

웹 기반의 데이터 전송 방법에 관한 연구동향

김 장 원*

1. 서 론

웹 2.0 의 시대에 이르러 참여, 공유, 개방의 모토 아래 다양하고 수많은 정보들이 인터넷 사용자들에 의해 생겨나게 되었다. 사용자들이 생각하고 있는 아이디어를 보다 편리하게 유통시킬 수 있는 환경이 마련되었기 때문에 다양한 데이터와 정보들이 생산되었으며, RIA와 같은 고급 웹 기술 등의 발전은 사용자의 경험을 중시하는 UX 환경을 풍부하게 해주었으며, 사람과 컴퓨터와의 정보교환 및 공유를 위한 휴먼 컴퓨터 인터랙션(HCI) 분야를 확장시키게 되었다. 이렇게 사용자들 간의 인터페이스가 중요한 시대 흐름 속에서 가장 중심에 HTML 이 기존의 단순한 웹 페이지가 아닌 커뮤니케이션을 위한 수단 혹은 플랫폼으로 자리매김하고 있다. HTML 은 최초 웹 에서 특정 텍스트와 자원에 대한 마크업을 위해 만들어진 언어에 불과하였으나, XML과 자바스크립트, CSS 등과 같은 언어들과 접목하여 사용자에게 효율적이며 편의성 높은 인터페이스를 제공해 줄 수 있는 플랫폼으로 진화하고 있다. 그 덕분에 소셜네트워크, 블로그 등이 다양한 인터페이스를 제공하고 편리한 방법으로 사용자들의 의견을 소통할 수

있게 되었다. 또한, 최근 등장한 Web Of Things (WOT) 의 경우 사물지능통신(M2M)의 한 가지 방법으로 인터넷 프로토콜을 기반으로 사물들 간의 정보를 교환하기 위한 연구 분야이다. WOT에서 제안하는 개념을 통해 가전기기 등이나 안경, 시계와 같은 휴대용 기기, 또는 모션 센서로부터 생성되는 데이터들을 교환하고 저장 및 관리를 할 수 있게 되었다. 이러한 환경 속에서 웹은 진화를 계속하여 HTML 은 HTML 5 버전까지 진화하게 되었으며, Google 과 Facebook 등 많은 회사들이 웹을 기반으로 자사의 서비스를 다양한 형태로 제공하기 위한 연구를 진행하고 있다. 그 결과 모질라 단체 에서는 웹 기반의 웹 OS를 개발하고, Google 의 크롬 브라우저를 통해 기존단일의 프로그램에서만 실행하였던 게임들과 많은 어플리케이션들을 웹 브라우저 안에서 사용할 수 있게 되어, 액티브 X 프로그램을 설치하지 않고서도 실시간으로 사용자들이 대화를 하고 메시지를 전송하고, 화상으로 채팅할 수 있게 되었다. 과거 Peer to Peer (P2P) 간의 통신 방법이 WOT에서는 웹 기반으로 가능하게 되었다. 본 논문에서는 웹을 기반하여 데이터를 전송하기 위해 추가된 기술들과 특징들을 살펴보고 사용자들 간의 통신 혹은 사물들간의 통신을 가능하게 할 수 있는 웹 기반 데이터 전송 방법 등의 연구 동향을 살펴본다.

* 교신저자(Corresponding Author) : 김장원, 주소 : 서울특별시 성북구 안암동, 고려대학교 정보통신대학 509호, 전화: 02) 925-3706, FAX: 02)921-9137, E-mail : ikaros1223@gmail.com
* 고려대학교 컴퓨터·전파통신공학부

2. 데이터 전송 기술 동향

웹을 기반으로 데이터를 전송하는 방법은 HTTP 규약을 기반으로 시작되었다. 웹상에서 정보를 주고받기 위해 TCP와 UDP 통신 규약을 사용하며 어플리케이션 레벨과 연결을 하기 위해 80포트를 사용한다[1]. HTTP는 클라이언트와 서버 사이에 요청과 응답방식으로 데이터를 주고 받는 방식이다. 그러므로 사용자가 웹 브라우저를 통해 그림이나 링크를 클릭 할 경우 웹 브라우저가 HTTP를 통하여 서버로 요청을 보내며, 서버는 해당 요청해 해당하는 응답을 HTTP 규약을 통해 사용자에게 전달한다. HTTP 1.0에서는 이러한 요청과 응답을 한 번에 주고받는 방식이어서 사용자가 웹 브라우저를 클릭할 경우 서버로부터 해당 페이지에 대한 모든 내용을 전송받는 시점까지 기다려야 했다. 이를 위해 브라우저와 서버간의 연결설정은 사용자의 요청과 그에 응답 횟수에 따라 증가되었다. HTTP 1.1로 진화하면서 한 번의 연결/해제 설정을 통해 연결설정에 대한 오버헤드는 줄어들

었으나, 여전히 한 개의 페이지에 대한 한 개의 요청/응답 방식을 따르고 있었다[2]. 그 결과 사용자들의 입력에 대한 서버의 연산이 빨라져도 사용자에게 실제 보여지는 화면은 느려지게 되는 문제점이 발생하였다. 그 결과 데이터의 전송에 있어 새로 고침(refreshing)이 없이 사용자의 웹 페이지의 데이터를 고칠 수 있는 기술이 등장하게 되었다. 이를 비동기적 자바스크립트와 XML, 즉 AJAX(Asynchronous Javascript and XML)라고 명명하였다[3,4]. AJAX를 통해 사용자들이 검색 엔진을 통해 검색을 하고자 할 경우, 페이지의 새로 고침 없이 서버에서 추천해 주는 검색어를 선택해서 넣을 수 있게 되었다. 그 결과동적으로 변화는 서비스들이 기하급수 적으로 증가하게 되었으며 사용자의 경험(UX)이 더욱 중요하게 되었다. 그림 1은 데이터 전송 방식의 변화를 표현하고 있는 그림으로 각 기술의 특징을 장단점을 보여준다[5-7]. AJAX 기술의 등장으로 많은 서비스들이 생성되고 사용자들의 요구사항이 증가되었으나 웹 브라우저에서 서버로 가는 한 방향통신으로

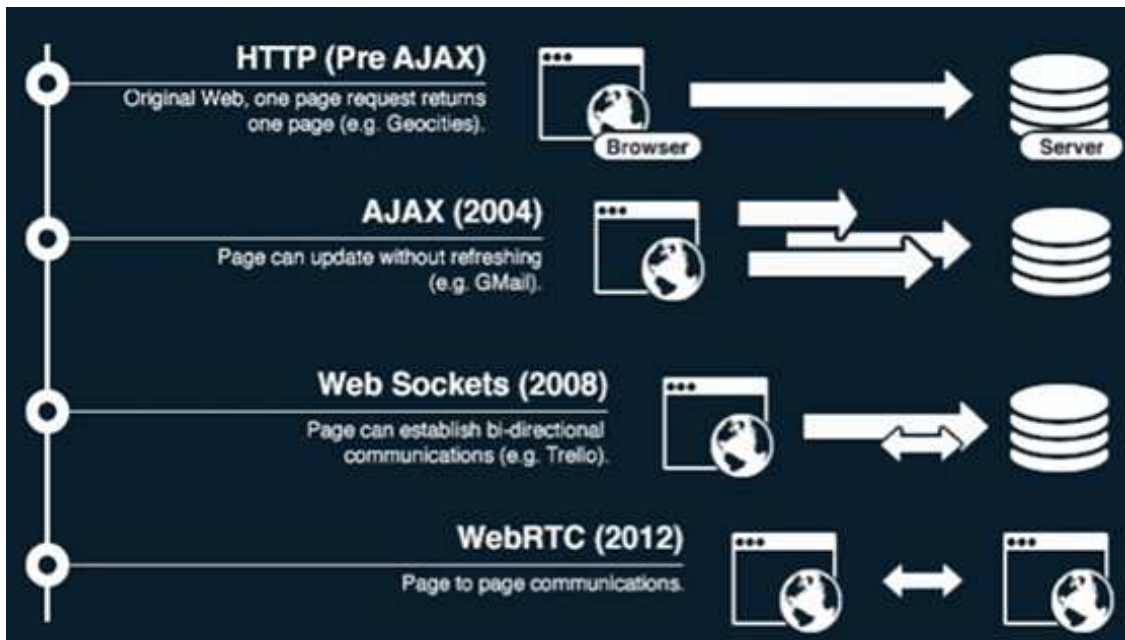


그림 1. 데이터 전송 방식의 변화

는 이러한 요구사항들을 수용할 수 없게 되었다. 이때 웹을 기반으로 양방향 통신을 할 수 있는 웹 소켓이 등장 하였다[8,9]. 웹 소켓은 AJAX와 달리 하나의 여러 개의 양방향 통신 채널을 만들 수 있다. 웹 소켓 프로토콜은 2011년 IETF를 통해 RFC 6455 로 채택되었다. 그리고 웹 소켓 API들에 대한 표준화가 W3C에 의해 제안되어 HTML 5 등의 웹 표준들과 융합되어 많은 기능들을 제공하는데 기여하고 있다[10]. 웹 소켓은 기존 HTTP에서 방식과 달리 웹 브라우저와 서버간의 연결 설정을 할 때 TCP 기반의 80 포트를 이용하지만 어플리케이션 레벨에서 HTTP로 바인딩을 수행하여 웹 브라우저와 서버간의 연결을 시도한다. 웹 소켓 프로토콜을 통해 보다 많은 사용자의 요구사항을 웹 브라우저로부터 받아 웹 서버로 보내고 받을 수 있다. 그러므로 실시간으로 사용자의 입력을 받아 서버로 전송해야 가능하였던 실시간 게임과 화상 회의 등이 특정 프로그램의 설치 없이 웹 브라우저만을 통해 가능하게 되었다. 또한 W3C에서 제안하고 있는 HTML 5을 통해 웹 브라우저로 표현할 수 있는 기술이 다양하게 정의되고 확장됨에 따라 웹 소켓의 영향력이 크게 증가하고 있다. 현재 웹 소켓은 크롬, 인터넷 익스플로러, 파이어폭스, 사파리, 오페라 등에서 사용이 가능하다. 하지만, 웹 소켓을 통한 웹 브라우저와 웹 서버간의 통신은 진정한 의미에 있어서 Peer to Peer (P2P) 통신을 하지 못한다. 현재 사물지능통신(Machine to Machine, M2M) 과 인터넷 사물통신(Internet of Things)과 같은 사물간의 대화에 초점이 맞춰지고 있다[11, 12]. 이러한 프로토콜을 통해 사물간의 통신이 이루어지고 세상에 존재하는 사물들이 서로 연결되어 새로운 융합된 정보를 가져오게 되는 환경을 만들기 위한 다양한 연구가 활발히 진행되고 있다. 이러한 기술들의 상호 교환을 위한 한 가지 방법으로 웹에서 사용

하고 있는 HTTP 프로토콜을 이용한다. 이를 Web of Things(WOT)라고 명명하며, HTTP 프로토콜에서 사용되는 4가지 함수인 POST, GET, PUT/PATCH, DELETE를 사용하여 데이터들의 생산, 읽기, 업데이트, 삭제(CRUD (create, read, update, delete) 기능을 수행한다[13]. 이를 위해 사물들 간에는 양 방향 통신이 실시간으로 이루어져야 한다. 그렇지만 웹 소켓은 기존 방식과 마찬가지로 웹 브라우저와 서버간의 연결 형태를 가지고 있기 때문에 제약이 따르게 된다. 그러므로 양방향 대화가 가능하며 사용자의 2차 3차 요구사항을 실시간으로 반영하기 위한 프로토콜이 필요하게 되었다. 그 결과 서로 다른 웹 페이지가 데이터의 교환을 가능하게 하는 WebRTC 개념이 등장하게 되었다. WebRTC (Web Real-Time Communication)는 웹 기반 실시간 대화를 가능하게 하는 프로토콜로써 W3C 에 의해 확장되고 있다. W3C 는 이를 위한 작업반(Working Group)을 만들었으며 구글, 페이스북 등 다수의 기업들이 해당 기술의 표준화 작업에 참여하고 있다[14].

3. WebRTC 기술 동향

3.1 WebRTC 구조

WebRTC는 웹 브라우저 상에서 실시간 대화를 위해 구글에서 최초로 오픈 소스 기반의 프로젝트로 만들었던 것으로 현재는 W3C에서 표준화 작업으로 채택되어 개발되고 있다. 현재까지는 크롬과 파이어폭스 브라우저에서만 지원하고 있으나 W3C의 권고안으로 완성되면 보다 많은 브라우저에서 사용가능할 것으로 판단된다. 그림 2는 WebRTC의 구조와 웹 브라우저 간의 메시지 전달 방법을 나타낸다. W3C에서 HTML 5에서 만들고 있는 Web API를 기반으로 하여 WebRTC

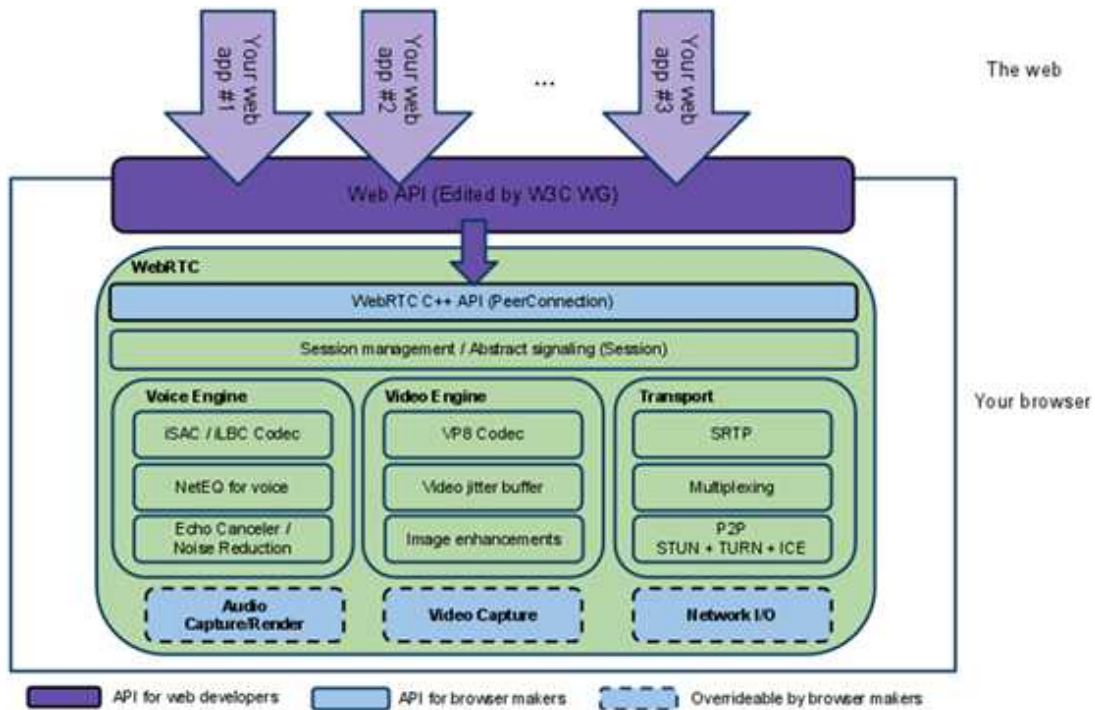


그림 2. WebRTC 구조

프레임워크 간 데이터의 전송이 이루어진다. WebRTC 프레임워크가 사용자의 브라우저마다 포함되어 있게 됨으로써 웹 브라우저에서 클라이언트 뿐만 아니라 서버의 기능을 수행할 수 있다. WebRTC 기반의 다자간 통신을 위해 WebRTC C++ API가 제공되며 오디오와 영상 메시지를 전송하기 위한 엔진들과 자바스크립트 API를 기반으로 데이터를 접근할 수 있는 API가 존재한다. 그 결과 기존까지 C언어를 기반으로 만들어졌던 어플리케이션과의 연동도 가능할 뿐만 아니라, 웹 브라우저 이외의 액티브 엑스 프로그램을 통한 플러그인 어플리케이션들을 추가 설치할 필요가 없다. 또한 WebRTC는 여러 개의 웹 앱을 동시에 지원하기 때문에 앱이 설치되어 있는 단말기와 브라우저에서 앱의 출력 결과들의 데이터의 전송을 지원한다. 그러므로 해당 WebRTC 표준안이 완성되면 Web of Things의 실현이 보다 앞당겨질 것이다.

3.2 WebRTC 기술 요소

WebRTC를 통한 실시간 데이터 전송을 위해 webRTC.io가 개발되었으며 데이터의 입/출력을 위해 크게 3가지 형태로 기술이 개발되고 있다. 첫 번째는 MediaStreams으로 웹 브라우저를 통해 사용자들이 가지고 있는 카메라와 마이크에 접근할 수 있는 API이다. 두 번째는 PeerConnection으로 음성 및 영상에 대한 연결설정을 위한 기술, 마지막으로 DataChannel이 있다. DataChannel을 통해 P2P 형식의 데이터 전송이 가능해진다. 데이터 전송 방식 측면에서 웹 소켓과 WebRTC의 차이점은 웹 소켓은 서버와의 이벤트를 공유함에 비해 WebRTC는 웹 서버로부터 세션에 대한 정보를 획득하고 이를 공유한다. 또한 Peer로부터 음성과 영상 데이터를 공유할 수 있다. 그렇기 때문에 세션에 대한 권한 및 보안 설정이 이슈화 되어 연구되고 있다[15]. 현재 webRTC.io는 자바스크립트 모듈로 개발되어 있

으며 Node.js 프레임워크를 통해서도 확장이 가능하기 때문에 특정 브라우저에서만 작동된다는 한계를 대신하기 위하여 웹 소켓 등과 혼합하여 개발할 수 있다[16,17]. 현재까지 개발된 데이터 전송 방법과 이를 반영하는 웹 브라우저는 다음 표 1과 같다.

그림 3은 webRTC.io를 기반으로 크롬, 파이어폭스에서 실행된 데모 영상으로 좌측의 상, 하 영상은 서로 다른 장소에서 서로 다른 브라우저를 기반으로 실시간 대화를 수행한 모습이며, 우측의 영상은 비디오 영상이 제공되는 순간 상대방의 모습을 캡처하는 화면이다. 이를 통해 실시간 영상을 감상할 수 있을 뿐만 아니라, 캡처도 가능하며, 실시간 화상채팅의 중간에 사용자가 그림을 그린 그림을 같이 띄워 서로 공유할 수 있다.

현재까지 개발된 대표적인 서비스로는 구글에

표 1.

	HTML 5	웹 소켓	WebRTC
크롬	○	○	○
파이어폭스	○	○	○
사파리	×	○	×
오페라	○	×	×
익스플로러	○	×	×



그림 3. WebRTC 데모

서 만든 행아웃 서비스 이다[18-20]. 이를 통해서 구글에서 제공되는 문서파일들을 동시에 공유하고 편집할 수 있으며 다자간 화상회의를 진행할 수 있다. 이러한 WebRTC 기술이 서로 다른 웹 브라우저에서 동작된다면 텍스트 기반의 소셜 네트워크 서비스 중심에서 멀티미디어 기반의 소셜 네트워크로의 이동이 확산될 것이다.

4. 결 론

본 논문에서는 웹 기반 데이터 전송 방식을 위해 HTTP 프로토콜에서부터 WebRTC 기술에 이르는 관련 동향을 살펴보고 해당 기술을 통해 얻을 수 있는 장점과 단점 등을 살펴보았다. 인터넷이 발달하고 사물간의 네트워크가 활성화 되는 사물 지능형 통신과 웹 기반의 사물 간 협력이 이뤄지는 미래 인터넷 환경에서 핵심적인 기술로서 자리매김할 WebRTC를 살펴보았으며, 이를 통해 가능한 새로운 시도와 연구 분야에 대해 기술하였다. 향후, W3C 등에 의해 해당 기술이 확장되고 많은 브라우저에 탑재가 된다면 언제 어디서든 웹 이 가능한 환경에서 데이터를 전송받고 전달할 수 있는 시대가 도래 할 것이라 기대되며 이와 관련한 연구가 활발히 진행될 것으로 기대된다.

참 고 문 헌

[1] T Berners-Lee, R Fielding, H Frystyk, HTTP <http://www.w3.org/Protocols/HTTP/1.0/>, 1996.

[2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, HTTP 1.1, <http://www.w3.org/Protocols/rfc2616/>, 1999.

[3] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, & F. Yergeau, "Extensible markup language (XML)," World Wide Web Journal, Vol.

2(4), pp. 27-66. 1997.

[4] J. J. Garrett, "Ajax: A new approach to web applications," 2005.

[5] WebRTC Demo, <http://generative.edb.utexas.edu/webrtc-demos/webrtc.ppt>

[6] W3C, WebRTC, <http://dev.w3.org/2011/webrtc/editor/webrtc.html>

[7] WebRTC, <http://www.webrtc.org/>

[8] I. Fette, "The WebSocket protocol, 2010."

[9] I. Hickson, "The websocket api. W3C Working Draft WD-websockets-20110929," 2011.

[10] M Pilgrim, "HTML5: Up and running," 2010.

[11] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S. Xie, & S. Gjessing, "Cognitive machine-to-machine communications: visions and potentials for the smart grid. Network," IEEE, vol. 26(3), pp. 6-13, 2012.

[12] D. Miorandi, S. Sicari, F. D. Pellegrini, & I. Chlamtac, "Internet of Things: Vision, Applications & Research Challenges. Ad Hoc Networks," 2012.

[13] S. S. Mathew, Y. Atif, Q. Z., Sheng & Z. Maamar, "The Web of Things-Challenges and Enabling Technologies," In Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence, pp. 1-23. Springer Berlin Heidelberg, 2013.

[14] W3C Working Group for WebRTC, <http://www.whatwg.org/specs/web-apps/current-work/multipage/>

[15] H. Alvestrand, "Cross Session Stream Identification in the Session Description Protocol," 2013.

[16] S. Tilkov, & S. Vinoski, "Node.js: Using JavaScript to build high-performance network programs," Internet Computing, IEEE, vol. 14(6), pp. 80-83, 2010.

[17] M. Iqbal, M. Husni, & H. Studiawan, "Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan Node.js. Jurnal Teknik ITS," Vol. 1(1), pp. A242-A245, 2012.

[18] K. Kahle, "Hangout with CERN: Lights, webcam, action!," BUL-NA-2013-034, p. 6, 2013.

[19] Google Hangout, <https://plus.google.com/hangouts>, 2013.

[20] C. Roseth, M. Akcaoglu, & A. Zellner, "Blending Synchronous Face-to-face and Computer-Supported Cooperative Learning in a Hybrid Doctoral Seminar," TechTrends, vol. 57(3), pp. 54-59, 2013.



김 장 원

- 2008년 고려대학교, 컴퓨터학과, 이학석사
- 2012년 고려대학교, 컴퓨터전파통신공학과, 공학박사
- 현재 고려대학교 컴퓨터전파통신공학과
- 관심분야: HTML 5, Web of Things, M2M, Semantic Web