The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 17, No. 6, pp.113-119, Dec. 31, 2017. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2017.17.6.113
JIIBC 2017-6-15

WebRSF: 1대 1 대화 서비스 제공을 위한 웹 기반 리치 커뮤니케이션 서비스 소프트웨어 프레임워크

WebRSF: A Web-based Rich Communication Service Software Framework for Providing the 1-to-1 Chat Service

이동철*

Dongcheul Lee*

요 약 차세대 모바일 메시징 서비스인 RCS(Rich Communication Service)는 통신사업자가 독점적으로 제공하는 서비스로 제3의 개발자가 해당 기능을 이용한 앱을 개발하기 어렵다. 이를 해결하기 위하여 몇몇 방법이 제시되어왔으나 실제로 응용 가능한 수준에는 못 미치는 실정이다. 본 논문은 제3의 개발자가 RCS 서비스를 이용한 앱을 쉽게 개발할 수 있도록 웹 인터페이스 기반의 RCS 소프트웨어 프레임워크인 WebRSF를 제안한다. WebRSF는 1대 1 대화 기능을 제공하기 위해 복잡한 RCS 프로토콜을 간단한 웹 기반 프로토콜 변환하여 RCS 기능을 제공한다. 이를 위해 WebRSF를 이용하는 클라이언트가 WebRSF와 메시지를 어떻게 주고받는지 정의하였고, WebRSF와 기존 RCS 노드들과의 네트워크 구성도 및 메시지 흐름도를 명시하였다. 또한 성능 평가를 통해 WebRSF 클라이언트를 사용할 경우 RCS 클라이언트를 사용하는 것에 비하여 디바이스가 얼마나 부하를 더 받는지 성능 평가를 통해 실험해 보았으며 세션 수립 시에는 17% 감소된, 세션 수립 후에는 25% 증가된 부하를 받는다는 것을 알 수 있었다.

Abstract The Rich Communication Service(RCS) is a next generation mobile messaging service. Since it has been developed and managed by a wireless service provider, 3rd party application developers cannot utilize the RCS features in their apps. A few studies have been proposed to solve this problem, even though they were not viable level of deployment. This paper presents a Web-based Rich Communication Service Software Framework(WebRSF) to help the 3rd party application developers adopt RCS features easily. WebRSF converts complicated RCS protocols to simple web-based protocols for providing the RCS 1-to-1 chat service to the 3rd parties. Communication protocols between WebRSF and its clients are defined in this paper. Also, a network configuration diagram and a message flow diagram are defined. Finally, performance evaluation between the WebRSF client and the RCS client are performed to simulate the load on clients' devices. It shows that the WebRSF client has 17% less loads than the RCS client while initiating sessions, and has 25% more loads after initiating sessions.

Key Words : RCS, WebRSF, Software framework

Ⅰ. 서 론

현대 사회에 접어 들면서 사람들 간에 서로 메시지를 주고받는 형식도 다양해 졌다. 편지, 전보, 팩스(FAX), 이메일(E-mail), SMS(Short Message Service), MMS(Mutimedia Message Service), 카카오톡(Kakao

*종신회원, 한남대학교 멀티미디어학과
접수일자: 2017년 10월 27일, 수정완료: 2017년 11월 27일
게재확정일자: 2017년 12월 8일

Received: 27 October, 2017 / Revised: 27 November, 2017 / Accepted: 8 December, 2017
*Corresponding Author: jackdclee@gmail.com
Department of Multimedia, Hannam University, Korea

Talk)이나 라인(Line)과 같은 OTT앱 등이 그것이다[1]. 특히 SMS나 MMS는 과거 스마트폰이 나오기 전 통신 수단으로 많이 사용되었으나 더 풍부한 기능을 가진 무료 OTT(Over-The-Top) 앱들이 대거 등장하면서 그 수요가 많이 줄어든 실정이다. 통신사들은 과거 통신 수단이었던 SMS와 MMS의 차세대 메시징 서비스인 RCS(Rich Communication Service)를 개발하여 OTT 서비스와 경쟁하고 있다.

OTT 서비스는 기능이 다양하고 대부분 무료라는 장점을 가지고 있으나 메시지를 주고받을 사람들끼리 반드시 똑같은 앱이 설치되어있거나 같은 서비스에 가입되어 있어야 한다는 한계를 가진다. RCS는 이러한 OTT와는 달리 전 세계 통신사들이 공통 규격을 사용하여 만들었고 스마트폰에 임베디드(Embedded)되어 출시되기 때문에 SMS처럼 전 세계 사용자들이 별도의 앱 설치나 서비스 가입 없이 자유롭게 메시지를 주고받는 것을 목표로 한다. 통신사들의 연합 기구인 GSMA(Groupe Speciale Mobile Association)가 중심이 되어 RCS 규격을 정하고 발전시키고 있다[2].

RCS의 사용을 확대하기 위해서는 RCS앱이 스마트폰에 임베디드되는 것뿐만 아니라 다른 앱에 RCS 기능이 포함되거나 웹(Web)이나 데스크탑(Desktop)에서도 이용 가능 하도록 해야 한다. 이를 위해서는 RCS 기능을 웹을 통해 공개하여 제3의 개발자가 이를 이용하여 다양하고 새로운 서비스를 개발할 수 있도록 해야 한다. 웹 기반으로 이러한 기능을 제공하게 되면 제3의 개발자들은 RCS를 개발하는데 필요한 SIP(Session Initiation Protocol), MSRP(Message Session Relay Protocol), XCAP(Extensible Markup Language Configuration Access Protocol)과 같은 어렵고 복잡한 프로토콜 대신 단순하고 간단한 웹 기반의 프로토콜을 사용하기 때문에 쉽고 빠르게 원하는 앱을 개발할 수 있다[3]. 본 논문은 RCS 서비스를 웹으로 제공하는 소프트웨어 프레임워크인 WebRSF(Web-based Rich Communication Service Software Framework)를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 RCS를 이용하여 1대 1 대화를 주고받을 때 사용되는 표준 RCS 규격에 대하여 알아보고 이를 제3의 개발자에게 공개하기 위한 관련 연구에 대하여 알아본다. 3장에서는 본 논문에서 제안하는 1대 1 대화를 제공하는 웹 기반 RCS 소프트웨어 프레임워크를 구현하기 위한 네트워크 구성 방법과 기존 RCS 노드와 WebRSF간 메시지를 주고받는 흐름에 대하여 알아본다. 4장에서는 표준 규격과 WebRSF를 사용했을 때를 비교한 성능 평가를 하고 5장에서 결론을 맺는다.

II. 관련 연구

GSMA는 2007년부터 RCS 규격을 발전시켜 왔으며 현재 가장 최신 규격은 버전 7.0이다[4]. RCS는 통신사의 IMS(IP Multimedia Subsystems) 인프라를 이용하여 동작하며 1대 1 대화를 위해 SIP을 사용하여 양 사용자간 MSRP 세션(Session)을 수립하고, 세션이 수립된 이후로는 MSRP를 사용하여 메시지나 각종 미디어를 주고받는다. 그림 1은 RCS 메시지를 발신할 사용자 UE1(User Equipment 1)이 다른 사용자 UE2에게 최초 대화 요청을 보낼 때 네트워크 노드들 간의 세션 수립을 위한 절차를 나타낸 것이며 그 세부 내용은 다음과 같다.

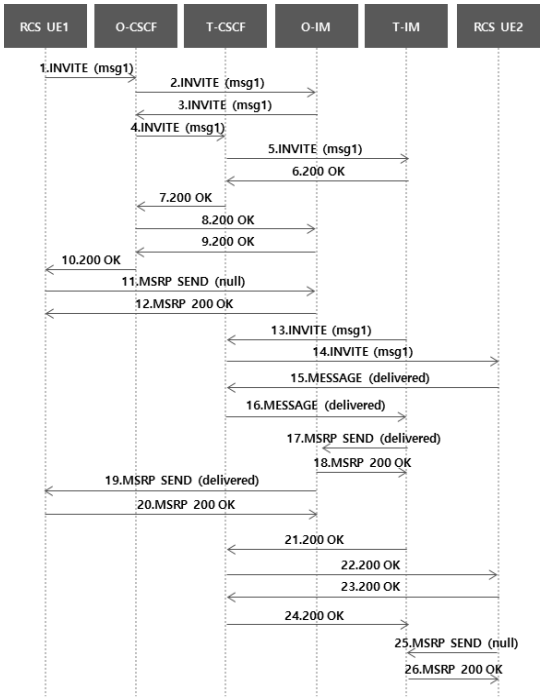

그림 1. 1대 1 대화 세션 수립을 위한 절차를 나타낸 시퀀스 다이어그램

Fig. 1. A sequence diagram showing procedures for the setup of 1-to-1 chat sessions

(1) RCS UE1이 사용자로부터 대화 내용을 입력 받으면 SIP INVITE에 대화 내용을 포함하여 이전에 등록되었던 O-CSCF(Originating-Call Session Control Function)로 전송한다.

(2) O-CSCF는 iFC(initial Filter Criteria)룰에 따라 SIP INVITE를 O-IM(Originating-Instance Messaging)으로 보낸다.

(3-4) O-IM은 INVITE를 RCS UE2가 등록된 T-CSCF(Terminating-Call Session Control Function)로 보내기 위하여 O-CSCF로 INVITE를 전달하고, O-CSCF는 라우팅(Routing) 규칙에 의해 INVITE를 T-CSCF로 전달한다.

(5) T-CSCF는 전달 받은 INVITE를 iFC에 따라 T-IM(Terminating-Instance Messaging)으로 전송한다.

(6-10) T-IM은 INVITE 내 발신자가 작성한 대화 내용을 DB(DataBase)에 저장하고, UE1에게 대화 내용을 잘 받았다고 알려주기 위해 SIP 200 OK를 INVITE가 온 경로로 전송한다.

(11-12) UE1은 O-IM과 직접 MSRP 세션을 맺기 위해 빈 내용의(null) MSRP SEND를 O-IM으로 보낸다. O-IM은 소켓(Socket) 정보를 DB에 저장한 후 MSRP 200 OK를 UE1에게 보낸다.

(13-14) T-IM은 UE2로 대화 내용을 전달하기 위해 대화 내용이 포함된 INVITE를 UE2가 등록된 T-CSCF로 보내고 T-CSCF는 이를 UE2로 전달한다.

(15-16) UE2는 INVITE 내 발신자가 보낸 대화 내용을 사용자 화면에 보여주고, 대화 내용을 잘 받았음을 T-IM에게 알려주기 위해 SIP MESSAGE에 CPIM/IMDN(Common Presence and Instant Messaging/Instant Message Disposition Notification) 형식으로 delivered 알림 정보를 담아 전송한다[5][6].

(17-20) T-IM은 받은 IMDN 정보를 MSRP SEND로 O-IM을 통해 UE1으로 전송하고 MSRP 200 OK를 받는다.

(21-22) T-IM은 (15-16)에 대한 SIP 200 OK를 전송한다.

(23-24) UE2는 (13-14)에 대한 SIP 200 OK를 전송한다.

(25-26) UE2은 T-IM과 직접 MSRP 세션을 맺기 위해 (11-12)와 같이 UE2의 소켓 정보를 저장하기 위해 내용이 없는 MSRP SEND를 전송하고 MSRP 200 OK를 받는다.

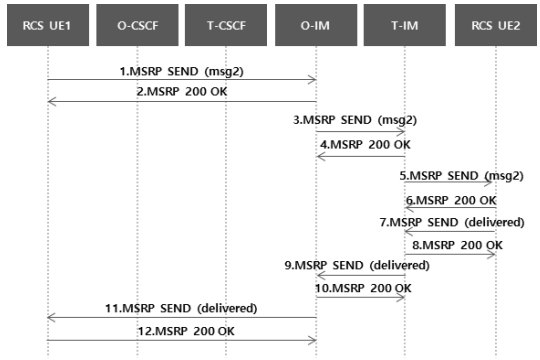

그림 2. MSRP 세션을 이용하여 대화를 주고받기 위한 절차를 나타낸 시퀀스 다이어그램

Fig. 2. A sequence diagram showing procedures for exchanging chat messages using the MSRP session

그림 2은 UE1이 UE2에게 두 번째 또는 그 이후의 대화 요청을 보낼 때 MSRP 세션을 이용한 네트워크 노드들 간의 메시지 교환 절차를 나타낸 것이다. MSRP 세션 수립 후 UE1이 두 번째 대화 요청을 보내면 O-IM, T-IM을 거쳐 UE2로 MSRP SEND를 보낸다. 이때 UE2로부터 CPIM/IMDN 형식으로 delivered 알림을 전송 받도록 요청한다. 따라서 UE2는 성공적으로 대화 내용을 사용자에게 보여 준 후, MSRP SEND에 delivered 알림 정보를 넣어 T-IM과 O-IM을 거쳐 UE1에게 전송한다.

GSMA에서는 이와 같이 RCS 표준 규격을 만들어 SMS를 대체할 수 있는 새로운 서비스를 만들었지만 이는 스마트폰용 앱을 사용해야만 하는 구조로 되어있어 RCS 사용을 확대시키기에 한계가 있다. 표준 규격에는 RCS 기능을 외부에 개방하기 위해 부가적으로 RCS API(Application Programming Interface)에 대한 언급이 있으나 개략적인 시스템 구조와 기능요구사항만 있을 뿐 API를 구현하기 위한 세부 방법은 다루고 있지 않아 이에 대한 방안을 마련하는 것이 필요한 실정이다[7]. [8]은 RCS 기능을 웹 기반으로 제공하기 위한 소프트웨어 프레임워크를 제안하였으나 대화 발신용 기능만 제공되어 대화를 수신하기 위한 기능이 별도로 필요한 상황이다.

Ⅲ. 제안하는 방법

본 논문은 RCS 1대 1 대화 서비스를 제3의 개발자가 쉽게 포함하여 개발할 수 있도록 제공하기 위한 소프트

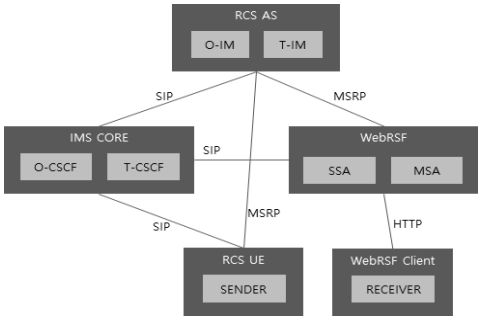

그림 3. RCS 구성을 위한 기존 노드와 WebRSF 간의 네트워크 구성도

Fig. 3. A network configuration diagram for existing RCS nodes and the WebRSF

웨어 프레임워크를 제안한다. RCS 서비스를 외부에 공개하기 위해서는 웹 인터페이스를 이용한 REST (Representational State Transfer)와 HTTP Streaming을 사용하기로 한다[9]. REST는 정보를 자원 형식으로 표현하고 URL(Uniform Resource Locator)을 통해 해당 자원을 접근하도록 하는 기술로 가볍고 확장성이 좋아 간단한 HTTP(Hypertext Transfer Protocol) 요청만으로도 서비스를 사용할 수 있도록 해 준다[10]. HTTP Streaming은 기존 HTTP 방식과는 달리 브라우저가 웹 서버에 특정 요청을 하지 않더라도 웹 서버가 브라우저로 데이터를 보낼 수 있게 해주는 웹 어플리케이션 모델이다. 이를 이용하면 chunked encoding을 사용하여 연결을 유지한 채로 다수의 이벤트를 전달할 수 있으며, 이를 통해 시스템 및 네트워크 부하 및 리소스 사용을 줄일 수 있는 장점이 있다[11]. 이를 RCS 대화 내용을 WebRSF 클라이언트로 전달하는데 사용하면 착신자가 웹 인터페이스를 이용하더라도 RCS 대화 내용을 실시간으로 전달할 수 있게 된다.

그림 3은 기존 RCS를 제공하기 위해 필요한 노드들과 WebRSF와의 네트워크 구성도를 나타낸 그림이다. WebRSF는 기존 RCS 노드들의 구성을 그대로 유지한 채 IMS 코어 노드들과 SIP 연결을 맺고, RCS AS(Application Server) 노드들과 MSRP 연결을 맺는다. WebRSF를 시용하는 클라이언트들은 RCS AS나 IMS 코어 노드와 직접 연결 없이 오직 WebRSF를 통해서만 서비스를 제공 받으면 된다. 즉, WebRSF는 기존에 RCS UE가 담당했던 SIP 및 MSRP 프로토콜을 이용한 RCS 서비스 요청을 대신해주고 이를 웹 인터페이스를 통해 다른 클라이언트에게 제공해주는 역할을 한다.

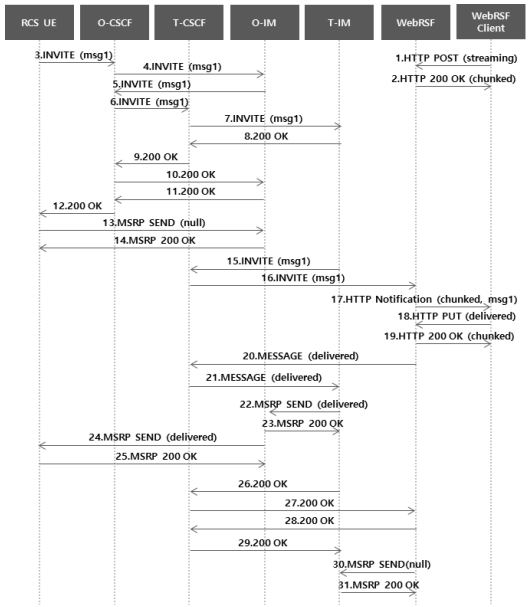

그림 4. WebRSF를 사용하여 1대 1 대화 세션 수립을 위한 절차를 나타낸 시퀀스 다이어그램

Fig. 4. A sequence diagram showing procedures for the setup of 1-to-1 chat sessions using WebRSF

그림 4는 위와 같은 네트워크 구성도에서 RCS UE 사용자가 WebRSF 클라이언트를 사용하는 사용자에게 최초 대화를 요청할 경우 각 노드들 간에 주고받는 메시지의 흐름을 나타내며 그 세부 흐름은 다음과 같다.

(1) WebRSF 클라이언트는 HTTP Streaming을 사용하여 착신 대화를 받기 위해 WebRSF와 스트리밍 채널을 생성을 요청한다. 이때 REST 방식으로 해당 리소스를 요청하게 되며 그 예는 다음과 같다.

HTTP: POST /{url}Notification/Streaming

(2) WebRSF는 클라이언트와 스트리밍 채널을 맺고 향후 발생하는 모든 이벤트 정보를 해당 채널을 이용하여 클라이언트에 전달한다. 여기서는 스트리밍 채널 요청을 잘 처리했다는 결과를 클라이언트에 보내기 위해 HTTP 200 OK 응답의 Transfer-Encoding 헤더 값을 chunked로 설정하여 전송한다.

(3-14) 그림 1의 (1-12) 흐름과 동일하다.

(15-16) WebRSF는 RCS 클라이언트처럼 T-CSCF에 등록되어 있기 때문에 T-IM은 해당 T-CSCF로 INVITE를 보내고, T-CSCF는 WebRSF에게 해당 메시지를 전달한다.

(17) WebRSF는 INVITE 내 From, To, imdn. Message-ID, Subject 그리고 Reqeust-URI 헤더를 읽어 들여 이를 기반으로 하여 chunked transfer encoding된 데이터를 만들고 스트리밍 채널을 이용해 클라이언트로 해당 데이터를 전송한다[12]. 표 1은 "01012341234"의 번호를 가진 RCS UE 사용자가 "Hello?"라는 메시지를 WebRSF 클라이언트 사용자에게 보낼 경우 스트리밍 채널로 전송되는 chunked 메시지 예이다.

(18-19) WebRSF 클라이언트는 대화 내용을 잘 받았다고 알려주기 위해 다음과 같이 해당 리소스를 호출하고 HTTP 200 OK 응답을 받는다.

HTTP: PUT /{url}/Chat/{InstanceId}

(20-31) WebRSF는 위 요청을 받은 후 해당 요청을 SIP MESSAGE로 변환하여 CPIM/IMDN 형식으로 delivered 알림을 전송한다. 나머지 흐름은 그림 1의 (15-26) 흐름과 동일하다.

그림 5는 RCS UE가 WebRSF 클라이언트에게 두 번째 또는 그 이후의 대화 요청을 보낼 때 네트워크 노드들 간의 메시지 교환 절차를 나타낸 것이며 그 세부 흐름은 다음과 같다.

(1-6) MSRP 세션 수립 후 과정은 그림 2와 동일하다.

(7) WebRSF는 MSRP SEND 내 Sender-URI, Content, From tag, To tag 헤더를 읽어 들인다. 이를 클라이언트로 전달하기 위해 일단 스트리밍 채널을 통해 받을 데이터가 있다는 것을 클라이언트에 알린다.

(8) 클라이언트는 받을 데이터가 있다는 것을 스트리밍 채널을 통해 알게 되고 WebRSF에 다음과 같이 해당 리소스를 요청한다.

HTTP GET /{uri}/Chat/Media

(9) WebRSF는 RCS UE가 보낸 SEND를 바탕으로 chunked transfer encoding된 데이터를 만들고 HTTP 200 OK를 사용하여 클라이언트에 해당 데이터를 전송한다.

(10-11) WebRSF 클라이언트는 대화 내용을 잘 받았다고 알려주기 위해 HTTP PUT으로 해당 리소스를 호출하고 HTTP 200 OK 응답을 받는다.

(12-17) 그림 2의 (7-12) 흐름과 같다.

표 1. 스트리밍 채널로 WebRSF 클라이언트에게 전송될 메시지의 예

Table 1. An example message that the WebRSF client would receive through the streaming channel

```
{
  "type" : "chat",
  "instance" : "https://{uri}/Chat/{instance-id}",
  "data" : {
    "status" : "initiate",
    "subject" : "Hello?",
    "message_id" : "OvKxVdCERdeqaRA",
    "sender" : "01012341234"
  }
}
```

Ⅳ. 성능 평가

WebRSF의 성능을 평가하기 위하여 RCS 클라이언트와 WebRSF 클라이언트에서 1대1 대화 요청 시 처리된 SIP과 MSRP 메시지 수를 비교하였다[13]. 그림 1, 그림 2, 그림 4, 그리고 그림 5의 메시지 교환 절차를 바탕으로 성능 평가를 위해 시뮬레이션을 수행하였다. 그림 3의 네트워크 구성도에서 각 노드들을 시뮬레이션하기 위해 C++ 프로그램을 작성하였다. 시뮬레이션 환경은 천만 명의 RCS 클라이언트 사용자와 WebRSF 클라이언트 사용자가 있다고 가정하였고 각 사용자는 Poisson(25) 분포로 1대 1 대화를 요청 한다고 가정하였다.

그림 6은 이 시뮬레이션의 결과를 보여준다. SIP 메시지를 처리할 때 WebRSF 클라이언트는 RCS 클라이언트보다 17% 감소된 메시지를 처리하였다. 그러나 세션이 수립된 후 MSRP 메시지를 처리할 때에는 WebRSF 클라이언트가 RCS 클라이언트보다 25% 많은 메시지를 처리하였다. 대부분의 1대 1 대화는 최초 한번만 SIP을 사용하고 나머지는 MSRP를 사용한다는 점을 감안하면 WebRSF 클라이언트를 사용할 경우 RCS 클라이언트를 사용하는 것보다 더 많은 클라이언트 부하를 요구할 것으로 예상된다.

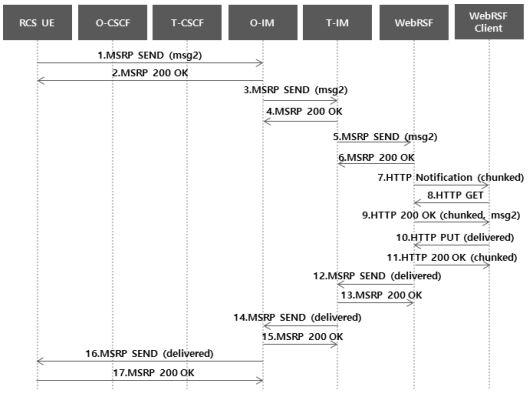

그림 5. WebRSF를 사용하여 MSRP 세션 기반의 대화를 주고받기 위한 절차를 나타낸 시퀀스 다이어그램
Fig. 5. A sequence diagram showing procedures for exchanging MSRP-session-based chat messages using WebRSF

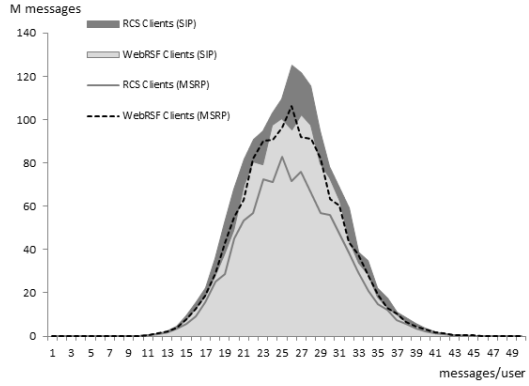

그림 6. 1대 1 대화 처리 시 RCS 클라이언트와 WebRSF 클라이언트에서 처리된 SIP과 MSRP 메시지의 수
Fig. 6. The number of SIP and MSRP messages sent and received by RCS clients and WebRSF clients during the 1-to-1 chatting

V. 결 론

WebRSF를 도입하면 제3의 개발자가 쉽게 RCS 클라이언트 기능을 갖춘 앱을 개발할 수 있도록 해 주는 장점이 있다. WebRSF는 웹 인터페이스를 이용하여 해당 기능을 제공해 주기 때문에 스마트폰뿐만 아니라 PC, 태블릿, TV, 자동차 등 다양한 기기에서 이용할 수 있다. 본 논문에서는 이를 위해 WebRSF 및 WebRSF 클라이언트와 기존 RCS 노드와의 네트워크 구성도를 설계하였고, 기존 노드들과의 메시지 교환 흐름을 정의하였다. 또한 WebRSF 클라이언트가 스트리밍 채널을 통해 받을 메시지의 예시를 제시하였다. 뿐만 아니라 성능 평가를 통해 WebRSF 클라이언트를 사용하는 디바이스가 RCS 클라이언트를 사용하는 디바이스보다 얼마나 많은 부하를 발생시킬지 성능 평가를 통해 알아보았다.

References

[1] B. B. Moshe, A. Dvir, and A. Solomon, “Analysis and optimization of live streaming for over the top video”, Proceedings of the 2011 IEEE Consumer Communications and Networking Conference, pp.60-64, 2011 (DOI: 10.1109/CCNC.2011.5766553)

[2] GSMA, “RCS Market Launches to date”, May 2014

[3] S.-S. Y, S.-T. Kim, "Implementation of IMS Core SIP Gateway based on Embedded", JIIBC, Vol. 14, No. 5, October 2014 (DOI: 10.7236/JIIBC.2014.14.5.209)

[4] GSMA, “Rich Communication Suite 7.0 Advanced Communications Services and Client Specification”, June 2017

[5] IETF, “Common Presence and Instant Messaging (CPIM): Message Format”, Request for Comments: 3862, 2004

[6] IETF, “Instant Message Disposition Notification”, Request for Comments: 5438, 2009

[7] GSMA, "Rich Communication Suite RCS API Detailed Requirements 2.5", March 2015

[8] D. Lee, “A Web-based Open API Framework for RCS-e Session Establishment”, JIIBC, Vol. 15, No. 5, October 2015 (DOI:110.7236/JIIBC.2015.15.5.125)

[9] C-Y. Lee, “A Study of M2M Platform Technologies based on REST”, Journal of the Korea Academia-Industrial cooperation Society (JKAIS), Vol. 12, No. 9, 2011 (DOI: 10.5762/KAIS.2011.12.9.4153)

[10] O. Liskin, L. Singer, and K. Schneider, "Welcome to the Real World: A Notation for Modeling REST Services", IEEE Internet Computing, Vol. 16, Iss.

4, 2012 (DOI: 10.1109/MIC.2012.59)

[11] M. Adeyeye, I. Makitla, and T. Fogwill, "Determining the signalling overhead of two common WebRTC methods: JSON via XMLHttpRequest and SIP over WebSocket", AFRICON, 2013 (DOI: 10.1109/AFRCON.2013.6757840)

[12] IETF, "SIP: Session Initiation Protocol", Request for Comments: 3261, 2002

[13] Z. Ding, Y. Sun, C. Jiang, M. Zhou, J. Liu and W. Song, "Performance Evaluation of Transactional Composite Web Services", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 46, Iss. 8, 2016 (DOI: 10.1109/TSMC.2015.2478886)

저자 소개

이 동 철(종신회원)

- 2002년 2월 : POSTECH 컴퓨터공학과 학사
- 2004년 2월 : POSTECH 전자컴퓨터공학과 석사
- 2004년~2012년 : KT책임연구원
- 2012년 8월 : 한양대학교 전자컴퓨터통신공학과 박사
- 2012년 9월~현재 : 한남대학교 교수

<관심분야 : 소프트웨어 프레임워크, 모바일 앱, RCS>

※ 이 논문은 2017년도 한남대학교 교비학술연구비지원으로 작성되었습니다.