

통합 SNS 게이트웨이를 위한 웹 서비스 어댑터 구현 및 성능 분석[☆]

Implementation and Performance Analysis of Web Service Adapter for Integrated SNS Gateway

정 인 식¹ 김 현 우¹ 권 동 우¹ 주 흥 택^{1*}
Insik Jung Hyeonwoo Kim Dongwoo Kwon Hongtaek Ju

요 약

본 논문은 모바일 SNS 트래픽을 줄이는 방안으로 통합 SNS 게이트웨이를 이용하는 방법을 제시한다. 통합 SNS 게이트웨이는 모바일 클라이언트와 SNS 서버 사이에서 통신 중계자 역할을 한다. 통합 SNS 게이트웨이 내에서 SNS 서버와의 통신을 담당하는 웹 서비스 어댑터의 역할을 제안하고 구현하였다. 또한, 통합 SNS 게이트웨이 내의 캐시 엔진을 사용하여 사용자가 통합 SNS 게이트웨이를 통하여 특정 SNS 서버에게 콘텐츠를 요청할 때 발생하는 트래픽량과 응답 시간에 대한 성능을 측정하고 분석하여 결과를 제시한다.

☞ 주제어 : SNS 게이트웨이, 웹 서비스 어댑터, SNS 트래픽 측정, 캐싱 성능 측정, SNS 콘텐츠 캐싱

ABSTRACT

In this paper, we present integrated SNS Gateway as way to reduce mobile SNS traffic. Integrated SNS gateway's role is communication broker that located between mobile clients and SNS server. In Integrated SNS gateway, web service adapter communicates with SNS Server. Experiment was performed in aspects of contents RTT and network traffic. We evaluate and propose performances of integrated SNS gateway using gateway.

☞ keyword : SNS Gateway, Web Service Adapter, SNS Traffic Measurement, Caching Performance Measurement, SNS Contents Caching

1. 서 론

Social Network Service(SNS)는 손쉽게 다른 사람들과 관계를 유지하거나 만들 수 있고 개인적인 의견을 공유할 수 있는 서비스이다. 이 때문에 현재도 전 세계적으로 사용자의 수가 꾸준히 증가하고 있다. 대부분의 SNS는 편리한 서비스 제공을 위해 스마트폰을 지원함으로써 모바일 SNS를 이용하는 사용자도 많아졌다. 실제 조사를 따르면 스마트폰의 사용자 60.3%가 SNS를 이용한 경험이 있는 것으로 나타났고, 또 SNS 경험자의 84.7%가 스

마트폰을 통해 SNS를 이용한다고 나타났다[1].

하지만 스마트폰으로 SNS를 이용하는 사용자가 증가하면서 고려할 사항도 늘어났다. 실제로 SNS를 이용하면서 사용자들의 트래픽이 과도하게 몰리는 현상으로 인해 장애가 일어나기도 했다. 특정 모바일 SNS의 사례를 보면, SNS의 메시지의 전달방식으로 폴링방식을 사용하였는데 SNS 사용자들의 빈번한 메시지 전송으로 인해 많은 메시지를 전달하는 과정에서 해당 메시지의 서버뿐만 아니라 모바일 통신사의 인터넷속도까지 느리게 만들었다. 또 다른 SNS는 사용자를 대상으로 한 이벤트로 인해 특정 페이지에 집중적으로 사용자가 몰리게 됨으로써 해당 페이지가 한동안 동작하지 않았다. 이처럼 사용자들이 한 서버에 많은 요청을 하게 되면 분산 서비스 거부(Distributed Denial-of-Service, DDoS)공격과 유사한 형태로 해당 서버 또는 네트워크가 마비되는 현상을 일으킬 수도 있다. 이러한 현상은 일시적으로 서버의 동작이 정지시킴과 사용자에게 불편함을 제공하고 운영기업의 서비스 신뢰성 하락을 가져올 수 있어 기업 이미지에 치명

1 Computer Engineering, Keimyung University., Daegu, 704-701, Korea.

* Corresponding author (juht@kmu.ac.kr)

[Received 9 January 2014, Reviewed 14 January 2014, Accepted 12 February 2014]

☆ This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation(2012H1B8A2025942)

적일 수 있다.

이러한 현상을 방지하기 위해 우리는 SNS 서버 콘텐츠의 응답 속도를 향상시키며 최근 증가하는 모바일 SNS 트래픽을 줄이는 방안으로, 모바일 클라이언트와 SNS 서버 사이에 통합 SNS 게이트웨이 서버를 설치하여 동작시키는 방법을 이용한다. 통합 SNS 게이트웨이 서버는 캐시 엔진을 사용하여 콘텐츠 요청에 대한 빠른 응답을 기대할 수 있다. 또 SNS 서버와 동작하는 웹 서비스 어댑터 모듈과 다수의 SNS를 통합하여 편리하게 사용할 수 있는 모바일 애플리케이션을 구현하여 한 번의 요청으로 여러 SNS의 응답을 받을 수 있게 하여 트래픽을 줄이는 효과를 얻을 수 있다.

따라서 본 논문에서는 통합 SNS 게이트웨이 서버의 구조를 설명하고 웹 서비스 어댑터의 구조와 통합 프로토콜을 제시한다. 그리고 게이트웨이의 유/무와 캐시의 유/무에 따른 통합 모바일 애플리케이션의 콘텐츠 응답 속도 실험을 수행하고 실험 결과를 제시한다. 본 논문에서는 여러 SNS 중 세계적으로 널리 쓰이는 페이스북을 대상으로 웹 서비스어댑터를 구현하고 페이스북 서버에 글, 사진, 동영상의 세 가지 콘텐츠를 요청하는 실험을 하였다. 실험은 현재 구현된 페이스북 웹 서비스 어댑터를 거쳐서 페이스북 서버에 콘텐츠를 요청하는 통신을 할 때, 게이트웨이를 거치는 경우와 그렇지 않은 경우, 또 게이트웨이를 거칠 때 캐싱이 동작할 경우와 동작하지 않은 경우를 비교하여 응답 속도를 측정한다. 또 페이스북 SNS 서버와 게이트웨이 사이의 트래픽량을 측정한다.

본 논문의 2장에서는 관련 연구로 휴대폰을 이용한 원격 모니터링 대한 게이트웨이의 사용 방법과 스마트 홈에서 게이트웨이의 사용법에 대하여 설명한다. 3장에서는 통합 SNS 게이트웨이와 웹 서비스 어댑터의 구현과정을 설명하고 4장에서 통합 SNS 게이트웨이의 사용 유/무에 따른 응답속도와 트래픽량을 비교하기 위해 구축한 실험 환경과 측정된 실험 결과를 이야기한다. 마지막으로 5장에서는 결론과 향후 연구를 제시하고 마무리한다.

2. 관련 연구

정인석 외 2명은 [2]에서 모바일 클라이언트에서 SNS 서버와 통신할 때 트래픽을 줄이는 방안으로 모바일 통합 SNS 게이트웨이를 제시하였다. 그리고 모바일 통합 게이트웨이 SNS서버 내 페이스북 어댑터를 설계하고 구현하였다. 본 논문에서는 이전 연구를 바탕으로 페이스북

어댑터의 기능을 추가하고 캐싱 서버를 두어 모바일 통합 SNS 게이트웨이가 클라이언트 통신에 미치는 영향을 제시한다.

Yoshiro Imai 외 3명은 [3]에서 네트워크 카메라와 모바일 클라이언트 사이에 애플리케이션 게이트웨이를 두고 네트워크 카메라에서 모니터링 하는 사진들을 게이트웨이로 전송한다. 전송되는 사진들은 모바일 기기에 맞게 크기를 조절하여서 전송하여 모바일 클라이언트와 게이트웨이 사이의 불필요한 트래픽을 감소시키는 방법을 제시한다. 우리의 연구에서는 사용자 입장에서 SNS의 콘텐츠의 필요한 부분만 요청하여 클라이언트에게 전송함으로써 트래픽을 감소시킬 수 있다.

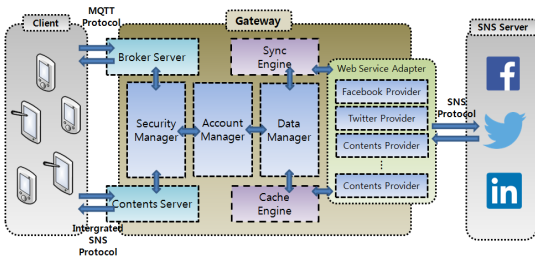
Dimitar Valtchev와 Ivailo Frankov의 연구 [4]는 Open Services Gateway initiative(OSGi)를 이용하여 송전선, 전화선 등의 서로 다른 외부의 프로토콜들을 통합하고 사용하는 통합된 프로토콜을 이용하여 스마트 홈을 사용하는 방법을 제시하였다. 게이트웨이를 사용함으로써 하나의 콘솔에서 스마트홈 내의 서로 다른 서비스들을 관리할 수 있다. 우리는 서로 다른 SNS들의 프로토콜들을 통합하는 방법으로 게이트웨이를 사용한다.

3. 통합 SNS 게이트웨이 서버

3.1 통합 SNS 게이트웨이의 기능

우리는 이전 연구[5]에서 설계된 모바일 통합 SNS 게이트웨이 서버의 구조를 (그림 1)과 같이 제안하였다. 통합 SNS 게이트웨이 서버는 클라이언트의 통합 SNS 애플리케이션과 SNS 서버 사이에 위치하여 클라이언트에서 요청하는 콘텐츠들을 캐싱하고 동기화하는 기능을 제공한다.

SNS 사용자가 SNS에 콘텐츠를 요청하면 SNS 서버는 매번 사용자의 요청에 응답하게 된다. 또한, 다수의 SNS를 사용할 경우 다수의 SNS 애플리케이션을 통해 각 콘텐츠를 각기 다른 SNS로 요청하게 되고 그만큼 트래픽이 증가한다. 하지만 SNS에 콘텐츠를 요청할 때 통합 SNS 게이트웨이를 이용하게 되면 여러 가지 장점이 있다. 먼저 사용자 입장에서는 하나의 통합 SNS 애플리케이션에서 한 번의 동작으로 다양한 종류의 SNS의 정보들을 얻을 수 있다. 또 게이트웨이 서버의 캐싱으로 인해 빠른 응답을 얻을 수 있다. 반면에 해당 SNS 서버 입장에서는 통합 SNS 게이트웨이의 캐싱 기능으로 인해 동일한 SNS 콘텐츠에 대한 요청 트래픽이 감소한다.



(그림 1) 통합 SNS 게이트웨이의 구조

(Figure 1) Structure of the Integrated SNS Gateway

3.2 웹 서비스 어댑터 구성

웹 서비스 어댑터란 게이트웨이 서버 내에서 게이트웨이와 SNS 서버 간의 데이터를 교환하는 어댑터 역할을 한다. 데이터를 교환하기 위해서 통합 SNS 애플리케이션과 통합 SNS 게이트웨이가 통신하는 통합 SNS 프로토콜을 게이트웨이와 SNS 서버가 통신하는 전용 프로토콜로 바꾸는 기능이 필요한데, 웹 서비스 어댑터가 각 SNS 통신에 맞게 프로토콜을 변환하는 기능을 한다. 게이트웨이가 SNS 서버와 통신하는 과정은 각 SNS의 공개된 API를 사용하여 통신한다. 이 때문에 웹 서비스 어댑터는 SNS 별로 해당 API로 구현된 콘텐츠 프로바이더들을 가지고 있고 독립적으로 각 SNS와 통신을 한다. 현재 구현되어 있는 페이스북 콘텐츠 프로바이더는 페이스북의 PHP SDK를 사용하여 구현하였다. 페이스북 이외의 콘텐츠 프로바이더들은 각각의 SNS가 지원하는 API를 이용하여 구현할 수 있다.

3.3 웹 서비스 어댑터 인증과정

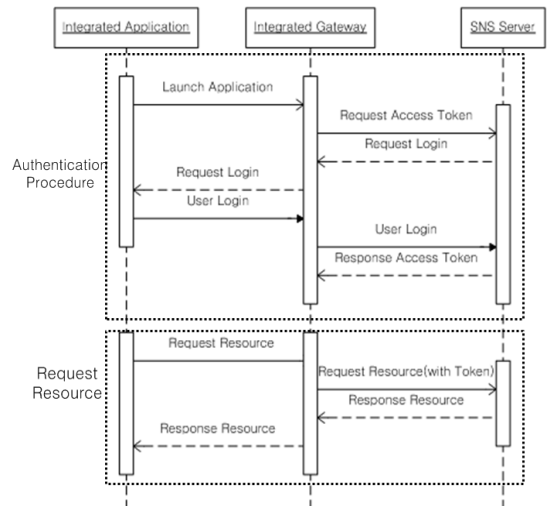
현재 SNS들은 자신들만의 안전한 인증과정을 사용하고 있다. 페이스북, 트위터 등 여러 SNS는 사용자 인증으로 OAuth를 사용한다. 통합 SNS 게이트웨이에서도 SNS에 접속하기 위해서는 OAuth 인증과정을 거쳐야 한다. 따라서 일반적인 SNS 클라이언트를 대신할 통합 SNS 게이트웨이에서의 인증과정을 설명한다.

OAuth는 ID/Password의 직접적인 입력으로 인한 유출을 방지하기 위하여 써드 파티 애플리케이션에서 Access Token을 받아 사용하는 인증 방법이다. Access Token은 사용자를 대신하여 서비스 프로바이더에 자원을 요청할 수 있게 한다. 다양한 SNS 중 트위터, My Space와 같은 서비스는 OAuth 1.0a 버전을 사용하고, 페이스북, 구글은

OAuth 2.0을 사용한다. 두 버전은 지원범위나, 요청방식, 인증모델 등의 차이가 있지만 핵심적인 기능은 동일하다.

통합 애플리케이션 및 게이트웨이와 SNS 서버에서 발생하는 인증과정은 (그림 2)와 같다. 인증 절차에서 먼저 사용자가 통합 SNS 애플리케이션을 시작하게 되면 통합 애플리케이션은 게이트웨이에 애플리케이션 실행을 알리고 게이트웨이는 SNS 서버에게 Access Token을 요청한다. 그 후 서버는 현재 사용되는 사용자에 대한 인증을 요청하게 되고 로그인을 함으로써 게이트웨이 서버는 사용자의 권한을 Access Token이란 이름의 토큰으로 게이트웨이가 가지게 된다. 토큰은 처음 접속할 때만 요청하게 되며 토큰이 만료되기 전까지 토큰 요청 없이 계속해서 사용할 수 있다.

통합 애플리케이션과 게이트웨이는 통합 SNS 프로토콜을 사용하여 통신하고 게이트웨이와 SNS 서버는 SNS API를 사용하여 통신한다. SNS 서버에 콘텐츠를 요청하게 될 때는 통합 애플리케이션에서는 통합 SNS 프로토콜로 게이트웨이에게 콘텐츠를 요청하고 게이트웨이는 여기에 인정 받은 토큰을 더해 SNS API로 요청하게 되고 SNS 서버는 토큰을 확인한 후 해당 콘텐츠를 응답해준다.



(그림 2) 통합 SNS 게이트웨이의 인증 절차

(Figure 2) Authentication Procedure of Integrated SNS Gateway

3.4 페이스북의 데이터 통신 방법

페이스북에서는 여러 종류의 API를 제공하는데 여러

방법 중에서도 콘텐츠를 요청/응답받는데 있어서 가장 많이 사용되는 Graph API 방법과 Facebook Query Language (FQL) 방법이 있다[9]. 페이스북은 사진, 글 등을 각각 하나의 개체로 취급하는데 Graph API 방법은 개체와 개체 사이의 연결을 일정한 방식으로 표현하여 요청하는 방식으로 간단하게 페이스북 데이터를 읽고 쓸 수 있다. 반면에 FQL은 SQL스타일의 인터페이스를 가지고 쿼리를 만들어 해당 테이블의 콘텐츠를 요청하는 방법이다. 이 방법은 Graph API보다 더 자세한 조건을 만들어서 콘텐츠를 요청할 수 있고 Graph API가 여러 번 요청하여 응답받는 콘텐츠를 FQL은 한 번의 쿼리를 호출함으로써 콘텐츠를 얻을 수 있기 때문에 보다 직관적이다. 그러므로 본 논문에서는 FQL 방식을 이용하여 실험용 페이스북 SDK를 사용한 애플리케이션과 웹 서비스 어댑터의 페이스북 프로바이더를 구현하였다.

페이스북은 FQL으로 콘텐츠를 요청받게 되면 응답 데이터 형식으로 JSON을 사용한다. 페이스북의 콘텐츠는 각각 테이블형식으로 되어있다. 사진 콘텐츠는 PHOTO 테이블에서 동영상은 VIDEO 테이블에서 질의할 수 있다. 따라서 SQL의 질의와 같이 클라이언트에서 FQL으로 페이스북 서버에 콘텐츠를 요청하게 되면 요청한 테이블 (FROM 절의 테이블)에서 원하는 컬럼들(SELECT 절의 컬럼)이 JSON 형식으로 응답이 오게 된다. FQL 방식에 예로 들면 만약 자신의 STREAM 테이블에 message라는 컬럼을 요청하게 되면 message라는 하나의 컬럼만 요청하였기 때문에 (그림 3)과 같은 포맷으로 응답을 받을 수 있고 해당 콘텐츠를 바로 사용할 수 있다. 반면에 사진과 동영상을 페이스북 서버로 요청하게 되면 응답되는 데이터는 사진과 동영상의 바이너리 데이터가 아니라 해당 사진, 동영상이 저장된 URL이다. 따라서 응답받은 URL을 통해 다시 한 번 콘텐츠를 요청해야 한다.

<pre> Message Column From Stream Table : { "data": [{ "message": "Hello world" }] } </pre>	<pre> SRC Column From Photo Table : { "data": [{ "src": "http://.....jpg" }] } </pre>
--	---

(그림 3) Facebook의 FQL 쿼리 응답 형식

(Figure 3) FQL Query Response Format from Facebook

요청한 FQL의 message가 여러 개 있을 경우에도 결과 값이 (그림 3)과 같이 'DATA'라는 JSON 배열로 정렬되어서 오기 때문에 한 번의 요청으로 모든 글을 불러올 수 있다.

3.5 통합 SNS 프로토콜 설계

우리는 이전 연구 [5]에서 우리가 개발한 통합 SNS 애플리케이션과 통합 SNS 게이트웨이 사이의 통신 규격인 통신 콘텐츠 로드 프로토콜을 제안하였다. 설계한 콘텐츠 로드 프로토콜은 웹 페이지에 대한 데이터 요청을 고려하여 여러 SNS에 대해 다양한 데이터를 요청하지 못할 뿐만 아니라 페이스북, 트위터와 같이 자신의 글이나 사진을 올릴 때에 대해서 고려하지 않았다. 그러므로 우리는 [6]에서 SNS의 프로토콜을 분석한 후 SNS를 고려하여 데이터 형식을 다시 설계하고 현재 웹 서비스 어댑터에 적용하였다.

(그림 4)는 설계한 통합 SNS 프로토콜이다. 통합 SNS 프로토콜은 통합 SNS 애플리케이션과 통합 SNS 게이트웨이간의 통신을 할 때 사용된다. 프로토콜 포맷은 Java Script Object Notation(JSON)포맷[7]을 따르고 있다. JSON 형식은 XML과 같은 객체 직렬화방법과 비교할 때 빠른 속도를 보여준다[8]. 뿐만 아니라 JSON은 서로 다른 시스템간의 객체를 교환하는데 편리하다. 이러한 이유로 우리는 JSON 형식을 사용하여 프로토콜을 설계하여 적용하였다.

<pre> Contents Request Format : { "ID": "Own Client ID", "SNS TYPE": "Request SNS Type", "DATA": { "URL": "Request URL", "USER": "Target User", "OBJECTIVE": "Request Objective", "MESSAGE": "Request Message" } } </pre>	<pre> Contents Response Format : { "ID": "Own Client ID", "SNS TYPE": "Request SNS Type", "DATA": { "MESSAGE": "Response Message", "IMGSRC": "Image URL", "USER": "Contents Owner", "DATE": "Created Time" } } </pre>
---	---

(그림 4) 통합 SNS 프로토콜

(Figure 4) Integrated SNS Protocol

4. 게이트웨이 성능 측정

4.1 실험 목적

이 실험은 기존의 페이스북의 콘텐츠 요청방법과 비교하여 설계된 통합 SNS 게이트웨이를 이용할 때의 트래픽량과 응답속도를 비교한다.

통합 SNS 게이트웨이의 유무에 따른 차이와 캐시 엔진의 사용 유/무에 따른 비교를 하기 위해 각 콘텐츠 요청에 대해 페이스북 안드로이드 SDK를 사용하여 구현한 페이스북 클라이언트 애플리케이션과 페이스북 서버의 통신에서 트래픽을 수집하고 통합 SNS 애플리케이션과 통합 SNS 게이트웨이와 페이스북 서버 사이의 트래픽을

수집하여 비교한다. 현재 구현된 통합 SNS 게이트웨이의 기능인 캐시 엔진과 웹 서비스 어댑터를 이용할 때, 사용자 측면에서 얼마나 응답 속도가 향상되는지 측정하고 트래픽이 감소하는지 분석한다.

4.2 실험 환경

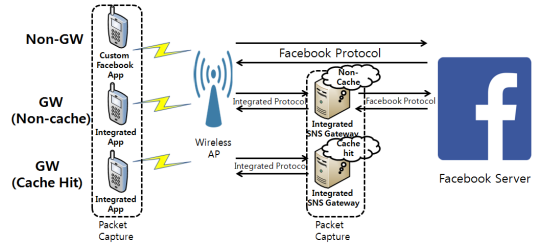
게이트웨이의 성능 측정을 하기 위해 (그림 5)와 같이 실험 환경을 구성하였다. 게이트웨이의 CPU는 3.40GHz 듀얼 코어이고, 메모리는 1GB이다. 운영체제는 CentOS release 5.10 이고 캐시 엔진으로는 Squid 2.6. STABLE21 을 사용하였으며 설정은 기본값으로 설정하여 실험하였다. 모바일 단말은 CPU Quad Core 1.4GHz와 2GM의 메모리를 사용하였다. 페이스북 안드로이드 SDK는 3.5.2버전을 사용하였고 PHP SDK는 3.2.3 버전을 사용하였다.

실험은 (그림 5)와 같은 세 가지 환경에서 수행하였다.

- Non-GW 실험은 페이스북에서 지원하는 안드로이드 SDK를 사용하여 페이스북 서버에 콘텐츠를 요청하는 페이스북 클라이언트 애플리케이션을 만들고 콘텐츠를 요청할 때 발생하는 패킷을 수집하고 응답받는 속도를 측정하였다(Non-GW).
- GW(Non-cache) 실험은 통합 SNS 게이트웨이와 통신하는 통합 모바일 SNS 애플리케이션을 만들어 통합 SNS 게이트웨이를 거쳐 SNS 서버에 콘텐츠를 요청한다. 이때 캐시를 사용하지 않는다(Non-cache).
- GW(Non-cache) 실험은 캐시가 적용되는 상황에서 통합 모바일 SNS 애플리케이션을 사용하여 콘텐츠를 요청한다(Cache Hit).

Non-Cache 실험과 Cache Hit 실험의 통합 SNS 애플리케이션은 통합 SNS 프로토콜을 사용하여 게이트웨이에게 콘텐츠를 요청하게 된다. 요청을 받게 되면 게이트웨이에 구현된 페이스북 어댑터는 페이스북과 통신을 하기 위해서 통합 SNS 프로토콜을 페이스북 프로토콜로 변환하는 작업을 한다. 그 후 변환된 페이스북 프로토콜을 사용하여 페이스북에 콘텐츠를 요청하게 된다. 이 통신과정에서 캐시의 유/무가 콘텐츠 응답속도에 미치는 영향과 애플리케이션과 페이스북 서버 사이에 발생하는 트래픽량을 비교한다.

Non-GW 실험의 페이스북 애플리케이션과 Non-Cache 실험, Cache Hit 실험의 게이트웨이의 페이스북 어댑터들은 페이스북의 FQL을 이용하여 콘텐츠를 요청하는 방식으로 구현하였다.



(그림 5) 페이스북 트래픽 측정 실험 환경
(Figure 5) Facebook Traffic Measurement Experiment Environment

패킷을 수집하는 방법으로는 모바일 단말기의 네트워크 환경을 제어할 수 있게 하도록 루트 권한을 얻은 후 안드로이드 애플리케이션을 설치하여 모바일 단말기에서 요청하고 응답받은 패킷들을 수집하였다.

4.3 실험 방법

페이스북에 저장된 글, 사진, 동영상을 요청하여 응답받을 때 게이트웨이의 성능 측정은 다음과 같이 실험하였다. 먼저 페이스북의 담벼락에 15개의 글을 작성하고 사진첩에 15개의 서로 다른 크기의 사진, 1개의 동영상을 업로드 하였다. (그림 5)의 세 가지 방법을 이용하여 각 콘텐츠를 요청하였다. 글 요청의 경우에는 한 번의 요청으로 15개의 글을 가져오고 사진은 한 번의 요청으로 15개의 사진을 모두 가져오도록 FQL을 작성하였다.

사진과 동영상의 경우는 페이스북에서 직접적인 콘텐츠 데이터가 아니라 해당 콘텐츠의 URL로 응답해주기 때문에 응답받은 URL로 다시 한 번 HTTP 요청을 한다. 그렇게 되면 게이트웨이의 캐시엔진은 통합 모바일 SNS 애플리케이션이 요청한 콘텐츠와 캐시된 콘텐츠를 비교하여 캐시가 적용한다면 직접 응답을 하고 캐시 실패한다면 게이트웨이에서 페이스북 서버에 요청한다. 요청한 콘텐츠의 응답이 오게 되면 캐시엔진은 해당 콘텐츠를 캐싱하며 통합 모바일 SNS 애플리케이션에 응답하게 된다.

클라이언트 응답 속도를 측정하기 위해서 3가지 실험 모두 클라이언트에서 패킷을 수집하고, 페이스북 서버로 향하는 트래픽량을 측정하기 위해서 Non-GW 실험은 모바일 클라이언트에서, Non-Cache와 Cache-Hit 실험은 게이트웨이에서 페이스북 서버로 요청하는 패킷들과 응답받는 패킷들을 수집하였다.

4.4 실험 결과

먼저 각 콘텐츠의 캐시 가능 여부를 확인하기 위해 각 콘텐츠들을 요청 하고 캐시가 되는지 확인하였다. 확인결과, 글과 동영상 요청에 대해서 캐시가 되지 않았다. 페이스북의 프로토콜로 동작하는 데이터들은 캐시엔진의 캐시로그가 남지 않았다. 하지만 사진과 같이 URL을 명시하여 요청하는 경우에는 캐시로그가 기록되며 캐시가 되는 것을 확인할 수 있었다. 동영상을 요청할 때는, 동영상 콘텐츠의 주소를 요청하지만, 페이스북 서버에서 해당 동영상의 주소를 동적 URL로 응답이 오기 때문에 캐시엔진에서 캐시로그를 확인할 때 전체 URL이 저장되어 남지 않고 동적 URL의 파라미터 전인 “?” 기호까지만 로그가 남는 것을 확인할 수 있었다. 또한, 캐시를 하게 되었을 때 생기는 캐시파일도 생기지 않았다.

반면에 사진에 대해서는 15개의 사진요청에 대한 응답이 누락되는 데이터 없이 모두 캐시되는 것을 확인할 수 있었다. 사진들에 대해 요청을 하여 응답받은 URL과 패킷들을 분석해 보면 각각 2, 1, 1, 3, 3, 1, 3, 1개로 각각 나뉘어 8개의 페이스북 서버에 저장된 것을 확인할 수 있다. 사진을 요청하게 될 경우 8개의 세션이 만들어져서 사진을 응답받기 때문에 우리는 세션별로 트래픽과 응답속도를 측정하였다. 사진 별로 트래픽을 측정하는 방안도 있지만 TCP 연결이 사진별로 일어나지 않고 세션마다 일어나기 때문에 트래픽량을 정확하게 측정하기 위해 세션별로 트래픽과 응답시간을 측정하였고 결과로 나타났다.

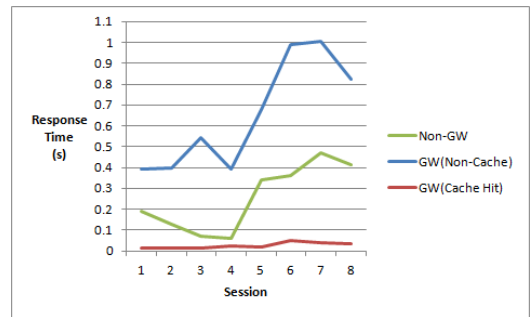
실험의 사진 응답 속도 측정은 수집한 패킷들을 각 서버의 세션별로 분류하여 해당 세션의 크기의 오름차순으로 정렬하였다. 또한 반복된 사진 요청의 응답시간 평균을 계산하고 각 실험방법의 응답시간을 그래프로 나타내었다. (표 1)은 사진을 요청하게 될 때 나뉘는 세션별 사진의 개수와 트래픽의 크기이다.

(그림 6)은 페이스북 서버에 사진을 요청하였을 때부터 사진을 모두 응답 받는 데까지 걸린 시간을 세션별로 나타낸 그래프이다. (그림 6)의 “Non-GW” 그래프는 페이스북 안드로이드 SDK를 사용하여 만든 페이스북 클라이언트 애플리케이션으로 SNS 게이트웨이 없이 사진을 요청했을 때의 응답시간이고 Non-GW실험과 같다. 캐시가 적용되지 않았을 때의 “Non-Cache”의 그래프는 게이트웨이를 거치지만 캐시가 되지 않을 때 페이스북 서버에 사진을 요청하는 것이고 실험 Non-Cache와 같다. “Cache Hit” 그래프는 게이트웨이에서 사진을 요청받았을 때 캐시가 적용해서 캐시에 저장된 사진들을 응답했을 때의 결과이고 Cache Hit 실험이다.

(표 1) 세션별 사진개수와 사이즈

(Table 1) Photos Count and Size for each Session

Session	The Number of Photos	Session Size (Byte)
1	1	6,308
2	1	6,401
3	1	6,474
4	1	6,623
5	2	10,334
6	3	15,215
7	3	16,508
8	3	18,147



(그림 6) 각 세션의 응답시간

(Figure 6) Reponse Time for each Session

Non-GW 실험은 페이스북 클라이언트 애플리케이션에서 게이트웨이를 거치지 않고 페이스북에 직접 통신하기 때문에 페이스북 서버와 통신하는 정상적인 속도를 보여준다. “Non-cache” 그래프는 캐싱이 되지 않지만 게이트웨이를 거치기 때문에 페이스북 클라이언트 애플리케이션의 그래프보다 느린 속도가 나타난다. “Cache Hit” 그래프는 캐시가 적용되었기 때문에 캐시가 되지 않은 그래프보다 빠른 속도를 보여줄 뿐만 아니라 페이스북 클라이언트 애플리케이션보다 빠른 응답시간을 보여준다. 이는 처음으로 해당 콘텐츠를 요청할 경우는 일반적인 페이스북 애플리케이션의 응답속도보다 느리지만 콘텐츠가 캐싱이 되고 다음 요청에서 캐시가 적용하게 되면 게이트웨이에서 페이스북 서버를 거치지 않고 직접 사진을 응답하게 되므로 더 빠르다는 결과를 나타낸다. SNS의 특성상 같은 콘텐츠를 반복적으로 요청하는 경우가 많고 또 여러 명의 사용자가 게이트웨이를 사용해서 콘텐츠를 요청할 경우 사용자가 증가하면 증가할수록 캐시 엔진에 의해서 응답속도가 빨라질 것이다.

또한 사진을 요청할 때 SNS 서버에 직접적으로 전송되는 트래픽량을 비교하기 위해 (그림 5)의 *Non-GW* 실험에서 사용한 페이스북 클라이언트 애플리케이션에서 페이스북 서버로 전송되는 트래픽량을 측정하였다. 그리고 실험한 *Non-Cache* 실험과 *Cache Hit* 실험에서 사용한 통합 SNS 게이트웨이에서 SNS 서버로 전달되는 트래픽을 측정하였다.

Non-GW, *Non-Cache* 방법으로 사진 요청 실험을 수행하였을 때는 사진의 용량과 TCP 연결에 따른 패킷들이 모두 발생함으로써 세션의 (사진용량요청/응답+세션연결) 크기 만큼의 트래픽이 모두 발생하였다. 하지만 *Cache Hit* 실험을 수행했을 때 발생한 패킷은 없었다. 사진의 경우 캐시 엔진의 캐시 용량이 충분하다면 100% 캐시가 되고 그에 따라 사진 요청의 캐시가 적절한 것을 확인하였고 게이트웨이가 SNS 서버에 요청하는 패킷은 전혀 발생하지 않았다. 따라서 본 논문에서 제안한 통합 SNS 게이트웨이를 사용하게 될 시 SNS 서버에 사진을 요청할 때 서버에 요청되는 트래픽량을 줄일 수 있다.

5. 결론 및 향후 연구

본 논문에서는 통합 SNS 게이트웨이의 필요성에 대하여 설명하고 통합 SNS 게이트웨이의 구성, 웹 서비스 어댑터의 기능과 웹 서비스 어댑터가 통신하는 프로토콜을 제시하였다. 그리고 페이스북의 콘텐츠 요청 방법에 대해서 설명하고 페이스북과 통신할 때 게이트웨이의 유/무와 캐시의 유/무에 따른 성능을 실험할 수 있는 환경을 구성한 후 실험결과를 표와 그래프로 나타냈다. 페이스북의 게이트웨이 성능 비교를 위해 세 가지의 방법(콘텐츠를 요청할 때 게이트웨이 없이 직접 페이스북 통신하는 방법, 게이트웨이를 거쳐 캐시하지 않고 통신하는 방법, 게이트웨이를 거쳐 캐시되는 통신방법)으로 모바일 단말에서 페이스북 서버로 사진을 요청할 때 생성되는 세션별로 응답 되는 시간에 대한 성능 측정하고 게이트웨이를 사용할 때 응답속도와 트래픽량 관점에서 향상된 결과를 제시하며 통합 SNS 게이트웨이의 성능을 증명하였다.

향후 계획으로 통합 애플리케이션에서 여러 가지 SNS를 지원하기 위해서 현재 이 논문에서 이용한 캐시 방법과 페이스북 어댑터의 방법을 바탕으로 여러 SNS를 통합하기 위해서 트위터 웹 서비스 어댑터 연구를 진행할 계획이며, 글, 동영상 같이 캐시가 되지 않는 콘텐츠에 대해서 캐싱 하는 방안을 연구하여 성능을 측정할 것이다.

참 고 문 헌(Reference)

- [1] J. Im, J. Yu, S. Jang, M. Kim, J. Yu “2012 Final Report Survey of smartphone usage”, Korea Internet and Security Agency, Dec. 2012
- [2] I. Jung, H. Kim, H. Ju, “Design and Implement Web Service Adapter of Mobile Integrated SNS Gateway”, 2013 The Korean Institute of Communications and Information Sciences Summer Conference”, pp. 643,644, Jun. 2013
- [3] Y. Imai, Y. Sugiue, Y. Hori, Y. Iwamoto, “An Enhanced Application Gateway for some Web services to Personal Mobile Systems,” International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol.2, Nov. 2005
- [4] D. Valtchev, I. Frankov, “Service gateway architecture for a smart home”, Communications Magazine, IEEE, vol. 40, Issue 4, Apr. 2002
- [5] S. H Lee, I. S Jung, H. W Kim, H. T Ju, “The design of integrated mobile SNS gateway structure”, in Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific, Sept. 2013,
- [6] I. S Jung, H. W Kim, D. K Hong, H. T Ju, “Protocol Reverse Engineering to Facebook Messages”, 2013 4th International Conference on, Intelligent Systems Modelling & Simulation (ISMS), Jan. 2013
- [7] D. Crockford, Introducing Json(2006), Retrieved Aug., 4, 2012, from <http://www.json.org>
- [8] K. Maeda, “Performance Evaluation of Object Serialization Libraries in XML, JSON and Binary Formats,” 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), pp.177,182, May 2012
- [9] Facebook, Facebook APIs, 2013, from <https://developers.facebook.com/docs/reference/apis/>

● 저 자 소 개 ●



정 인 식(Insik Jung)

2012년 계명대학교 컴퓨터공학과(컴퓨터공학사)
2012년 9월~현재 계명대학교 컴퓨터공학과 석사과정
관심분야 : 네트워크 및 시스템 관리, 모바일 네트워크 관리
E-mail : dlstlrdl77@kmu.ac.kr



김 현 우(Hyeonwoo Kim)

2010년 계명대학교 컴퓨터공학과(컴퓨터공학사)
2012년 계명대학교 컴퓨터공학과(공학석사)
2012년 9월~현재 계명대학교 컴퓨터공학과 박사과정
관심분야 : 네트워크 및 시스템 관리, 방화벽, 네트워크 보안
E-mail : hwkim84@kmu.ac.kr



권 동 우(Dongwoo Kwon)

2010년 계명대학교 컴퓨터공학과(공학사)
2012년 계명대학교 컴퓨터공학과(공학석사)
2013년 9월~현재 계명대학교 컴퓨터공학과 박사과정
관심분야 : 인터넷 침입 예측, SDN 관리, 네트워크 관리
E-mail : dwkwon@kmu.ac.kr



주 흥 택(Hongtaek Ju)

1989년 한국과학기술원 전자계산학과 학사
1991년 포항공과대학교 컴퓨터공학과 석사
2002년 포항공과대학교 컴퓨터공학과 박사
2002년~현재 계명대학교 컴퓨터공학과 부교수
관심분야 : 네트워크 및 시스템 관리, Embedded 웹 서버를 이용한 네트워크 요소관리, SyncML,
인터넷 침입 예측, 네트워크 보안
E-mail : juht@kmu.ac.kr