jan_tcp.groupby(by='conn_tech').size()
njantcpv6.groupby(by='conn_tech').size()
81 fev_tcp.groupby(by='conn_tech').size()
nfevtcpv6.groupby(by='conn_tech').size()
83 #BRAND
mar_tcp.groupby(by='brand').size()
85 nmartcpv6.groupby(by='brand').size()
abr_tcp.groupby(by='brand').size()
87 nabrtcpv6.groupby(by='brand').size()
mai_tcp.groupby(by='brand').size()
89 nmaitcpv6.groupby(by='brand').size()
jun_tcp.groupby(by='brand').size()
91 njuntcpv6.groupby(by='brand').size()
jul_tcp.groupby(by='brand').size()
93 njultcpv6.groupby(by='brand').size()
ago_tcp.groupby(by='brand').size()
95 nagotcpv6.groupby(by='brand').size()
set_tcp.groupby(by='brand').size()
97 nsettcpv6.groupby(by='brand').size()
```

```
out_tcp.groupby(by='brand').size()  
99 nouttcpv6.groupby(by='brand').size()  
nov_tcp.groupby(by='brand').size()  
101 nnovtcpv6.groupby(by='brand').size()  
dez_tcp.groupby(by='brand').size()  
103 ndeztcpv6.groupby(by='brand').size()  
jan_tcp.groupby(by='brand').size()  
105 njantcpv6.groupby(by='brand').size()  
fev_tcp.groupby(by='brand').size()  
107 nfevtcpv6.groupby(by='brand').size()
```

ANEXO D - CÓDIGO UDP DA SESSÃO 3.2.2

```
1 import pandas as pd
pd.set_option('display.max_rows', None)
3
#leitura dos dados
5 mar_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_03_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
abr_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_04_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
7 mai_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_05_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
jun_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_06_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
9 jul_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_07_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
ago_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_08_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
11 set_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_09_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
out_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_10_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
13 nov_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_11_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
dez_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_12_2019_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
15 jan_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_01_2020_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
fev_udp = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/
    Netmetric_02_2020_UDP.csv', usecols=['lastip6', 'conn_tech', '
    osversion', 'brand'])
```

```
    'osversion', 'brand'] )
17
nmarudpv6 = mar_udp.query('lastip6.isnull()', engine='python')
19 nabrudpv6 = abr_udp.query('lastip6.isnull()', engine='python')
nmaiudpv6 = mai_udp.query('lastip6.isnull()', engine='python')
21 njunudpv6 = jun_udp.query('lastip6.isnull()', engine='python')
njuludpv6 = jul_udp.query('lastip6.isnull()', engine='python')
23 nagoudpv6 = ago_udp.query('lastip6.isnull()', engine='python')
nsetudpv6 = set_udp.query('lastip6.isnull()', engine='python')
25 noutudpv6 = out_udp.query('lastip6.isnull()', engine='python')
nnovudpv6 = nov_udp.query('lastip6.isnull()', engine='python')
27 ndezudpv6 = dez_udp.query('lastip6.isnull()', engine='python')
njanudpv6 = jan_udp.query('lastip6.isnull()', engine='python')
29 nfevudpv6 = fev_udp.query('lastip6.isnull()', engine='python')

31 #analise dos dados

33 #OS VERSION
mar_udp.groupby(by='osversion').size()
35 nmarudpv6.groupby(by='osversion').size()
abr_udp.groupby(by='osversion').size()
37 nabrudpv6.groupby(by='osversion').size()
mai_udp.groupby(by='osversion').size()
39 nmaiudpv6.groupby(by='osversion').size()
jun_udp.groupby(by='osversion').size()
41 njunudpv6.groupby(by='osversion').size()
jul_udp.groupby(by='osversion').size()
43 njuludpv6.groupby(by='osversion').size()
ago_udp.groupby(by='osversion').size()
45 nagoudpv6.groupby(by='osversion').size()
set_udp.groupby(by='osversion').size()
47 nsetudpv6.groupby(by='osversion').size()
out_udp.groupby(by='osversion').size()
49 noutudpv6.groupby(by='osversion').size()
nov_udp.groupby(by='osversion').size()
51 nnovudpv6.groupby(by='osversion').size()
dez_udp.groupby(by='osversion').size()
53 ndezudpv6.groupby(by='osversion').size()
jan_udp.groupby(by='osversion').size()
55 njanudpv6.groupby(by='osversion').size()
fev_udp.groupby(by='osversion').size()
```

```
57 nfevudpv6.groupby(by='osversion').size()
#CONN_TECH
59 mar_udp.groupby(by='conn_tech').size()
nmarudpv6.groupby(by='conn_tech').size()
61 abr_udp.groupby(by='conn_tech').size()
nabrudpv6.groupby(by='conn_tech').size()
63 mai_udp.groupby(by='conn_tech').size()
nmaiudpv6.groupby(by='conn_tech').size()
65 jun_udp.groupby(by='conn_tech').size()
njunudpv6.groupby(by='conn_tech').size()
67 jul_udp.groupby(by='conn_tech').size()
njuludpv6.groupby(by='conn_tech').size()
69 ago_udp.groupby(by='conn_tech').size()
nagoudpv6.groupby(by='conn_tech').size()
71 set_udp.groupby(by='conn_tech').size()
nsetudpv6.groupby(by='conn_tech').size()
73 out_udp.groupby(by='conn_tech').size()
noutudpv6.groupby(by='conn_tech').size()
75 nov_udp.groupby(by='conn_tech').size()
nnovudpv6.groupby(by='conn_tech').size()
77 dez_udp.groupby(by='conn_tech').size()
ndezudpv6.groupby(by='conn_tech').size()
79 jan_udp.groupby(by='conn_tech').size()
njanudpv6.groupby(by='conn_tech').size()
81 fev_udp.groupby(by='conn_tech').size()
nfevudpv6.groupby(by='conn_tech').size()
83 #BRAND
#"P30 Pro" se refere a uma marca especifica , sendo trocada para as
    demais marcas que aparecem na Tabela 4.11.
85 len(nmarudpv6[nmarudpv6['brand'] == 'P30 Pro'])
len(abr_udp[abr_udp['brand'] == 'P30 Pro'])
87 len(nabrudpv6[nabrudpv6['brand'] == 'P30 Pro'])
len(mai_udp[mai_udp['brand'] == 'P30 Pro'])
89 len(nmaiudpv6[nmaiudpv6['brand'] == 'P30 Pro'])
len(jun_udp[jun_udp['brand'] == 'P30 Pro'])
91 len(njunudpv6[njunudpv6['brand'] == 'P30 Pro'])
len(jul_udp[jul_udp['brand'] == 'P30 Pro'])
93 len(njuludpv6[njuludpv6['brand'] == 'P30 Pro'])
len(ago_udp[ago_udp['brand'] == 'P30 Pro'])
95 len(nagoudpv6[nagoudpv6['brand'] == 'P30 Pro'])
len(set_udp[set_udp['brand'] == 'P30 Pro'])
```

```
97 len(nsetudpv6[nsetudpv6['brand'] == 'P30 Pro'])
   len(out_udp[out_udp['brand'] == 'P30 Pro'])
99 len(noutudpv6[noutudpv6['brand'] == 'P30 Pro'])
   len(nov_udp[nov_udp['brand'] == 'P30 Pro'])
101 len(nnovudpv6[nnovudpv6['brand'] == 'P30 Pro'])
   len(dez_udp[dez_udp['brand'] == 'P30 Pro'])
103 len(ndeudpv6[ndeudpv6['brand'] == 'P30 Pro'])
   len(jan_udp[jan_udp['brand'] == 'P30 Pro'])
105 len(njanudpv6[njanudpv6['brand'] == 'P30 Pro'])
   len(fev_udp[fev_udp['brand'] == 'P30 Pro'])
107 len(nfevudpv6[nfevudpv6['brand'] == 'P30 Pro'])
```