

Throw – 웹 브라우저 기반 크로스플랫폼 파일전송 시스템 개발

이호선¹, 임규정¹, 유현창¹
¹고려대학교 정보대학 컴퓨터학과
 {hoanun1804, gjilim2485, yuhc}@korea.ac.kr

Throw – A Browser Based Cross Platform File Transfer System

HoSun Lee¹, Gyujeong Lim¹, Heonchang Yu¹
¹Dept. of Computer Science and Engineering, Korea University

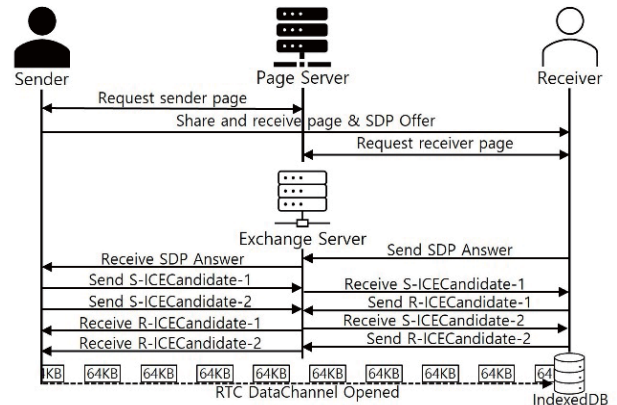
요 약

본 연구는 브라우저 기반 P2P 파일 전송 시스템인 Throw 를 개발하고, 기존 클라우드 스토리지 서비스와의 성능을 비교하였다. Throw 는 WAN 환경에서 평균 2 배 빠른 전송 속도를 보였으며, 다양한 운영체제에서 사용 가능한 장점이 있음을 확인하였다. Throw 는 브라우저 기반 P2P 파일 전송 기술의 가능성을 보여주었으며, 향후 성능 최적화와 사용자 편의성 증대를 통해 더욱 완성도 높은 시스템을 구축할 수 있을 것으로 기대된다.

1. 서론

현대 사회에서 파일 공유는 필수적인 활동이지만, 다양한 운영체제 간의 호환성 문제는 여전히 해결되지 않은 과제이다. macOS 의 AirDrop, Android 의 QuickShare 등 각 운영체제는 별도의 프로그램 설치 없이 자체 파일 전송 기능을 제공한다. 해당 프로그램은 하드웨어 단위의 최적화를 통해 뛰어난 파일 전송 속도를 제공하지만[1], 이종 운영체제 간의 원활한 파일 교환은 제한적이다. 이를 해결하기 위해 Google Drive, KakaoTalk 등의 다양한 솔루션이 존재하지만, 저장 공간 제한과 프라이버시 문제와 같은 한계점이 있다. 본 연구는 이러한 문제를 해결하고자 브라우저를 기반으로 한 P2P 크로스플랫폼 파일 전송 프로그램인 Throw 를 개발하였다. P2P 파일 전송은 클라우드 스토리지 서비스 대비 빠르고 효율적인 파일 전송 방식이 될 수 있으며, 브라우저 기반 시스템은 운영체제 간 호환성 문제를 해결하고 사용자 편의성을 높일 수 있다. 또한, P2P 파일 전송 시스템의 성능은 네트워크 환경, 클라이언트 디바이스 사양 등 다양한 요인에 영향을 받기 때문에, 이러한 요인들을 고려하여 시스템을 설계하고 성능을 평가하였다.

2. Throw 시스템 설계 및 구현



[그림 1] Throw 시스템 구성도

[그림 1]은 Throw 는 서버 및 발신자 측과 수신자 측의 클라이언트 간 관계 구성을 보여준다. 발신자는 서버에서 발신자측 클라이언트 페이지를 다운로드 받고, QR 코드 또는 클립보드 복사를 통해 수신자에게 수신자측 클라이언트 페이지를 공유한다. 이 과정에서 자동적으로 WebRTC 연결에 필요한 ICECandidate 정보를 교환하게 되고, WebRTC 연결이 수립되어 DataChannel 이 생성된다.

DataChannel 을 통해 64KB 단위의 파일 조각들을

전송하게 되는데, 이 과정에서 보안이 고려된 SCTP over DTLS over UDP 프로토콜[2]이 사용된다. 발신자 측에서 전송된 조각을 수신자 측에서 받음과 동시에 브라우저의 IndexedDB 에 저장한다. 모든 파일이 조각이 전송되면 IndexedDB 에 저장된 조각들을 합쳐 완전한 파일을 만든다.

Throw 는 다음의 기술을 사용한다.

- WebRTC[3]: 클라이언트 간의 P2P 연결을 생성하고, 데이터를 전송하기 위해 사용한다. 연결 수립을 위해 클라이언트가 서로 정보를 교환해야 한다.
- Node.js: 클라이언트 페이지를 다운로드 받고, 발신자와 수신자 간의 WebRTC 연결을 수립하기 위해 필수적인 정보들을 교환한다.
- IndexedDB[4]: 브라우저에서의 유사 파일시스템으로 동작한다. 발신자 측에서 전송받는 파일 조각들을 전부 메모리에 들고 있을 수 없으므로, 이를 디스크에 써서 저장하는 역할을 수행한다.

3. 실험 환경

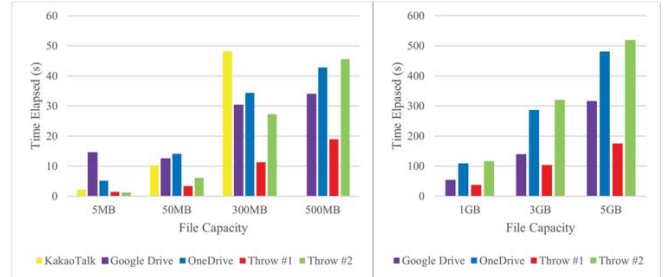
Throw 의 장거리(WAN) 네트워크 환경에서의 파일 교환을 실험한다. WAN 환경으로의 통신은 1Gbps 네트워크 망을 통해 이루어진다. 실험은 KakaoTalk, Google Drive 와 One Drive 에 대해, 파일이 스토리지 서비스에 업로드 되는 시간도 포함하여 다운로드 시간을 측정한다. 파일의 크기는 저장량인 5MB, 50MB, 300MB, 500MB 과, 대용량인 1GB, 3GB, 5GB 두 가지로 준비한다. ISP 네트워크의 영향을 확인하기 위해 Throw 는 수신자를 1Gbps 망(Throw #1)과 95Mbps 망(Throw #2)에 존재하는 경우를 상정해 각각 측정한다.

4. 실험 결과

[그림 2]는 파일 용량에 따른 주요 파일전송 서비스와 Throw 의 성능 비교 실험 결과를 보여준다. 저용량 파일에서는 Throw #1 이 가장 높은 성능을, KakaoTalk 이 가장 낮은 성능을 보였다. 이는 KakaoTalk 에서는 300MB 이상의 파일을 보낼 수 없기 때문이다. Google Drive 는 5MB 파일에 대해 바이러스 검사를 수행해 가장 늦은 다운로드 속도를 보여주었다. Throw #2 는 업로드와 다운로드가 동시에 이루어져 50MB 의 파일에 대해서는 두번째로 높은 성능을 보였다. 하지만, 500MB 파일을 다운로드 받을 때는 네트워크 대역폭 성능이 병목현상으로 작용하여 가장 낮은 성능을 보여주었다.

대용량 파일에서도 Throw #1 이 가장 높은 성능을, Throw #2 가 가장 낮은 성능을 보였다. 이는 Throw 를

비롯한 P2P 통신이 ISP 의 대역폭 제한에 대단히 큰 영향을 받는다는 것을 보여준다. 다만 Throw #2 는 대역폭 제한을 받는 상황에서도 업로드와 다운로드가 동시에 이루어지기 때문에, OneDrive 와 10% 정도의 성능차이만을 보인다.



[그림 2] 주요 파일전송 솔루션과 Throw 와의 성능 비교

Throw 는 WAN 환경에서 다른 스토리지 서비스보다 최대 2 배의 빠른 속도를 나타냈다. 추가로 Throw 는 LAN 환경에서도 작동할 수 있지만, Airdrop 과 Quickshare 보다는 최대 6 배까지 낮은 성능을 보인다.

5. 결론 및 의의

본 연구는 브라우저 기반 P2P 파일 전송 시스템인 Throw 를 개발하고, 기존 클라우드 스토리지 서비스와 성능을 비교하였다. 그 결과, Throw 는 WAN 환경에서 평균 2 배 더 빠른 전송 속도를 보였으며, 다양한 운영체제에서 사용 가능한 장점이 있음을 확인하였다. 하지만, 브라우저 샌드박스 환경이라는 오버헤드가 존재하며, NAT 문제로 인해 P2P 연결이 제약을 받을 수도 있다는 한계도 존재한다.

Acknowledgment

본 연구는 2024 년 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업 지원을 받아 수행되었음(2023-0-00044)

참고문헌

- [1] TUEXEN, M., et al. *Datagram Transport Layer Security (DTLS) Encapsulation of SCTP Packets*. 2017.
- [2] HEINRICH, Alexander, et al. {PrivateDrop}: Practical {Privacy-Preserving} Authentication for Apple {AirDrop}. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021. p. 3577-3594.
- [3] SUCIU, George, et al. WebRTC role in real-time communication and video conferencing. In: *2020 Global Internet of Things Summit (GIoTS)*. IEEE, 2020. p. 1-6.
- [4] MOZILLA DEVELOPER NETWORK. Using IndexedDB - Web APIs. MDN Web Docs [online]. [no date] [viewed 13 September 2024]. Available from: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB