UNIVERSIDADE FEEVALE

MATHEUS RODRIGUES KAUTZMANN

ANÁLISE DO USO DE WEBRTC PARA COMPARTILHAMENTO DE ARQUIVOS P2P

(Título Provisório)

Anteprojeto de Trabalho de Conclusão

Novo Hamburgo
2021

MATHEUS RODRIGUES KAUTZMANN

ANÁLISE DO USO DE WEBRTC PARA COMPARTILHAMENTO DE ARQUIVOS P2P

(Título Provisório)

Anteprojeto de Trabalho de Conclusão de Curso, apresentado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação pela Universidade Feevale

Orientador: Dr. Gabriel da Silva Simões

Novo Hamburgo
2021

RESUMO

WebRTC é um conjunto de tecnologias que possibilitam a comunicação direta entre diferentes clientes pela internet. O conceito de redes de comunicação direta não é novo, conhecido como Peer to Peer ou simplesmente P2P, pois possui implementações conhecidas como a rede BitTorrent e outras. A grande vantagem do WebRTC é o fato de se tratar de um projeto de código aberto, com um conjunto de APIs flexíveis e que já é considerado um verdadeiro padrão para comunicação em tempo real entre *browsers*. Este trabalho busca analisar aplicações para o WebRTC no compartilhamento de arquivos de forma segura e privada entre dispositivos conectados diretamente, enquanto compara com outras soluções comumente usadas. O objetivo é desenvolver um aplicativo utilizando as APIs do WebRTC a fim de averiguar as diferenças da aplicação desta tecnologia em comparação com o modelo cliente-servidor utilizando HTTPS já presente nos discos virtuais na nuvem, serviços de conversas por mensagem que também transferem arquivos como WhatsApp, Telegram e Signal e protocolos proprietários de transferência local entre dispositivos como os presentes no AirDrop e no Google Files. Para isso, o trabalho pretende comparar soluções de mercado com este aplicativo a ser desenvolvido, buscando destacar os pontos negativos e positivos de cada abordagem através de testes em diferentes tipos de rede. Acredita-se que o uso de WebRTC para compartilhamento de arquivos é promissor, pois o mesmo já se provou muito útil e popular para uso em aplicações de comunicação em tempo real, como *chats*, aplicativos de videoconferência e VoIP.

Palavras-chave: Rede P2P. WebRTC. Compartilhamento de arquivos.

SUMÁRIO

MOTIVAÇÃO

O modelo de cliente-servidor é amplamente utilizado (TANENBAUM, 2002) e serve para vários casos de uso na *internet*, inclusive para compartilhamento de arquivos, onde o servidor disponibiliza arquivos a serem acessados por diversos clientes diferentes. Para este fim, o protocolo comumente usado é o HTTP ou sua variante segura HTTPS. No entanto, existem cenários onde um compartilhamento efêmero é mais interessante, como no caso de uma transferência de arquivo pontual ou em um cenário descentralizado que não pode contar com servidores fixos sob controle de alguma entidade específica.

Existem os mais diferentes serviços que servem como disco virtual na nuvem hoje, como Google Drive, Dropbox, Onedrive e outros. Porém todos eles dependem de servidores fixos e sob controle das entidades que controlam os respectivos serviços.

Apesar de possuírem uma transferência de dados segura entre o cliente e servidor, usando HTTPS e também criptografar os dados em descanso, os provedores desses serviços ainda controlam as chaves necessárias para criptografar os arquivos, e não possuem a chamada criptografia de ponta-a-ponta. Segundo Ermoshina et. al (2016), esta permite que somente o emissor e o destinatário dos dados consigam ler seu conteúdo e está presente em serviços de mensagem como Telegram, WhatsApp e Signal. No entanto, estes serviços de mensagem possuem servidores armazenando os arquivos enviados, pois eles operam de forma assíncrona, eles precisam aguardar que o destinatário apareça e faça o download do arquivo para então decidir se o mesmo será removido ou não.

Por outro lado, há também os serviços de compartilhamento local de forma direta, que servem para transferências pontuais, como o AirDrop e Google Files. Estas soluções contemplam dispositivos específicos, no caso do AirDrop a compatibilidade se restringe a dispositivos rodando iOS ou macOS e o Google Files funciona entre dispositivos Android.

O nível de comunicação entre os modelos também diferem, no modelo cliente-servidor dos discos na nuvem utiliza-se a *internet* como meio pelo qual os arquivos são trafegados, sendo necessária uma conexão constante à *internet* por parte do cliente e do servidor. Já no compartilhamento via AirDrop ou Google Files só é possível realizar a transferência em rede local . No caso do AirDrop, é criada uma rede Wi-Fi localmente entre os dispositivos de maneira proprietária, ambos os dispositivos devem estar na mesma rede e o protocolo utilizado é de código fechado e desenvolvido pela Apple. O Google Files, assim como o AirDrop, cria uma rede Wi-Fi privada a partir de uma comunicação inicial via Bluetooth e utiliza um protocolo próprio desenvolvido pela Google para a transferência de

arquivos. Um item em comum entre as soluções deste tipo é que exigem proximidade física entre os dispositivos, pois utilizam comunicação direta entre os *hardwares* dos aparelhos.

Acredita-se que o WebRTC pode ser uma boa alternativa para compartilhamento de arquivos P2P, pois permite a flexibilidade de enviar os arquivos tanto por rede local ou através da *internet*. Sendo necessária apenas uma conexão momentânea a um servidor inicial a fim de estabelecer a conexão P2P entre as partes, este servidor é conhecido como servidor de sinalização. Após isso a comunicação é direta e síncrona entre os dispositivos participantes, com a transferência de arquivos ocorrendo através de APIs do WebRTC em cima do protocolo da camada de transporte UDP ou TCP, utilizando um protocolo de aplicação a ser desenvolvido para cada caso de uso.

Segundo Loreto e Romano (2014), a suíte WebRTC conta com criptografia ponta-a-ponta através de DTLS, que é uma variação compatível com UDP do conhecido TLS usado no HTTPS. Com isso, proporciona privacidade aos pares, pois toda a ligação e comunicação se dá entre as partes diretamente e depois de feita a transferência dos dados não há registros salvos em qualquer servidor. Portanto, pode-se dizer que a utilização de WebRTC é segura, privada e confiável.

O conjunto de APIs WebRTC proporciona maneiras de escolher a melhor rota para a conexão entre os pares, então se os mesmos estiverem em uma mesma LAN, estes usarão o caminho mais curto para se comunicar, sem precisar passar pela *internet* (WAN). A escolha do melhor caminho é feita através de um processo de seleção de candidatos, sendo o melhor deles o escolhido para realizar a conexão.

Existem cenários em que uma conexão P2P direta pode não ser possível, como no caso da presença de um *firewall* na rede ou de alguma configuração de NAT que impossibilite a conexão entre as duas partes. Para estes casos se faz necessário o uso de algum servidor de encaminhamento, que apenas servirá de ponte entre as duas partes, os tipos mais comuns são os servidores do tipo STUN e TURN.

Mesmo em cenários de utilização de servidores de encaminhamento, os dados trafegados não podem ser lidos, pois, como descrito anteriormente, a criptografia do DTLS garante a confidencialidade dos dados em tráfego. Os serviços de STUN ou TURN apenas usam o cabeçalho do UDP para obter o endereço do destino para onde os pacotes devem ser enviados. Como o WebRTC é flexível, existe também a possibilidade de usar TCP no lugar de UDP, que apesar de mais lento, pode auxiliar a contornar a existência de *firewalls* bloqueando tráfego UDP.

Atualmente o uso de WebRTC é bastante popular para aplicações que necessitam de comunicação em tempo real, como aplicações de *chat*, VoIP, chamadas de vídeo e até mesmo em aplicações destinadas à realização de conferências virtuais. Já pode-se observar o uso das tecnologias em programas conhecidos como o Twilio (SUMRAK, 2020) e Discord (VASS, 2018).

Graças à acessibilidade abrangente do WebRTC, que vai desde os navegadores modernos, que já possuem o suporte às APIs de forma embarcada, até bibliotecas de código aberto que implementam a suíte de APIs, é possível se observar que essa tecnologia é uma alternativa viável para estudo visando obter uma forma confiável, privada e segura para o envio de arquivos P2P.

Portanto, o objetivo deste trabalho é verificar a viabilidade e o potencial do uso de WebRTC para transferências de arquivos entre dois ou mais clientes de forma direta. Isto será feito através do desenvolvimento de um aplicativo real utilizando a suíte de tecnologias do WebRTC e posterior comparação dos pontos positivos e negativos da solução com as outras alternativas populares disponíveis no mercado.

OBJETIVOS

Objetivo geral

O objetivo deste trabalho é analisar o uso de WebRTC como alternativa para o envio e recebimento de arquivos de forma P2P. Para tanto, será criado um aplicativo que utilize o conjunto de APIs WebRTC a fim de demonstrar suas funcionalidades e comparar com outras soluções similares existentes.

Objetivos específicos

- Desenvolver um aplicativo demonstrando o uso de WebRTC para transferir arquivos entre dispositivos;
- Comparar solução com uso de AirDrop e Google Files;
- Comparar solução com os discos virtuais na nuvem: Google Drive e Dropbox;
- Entender as vantagens e limitações do uso de WebRTC;
- Verificar o comportamento da solução encontrada em cenários com firewalls e diferentes configurações de NAT;
- Investigar os cenários nos quais um envio de arquivo através do método proposto é vantajoso.

METODOLOGIA

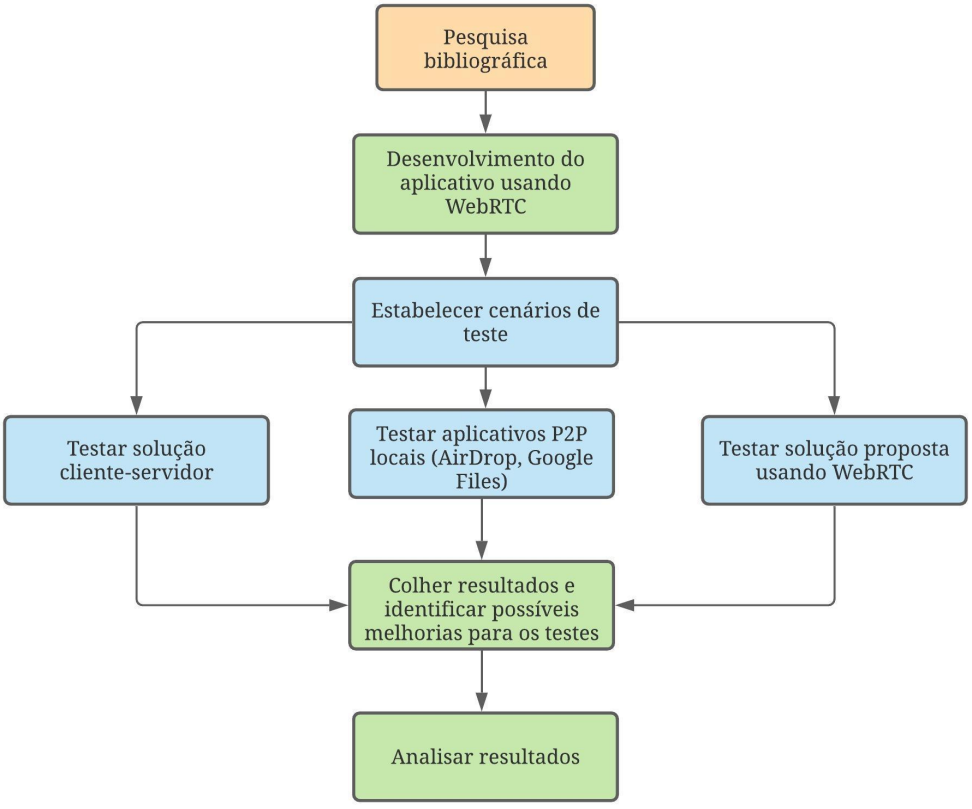
O objetivo do trabalho é testar o uso de WebRTC em casos reais através de uma aplicação a ser desenvolvida utilizando a suíte de tecnologias do WebRTC, então pode-se dizer que este trabalho constitui-se de uma pesquisa aplicada sobre o uso das APIs WebRTC para a transferência de arquivos P2P.

A pesquisa será feita via observação sistemática em conjunto com experimentação em ambiente local e na nuvem, visando ter uma comparação justa entre os diferentes métodos atuais do mercado e o método a ser desenvolvido em cima da suíte WebRTC.

O processo será dividido em três macro etapas: pesquisa bibliográfica, desenvolvimento do aplicativo utilizando WebRTC e, por fim, será feita a análise e comparação dos diferentes métodos de transferência de arquivos em diferentes cenários.

O fluxograma resumido do processo de pesquisa está presente na figura 1.

Figura 1 - Fluxograma que representa o desenvolvimento do trabalho.



Fonte: Autor

A etapa de pesquisa bibliográfica contempla buscas em bibliografia envolvendo obras que mencionam redes de computadores, WebRTC, redes P2P e assuntos relacionados. O objetivo desta etapa é levantar conhecimento para embasar as etapas posteriores da pesquisa.

O desenvolvimento do aplicativo utilizando WebRTC está presente na segunda etapa do trabalho. Esta etapa servirá para obter o aplicativo base que será comparado com as outras soluções de mercado na etapa seguinte. O principal requisito para este aplicativo é que ele consiga enviar um arquivo de forma privada e segura entre dois clientes, sem precisar passar por um servidor para transferir o arquivo em si, configurando uma rede P2P.

A terceira etapa é a principal etapa do trabalho, onde serão avaliados e coletados os resultados da pesquisa. Aqui se aplicará a observação sistemática para a comparação da ferramenta desenvolvida com as já presentes no mercado, visando abranger, entre outras, as seguintes variáveis: desempenho considerando tempo de transferência, privacidade (quem pode ver os dados), conectividade (em que cenários de conexão a solução funciona) e acessibilidade (que elementos de *hardware* e *software* são necessários para usar a solução).

CRONOGRAMA

Trabalho de Conclusão I

Etapa	Meses			
	Mar	Abr	Mai	Jun
Redação do anteprojeto				
Revisão do anteprojeto				
Entrega do anteprojeto				
Levantamento bibliográfico				
Pesquisa sobre WebRTC				
Análise das ferramentas de mercado				
Desenvolvimento inicial do aplicativo				
Redação TC I				
Revisão TC I				
Entrega do TC I				

Trabalho de Conclusão II

Etapa	Meses			
	Ago	Set	Out	Nov
Finalização do desenvolvimento do aplicativo				
Criação dos cenários para teste				
Execução dos cenários de teste no aplicativo				
Comparação de resultados com outras ferramentas				
Redação TC II				
Revisão TC II				
Entrega do TC II				
Apresentação do trabalho à banca				

BIBLIOGRAFIA

TANENBAUM, Andrew S. Computer Networks. 4th. ed. Upper Saddle River, NJ: Prentice Hall, 2002. 912 p.

ERMOSHINA, K.; MUSIANI, F.; HALPIN, H.. End-to-End Encrypted Messaging Protocols: An Overview. In: Transactions on Petri Nets and Other Models of Concurrency XV, 2016. p. 244–254.

LORETO, Salvatore; ROMANO, Simon Pietro. Real-Time Communication with WebRTC. 1st. ed. Sebastopol, CA: O'Reilly, 2014. 224 p.

VASS, Jozsef. How Discord Handles Two and Half Million Concurrent Voice Users using WebRTC. Discord Blog. Estados Unidos, 2018. Disponível em: <https://blog.discord.com/how-discord-handles-two-and-half-million-concurrent-voice-users-using-webrtc-ce01c3187429>. Acesso em: 23 mar. 2021.

SUMRAK, Jesse. How to Get Started With WebRTC: Intro to Browser APIs. Twilio Blog. Estados Unidos, 2020. Disponível em: <https://www.twilio.com/blog/get-started-webrtc>. Acesso em: 23 mar. 2021.