



ANTONIO VINICIUS FERREIRA E SILVA

**UMA ANÁLISE COMPARATIVA DAS VERSÕES DO PROTOCOLO HTTP:
EVOLUÇÃO E PONTOS QUE AMPLIEM O USO DO HTTP/3.**

FORTALEZA

2021

ANTONIO VINICIUS FERREIRA E SILVA

UMA ANÁLISE COMPARATIVA DAS VERSÕES DO PROTOCOLO HTTP: EVOLUÇÃO E
PONTOS QUE AMPLIEM O USO DO HTTP/3.

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Sistemas de Informação do Centro Universitário Christus, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Me. Luiz Gonzaga
Mota Barbosa

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Centro Universitário Christus - Unichristus
Gerada automaticamente pelo Sistema de Elaboração de Ficha Catalográfica do
Centro Universitário Christus - Unichristus, com dados fornecidos pelo(a) autor(a)

S586a Silva, Antonio Vinicius Ferreira e.
UMA ANÁLISE COMPARATIVA DAS VERSÕES DO
PROTOCOLO HTTP : EVOLUÇÃO E PONTOS QUE AMPLIEM O
USO DO HTTP/3. / Antonio Vinicius Ferreira e Silva. - 2021.
38 f. : il. color.

Trabalho de Conclusão de Curso (Graduação) - Centro
Universitário Christus - Unichristus, Curso de Sistemas de
Informação, Fortaleza, 2021.

Orientação: Prof. Me. Luiz Gonzaga Mota Barbosa.

1. HTTP/3. 2. QUIC. 3. UDP. I. Título.

CDD 005

ANTONIO VINICIUS FERREIRA E SILVA

UMA ANÁLISE COMPARATIVA DAS VERSÕES DO PROTOCOLO HTTP: EVOLUÇÃO E
PONTOS QUE AMPLIEM O USO DO HTTP/3.

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Sistemas de Informação do Centro Universitário Christus, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Aprovada em: 23/06/2021

BANCA EXAMINADORA

Prof. Me. Luiz Gonzaga Mota Barbosa (Orientador)
Centro Universitário Christus (Unichristus)

Prof. Dr. Daniel Nascimento Teixeira
Centro Universitário Christus (Unichristus)

Prof. Me. Matheus Magalhães de Carvalho
Centro Universitário Christus (Unichristus)

RESUMO

Este trabalho apresenta uma visão comparativa das versões do protocolo HTTP, descrevendo sua evolução, melhorias, dificuldades e apresentando sugestões para ampliar o uso do HTTP/3. Descobriu-se que em sua última versão, o HTTP introduz melhorias consistentes, decorrente da mudança em sua pilha de protocolos, como o TCP ao QUIC e UDP. Também foi visto que essas melhorias são percebidas em ambientes específicos e que o protocolo ainda apresenta baixa utilização. Foi possível sugerir medidas que procuram viabilizar o uso do HTTP/3 e concluir que hoje esse protocolo já pode ser considerado uma alternativa para uso, em especial para redes de alta latência.

Palavras-chave: HTTP/3. QUIC. UDP. Redes de alta latência.

ABSTRACT

This work presents a comparative view of the HTTP protocol versions, describing evolution, improvements, difficulties and presenting suggestions to expand the use of HTTP/3. It was found that in its latest version, HTTP introduces consistent improvements as a result of the change in its protocol stack, such as TCP to QUIC and UDP. It was also seen that these improvements are watched in specific environments and that the protocol still has low usage. It was possible to suggest measures that seek to enable the use of HTTP/3 and conclude that today this protocol can already be considered an alternative for use, especially for high latency networks.

Keywords: HTTP/3. QUIC. UDP. High latency networks.

LISTA DE ILUSTRAÇÕES

Figura 1 – Camadas da arquitetura da internet e principais protocolos.	14
Figura 2 – Estrutura dos segmentos.	17
Figura 3 – Representação <i>3-way handshake</i>	22
Figura 4 – Conexão inicial HTTP/2 x HTTP/3.	25
Figura 5 – Pilha de protocolo para HTTP/2 e HTTP/3	26
Figura 6 – Porcentagem de uso de vários elementos em websites.	28
Figura 7 – Porcentagem de uso dos protocolos HTTP nos últimos 7 dias.	28
Figura 8 – Balanceamento de requisições por geolocalização	32

LISTA DE TABELAS

Tabela 1 – Classificação de trabalhos: natureza e abordagem	21
Tabela 2 – Melhorias do protocolo HTTP	27
Tabela 3 – Problemas por versão	27

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Contextualização e delimitação do tema	9
1.2	Problematização	9
1.3	Pressupostos	10
1.4	Objetivos	10
1.4.1	<i>Objetivo geral</i>	10
1.4.2	<i>Objetivos específicos</i>	10
1.5	Justificativa	10
1.6	Estrutura do trabalho	10
2	REFERENCIAL TEÓRICO	12
2.1	A Internet e seu crescimento ao longo dos anos	12
2.2	Camadas de rede: Arquitetura da Internet	13
2.2.1	<i>Camada de aplicação</i>	14
2.2.2	<i>Camada de transporte</i>	14
2.2.3	<i>Camada de rede</i>	14
2.2.4	<i>Camada de enlace</i>	15
2.2.5	<i>Camada física</i>	15
2.3	UDP vs TCP	15
2.3.1	<i>Características</i>	16
2.3.2	<i>Estrutura dos segmentos</i>	16
2.3.3	<i>UDP ou TCP?</i>	18
3	METODOLOGIA	20
3.1	Tipo de pesquisa	20
3.2	Procedimentos de coleta de dados	20
3.3	Classificação de trabalhos consultados	20
4	RESULTADOS	22
4.1	A Evolução do protocolo HTTP	22
4.1.1	<i>HTTP/1.1</i>	23
4.1.2	<i>HTTP/2</i>	24
4.1.3	<i>HTTP/3</i>	25

4.2	Fatores que dificultam a implementação do HTTP/3	27
4.2.1	<i>Alto consumo de CPU</i>	28
4.2.2	<i>Bloqueio do tráfego UDP</i>	29
4.2.3	<i>Protocolos ossificados e Implantação lenta</i>	29
4.2.4	<i>Variação de performance por região</i>	29
4.2.5	<i>Variação de performance por arquitetura</i>	30
4.3	Medidas para promover a adoção do HTTP/3	30
4.3.1	<i>Utilizar geolocalização para melhorar o atendimento às requisições</i>	30
4.3.2	<i>Utilizar entrega não ordenada de dados em cada fluxo para evitar aumento de latência</i>	32
4.3.3	<i>Pontos que podem ser melhorados no futuro</i>	33
5	CONCLUSÕES E TRABALHOS FUTUROS	34
	REFERÊNCIAS	35

1 INTRODUÇÃO

1.1 Contextualização e delimitação do tema

O *Hypertext Transfer Protocol* (HTTP) é um protocolo utilizado para envio de textos, mídias e outros dados pela Internet. Ele utiliza o modelo requisição/resposta onde um cliente estabelece comunicação com um servidor, realiza uma solicitação e aguarda até que o servidor retorne sua resposta (BERNERS-LEE *et al.*, 1996). Padronizado em 1996 pela *Internet Engineering Task Force* (IETF) na versão 1.0, foi substituído por sua versão 1.1 três anos depois (STENBERG, 2014).

O crescimento do tráfego da internet dos últimos anos, a grande quantidade de dados carregados em páginas web e a necessidade de garantir segurança na comunicação entre cliente-servidor, tem tornado o uso do HTTP/1.1 cada vez mais obsoleto (CARLUCCI *et al.*, 2015). Em resposta a isso, grandes empresas investem esforços para implementar novos protocolos visando ganhos em tempo de resposta e redução de latência, sem deixar de garantir segurança e prevenção de perdas (BELSHE *et al.*, 2015; LANGLEY *et al.*, 2017). Baseado nesses protocolos, mais especificamente o *Speedy* (SPDY), foi implementada a segunda versão do HTTP.

O HTTP/2 permite um uso mais eficiente dos recursos de rede e uma percepção reduzida de latência, introduzindo a compactação do campo de cabeçalho e permitindo várias trocas simultâneas na mesma conexão. Ele também introduz o envio não solicitado de representações de servidores para clientes. (BELSHE *et al.*, 2015, s/p).

Ainda assim o HTTP/2 possui ineficiências, das quais podemos citar o bloqueio de fila e uma grande carga de *handshakes* feitos por sua camada de transporte e segurança. Essas ineficiências são contornadas por sua terceira versão, o HTTP/3. De acordo com a W3Techs (2021) o HTTP/3, em padronização pela IETF com base no protocolo experimental *Quick UDP Internet Connections* (QUIC), possui uma taxa de utilização de 18,4% contra 50,7% do HTTP/2.

1.2 Problematização

Dessa maneira tem-se como pergunta da pesquisa “O que torna a taxa de utilização do HTTP/3 baixa tendo em vista os benefícios que ele apresenta sobre suas versões anteriores?”.

1.3 Pressupostos

Parte-se do pressuposto que existem diversos fatores que interferem na ampla utilização do HTTP/3 ou que tornam necessário a utilização de suas versões anteriores, como: uma grande carga de mudanças necessárias para implementar modificações sem ocasionar interrupção nos serviços existentes; se os benefícios de sua utilização são de fato satisfatórios para implementá-lo; incompatibilidade de ambientes com a pilha utilizada pelo HTTP/3; se existe resistência ou falta de interesse a mudanças por parte de empresas e profissionais;

1.4 Objetivos

1.4.1 Objetivo geral

O objetivo geral deste trabalho é analisar as versões do protocolo HTTP, descrever os motivos da baixa utilização da versão mais recente, e então propor medidas que ampliem seu uso.

1.4.2 Objetivos específicos

1. Caracterizar as versões do protocolo HTTP.
2. Apresentar os benefícios das novas versões.
3. Entender os fatores que impedem a utilização do HTTP/3.
4. Propor medidas para promover a adoção do protocolo.

1.5 Justificativa

Diante de todos os benefícios apontados pelo uso do HTTP/3, por pesquisas e estatísticas demonstradas em artigos, conferências e outros meios de veiculação da informação (LANGLEY *et al.*, 2017), torna-se necessário elencar os motivos de sua baixa utilização e apontar medidas para promovê-la.

1.6 Estrutura do trabalho

O capítulo 2 trata do referencial teórico, onde aborda pontos para a contextualização do objetivo desta pesquisa, bem como auxiliar o leitor, lembrando ou ampliando o

conhecimento em alguns assuntos inerentes ao tema da pesquisa.

O capítulo 3, trata da metodologia escolhida para realizar a pesquisa.

O capítulo 4 aborda mais detalhadamente a evolução do protocolo HTTP citada no capítulo 1, os pontos que dificultam a implementação do HTTP/3 e então propõe medidas para promover sua utilização.

O capítulo 5 apresenta as considerações finais quanto ao HTTP/3, seus benefícios e usabilidade, e apresenta os trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 A Internet e seu crescimento ao longo dos anos

Segundo Kurose e Ross (2013) “A Internet é uma rede de computadores que interconecta centenas de milhões de dispositivos de computação ao redor do mundo.” O nascimento de redes de dispositivos conectados ocorreu no início da década de 1960 com a interconexão de universidades e institutos educacionais nos Estados Unidos, sendo liderada pela *Advanced Research Projects Agency* (ARPA), ficando conhecida como ARPAnet. Anos depois, outras redes independentes começaram a surgir, como a Telenet, uma rede comercial que usava tecnologia baseada na ARPAnet.

Na década de 1970 foi realizada a interconexão dessas redes, surgindo o termo *interneting* (CERF; KAHN, 1974). Contudo, foi somente na década de 1990 que a Internet teve seu grande crescimento com a criação da *World Wide Web* (WWW), e de protocolos e ferramentas essenciais para comunicação Web, como as versões iniciais do HTTP e HTML, bem como um servidor Web e um navegador (KUROSE; ROSS, 2013).

Uma infraestrutura que provê serviços a aplicações. Tais aplicações incluem correio eletrônico, navegação na Web, redes sociais, mensagem instantânea, Voz sobre IP (VoIP), vídeo em tempo real, jogos distribuídos, compartilhamento de arquivos *peer-to-peer* (P2P), televisão pela Internet, login remoto e muito mais. (KUROSE; ROSS, 2013, p. 4).

De acordo com Kurose e Ross (2013), no início, a internet era utilizada para transferência de mensagens e era utilizada especificamente para meios acadêmicos, contudo, o aumento de sua utilização para fins comerciais, o desenvolvimento de navegadores com interfaces gráficas, e o crescimento na implementação de internet banda larga em redes cabeadas, e móveis, contribuiu para o:

1. Surgimento de uma gama de serviços fornecidos pela internet, como pesquisa, comércio eletrônico, redes sociais, compartilhamento de arquivos, jogos e reprodução de vídeo e voz.
2. Crescimento no número de dispositivos conectados, bem como no surgimento de novos dispositivos como *Internet of Things* (IoT), ou mesmo usuários autônomos, como *bots* que são responsáveis por cerca de 45% do tráfego da internet (CLOUDFLARE, 2021).
3. A forma como os usuários interagem e encaram a internet e o período em que usuários ficam conectados a rede.

Uma página web (também denominada documento) é construída de objetos. Um objeto é simplesmente um arquivo – tal como um arquivo HTML, uma imagem JPEG, uma imagem GIF, um Applet Java, um clipe de áudio e assim por diante – que se pode acessar com um único URL. A maioria das páginas web é construída de um arquivo-base HTML e diversos objetos referenciados. (KUROSE; ROSS, 2013, p. 72).

Outro fator importante é a quantidade de objetos presentes em uma página web. Enquanto no início, era comum somente textos serem exibidos em páginas web, hoje, somente uma pequena parte de um site é composta por um arquivo de texto. Imagens, animações, áudios, vídeos, e *plugins* para outros serviços correspondem a maior parte do tráfego de internet ultimamente. (PETERSON; DAVIE, 2012)

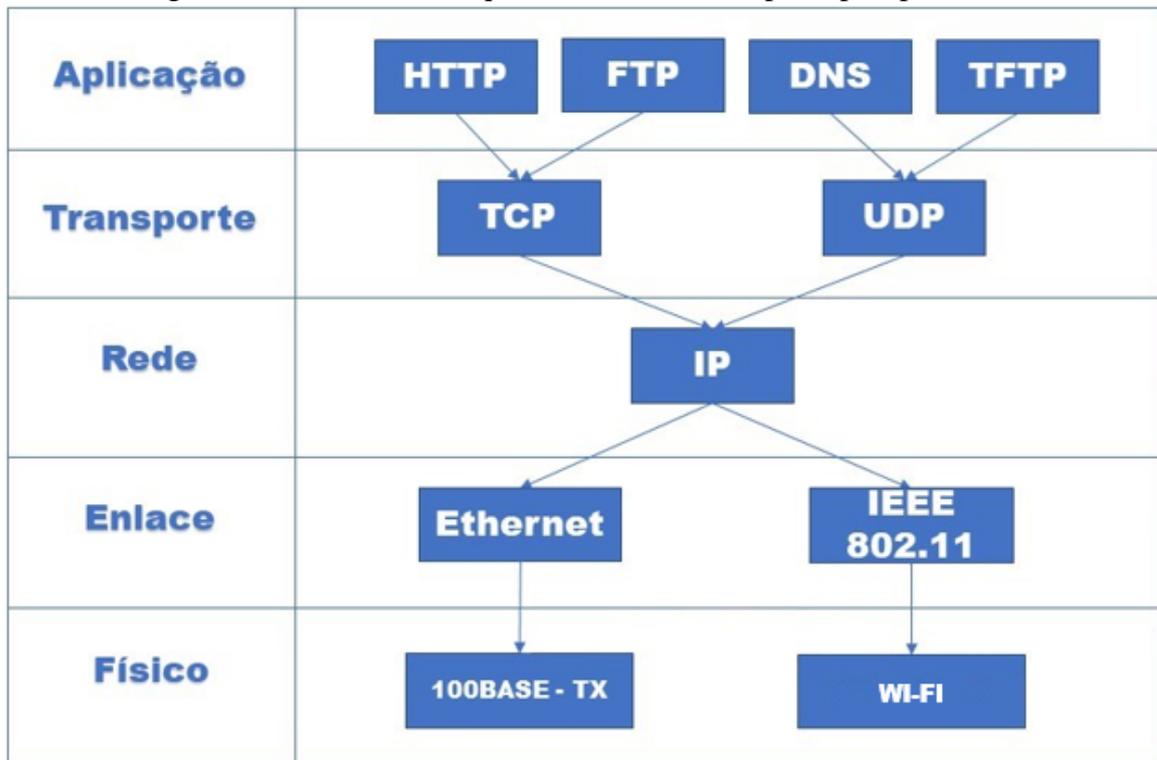
2.2 Camadas de rede: Arquitetura da Internet

A arquitetura da internet é um modelo padrão de camadas definidas para implementar a comunicação entre computadores em rede, também conhecida como pilha TCP/IP, TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão) e IP (*Internet Protocol* - Protocolo de Internet), devido seus protocolos mais conhecidos (DUNKELS, 2001). Foi utilizado o modelo apresentado por Kurose e Ross (2013) como mostra a Figura 1, onde cada camada utiliza os protocolos da camada abaixo para o envio de informações através da rede.

Muitos autores apresentam um modelo de quatro camadas ao unir a camada de enlace e a física. Um modelo semelhante e bem conhecido é o OSI (*Open Systems Intercommunication* - Intercomunicação de código aberto), definido pela ISO (*International Organization for Standardization* - Organização Internacional de Normalização), que apresenta 7 camadas ao incluir 2 camadas entre aplicação e transporte. Contudo, o modelo TCP/IP foi mais difundido e utilizado desde o início da Internet (PETERSON; DAVIE, 2012).

O conceito de camadas oferece dois recursos úteis. Primeiro, elas decompõem o problema de construção de uma rede em componentes mais gerenciáveis. Em vez de implementar um software monolítico que faz tudo o que você sempre deseja, você pode implementar várias camadas, cada uma resolvendo uma parte do problema. Em segundo lugar, ela fornece um design mais modular. Se você decidir que deseja adicionar algum novo serviço, pode precisar apenas modificar a funcionalidade em uma camada, reutilizando as funções fornecidas em todas as outras camadas. (PETERSON; DAVIE, 2012, s/p).

Figura 1 – Camadas da arquitetura da internet e principais protocolos.



Fonte: Elaborado pelo autor (2021)

2.2.1 Camada de aplicação

A camada de aplicação é a quinta camada na arquitetura da Internet. É nessa camada onde ficam aplicações como os navegadores e servidores web. Aplicações conseguem através dos protocolos, realizar transferência de documentos pela web, trocando pacotes de informação com a aplicação em outro sistema final (KUROSE; ROSS, 2013).

2.2.2 Camada de transporte

A quarta camada na arquitetura da internet é a camada de transporte, de acordo com Kurose e Ross (2013), o “protocolo da camada de transporte fornece comunicação lógica entre processos de aplicação que rodam em hospedeiros diferentes”, os protocolos dessa camada são os responsáveis pelo envio das mensagens da camada de aplicação, podendo ou não ter um controle nesse fluxo de envio ou uma garantia de entrega.

2.2.3 Camada de rede

A camada de rede suporta a interconexão de várias tecnologias de rede em uma única rede lógica fazendo o roteamento e encaminhamento dos datagramas na rede. Os pacotes são

atribuídos do endereço de origem e destino da camada de rede e encaminhados para o envio (KUROSE; ROSS, 2013). Segundo Tanenbaum (2003) o controle de congestionamento também pode ser uma responsabilidade da camada de rede.

2.2.4 Camada de enlace

A camada de enlace é a responsável pela transmissão de pacotes, ponto a ponto, entre cada trecho da origem ao destino. Diferente dos protocolos das camadas acima, esses protocolos existem em cada nó do percurso (KUROSE; ROSS, 2013).

Os protocolos da camada de enlace, podem oferecer entrega garantida entre nós adjacentes, e serviços que regulam o tráfego de dados no nó emissor, a fim de evitar gargalo no buffer do nó receptor (TANENBAUM, 2003).

2.2.5 Camada física

A primeira camada trata do envio de todos os *bits* que compõem um quadro entre um trecho da origem ao destino, devendo garantir que o *bit* enviado chegará no próximo ponto de forma íntegra (TANENBAUM, 2003). Segundo Kurose e Ross (2013), o modo de envio desses bits depende do meio de transmissão do enlace.

2.3 UDP vs TCP

“A Internet tem dois protocolos principais na camada de transporte, um protocolo sem conexões e outro orientado a conexões.” (TANENBAUM, 2003) O protocolo orientado a conexões é o TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão), e o sem conexão é chamado de UDP (*User Datagram Protocol* - Protocolo de Datagrama do Usuário) .

Ambos os protocolos são usados para envio de mensagens fim a fim. Contudo, eles possuem diferenças nas formas de execução, e por mais que possam ser usadas em qualquer ocasião, algumas aplicações se adaptam melhor a um ou outro protocolo (KUROSE; ROSS, 2013).

2.3.1 Características

O UDP é um protocolo de transporte simples que não adiciona nenhuma função ao envio do segmento a não ser o de multiplexação/demultiplexação e de verificação de erros. Entretanto, mesmo fornecendo verificação de erros, o UDP não realiza nenhuma ação de recuperação, em alguns casos esses segmentos são descartados ou enviados à aplicação acompanhados de um aviso (KUROSE; ROSS, 2013).

Em contraste com um protocolo de demultiplexação simples como o UDP, um protocolo de transporte mais sofisticado é aquele que oferece um serviço de fluxo de bytes confiável e orientado para conexão. Esse serviço tem se mostrado útil para uma ampla variedade de aplicações, pois libera a aplicação de ter que se preocupar com dados perdidos ou reordenados. (PETERSON; DAVIE, 2012, s/p).

O TCP é orientado para conexão pois, antes que um processo de aplicação possa começar a enviar dados a outro, os dois processos precisam primeiro se “apresentar” — isto é, devem enviar alguns segmentos preliminares um ao outro para estabelecer os parâmetros da transferência de dados [...] Como três segmentos são enviados entre dois hospedeiros, esse procedimento de estabelecimento de conexão é muitas vezes denominado apresentação de três vias (*3-way handshake*). (KUROSE; ROSS, 2013, p. 169).

Além de uma “entrega confiável e em sequência”(TANENBAUM, 2003), o TCP também desempenha outras funções;

1. "Controle de fluxo: para cada um desses fluxos de bytes que permite ao receptor limitar a quantidade de dados que o remetente pode transmitir em um determinado momento."
2. "Controle de congestionamento altamente ajustado: a ideia desse mecanismo é restringir a rapidez com que o TCP envia dados, não para evitar que o remetente ultrapasse o receptor, mas para evitar que o remetente sobrecarregue a rede."

2.3.2 Estrutura dos segmentos

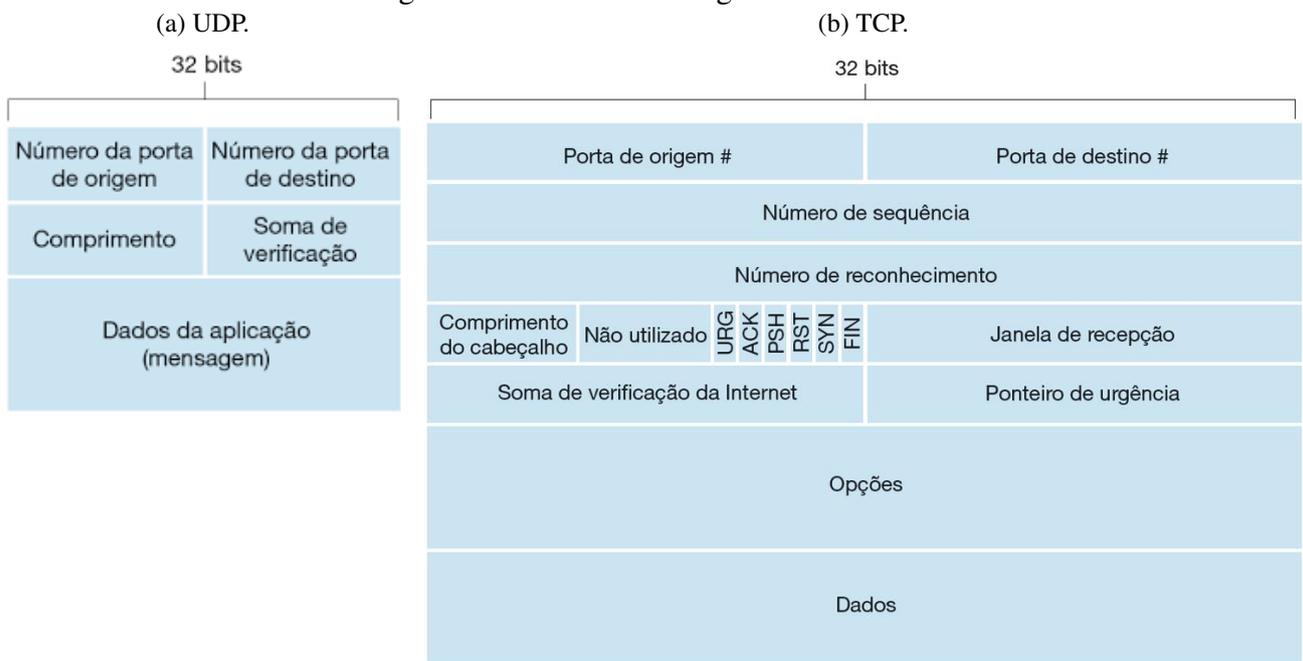
A estrutura do segmento UDP definida na Figura 2a possui cinco campos. Um campo de tamanho variável para dados e quatro campos destinados ao cabeçalho do segmento, com 2 bytes (ou 16 bits) cada. Esses campos determinam porta de origem e destino, possibilitando a função de multiplexação/demultiplexação, comprimento em bytes do segmento, e soma de verificação usada pelo destinatário para verificar se o segmento está íntegro (POSTEL, 1980).

A estrutura do TCP, por sua vez, definido na Figura 2b, possui geralmente um cabeçalho de 20 bytes, mas pode variar devido ao campo Opções. Note que ambas as estruturas são dispostas verticalmente, identificando que cada linha possui 4 bytes (ou 32 bits).

O cabeçalho TCP possui, como no UDP, os campos de porta de origem e destino, o campo comprimento do cabeçalho, usado para indicar onde começam os dados, e o campo de soma de verificação para verificar a integridade do segmento. Os outros campos do TCP são:

1. Número de sequência e número de reconhecimento: são usados para carregar o número do primeiro byte do segmento enviado e o número do próximo byte que um ponto espera receber do ponto do lado oposto (considerando que o protocolo TCP aceita fluxos de dados seguindo em ambas as direções de um canal). Esses campos possibilitam a execução de envio dos segmentos em paralelo sem a necessidade de confirmação de cada segmento para enviar o seguinte.
2. Janela de recepção: armazena o número de bytes que um destinatário se dispõe a receber.
3. Campo de flag: contém 6 flags de 1 bit cada.
 - a) URG: define se dados nesse segmento são urgentes, e alimentam o campo ponteiro de urgência com o último byte dos dados urgentes.
 - b) ACK: Indica se o campo de reconhecimento possui um valor válido.
 - c) SYN, FIN e RST: usados para estabelecer, encerrar ou reiniciar uma conexão.
 - d) PSH: Indica ao destinatário que o envio dos dados à camada superior deve ser feito imediatamente.

Figura 2 – Estrutura dos segmentos.



Melhorias sobre os protocolos IP e TCP foram propostas e implementadas com a intenção de melhorar o controle de congestionamento. O que anteriormente era inferido pelos nós finais devido a perda de pacotes, passou a ser notificado pela camada de rede. Essa notificação é conhecida como Notificação Explícita de Congestionamento ou *Explicit Congestion Notification* (ECN).

Floyd *et al.* (2001) definem o ECN e atribui dois bits no cabeçalho IP e dois bits no cabeçalho TCP, sendo os dois últimos chamados de CWR e ECE, definidos utilizando 2 bits do campo não utilizado” apresentado na Figura 2b.

De acordo com Floyd *et al.* (2001), para utilizar o ECN em uma transmissão TCP, o remetente envia um pacote SYN com as flags ECE e CWR definidas, e o receptor devolve um pacote SYN-ACK com as flags ECE definida e CWR não definida, indicando que ambos suportam ECN. Isso permite que os nós finais reconheçam a notificação de congestionamento apontada pelos dispositivos de camada três presentes no trajeto, caso exista. Essa notificação é feita ao marcar a flag CE do cabeçalho IP nos datagramas desta conexão. Qualquer outra configuração para pacotes SYN e SYN-ACK indicam que os nós finais não suportam ECN).

Uma vez que a flag CE seja marcada em um cabeçalho IP, o receptor interpretará que o trajeto está congestionado e avisará o outro emissor, definindo a flag ECE de seu pacote ACK. O emissor receberá esse pacote e responderá definindo a flag CWR informando que sua janela de congestionamento foi reduzida.

2.3.3 UDP ou TCP?

“O TCP foi projetado especificamente para oferecer um fluxo de bytes fim a fim confiável em uma inter-rede não confiável” (TANENBAUM, 2003). Grande parte dos protocolos utilizados para troca de mensagens na internet necessitam de garantia de entrega, e de integridade dos dados, por isso tendem a utilizar o TCP, permitindo que a camada de aplicação abstraia esses serviços.

Embora o TCP seja preferível ao UDP em muitos casos, por fornecer um serviço de entrega de dados confiável. Kurose e Ross (2013) afirmam que muitas aplicações se adaptam melhor ao UDP devido a sua estrutura de cabeçalho enxuta, e a inexistência de funções de controle de transmissão, congestionamento, reconhecimento de recepção e estabelecimento de conexão que tendem a tornar o tempo de transporte maior. Serviços que precisam realizar rápidas conexões e que não necessitam de confirmação de recepção como o *Domain Name Service*

(DNS), ou aplicações de Voz por IP (VoIP), e *streamings* em tempo real que tendem a reagir mal ao controle de congestionamento ou reenvio de pacotes perdidos, são exemplos de uso do UDP.

3 METODOLOGIA

3.1 Tipo de pesquisa

A pesquisa realizada foi de natureza descritiva e explicativa, com abordagem quantitativa e qualitativa, utilizando o método de pesquisa bibliográfica.

A pesquisa descritiva preocupa-se em observar os fatos e registrá-los, analisá-los, classificá-los e interpretá-los, e o pesquisador não interfere neles. (ANDRADE, 2002).

Segundo Gil (2002, p.48) “Uma pesquisa explicativa pode ser a continuação de outra descritiva, visto que a identificação dos fatores que determina o fenômeno exige que esteja suficientemente escrito e detalhado.”

Os conceitos de quantidade e qualidade não são totalmente dissociados, na medida em que de um lado a quantidade é uma interpretação, uma tradução, um significado atribuído à grandeza com que um fenômeno se manifesta (portanto, é uma qualificação dessa grandeza), e de outro ela precisa ser interpretada qualitativamente, pois, sem relação a algum referencial, não tem significação em si. (BASTOS, 2008, s/p).

Para Richardson (1999), utilizar a metodologia qualitativa possibilita ao pesquisador descrever a complexidade de determinado problema, e analisar a interação das variáveis.

3.2 Procedimentos de coleta de dados

O uso de livros e RFCs para a contextualização e para determinar a evolução do tema foi de extrema importância, considerando todo o detalhamento com o qual eles abordam seu conteúdo.

Os dados apresentados foram obtidos a partir da análise dos resultados apresentados em artigos periódicos, bem como em ferramentas de monitoramento que apresentam resultados constantemente atualizados.

3.3 Classificação de trabalhos consultados

Os principais trabalhos consultados, descrevem de forma detalhada e didática, a adoção e análise de performance do HTTP/3, ou do QUIC em comparação a outras versões do HTTP. Apresentando graficamente seus resultados, além de fornecer importantes observações para embasá-los.

Dentre eles, podemos citar a pesquisa realizada por (LANGLEY *et al.*, 2017) em conjunto a outros pesquisadores. Descrevendo a implantação do QUIC em diferentes ambientes e apresentando sua performance em diferentes dispositivos e regiões geográficas. Seu trabalho serviu como base para outras pesquisas como as feitas por (TREVISAN *et al.*, 2021) e (SAIF *et al.*, 2020). A Tabela 1 apresenta os principais trabalhos utilizados em relação a sua natureza e abordagem.

Tabela 1 – Classificação de trabalhos: natureza e abordagem

Titulo	Natureza	Abordagem
The quic transport protocol: Design and internet-scale deployment	D/E	QT/QL
Http over udp: an experimental investigation of quic	D/E	QT/QL
An Early Benchmark of Quality of Experience Between HTTP/2 and HTTP/3 using Lighthouse	D	QT/QL
Measuring HTTP/3: Adoption and Performance	D/E	QT/QL

Legenda: D - Descritiva E - Explicativa QT - Quantitativo QL - Qualitativo

Fonte: Elaborado pelo autor (2021)

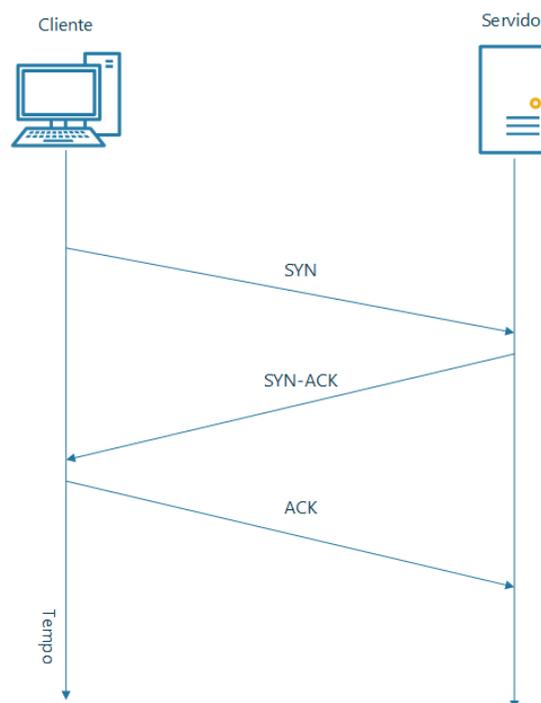
4 RESULTADOS

4.1 A Evolução do protocolo HTTP

O HTTP é o protocolo mais utilizado em toda a Internet, parte desse resultado vem do seu modo de operação. O método de requisição/resposta foi bem aceito por todos, pois permite o consumo de recursos e conteúdo sob demanda. Muitos autores também atribuem o fato do HTTP ser executado sobre o TCP ser responsável por esse resultado, pois com ele nem uma das aplicações, cliente ou servidor, precisa se preocupar com mensagens perdidas, mensagens duplicadas, mensagens longas ou confirmações.

Como apresentado brevemente na introdução, com o HTTP o cliente estabelece comunicação com servidor, realiza uma solicitação e aguarda até que o servidor retorne sua resposta. Sendo mais específico, o cliente estabelece uma conexão TCP através de uma porta, por padrão 80 ou a 443 (HTTPS). Essa conexão, conhecida como apresentação de três vias ou *3-way handshake*, consiste no envio de um segmento TCP ao servidor, que o reconhece e envia um segmento em resposta ao cliente, que também o reconhece enviando novamente para o servidor como mostra a Figura 3. Esse último segmento também é acompanhado da requisição HTTP (KUROSE; ROSS, 2013).

Figura 3: Representação *3-way handshake*.



Fonte: Elaborado pelo autor (2021)

No HTTP/1.0 uma conexão TCP era estabelecida para cada requisição/resposta, sendo encerrada pelo servidor após o envio da resposta. (BERNERS-LEE *et al.*, 1996).

Em um mundo no qual as páginas da Web típicas consistiam inteiramente em texto HTML, esse método era adequado. Após alguns anos, uma página da Web média continha grandes números de ícones, imagens e outros atrativos visuais, e assim o estabelecimento de uma conexão TCP para transportar um único ícone tornou-se uma operação muito dispendiosa. (TANENBAUM, 2003, s/p).

4.1.1 HTTP/1.1

Como o uso do HTTP/1.0 apresentava grande latência, levando em consideração que uma página com vários elementos demorava muito tempo para ser carregada inteiramente, o HTTP/1.1 foi lançado, suportando conexões persistentes. Nielsen *et al.* (1999) cita vantagens no uso de conexões persistentes que podemos resumir em duas, são elas:

1. Redução de latência nas solicitações, economia de CPU e memória, em todos os nós, redução do congestionamento de rede.
2. Solicitações e respostas HTTP podem ser canalizadas em uma conexão. Um cliente pode fazer várias solicitações sem esperar cada resposta, tornando a conexão TCP mais eficiente. Esse recurso é chamado de *pipeline*.

Embora o *pipeline* oferecesse alguma eficiência, seu uso não resolvia um problema evidenciado pelo uso de conexões persistentes. Mesmo realizando requisições sem de fato ter recebido a resposta anterior, o TCP entrega as requisições em ordem para a camada de aplicação, isso implica que caso um segmento seja maior que os demais, ou mesmo que algum segmento tenha se perdido, ou entregue corrompido, os segmentos posteriores esperarão até que esse segmento seja entregue corretamente ao destino. Esse problema é conhecido como bloqueio de cabeça de fila ou *head of line blocking*, utilizaremos o termo bloqueio de fila em referência a esse problema no decorrer do trabalho. Segundo Stenberg (2014), o *pipelining* HTTP é desabilitado em muitas aplicações clientes.

Outro mecanismo para maximizar o tempo de resposta e mitigar o bloqueio de fila é o paralelismo de conexões TCP, várias conexões abertas no mesmo host e a adesão de outros hosts dividindo as requisições entre eles, diminuindo o tempo de espera total. ((STENBERG, 2014)(BELSHE *et al.*, 2015)).

O preço dessa prática é um aumento no processamento e de consumo de memória dos nós finais e intermediários, que por consequência pode levar a um congestionamento e um aumento nos custos das empresas.

4.1.2 HTTP/2

O crescimento no número de dispositivos conectados à Internet e o aumento em sua utilização tornaram as melhorias do HTTP/1.1 obsoletas, e evidenciaram a necessidade de mais medidas para diminuir o tempo de resposta de páginas. O HTTP/2 foi proposto com base no protocolo experimental SPDY, que na época, já possuía evidências de melhorias sobre o HTTP/1.1.

Os resultados iniciais foram muito animadores: quando fizemos o download dos 25 principais sites por conexões de rede doméstica simuladas, vimos uma melhoria considerável no desempenho, as páginas carregavam até 55% mais rápido. (BELSHE; PEON, 2009).

Segundo Stenberg (2014) o HTTP/2 trouxe as seguintes melhorias em performance e em redução de latência:

1. Multiplexação de fluxos em uma única conexão: os pacotes de vários fluxos são misturados na mesma conexão e depois divididos no destinatário.
2. Priorização de fluxos: cada fluxo possui uma prioridade. Em alguns casos cliente ou servidor podem precisar forçar o envio ou recebimento de um fluxo.
3. Compressão de cabeçalho: dados dos cabeçalhos das mensagens são comprimidos, o destinatário recebe a mensagem e descomprime o cabeçalho através de uma lista de decodificação trocada anteriormente. (PEON; RUELLAN, 2015)
4. *Server push*: O servidor pode enviar recursos para o cliente sem solicitação anterior.
5. Cancelamento de mensagem: interromper a troca de uma mensagem sem a necessidade de encerrar a conexão TCP.
6. Controle de fluxo: cada um dos vários fluxos da conexão tem controles individuais para evitar que o número de pacotes enviados ultrapasse o estabelecido pelo controle de fluxo TCP, garantindo que nenhum fluxo seja bloqueado. (BELSHE *et al.*, 2015)

Embora o HTTP/2 diminua a latência da comunicação criando vários fluxos em uma única conexão, ele não mitiga o bloqueio de fila, uma vez que o TCP irá bloquear a entrega de um fluxo de pacotes caso um pacote anterior ainda esteja sendo recebido.

Em redes com altas taxas de perda de pacotes, a utilização do HTTP/1.1 mostrou-se melhor por utilizar mais conexões TCP para distribuir pacotes perdidos, uma vez que o HTTP/2 tipicamente só implementa uma conexão por host. (STENBERG, 2018).

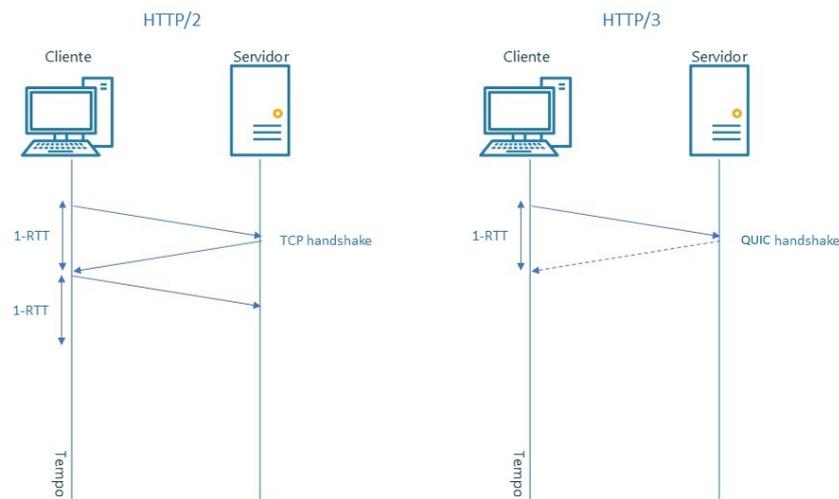
4.1.3 HTTP/3

O HTTP/3 vem sendo padronizado utilizando muitos conceitos do HTTP/2, mas também segue muitos conceitos do QUIC, protocolo experimental que apresentou reduções de latência em média de 8% em respostas de pesquisas, e de 18% em reprodução de vídeos, em comparação com o HTTP/2 (LANGLEY *et al.*, 2017). Bem como apresentou uma redução no tempo de recuperação de pacotes, em redes com perdas induzidas, quando comparado ao SPDY (CARLUCCI *et al.*, 2015).

Das diferenças e melhorias que Stenberg (2018) cita ao HTTP/3 em comparação com sua versão anterior, podemos citar:

1. A utilização de ID de conexões: com intuito de garantir a continuidade de uma comunicação com servidor, mesmo que o cliente alterne de IP.
2. Redução de handshakes para estabelecer conexão: como mostra a Figura 4, HTTP/3 utiliza somente um tempo de ida e volta ou *1 Round Trip Time* (1-RTT) para estabelecer conexão. Se já existir um ID de conexão, o estabelecimento da conexão é de 0-RTT.
3. Uso do TLS 1.3: além de garantir mais segurança que a versão utilizada no HTTP/2, também apresenta um redução de RTT em seu *handshake*.

Figura 4: Conexão inicial HTTP/2 x HTTP/3.

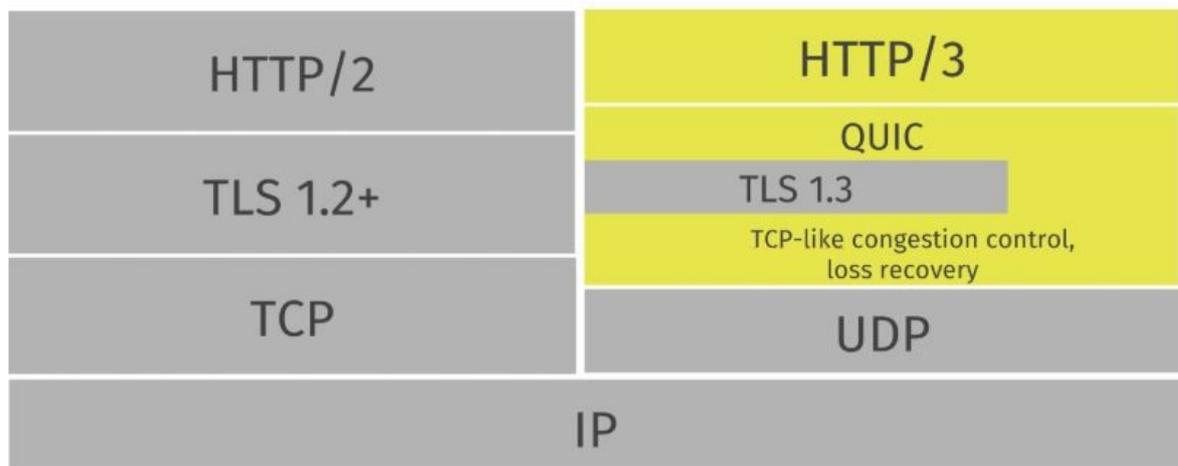


Além dos pontos citados acima, os pontos que mais possuem destaque nas mudanças implementadas pelo HTTP/3 são: O uso do protocolo UDP ao invés do TCP na camada de transporte de sua pilha, e a entrega ordenada em cada fluxo, mas não entre eles.

Ao usar o UDP, o HTTP/3 mitiga o problema de bloqueio de fila que ainda afetava o HTTP/2. De forma semelhante, ao realizar uma entrega de pacotes em ordem em cada fluxo, mas não entre eles, o HTTP/3 limita somente aquele fluxo a uma espera, caso algum pacote precise ser retransmitido.

Como o UDP não oferece retransmissões de pacotes, controle de congestionamento, e outros recursos presentes no TCP, essas atribuições foram absorvidas pelo QUIC, que compõe parte da camada de aplicação e de transporte, como mostra a Figura 5.

Figura 5: Pilha de protocolo para HTTP/2 e HTTP/3



Fonte: Stenberg (2018)

Analisando a Tabela 2 podemos ter uma visão mais clara das melhorias apresentadas, na evolução do protocolo HTTP. Dentre elas, é importante reforçar que:

1. O *Pipeline* apresentado desde o HTTP/1.1 não era amplamente utilizado devido a recepção rigorosa do TCP. De acordo com suas RFCs, o HTTP/2 e 3 também não fazem uso desse recurso.
2. HTTP/3 não utiliza priorização de fluxos, de acordo com Stenberg (2018) a abordagem de priorização para HTTP/2 foi considerada muito complicada, ou mesmo uma falha absoluta.
3. A compressão de cabeçalhos do HTTP/3 não utiliza o mesmo recurso do HTTP/2. QPACK é uma variação do HPACK, ainda em desenvolvimento com intuito de reduzir o bloqueio de fila (KRASIC *et al.*, 2021).

4. O TLS1.3 também é utilizado no HTTP/2, contudo, ele é disponibilizado como uma opção, enquanto seu uso no HTTP/3 é mandatório.

Tabela 2: Melhorias do protocolo HTTP

Versão	HTTP/1.0	HTTP/1.1	HTTP/2	HTTP/3
Conexões persistentes		✓	✓	✓
<i>Pipeline</i>		✓		
Fluxos Multiplexados			✓	✓
Priorização de fluxos			✓	
Compressão de cabeçalhos			✓	✓
<i>Server Push</i>			✓	✓
Controle de fluxo			✓	✓
ID de conexões				✓
0-RTT				✓
TLS1.3				✓

Fonte: Elaborado pelo autor (2021)

Da mesma forma, na Tabela 3 podemos perceber melhor que problemas existem em cada versão. A latência era um problema no HTTP/1.0 mas passou a ser tratado por todas as versões posteriores. O uso de CPU nas primeiras versões era alto devido a constante necessidade de estabelecimento de conexão e posteriormente por conta do paralelismo de conexões TCP. O bloqueio de fila só passou a ser tratado no HTTP/3, mas o uso de recursos do HTTP/3 torna o consumo de CPU alto relacionado ao HTTP/2.

Tabela 3: Problemas por versão

Versão	HTTP/1.0	HTTP/1.1	HTTP/2	HTTP/3
Latência	×			
CPU	×	×		×
Bloqueio de fila		×	×	

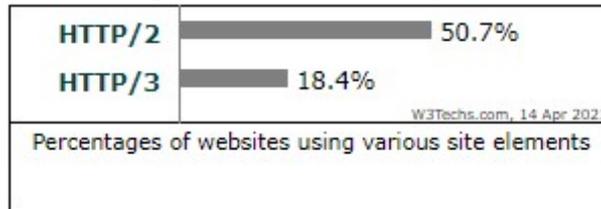
Fonte: Elaborado pelo autor (2021)

4.2 Fatores que dificultam a implementação do HTTP/3

Embora o HTTP/3 possua melhorias sobre suas versões anteriores, e os protocolos experimentais possuam ganhos em redução de latência, recuperação de pacotes, controle de congestionamento e segurança. Seu uso ainda não foi amplamente difundido, como podemos ver nas Figuras 6 e 7, que apresentam as taxas de utilização das versões do HTTP, apresentado por diferentes sites que realizam esse monitoramento.

As diferenças nos resultados, são baseadas nos períodos de análise, e no universo de ambos os testes. Contudo, a proximidade de ambas as pesquisas reforçam um consentimento em relação à taxa de uso principalmente das versões 2 e 3 do HTTP.

Figura 6: Porcentagem de uso de vários elementos em websites.

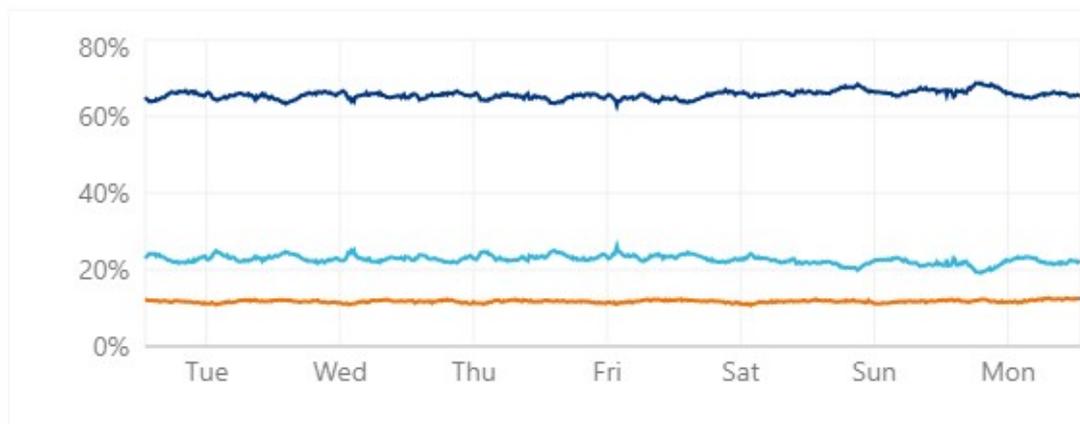


Fonte: W3Techs (2021)

Figura 7: Porcentagem de uso dos protocolos HTTP nos últimos 7 dias.

HTTP/1.x vs HTTP/2 vs HTTP/3

[Learn](#) how Cloudflare supports HTTP/2 and HTTP/3 to speed up your websites without requiring any changes to your existing codebase.



Fonte: Cloudflare (2021)

4.2.1 Alto consumo de CPU

De acordo com Stenberg (2018), cargas de requisições sobre UDP utilizam aproximadamente o dobro de processamento necessário para as mesmas cargas de requisições sobre TCP. A explicação apresentada por ele é que pilhas UDP principalmente no Linux não são tão otimizadas quanto a pilha TCP, uma vez que não são tradicionalmente usadas para transferências

de alta velocidade como essa. Isso ocorre em parte porque o uso de TCP em todo o mundo teve mais tempo para amadurecer, melhorar e obter assistência de hardware.

4.2.2 Bloqueio do tráfego UDP

Várias organizações limitam ou bloqueiam o tráfego UDP, exceto na porta 53 que é usada para DNS. E o fazem com intuito de proteger-se de ataques de amplificação, quando um invasor pode fazer uma grande quantidade de tráfego de saída para atingir vítimas inocentes.

4.2.3 Protocolos ossificados e Implantação lenta

Além do bloqueio de tráfego UDP por grandes organizações. Ainda devemos considerar o bloqueio e atraso na comunicação por roteadores, firewalls, e outros equipamentos que compõem a interconexão de redes.

Essas caixas foram implantadas há algum tempo, quando os protocolos tinham um conjunto de recursos daquela época. A introdução de novos recursos ou mudanças no comportamento que não eram conhecidos antes pode acabar sendo considerada ruim ou ilegal por essas caixas. Esse tráfego pode muito bem ser interrompido ou atrasado até o ponto em que os usuários realmente não queiram usar esses recursos. Isso é chamado de ossificação de protocolo. (STENBERG, 2018, s/p).

Conhecer os protocolos que eles roteiam, monitoram, ou modificam não é fácil, uma vez que o software necessário para adquirir esse conhecimento nem sempre está atualizado. Atualizar esses equipamentos também não é uma tarefa fácil, levando em consideração o número de equipamentos, ou quais estão de fato aptos a receber atualizações.

Além disso, mudanças em protocolos geram mudanças em *kernels* de sistemas operacionais, que também levam um tempo considerável para serem implantados. Segundo Stenberg (2018), mesmo aprimoramentos no TCP não são amplamente suportados atualmente.

4.2.4 Variação de performance por região

De acordo com Langley *et al.* (2017), nos testes realizados com QUIC, ocorreram variações de performance e de retransmissão de pacotes em diferentes regiões geográficas. Para regiões com menor perda de pacotes e menor tempo de resposta, a performance do QUIC foi bem aproximada do TCP, chegando algumas vezes a apresentar impacto negativo no desempenho. A

performance aumentava à medida que uma determinada região possuía maior perda de pacotes e possuía tempo de resposta maior.

Acredita-se que essa performance pode estar atrelada aos limites de CPU do cliente, ou a ineficiências do planejador do lado do sistema operacional do cliente (LANGLEY *et al.*, 2017). Como o HTTP/3 utiliza QUIC em sua pilha, é possível considerar que um resultado semelhante ocorre nele também.

4.2.5 Variação de performance por arquitetura

Langley *et al.* (2017) também aponta que a performance do QUIC para dispositivos móveis é geralmente menor do que para usuários de desktop. Isso se deve ao fato de que os aplicativos móveis costumam ser ajustados para seu ambiente. Por exemplo, quando os aplicativos limitam o conteúdo para pequenas telas móveis, as otimizações de transporte têm menos impacto. Os telefones celulares também são mais restritos à CPU do que os dispositivos de desktop, fazendo com que a CPU seja o gargalo quando a largura de banda da rede é abundante.

4.3 Medidas para promover a adoção do HTTP/3

O HTTP/3 ainda está em padronização e embora muitos trabalhos já o considerem nos estágios finais, podem existir modificações que apresentem melhorias ainda mais significativas para promover sua adoção em larga escala. Contudo, conforme ocorreu no passado com o uso do paralelismo de conexões no mesmo host e a utilização de outros hosts dividindo as requisições para mitigar uma deficiência do HTTP/1.1, podemos propor medidas que promovam a implantação do HTTP/3 em face de suas atuais deficiências.

4.3.1 Utilizar geolocalização para melhorar o atendimento às requisições

Essa sugestão utiliza conceitos bastante abordados em redes, e embora teórica, apresenta uma nova visão para eles além da intenção principal de maximizar o uso do protocolo HTTP/3. Contudo, antes de apresentá-la é necessário conhecer e reforçar alguns pontos.

1. Um dos fatores que diminui os benefícios do HTTP/3 é sua baixa performance em relação a redes com baixa latência. Langley *et al.* (2017) apresenta esse resultado ao comparar a performance do QUIC e HTTP/2 em regiões que em geral possuíam redes com alta latência, e regiões com redes de baixa latência.

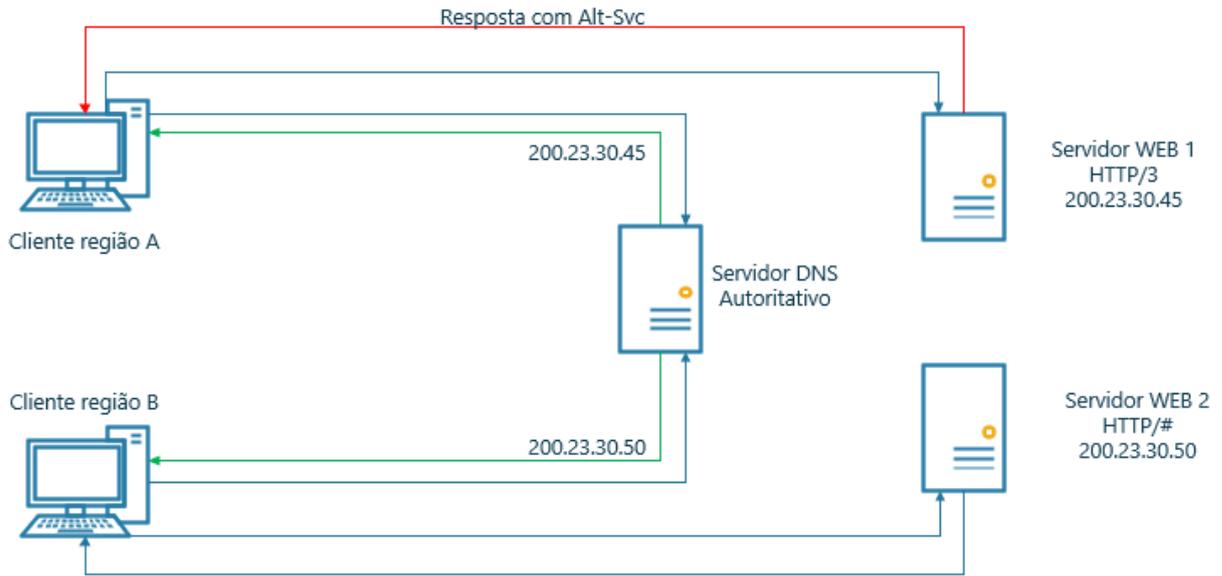
2. É possível obter informações de geolocalização de um endereço IP. Essas informações são disponibilizadas em listas ou mapas de IP geográfico como são chamadas por Gerend (2021), e existem muitos fornecedores que os disponibilizam de forma gratuita ou comercial (IP2LOCATION, 2021; WONDERNETWORK, 2021). Essas listas podem conter localização, dados do provedor e RTT médio de cada endereço.
3. Através de serviços DNS, é possível responder a consultas DNS com base na localização geográfica dos clientes. Usando listas citadas anteriormente, essas políticas DNS podem ser configuradas manualmente indicando todas as localizações necessárias em um padrão nome e endereço IP (GEREND, 2021), bem como é possível utilizar serviços DNS que já apresentem esse conhecimento de localizações, como é o caso de algumas ferramentas de DNS em nuvem (AZURE, 2021; AWS, 2021).
4. Um servidor web pode anunciar a disponibilidade de HTTP/3 por meio do campo de cabeçalho de resposta Alt-Svc (*Alternative Services*) HTTP caso o cliente não esteja usando esse protocolo por padrão (BISHOP, 2021).

Com base nisso podemos sugerir a utilização da geolocalização do cliente para designar o servidor mais indicado para atender suas requisições, onde um servidor DNS analisa o endereço de origem de uma solicitação de resolução DNS e resolveria essa solicitação, indicando um servidor web que oferece comunicação com HTTP/3 caso o cliente esteja em uma rede de alta latência.

Como podemos ver na Figura 8, quando o cliente região A realiza uma solicitação de resolução DNS, ela é recebida pelo servidor DNS autoritativo do domínio requisitado, que analisa o endereço de origem e verifica se ele se enquadra em alguma política de roteamento por geolocalização ou não. Sendo o IP de origem pertencente a uma localização que possui um RTT alto, o servidor DNS responde a solicitação de resolução DNS passando o endereço do servidor web 1, que possui HTTP/3 habilitado. Se o endereço de origem não atender aos critérios, o IP do servidor web 2 é passado em resposta a solicitação de resolução DNS.

Após resolver o endereço do servidor web, o cliente A envia sua requisição, que ao ser recebida pelo servidor web 1, verifica se o cliente enviou a solicitação com HTTP/3. Caso a requisição esteja em outra versão do protocolo, o servidor responde, sugerindo que eles utilizem o HTTP/3 nas próximas comunicações através do cabeçalho de resposta Alt-Svc. Se o cliente reconhecer HTTP/3, ele pode tentar estabelecer uma conexão QUIC com o host e a porta indicados; se a conexão for bem-sucedida, o cliente pode enviar requisições HTTP/3.

Figura 8: Balanceamento de requisições por geolocalização



Fonte: Elaborado pelo autor (2021)

Embora a Figura 8 abstraia essa informação, o servidor DNS autoritativo não vê o endereço IP do cliente mas sim o endereço do IP local do cliente, que em sua grande maioria pertence ao provedor de acesso do cliente, permitindo uma estimativa da localização geográfica do usuário. Vale ressaltar que, se uma consulta atingir o servidor DNS autoritativo por meio de vários saltos, a política considerará apenas o IP do último resolvedor do qual o servidor DNS recebe a consulta.

4.3.2 Utilizar entrega não ordenada de dados em cada fluxo para evitar aumento de latência

Embora autores como Stenberg (2018) tenham abordado a entrega ordenada de dados em fluxo como um recurso obrigatório do QUIC, Iyengar e Thomson (2021) afirmam que o QUIC permite que a camada de aplicação decida se essa entrega será ordenada ou não. De acordo Bishop (2021), “embora o QUIC permita a entrega fora de ordem em um fluxo, o HTTP/3 não usa esse recurso”. Utilizando os campos ID de fluxo e Deslocamento para colocar os dados em ordem em um fluxo.

Devido ao mecanismo de entrega ordenada do TCP, a multiplexação no HTTP/2 ainda sofre com os efeitos do bloqueio de fila. No HTTP/3 esse efeito é drasticamente reduzido com o transporte por UDP e a independência de entrega entre fluxos. No entanto, ao ordenar dados em cada fluxo, o bloqueio de fila ainda se torna possível (IYENGAR; THOMSON, 2021).

Bishop (2021) não descreve os motivos para a ordenação de dados em cada fluxo no HTTP/3, limitando-se somente a afirmar que o protocolo faz isso.

Deste modo, sugere-se o uso de fluxos de dados não ordenados com a intenção evitar o bloqueio de fila, fazendo com que somente aqueles dados que foram perdidos ou corrompidos demorem mais a ser entregues a aplicação. Essa sugestão pode trazer efeitos sobretudo em redes de altas latências onde o QUIC já possui um rendimento superior ao HTTP/2, podendo fazer sua curva de performance ser ainda mais positiva.

4.3.3 Pontos que podem ser melhorados no futuro

Além das sugestões propostas neste trabalho, existem fatores que só podem ou devem ser efetivamente tratados com a ampla implementação do HTTP/3. O alto consumo de processamento devido a cargas de requisições sobre UDP nos sistemas finais, o bloqueio de tráfego UDP por organizações e por nós intermediários são exemplos disso. Ações direcionadas a esses fatores devem se tornar mais visíveis, levando em consideração as iniciativas de grandes *players* com HTTP/3 e QUIC nos últimos anos, onde podemos citar (CLOUDFLARE, 2020) e Google através do trabalho de (LANGLEY *et al.*, 2017). No entanto, definir um prazo para essas mudanças está além do escopo deste trabalho.

5 CONCLUSÕES E TRABALHOS FUTUROS

O HTTP/3 oferece uma abordagem diferente das suas versões anteriores. Utilizar UDP na camada de transporte e tratar controles de congestionamento, retransmissão e correção de erros a nível de aplicação torna-o um protocolo de rápida e fácil implantação, indo no sentido contrário a sua contrapartida o TCP.

Além disso, a inclusão de IDs de conexões e uso de TLS 1.3 apresenta uma redução de latência grande comparada às versões anteriores no quesito estabelecimento de conexão. Essas e outras melhorias implementadas apresentam benefícios quando comparados com HTTP/2 em circunstâncias específicas como em redes de alta latência, e apresentam rendimentos iguais ou inferiores em redes com pouca ou nenhuma perda de pacote e baixa latência, sendo este último caso, ligado fortemente incompatibilidade de ambientes com a pilha utilizada pelo HTTP/3.

Diante disso, infere-se que até o momento, o HTTP/3 funciona bem como uma opção ao HTTP/2, permitindo que quem implemente ambos consiga obter melhores resultados em diferentes circunstâncias de rede. Novas atualizações definirão se o potencial do HTTP/3 superará completamente o HTTP/2.

Em trabalhos futuros poderá ser considerado a implementação e avaliação das medidas sugeridas neste trabalho. Em especial, ao implementar e avaliar uma versão do protocolo HTTP/3 que permita a entrega de dados fora de ordem em cada fluxo, buscando entender se de fato ele apresenta os resultados sugeridos.

Outro trabalho a ser considerado está relacionado a um ponto que ainda não foi discutido em trabalhos científicos ou outras fontes de informação que este trabalho buscou, como resistência ou falta de interesse a mudanças por parte de empresas e profissionais em utilizar o HTTP/3. Para trabalhos futuros pode-se considerar o desenvolvimento de uma pesquisa para entender o quanto esse fator limita a implementação do HTTP/3.

REFERÊNCIAS

- ANDRADE, M. M. d. **Como preparar trabalhos para cursos de pós-graduação: Noções práticas**. 5. ed. [S.l.]: São Paulo: Atlas, 2002.
- AWS. **Amazon Route 53**. 2021. <<https://aws.amazon.com/pt/route53/>>. Acesso em: 15/06/2021.
- AZURE, M. **Traffic Manager - Cloud Based DNS Load Balancing | Microsoft Azure**. 2021. <<https://azure.microsoft.com/en-us/services/traffic-manager/>>. Acesso em: 15/06/2021.
- BASTOS, N. M. G. Introdução à metodologia do trabalho acadêmico. **Fortaleza: Nacional**, 2008.
- BELSHE, M.; PEON, R. **A 2x Faster Web**. 2009. <<https://blog.chromium.org/2009/11/2x-faster-web.html>>. Acesso em: 18/05/2021.
- BELSHE, M.; PEON, R.; THOMSON, M. **Hypertext Transfer Protocol Version 2 (HTTP/2)**. RFC Editor, 2015. RFC 7540. (Request for Comments, 7540). Disponível em: <<https://rfc-editor.org/rfc/rfc7540.txt>>.
- BERNERS-LEE, T.; NIELSEN, H.; FIELDING, R. T. **Hypertext Transfer Protocol – HTTP/1.0**. RFC Editor, 1996. RFC 1945. (Request for Comments, 1945). Disponível em: <<https://rfc-editor.org/rfc/rfc1945.txt>>.
- BISHOP, M. **Hypertext Transfer Protocol Version 3 (HTTP/3)**. [S.l.], 2021. Work in Progress. Disponível em: <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>>.
- CARLUCCI, G.; CICCIO, L. D.; MASCOLO, S. Http over udp: an experimental investigation of quic. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. [S.l.: s.n.], 2015. p. 609–614.
- CERF, V. G.; KAHN, R. A protocol for packet network interconnection. **IEEE Transactions of Communications**, v. 22, n. 5, p. 627–641, 1974.
- CLOUDFLARE. **HTTP/3: the past, the present, and the future**. 2020. <<https://blog.cloudflare.com/http3-the-past-present-and-future/>>. Acesso em: 18/06/2021.
- CLOUDFLARE. **Automated Traffic**. 2021. <<https://radar.cloudflare.com/>>. Acesso em: 12/05/2021.
- DUNKELS, A. Design and implementation of the lwip tcp/ip stack. **Swedish Institute of Computer Science**, v. 2, n. 77, 2001.
- FLOYD, S.; RAMAKRISHNAN, D. K. K.; BLACK, D. L. **The Addition of Explicit Congestion Notification (ECN) to IP**. RFC Editor, 2001. RFC 3168. (Request for Comments, 3168). Disponível em: <<https://rfc-editor.org/rfc/rfc3168.txt>>.
- GEREND, J. **Usar política de DNS para gerenciamento de tráfego baseado em localização geográfica com servidores primários**. 2021. <<https://docs.microsoft.com/pt-br/windows-server/networking/dns/deploy/primary-geo-location>>. Acesso em: 14/06/2021.
- GIL, A. C. **Como elaborar projetos de pesquisa**. [S.l.]: São Paulo: Atlas, 2002, p.48. v. 4.

IP2LOCATION. **IP Address Ranges by Country**. 2021. <<https://lite.ip2location.com/ip-address-ranges-by-country>>. Acesso em: 15/06/2021.

IYENGAR, J.; THOMSON, M. **QUIC: A UDP-Based Multiplexed and Secure Transport**. RFC Editor, 2021. RFC 9000. (Request for Comments, 9000). Disponível em: <<https://rfc-editor.org/rfc/rfc9000.txt>>.

KRASIC, C. B.; BISHOP, M.; FRINDELL, A. Internet-Draft, **QPACK: Header Compression for HTTP/3**. Internet Engineering Task Force, 2021. Work in Progress. Disponível em: <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-qpack-21>>.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet uma abordagem top-down**. 6. ed. [S.l.]: Pearson, 2013.

LANGLEY, A.; RIDDOCH, A.; WILK, A.; VICENTE, A.; KRASIC, C.; ZHANG, D.; YANG, F.; KOURANOV, F.; SWETT, I.; IYENGAR, J. *et al.* The quic transport protocol: Design and internet-scale deployment. In: **Proceedings of the conference of the ACM special interest group on data communication**. [S.l.: s.n.], 2017. p. 183–196.

NIELSEN, H.; MOGUL, J.; MASINTER, L. M.; FIELDING, R. T.; GETTYS, J.; LEACH, P. J.; BERNERS-LEE, T. **Hypertext Transfer Protocol – HTTP/1.1**. RFC Editor, 1999. RFC 2616. (Request for Comments, 2616). Disponível em: <<https://rfc-editor.org/rfc/rfc2616.txt>>.

PEON, R.; RUELLAN, H. **HPACK: Header Compression for HTTP/2**. RFC Editor, 2015. RFC 7541. (Request for Comments, 7541). Disponível em: <<https://rfc-editor.org/rfc/rfc7541.txt>>.

PETERSON, L. L.; DAVIE, B. S. **Computer Networks: A Systems Approach**. [S.l.]: Elsevier, 2012.

POSTEL, J. B. **User Datagram Protocol**. RFC Editor, 1980. RFC 768. (Request for Comments, 768). Disponível em: <<https://rfc-editor.org/rfc/rfc768.txt>>.

RICHARDSON, R. J. **Pesquisa social: métodos e técnicas**. 3. ed. [S.l.]: São Paulo: Atlas, 1999.

SAIF, D.; LUNG, C.-H.; MATRAWY, A. An early benchmark of quality of experience between http/2 and http/3 using lighthouse. **arXiv preprint arXiv:2004.01978**, 2020.

STENBERG, D. **HTTP2 explained**. [S.l.]: GitBook, 2014. <<https://http2-explained.haxx.se/en>>. Acesso em: 14/04/2021.

STENBERG, D. **HTTP/3 explained**. [S.l.]: GitBook, 2018. <<https://http3-explained.haxx.se/en>>. Acesso em: 14/04/2021.

TANENBAUM, A. S. **Computer Networks**. 4. ed. [S.l.]: Prentice Hall- PTR, New Jersey, 2003.

TREVISAN, M.; GIORDANO, D.; DRAGO, I.; KHATOUNI, A. S. Measuring http/3: Adoption and performance. **arXiv preprint arXiv:2102.12358**, 2021.

W3TECHS. **Usage statistics of site elements for websites**. 2021. <https://w3techs.com/technologies/overview/site_element>. Acesso em: 13/04/2021.

WONDERNETWORK. **Global Ping Statistics**. 2021. <<https://wondernetwork.com/pings>>. Acesso em: 15/06/2021.