



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

모바일 데이터 트래픽을 줄이기 위한
클라이언트 캐시 시스템 설계 및 구현



2012년 2월

부경대학교 대학원

시스템경영공학과

박지원

공학석사 학위논문

모바일 데이터 트래픽을 줄이기 위한 클라이언트
캐시 시스템 설계 및 구현

지도교수 옥 영 석

이 논문을 공학석사 학위논문으로 제출함.

2012년 2월

부경대학교 대학원

시스템경영공학과

박지원

박지원의 공학석사 학위논문을 인준함

2012년 02월 24일



주심 공학박사 구 평 회 인

위원 공학박사 김 영 진 인

위원 공학박사 옥 영 석 인

목 차

제 1장 서론	1
1.1 연구 배경 및 목적	1
제 2장 이론적 배경	7
2.1 모바일 플랫폼	7
2.2 캐시 시스템	9
2.2.1 서버측면 캐시 시스템	10
2.2.2 클라이언트측면 캐시 시스템	12
2.3 객체 풀링 메커니즘	13
제 3장 MACS 구축 방법론	16
3.1 데이터 트래픽 분석	16
3.2 데이터 트래픽 중요도 선정	18
3.3 모바일 환경 최적화	19
제 4장 프레임워크 개발	21
4.1 MACS의 디자인 및 구현	21
4.2 MACS의 캐시 구조	22
4.3 MACS의 풀링 구조	23

제 5장 실험 및 분석	25
5.1 버스정보 시스템에 MACS의 적용	25
5.2 무선망 사용 시 배터리 소모량 실험	27
5.3 MACS 프레임워크 성능 실험	29
제 6장 결론 및 향후과제	32
참고문헌	33
Abstract	36



그림 목차

[그림 1.1] 모바일 데이터 트래픽 증가 추세	1
[그림 1.2] 모바일 클라우드 구성도	3
[그림 1.3] 북미지역의 모바일 데이터 트래픽 구조	4
[그림 2.1] 모바일 플랫폼 구조	7
[그림 2.2] 월드와이드웹 캐시 시스템	9
[그림 2.3] 기본적인 캐시 서버의 동작 원리	10
[그림 2.4] 클라이언트 측면 캐시 시스템	12
[그림 2.5] 커넥션 라이프사이클	14
[그림 3.1] 모바일 애플리케이션의 성장률	16
[그림 3.2] 모바일 애플리케이션의 작동 실패 원인	17
[그림 3.3] 데이터 요청 방법론	18
[그림 4.1] MACS 아키텍처	21
[그림 4.2] HTTP Request Header 정보 예	22
[그림 5.1] BISB 아키텍처	25
[그림 5.3] 무선과 내부저장소 사용의 배터리 소모량 차이	27
[그림 5.4] 시간의 증가에 따른 배터리 감소율	29
[그림 5.5] 캐시 시스템과 비캐시 시스템의 성능 차이	30
[그림 5.6] 폴링 시스템과 비폴링 시스템의 차이	31

표 목차

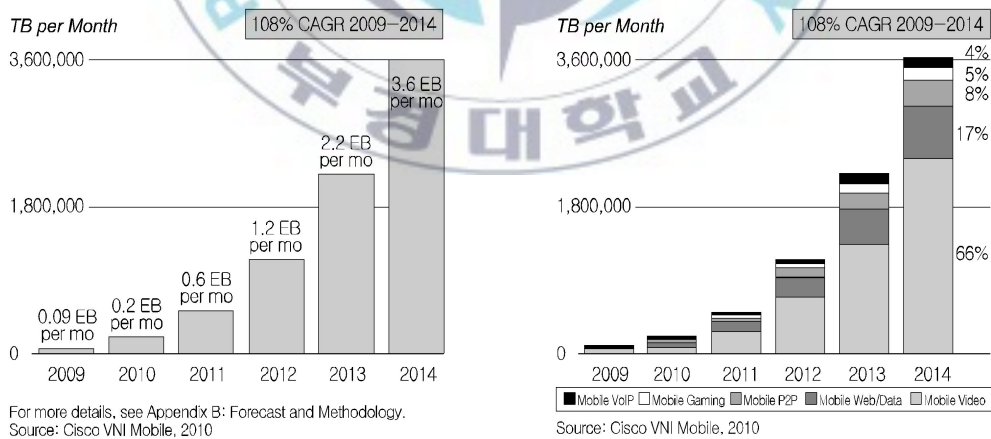
[표 2.1] 세계 스마트폰 OS 점유율	8
[표 2.2] 캐시 오브젝트 유효성 관련 항목	11
[표 5.1] 무선과 내부저장소 사용의 배터리 소모량 차이	28



제 1 장 서 론

1.1 연구 배경 및 목적

오늘날 모바일 환경의 전망에 의하면 데이터 트래픽은 기하급수적으로 증가하고 있다. 초기 음성과 간단한 문자 정보만 전송하던 모바일 통신은 동영상 서비스의 성장으로 인하여 가장 큰 트래픽을 점유하며 성장세가 가속화 될 것으로 예측되며 혁신적인 단말기와 서비스의 등장은 모바일 서비스의 성장을 더욱 더 가속화시킬 것이다.[1][2][3]



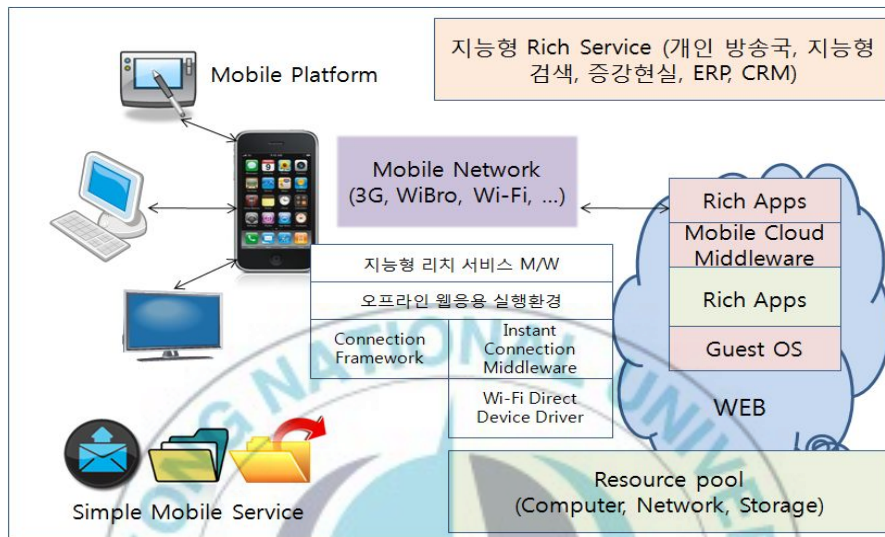
[그림 1.1] 모바일 데이터 트래픽 증가 추세

[그림 1.1]은 모바일 데이터 트래픽 증가 추세에 관한 자료이다.[1] [그림 1.1]에서 보듯 Cisco보고서에서 세계 모바일 트래픽 증가량은 매년 108%씩 폭발적으로 성장할 것이라 예측되고 있다. 또한 보고서는 국내 모바일(3G)시장이 2009년에서 2010년 사이 KT는 약 344%, SK telecom은 232% 그리고 LG는 114%의 트래픽 성장률을 나타냈으며 특히 KT는 2009년 대비 2012년에는 약 49배의 성장률을 보여 줄 것으로 예측하였다. 또한 4G망인 WIMAX나 LTE의 보급으로 인해 모바일 데이터 트래픽이 매년 약 40배씩 성장할 것으로 예측된다.[1][2]

무선인터넷이 급속히 성장함에 따라 이동통신업계는 태블릿PC, 노트북, 넷북 등의 단말을 통해 새로운 형태의 모바일 데이터전송 서비스를 제공하는 추세이다. 특히 배터리 기술의 발달과 경량화 등으로 인해 최근 개발되고 있는 PC류 등이 통신모듈 등을 내장하여 이동성을 가지게 되면서 모바일 망에 접속할 수 있는 기능을 가질 수 있거나 스마트폰이 지원하는 모듈 등을 이용하여 접속할 수 있는 등 이동전화뿐만 아니라 각종 기기들 역시 점차 모바일 망을 사용하기 시작하였다.[4]

모바일 망을 지원하는 각종 기기들이 증가되어 멀티단말기기 생태계가 구축됨에 따라 모바일 클라우드(Mobile Cloud)의 보급 역시 가속화 되고 있다. 모바일 클라우드란 스마트폰, 노트북, 넷북과 같은 휴대용 모바일 단말에서 애플리케이션을 구동할 때 단말 외부의 데이터 스토리지와 프로세싱을 활용하는 인프라를 말한

다.[6]



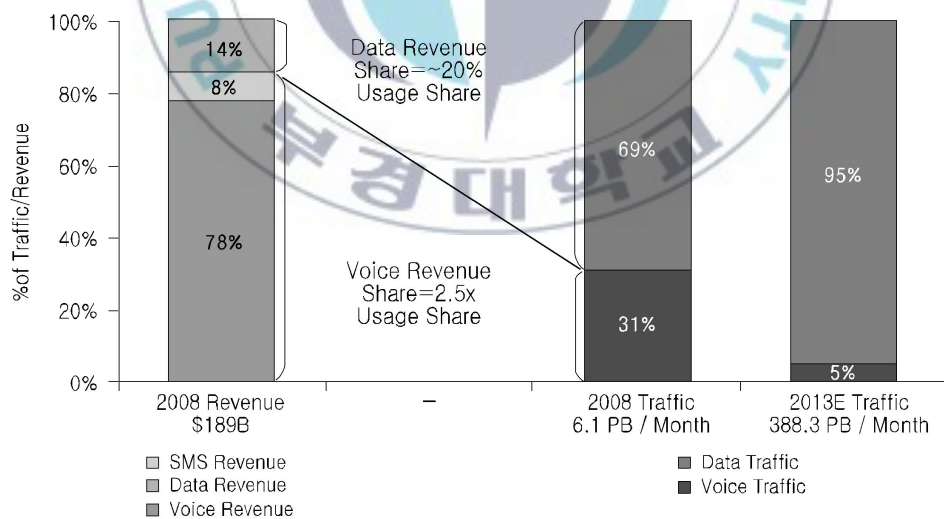
[그림 1.2] 모바일 클라우드 구성도

[그림 1.2]는 모바일 클라우드 구성도에 관한 것이다.[5] [그림 1.2]에서 보듯 앞으로 사용자들은 모바일 클라우드를 통해 고성능의 장치 없이도 대용량 자료 등을 공유하거나 다양한 서비스를 이용할 수 있는 등 그 잠재력이 크다. AT&T는 2010년 3월 일반 휴대폰에서도 스마트폰과 같은 데이터 관리 및 업로드 기능을 PC와 연동해 제공하는 클라우드 기반 서비스인 'Mobile Share'를 제공하고 있으며 Google은 메일에 있는 자료들을 클라우드 환경에서 읽거나 다양한 문서 파일 형태로 저장을 할 수 있도록 하는 기

능인 'Cloud Print'를 서비스하고 있다. 문서나 자료에 대한 것뿐만이 아니라 게임서비스도 모바일 클라우드를 지원하기 시작하였다.

일반적으로 게임을 하기 위해서는 게임을 실행시키기 위한 모듈을 다운로드 하여 실행 환경을 구축한 후에 사용자의 시스템 자원을 사용하여 실행되었지만 NHN Japan이 제공하는 GCLOUD는 서버의 자원을 사용하여 게임이 실행되며 사용자는 화면만 전송받아 이용할 수 있다.[7]

이런 모바일 망을 사용 하는 것에 따른 데이터 트래픽은 계속적으로 급증 하게 되어 사용자뿐만이 아니라 모바일 회선망 서비스 제공자에게 까지 그 피해를 미친다.



[그림 1.3] 북미지역의 모바일 데이터 트래픽 구조

[그림 1.3]은 북미지역의 모바일 데이터 트래픽 구조를 보여준 그림이다.[2] [그림 1.3]에서 보듯 2008년 발생한 데이터 트래픽 중에 31%만에 음성데이터였으며 이는 총 매출액의 86%를 차지하고 있다. 만약 현재의 매출 시스템으로 계속 간다면 앞으로 폭발적으로 증가하는 데이터 트래픽 성장률에 비춰 2013년 약 5%의 데이터 트래픽에서 발생하는 수익이 전체 매출액의 86%를 차지하게 될 것이다. 이러한 현상은 세계적으로도 유사하게 나타나는데 Chetan sharma consulting사의 2010년도 자료를 보면 전체 매출액에서 데이터 매출액 비중 현황이 균형점을 벗어나 있는 상태이다.[8]

그렇기 때문에 서비스 제공자들은 매출과 데이터 이용량의 불균형에 대응하기 위해 비용과 네트워크 효율성을 제고하면서 트래픽 증가에 적절히 대응하는 전략을 취할 것이다. 특정 트래픽의 차단 및 제한을 한다거나 요금제 개선, 우회망으로의 트래픽 분산, 망진화에 의한 네트워크 용량 확대, 등 각 서비스 제공자들이 처한 환경에 맞게 선택하여 대응할 것이다.[9]

물론 서비스 제공자가 가장 손쉽게 데이터 트래픽에 대한 수익을 창출 할 수 있는 것은 요금제 개선이다. 사용량에 따라 과금정책을 펼치거나 많은 데이터 트래픽을 발생시키는 기업에 대해 추가적인 비용을 청구하는 등의 정책을 펼칠 수 있지만 성장과 신규매출 기회의 대부분이 데이터 사용량에서 창출될 것이라는 점을 감안할 때, 데이터 트래픽을 과금에 의해 줄여 나가는 정책은 중장

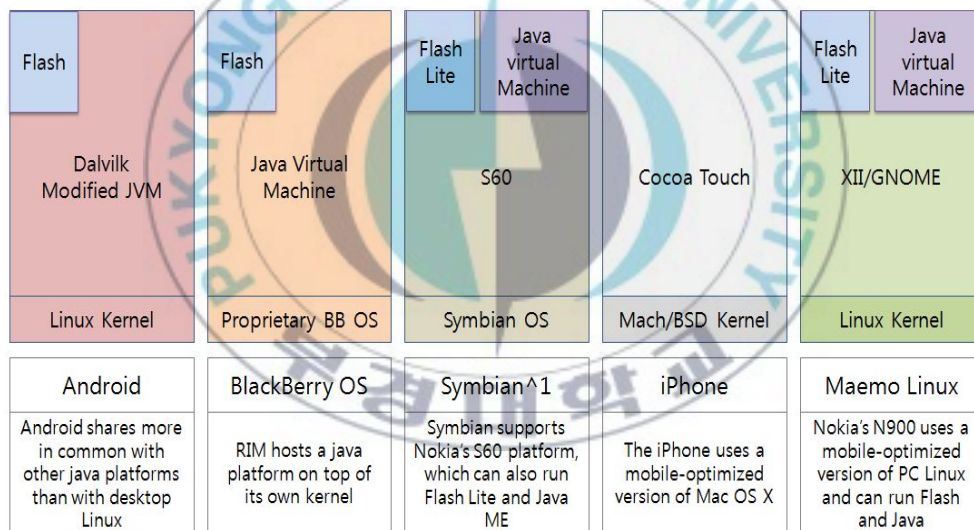
기적으로 발전을 저해하는 정책이라 할 수 있다. 그 외에 우회망 및 망진화 전략은 초기 설비 투자비용 및 서비스 사용 지역의 제한 등의 제약 때문에 결국 가장 저비용 고효율적인 전략인 사용자 데이터 트래픽 발생 제어 전략이 가장 근본적인 해결책이라 판단된다.

그러므로 본 논문에서는 사용자가 전송하는 데이터 트래픽을 분석하여 줄이고자 하는 트래픽의 종류를 파악하고 이를 해결하기 위해 기존 캐시시스템에 사용되는 커넥션 풀, 오브젝트 풀링 등을 모바일 환경에 맞게 적용한 MACS(Mobile Architecture for Client-side caching Service)를 개발하여 데이터 트래픽의 발생을 최소화 시키며 이로 인해 발생하는 망 구축 및 유지비용을 최소화하고 애플리케이션의 성능을 극대화 하는데 그 목적을 둔다.

제 2 장 이론적 배경

이 절에서는 본 논문에서 연구하고자 하는 문제에 대한 기존연구의 분석 및 문제에 대한 정의를 소개하고자 한다.

2.1 모바일 플랫폼



[그림 2.1] 모바일 플랫폼 구조

오늘날 모바일 시장은 치열한 모바일 플랫폼 전쟁 양상을 보여준다. 기존 마이크로소프트, 심비안 뿐만이 아니라 구글, 애플, 모토

를라, 삼성 등 인터넷 서비스, 디바이스 제조업체까지 모바일 플랫폼 시장에 뛰어들고 있다. 이런 모바일 플랫폼이 지원하는 환경은 음성통화, SMS 뿐만이 아니라 검색, 위치기반서비스, 인터넷 등 언제 어디서나 웹기반 서비스를 사용할 수 있도록 도와주는 등 복잡해지며 다양해지고 있다. [그림 2.1]은 모바일 플랫폼 구조에 관한 것을 보여준 것이다.[10] [그림 2.1]에서와 같이 모바일 플랫폼들은 리눅스나 자체적인 OS를 기반으로 사용하며 iPhone을 제외한 플랫폼에서 java를 사용할 수 있다. 하지만 여타 회사들 역시 자바 가상머신을 각자 다르게 구현했기 때문에 새로운 휴대폰이 나올 때마다 각 앱을 테스트해야하고 최적화도 따로 시켜야 하는 단점이 있다. 안드로이드는 이런 단점을 극복하여 Dalvik이라는 새로운 개발 플랫폼을 적용하여 만들어 단일화와 표준화를 이룬 자바 기술의 구현이라 볼 수 있다.

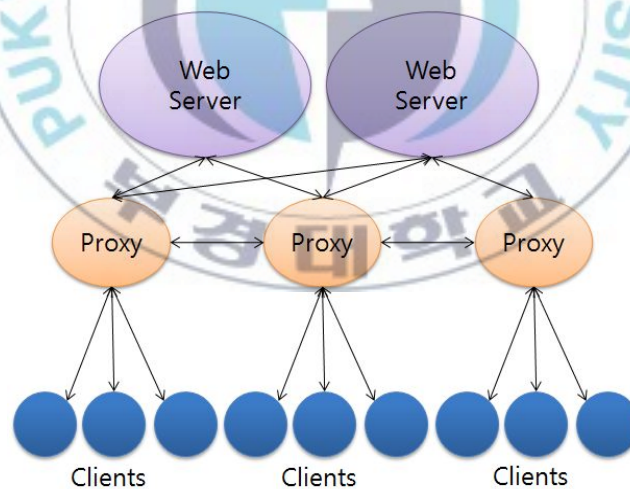
[표 2.1] 세계 스마트폰 OS 점유율

Company	1Q11 Thousands of Units	1Q11 Market Share (%)	1Q10 Thousands of Units	1Q10 Market Share (%)
Andorid	36,267,80	36	5,226.60	9.6
Symbian	27,598.50	27.4	24,067.70	44.2
iOS	16,883.20	16.8	8,359.70	15.3
RIM	13,004.00	12.9	10,752.50	19.7
Microsoft	3,658.70	3.6	3,696.20	6.8
Other OS	3,357.20	3.3	2,402.90	4.4
Total	100,769.30	100	54,505.50	100

[표 2.1]은 세계 스마트폰 OS 점유율을 나타낸 것이다.[10] [표 2.1]에서 보면 2010년 1분기에 비해 2011년 1분기에는 안드로이드의 폭발적인 성장력으로 인하여 세계 36%의 점유율 1위를 달성하였다. 이는 개방형 플랫폼과 범용성 등으로 인한 효과로 보이며 앞으로도 안드로이드가 점유율을 점차 늘려 2위와의 격차를 벌려 나갈 것이라 예측된다.[10]

본 논문에서는 범용성 및 개방성을 감안하여 안드로이드 모바일 플랫폼 환경에서 우선적으로 적용 한다.

2.2 캐시 시스템

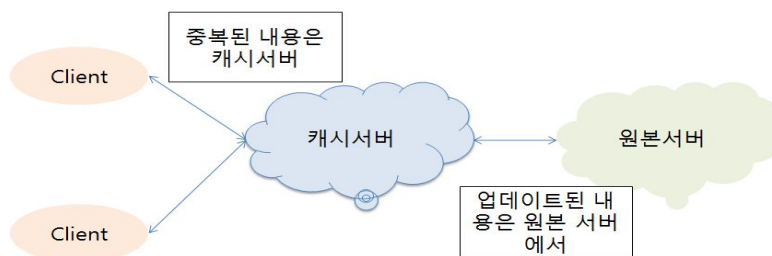


[그림 2.2] 월드와이드웹 캐시 시스템

캐시 시스템은 보통 인터넷 브라우저에서 한번 봤던 이미지 데이터나 여러 가지 정보 들을 로컬저장소의 임시 저장 폴더에 저장시켰다가 나중에 해당 페이지를 다시 읽어 들일 경우에 서버로부터 데이터를 다시 요청할 때에 일부 또는 전체의 데이터를 로컬저장소로부터 읽는 시스템이다. 로컬저장소를 사용함에 따라 서버에 대한 요청 횟수 감소 및 회선사용 비용 등을 감소시키는 효과를 얻을 수 있다. 캐시시스템은 크게 서버, 프록시(Proxy), 클라이언트 측면으로 나눌 수 있다.[11]

[그림 2.2]는 월드와이드웹(World Wide Web, WWW) 캐시 시스템에 관한 것이다.[11] [그림 2.2]에서와 같이 클라이언트의 요청을 받아 서버와 프록시가 서로 정보를 교환하며 데이터를 업데이트 시킨다는 측면에서 캐시 시스템은 볼 때 이를 다시 서버측면 캐시와 클라이언트측면 캐시 두 가지로 나눌 수 있다.

2.2.1 서버측면 캐시 시스템



[그림 2.3] 기본적인 캐시 서버의 동작 원리

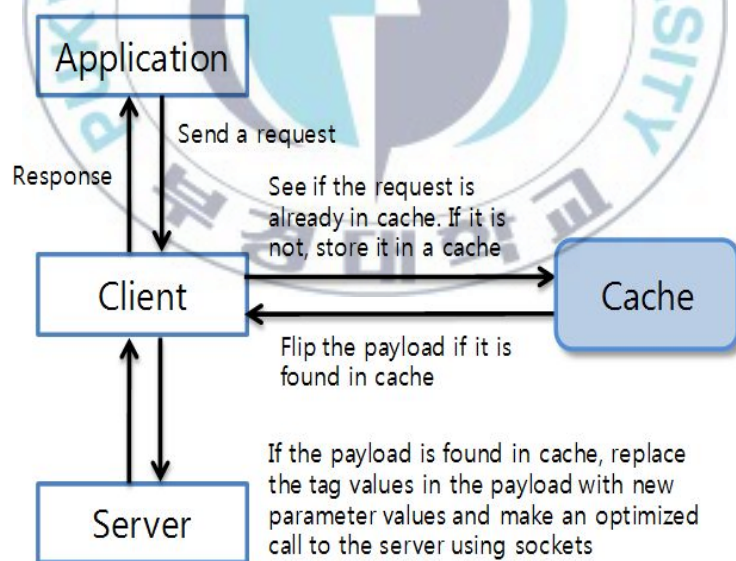
[그림 2.3]은 일반적인 캐시 서버의 동작 원리에 관한 것이다. [그림 2.3]과 같이 서버측면 캐시 시스템은 서버에서 사용되거나 반복 요청되어 지는 것들에 대해 캐시를 사용하여 정보를 프록시에 전송하며 이 때 프록시는 자주 사용되는 웹 페이지의 생명주기 또는 TTL(time-to-live)를 확인하여 복사본을 저장하여 웹 성능을 향상시키는데 도움을 준다.[12]

[표 2.2] 캐시 오브젝트 유효성 관련 항목

항목	내용	비고
마지막 변경 사항	해당 콘텐츠의 마지막 변경 시간을 기록	G M T 기준
E-Tag	서버에서 생성되는 유일한 인자로 오브젝트의 중복성과 캐시 유효성 여부를 확인할 수 있는 항목	HTTP 1.1
캐시-컨트롤 (Cache-Control)	해당 오브젝트에 대한 캐시 관련 정책을 제어하는 항목 (max-age, no-cache, must-revalidate 등이 있음)	HTTP 1.1
Age	해당 오브젝트가 갱신된 시점을 기준으로 초기화 되며 새롭게 갱신되는 시점까지 1 초 단위로 증가함	
Expires	해당 오브젝트가 얼마동안 유효한지를 표시해주는 항목으로 명시된 시간 이후 캐시 유효성 재확인	G M T 기준

[표 2.2]는 캐시 오브젝트 유효성 관련 항목에 관한 것이다.[13]
 [표 2.2]에서 보는 내용들은 일반적인 웹 페이지 구성 중 캐시와 관련된 것이며 이외에도 다양한 항목들이 오브젝트 유효성 검사를 위한 캐시 정책에 영향을 미친다. 웹 서비스는 연결 기반 서비스이므로 사용자와 서버간 연결 관리를 어떻게 하느냐에 따라 성능이 배로 차이가 날 수 있다. 데이터의 재사용 성을 높이거나 불필요한 응답시간을 배제 및 트래픽 할당량을 조절하는 등을 통해 성능을 확연하게 달리 할 수 있다.[12][13]

2.2.2 클라이언트 측면 캐시 시스템



[그림 2.4] 클라이언트 측면 캐시 시스템

[그림 2.4]는 클라이언트 측면 캐시 시스템에 대한 예를 보여준다.[14] 클라이언트 측면 캐시 시스템은 서버로부터 받은 데이터를 로컬저장소에 임시로 저장을 하였다가 재요청시 서버에 요청하지 않고 로컬저장소에 있는 데이터를 읽어 들이는 역할을 한다. 로컬저장소를 사용하기 때문에 클라이언트 측면 캐시 시스템은 서버에 직접 요청하는 것보다 많은 이점을 얻을 수 있다. 서버에 요청을 작업을 하지 않아도 되므로 상대적으로 적은 CPU자원을 사용하며 전력소비를 감소시키고, 네트워크 접속 등으로 인한 지연현상 등을 고려하지 않아도 된다. 이러한 장점 때문에 클라이언트 측면 캐시 시스템을 사용하면 애플리케이션의 성능을 보다 더 효율적이고 빠르게 향상 시킬 수 있다.

2.3 객체 풀링 메커니즘

객체 풀링이란 자원들을 미리 사용 준비를 시켰다가 해당 자원을 요청할 때 풀에서 이를 꺼내어 사용하고 사용이 끝나면 이를 다시 반환하거나 종료시키는 방법론을 말한다. 객체의 생성을 미리 시켜놓았다가 이를 꺼내어 사용하게 되면 객체 생성 시간이 소모되지 않기 때문에 매우 빠른 속도로 작업을 할 수 있다.

[그림 2.5] Connection Lifecycle에 대한 메커니즘을 나타낸 것이다.[15] [그림 2.5]의 기반이 되는 메커니즘이 객체 풀링 메커니즘이다. 커넥션 풀은 데이터베이스에 일정량의 커넥션을 미리

만들어 사용자가 커넥션을 요청할 때 이를 빠르게 제공하며 커넥션의 생성 수를 미리 설정하여 과도한 커넥션의 수가 만들어져서 데이터베이스에 부하를 주는 현상을 사전에 차단한다.[11]



[그림 2.5] 커넥션 라이프사이클

네트워크 커넥션 역시 객체 풀링 메커니즘을 응용하게 되면 네트워크를 접속하거나 종료할 때마다 사용되는 배터리 소모와 객체 생성에 대한 비용 감소 및 빠른 접속 등의 효과를 얻을 수 있다.[16][17]

본 논문에서는 MACS모듈을 만들어 앞서 설명한 캐시 측면 시스템과 객체 풀링 메커니즘을 활용하여 서버 측에 대한 요청을 최소

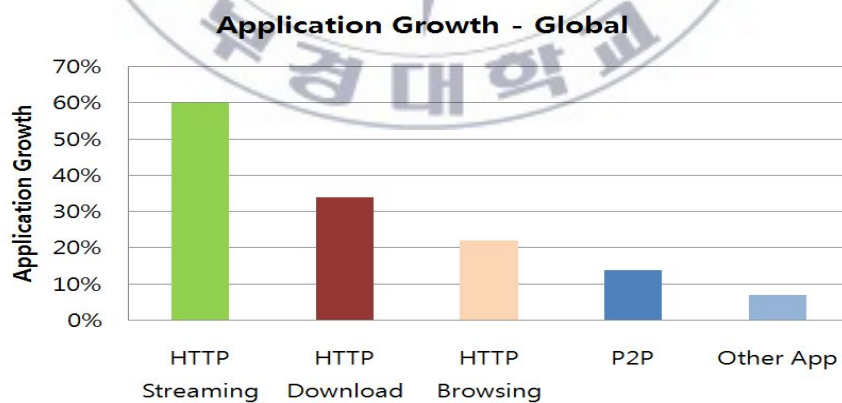
화 시켜 배터리 소모 감소 및 빠른 애플리케이션 속도 등을 구현
해 나갈 것이다.



제 3 장 MACS 구축 방법론

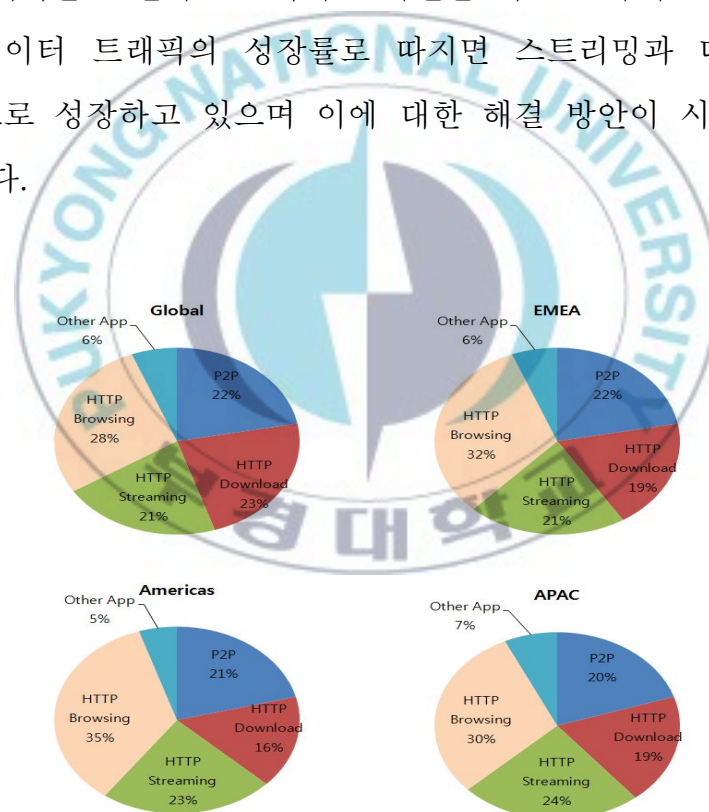
데이터 트래픽 발생량을 조절하기 위한 MACS구축 방법론은 트래픽 분석, 중요도 선정, 최적화로 구성 된다. 데이터 트래픽 분석은 전체 전송되는 트래픽의 성격 중에 어떤 데이터들로 구성되어 있는지를 분석한다. 그리고 분석된 데이터 중에서 데이터의 성격을 파악하여 중요성을 선정 후에 선정된 데이터를 분류하여 TTL을 적용한다. 마지막으로 모바일 환경에 최적화된 프레임워크를 설계한다.

3.1 데이터 트래픽 분석



[그림 3.1] 2011년 세계 모바일 애플리케이션의 성장률

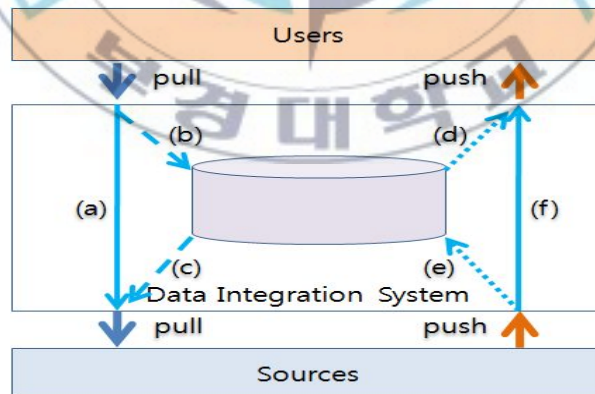
[그림 3.1]은 2011년 세계 모바일 애플리케이션의 성장률을 나타낸 것이다.[18] 그림에서 보듯이 트래픽은 HTTP 스트리밍, HTTP 파일다운로드, HTTP 브라우징, P2P, 그리고 기타 등으로 이루어져 있다. HTTP 스트리밍은 동영상등을 재생하는 부분이며 HTTP Download는 웹상의 파일들을 다운로드 받는 것에 해당된다. HTTP Browsing은 스트리밍과 다운로드를 제외한 HTTP 데이터 트래픽을 말한다. 그 외에는 파일을 서로 공유하는 P2P 등이 있다. 데이터 트래픽의 성장률로 따지면 스트리밍과 다운로드가 폭발적으로 성장하고 있으며 이에 대한 해결 방안이 시급한 것으로 보인다.



[그림 3.2] 모바일 애플리케이션의 작동 실패 원인

[그림 3.2]는 모바일 애플리케이션의 작동 실패 원인에 대해 나타낸 것으로 [그림 3.2]에서 보면 모바일 애플리케이션의 작동 오류를 발생시키는 주요 원인은 HTTP Browsing시에 발생된다.[19] 이는 데이터 스트리밍이나 파일 다운로드를 제공하는 업체의 경우 대용량의 전송을 처리 할 수 있는 기술 등이 이미 갖추어져 있지만 중소기업 또는 일반적인 웹 정보 제공 업체의 경우 폭발적인 모바일 수요를 미처 예측하지 못한 원인과 많은 수의 커넥션을 처리해야 하는 부담감 등을 예측할 수 있다. 그러므로 본 논문에서는 HTTP Browsing의 문제를 우선적으로 해결하도록 데이터 트래픽을 분석한다.

3.2 데이터 트래픽 중요도 선정



[그림 3.3] 데이터 요청 방법론

[그림 3.3]은 데이터 요청 방법론에 관한 것이다.[20] 사용자는 데이터를 (a)와 (f)처럼 바로 데이터를 전송 및 수신 할 수 있지만 (b),(c),(d),(e)처럼 데이터를 별도의 저장소에서 가져올 수도 있다. 데이터를 갱신해야 할지 아니면 별도의 저장소에서 가져와야 할지에 대해 결정해 주는 것이 데이터 신선도이다.[20]

사용자 또는 프로그램은 데이터 신선도를 판단하여 정보를 외부에서 가져올 것인지 아니면 데이터 저장소에서 가져올 것인지를 결정한다.

3.3 모바일 환경에 최적화

모바일환경은 일반적인 PC의 작동 환경과 다르다. 가장 큰 차이점은 전력 소모량이다. 대부분의 모바일 장치는 배터리를 사용하여 작동하며 PC는 전력 케이블을 사용하여 작동된다. 그렇기 때문에 모바일 환경에서의 소프트웨어 설계는 최소한의 자원을 소모하도록 설계하여야 한다. 사용자에게 애플리케이션을 얼마나 오랫동안 사용할 수 있는냐는 결국 배터리 수명과 관계되어 있기 때문에 개발자는 항상 배터리 소모를 최소화 할 수 있도록 이를 고려해서 개발 하여야 한다.[21]

배터리 소모를 최소화 하도록 개발하기 위해서는 첫째, 불필요한 객체의 생성을 피하도록 한다. 객체를 생성하는 것은 결코 자원을 공짜로 사용해서 생성하는 것은 아니다. 객체를 한번 만들 때마다

자원은 소모되며 심지어 개발자가 문자를 입력받는 함수를 생성할 때에도 자원은 소모된다. 예를 들어 PC에서 개발을 하는 경우에는 String을 사용하여 문자열을 계속적으로 입력 받더라도 전력 자원 소모량에 대한 제한이 없지만 모바일 환경에서 String을 사용한 문자 객체를 생성하는 것은 많은 자원 소모를 발생시킨다. 그래서 String이 아닌 StringBuffer 같이 문자열을 추가 할 수 있는 메소드를 사용하여 자원 사용을 최소화 할 수 있도록 개발 하여야 한다.

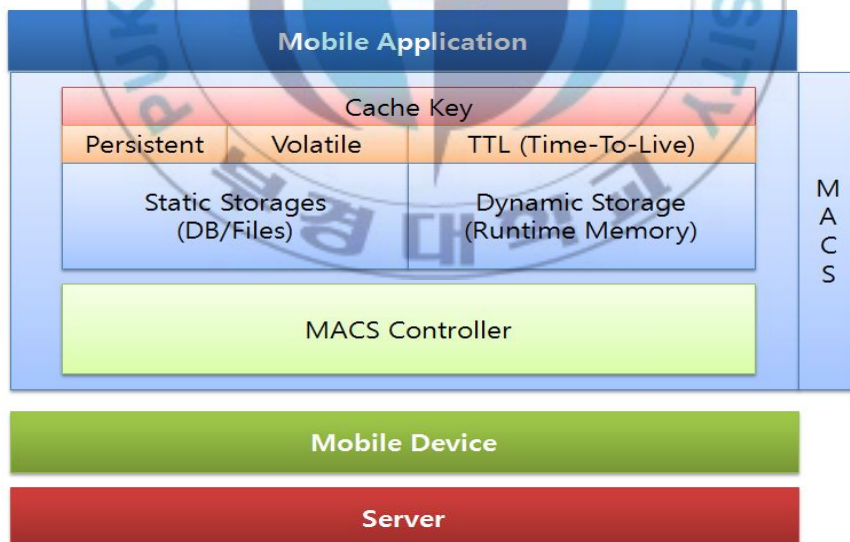
둘째, 객체 필드가 아니라면 메소드를 정적(Static)으로 선언하여 이를 사용한다. 메소드를 정적으로 선언하게 되면 미리 메모리상에 생성되어 있기 때문에 보다 빠르게 불러 올 수 있다.

셋째, Getters/Setters 사용을 피해야 한다. Getters/Setters가 포함된 DTO(Data Transfer Object)와 같은 객체를 만들어 데이터를 전송하는 기법은 일반적으로 네트워크 부하를 줄이고 응답 시간 향상을 얻을 수 있지만 모바일 환경에서 DTO를 생성하는 것은 추가 객체를 생성하는 것이므로 많은 자원을 소모하게 된다. 그러므로 객체 생성을 최대한 피하는 다이렉트 접근법으로 설계를 하여 모바일 자원 사용을 최소화 하여야 한다.

제 4 장 프레임워크 개발

4.1 MACS의 디자인 및 구현

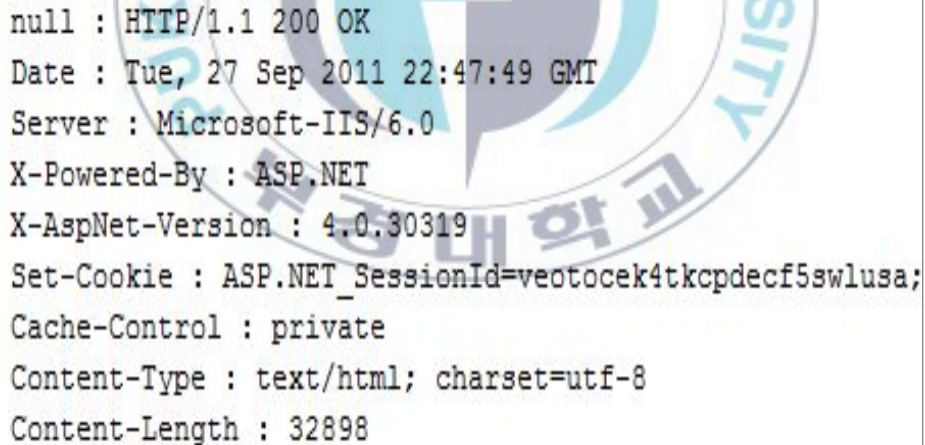
[그림 4.1]은 MACS의 아키텍처이다. MACS Controller가 캐시 시스템과 풀링 시스템을 함께 관리한다. 모바일 애플리케이션은 Cache Key를 통하여 필요한 자료들이나 커넥션 객체들을 가져온다. 커넥션 같은 객체들은 메모리 같은 동적인 저장소를 사용하여 이를 저장하며 TTL값을 이용하여 자동적으로 이를 종료시킨다.



[그림 4.1] MACS 아키텍처

4.2 MACS의 캐시 구조

MACS는 라이브러리로 구현되어 있기 때문에 원하는 애플리케이션에 라이브러리를 추가하면 애플리케이션 내에서 기능을 사용할 수 있다. 언제든지 애플리케이션에서 정보를 요청하면 MACS Controller가 이를 체크한다. 만약 해당 정보에 대한 TTL가 초과하였거나 캐시 안에 데이터가 없으면 이를 서버에 요청하게 되며 서버로부터 데이터를 전송받아 이를 캐시 안에 저장하고 정보를 전송하게 된다. MACS Controller는 Cache Key정보를 확인해서 정보의 유용성을 확인한다.



```
null : HTTP/1.1 200 OK
Date : Tue, 27 Sep 2011 22:47:49 GMT
Server : Microsoft-IIS/6.0
X-Powered-By : ASP.NET
X-AspNet-Version : 4.0.30319
Set-Cookie : ASP.NET_SessionId=veotocek4tkcpdecf5swlusa;
Cache-Control : private
Content-Type : text/html; charset=utf-8
Content-Length : 32898
```

[그림 4.2] HTTP Request Header정보 예

이때 사용되는 Cache Key는 HTTP Request Header정보도 사용할 수 있다. [그림 4.2]는 HTTP Request Header정보에 관한 예를 나타낸 것으로 [그림 4.2]와 같은 정보들이 Cache Key로 사용 될 수 있다. Header정보에 포함되는 것은 해당 정보에 대한 전송 완료 여부, 시간정보, 서버정보, 버전정보, 타입 및 언어 정보 등이 포함되는데 시간에 대한 정보 등을 활용하여 캐시정보를 사용할 수 있다. 캐시에 대한 정보들은 MACS의 저장소에 저장된다. 저장소는 두 가지 타입이 있는데 하나는 정적인 저장소이며 다른 하나는 동적인 저장소이다. 정적인 저장소는 휘발성과 비휘발성으로 구분할 수 있는데, 애플리케이션이 실행중일 때에만 필요한 정보의 경우 휘발성에 저장하며 애플리케이션이 종료되어도 여전히 유지해야할 필요성이 있는 사용자 설정 정보 같은 것들은 비휘발성에 저장한다. 동적인 저장소는 업데이트 빈도가 짧으며 특정 시간마다 새로 갱신해 줘야 할 필요성이 있는 정보를 저장한다.

4.3 MACS의 폴링 구조

모바일 환경에서의 폴링구조는 일반적으로 사용하는 것과 다르게 디자인 되어야 한다. 모바일 환경이라는 특수성을 고려하여 제한된 모바일 자원을 최소한으로 소모하도록 이를 고려해서 설계하여야 하기 때문이다. 설계 방법에 따라 배터리의 소모량 등 성능

에 큰 영향을 미치기 때문이다.[22] MACS 프레임워크에서는 폴
에서 객체를 가져온 다음에 장시간 사용하지 않을 경우 객체를 자
동적으로 파괴하도록 구조를 설계한다. 해당 객체를 파괴하지 않
는다면 지속적으로 이를 유지하기 위해 모바일 장치 내에서 메모
리 및 커넥션에 필요한 자원들을 지속적으로 소모하기 때문이다.
객체 풀의 크기는 MACS Controller에서 설정되어 관리한다.

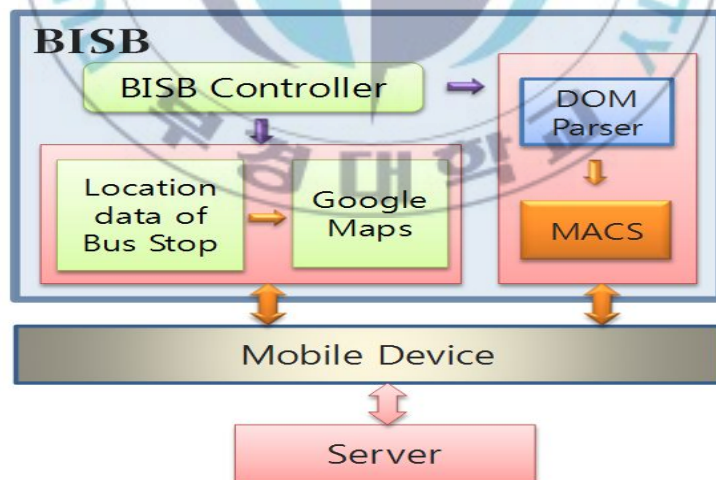
해당 풀링에 대한 사용 설정은 개발자가 해당 애플리케이션의 사
용목적에 맞게 변형하여 사용하여야 한다.



제 5 장 실험 및 분석

5.1 버스정보 시스템에 MACS 적용

BIS(Bus Information System)이란 버스의 운행정보를 실시간으로 제공하는 첨단 교통 시스템으로 각 버스에 위성항법장치(GPS)를 설치하여 인공위성과 연결시켜, 운행상황을 교통정보 센터로 보내면 이를 웹페이지 등을 통하여 정보를 표시해주는 시스템이다.[23] 이용자들은 버스의 위치나 정류장정보, 도착예상시간 등을 얻을 수 있으며 이 시스템은 버스 내부에도 설치하여 배차간격을 조절하는 등에 사용되기도 한다.

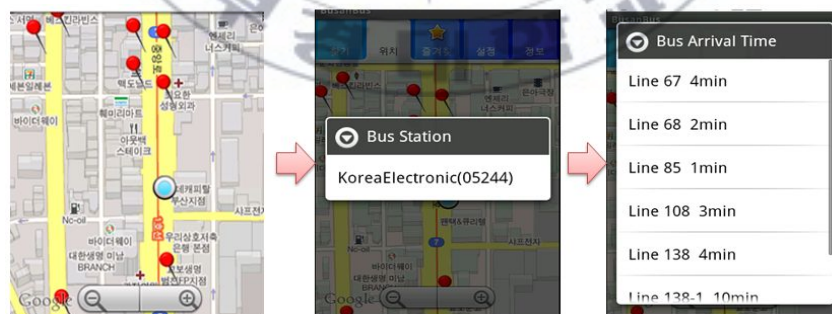


[그림 5.1] BISB 아키텍처

BISB(Bus Information System for Busan city)는 부산에 거주하는 모바일 사용자를 위해 개발되었다. 본 논문에서는 MACS모듈을 BISB에 설치하여 클라이언트 측면 캐시 시스템과 폴링 시스템을 테스트할 것이다. BISB는 Android v1.6버전을 사용하여 개발하였으며 테스트는 G1개발자 핸드폰을 사용하였다.

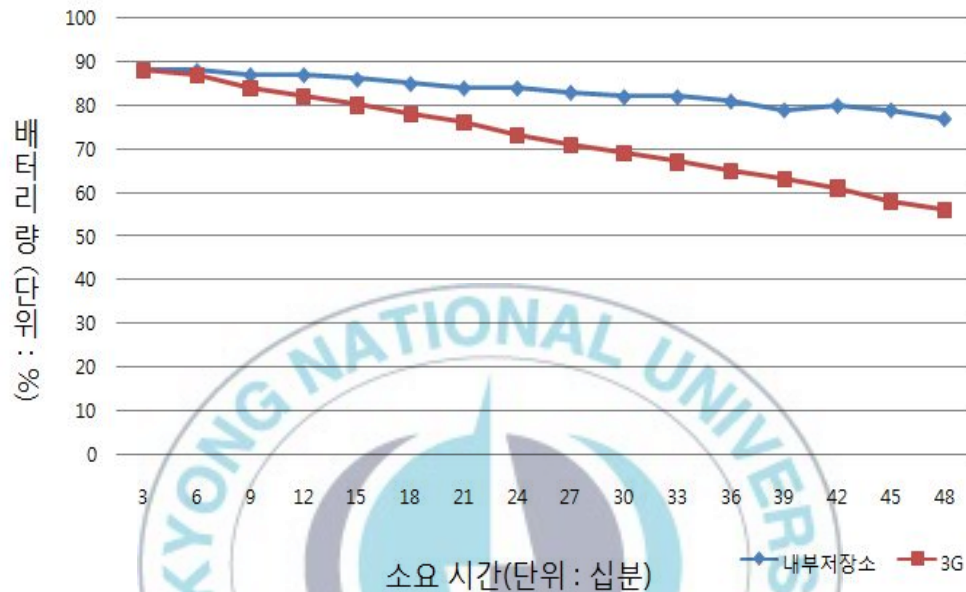
[그림 5.1]은 BISB아키텍처에 관한 설명이다. 그림에서 보듯이 BISB Controller를 통해 버스정류장, 지도, 도착시간 등 애플리케이션에서 필요한 정보들을 가져온다. 서버에서 제공하는 버스 도착 시간들은 MACS 프레임워크를 통해 읽어 들여 지며 읽은 정보들은 DOM Parser를 통해 필요한 정보를 필터링 하여 얻는다.

[그림 5.2]는 BISB 애플리케이션 실행화면에 관한 화면이다. [그림 5.2]에서 보듯 사용자가 지도상의 버스 정류장의 정보를 선택하여 조회를 하면 해당 버스정류장에 대한 정보들이 나오게 된다.



[그림 5.2] BISB 애플리케이션 실행화면

5.2 무선망 사용 시 배터리 소모량 실험



[그림 5.3] 무선과 내부저장소 사용의 배터리 소모량 차이

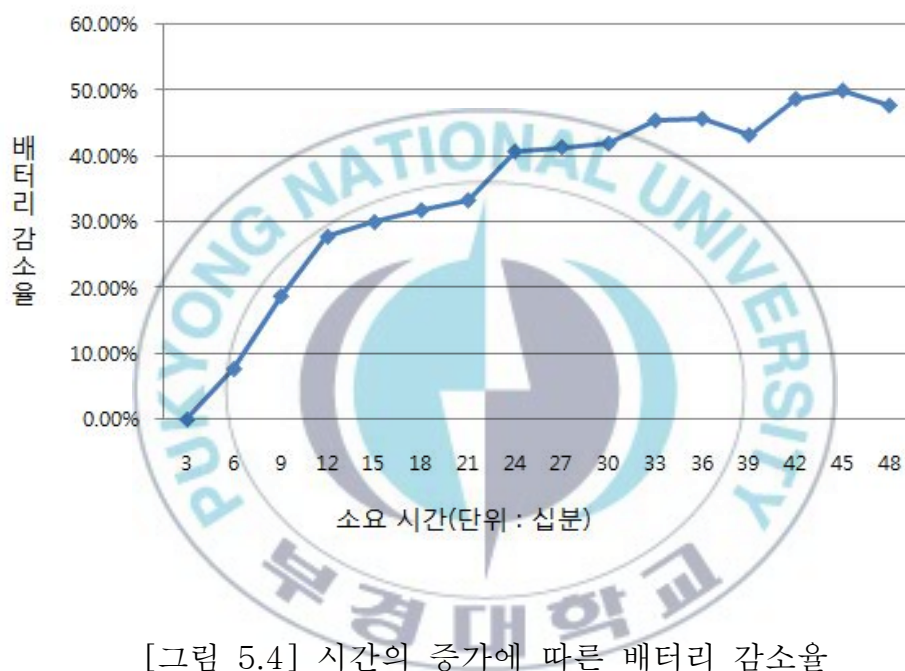
[그림 5.3]는 무선을 사용할 때와 메모리를 사용할 때의 배터리 소모량의 차이에 관한 결과이다. 해당 데이터는 Android에서 제공하는 `Android.os.BatteryManager` 메소드를 이용하여 배터리 결과 값을 `Sqlite Database`에 입력하였다가 이를 추출하여 측정하였으며 측정 데이터의 종류는 약 80KB용량의 <http://www.naver.com> (네이버) 데이터를 사용하였다. 실험 방법은 네이버 홈페이지 데이터를 내부저장소를 이용한 호출하는 방법과 이를 3G를 이용하여 데이터를 호출하는 방식을 선택하였다.

[표 5.1 무선과 내부저장소 사용의 배터리 소모량 차이]

소요시간 (단위: 십분)	내부저장소 사용 시 배터리 소모량(A)	3G 사용 시 배터리 소모량(B)	감소율 $((A-B) \times 100 / B)$
3	12%	12%	0.00%
6	12%	13%	7.69%
9	13%	16%	18.75%
12	13%	18%	27.78%
15	14%	20%	30.00%
18	15%	22%	31.82%
21	16%	24%	33.33%
24	16%	27%	40.74%
27	17%	29%	41.38%
30	18%	31%	41.94%
33	18%	33%	45.45%
36	19%	35%	45.71%
39	21%	37%	43.24%
42	20%	39%	48.72%
45	21%	42%	50.00%
48	23%	44%	47.73%

[표 5.1]은 무선과 내부저장소 사용의 배터리 소모량 차이를 나타낸 것이며 위의 감소율을 보다 쉽게 파악하기 위해 그림으로 나

타낸 것이 [그림 5.4]이다. [그림 5.4]는 시간의 증가에 따른 배터리 감소율을 그림으로 보여준다. [표 5.1]과 [그림 5.4]에서 알 수 있듯이 3G망을 사용 할 경우 시간이 흐를수록 배터리 감소율의 차이가 최대 50%에 가까워진다는 것을 알 수 있었다.

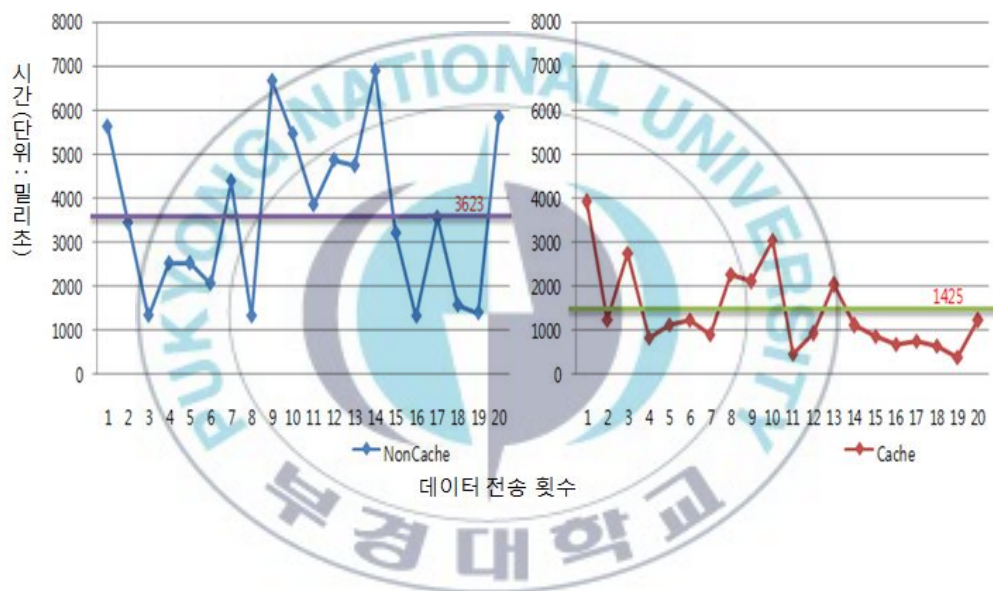


[그림 5.4] 시간의 증가에 따른 배터리 감소율

위와 같은 결과를 통해 내부저장소를 사용하는 것이 3G망을 사용하는 것에 비해 배터리 소모량이 적다는 것을 알 수 있었으며, 모바일 프로그램 설계를 할 경우에 3G망사용을 최소화 하도록 설계하여야 한다는 것을 알 수 있었다.

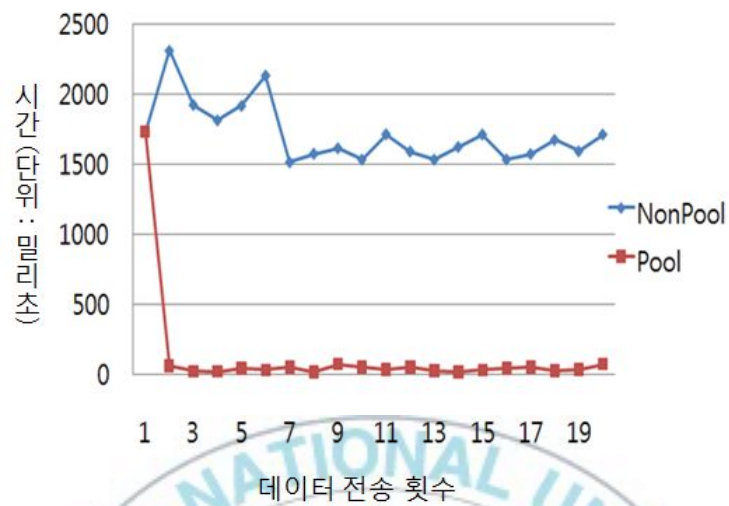
5.3 MACS 프레임워크 성능 실험

실험 정보는 다음과 같이 설정하였다. BISB에서 모든 버스 도착 정보가 관리된다. 버스 도착 시간이 10분 이상인 도착 정보에 대해서는 TTL을 2분으로 설정하였으며 도착시간이 4~10분인 경우는 TTL을 1분으로 설정하였다. 만약 도착시간이 4분이하일 경우에는 TTL을 0으로 설정하여 항상 데이터를 새로 가져오도록 설정 하였다.



[그림 5.5] 캐시 시스템과 비캐시 시스템의 차이

캐시 시스템을 사용할 경우와 비 캐시 시스템을 사용할 경우의 차이는 [그림 5.5]과 같다. 위 실험결과에서 보듯이 약 2.54배 캐시를 사용한 애플리케이션이 비 캐시 애플리케이션에 비하여 빠르다는 것을 알 수 있다.



[그림 5.6] 풀링 시스템과 비풀링 시스템의 차이

[그림 5.6]는 BISB애플리케이션에서 MACS 프레임워크 상의 풀링 시스템을 사용한 것과 사용하지 않은 것의 차이를 보여준다. 실험 결과에서 보듯 약 14.4배의 성능 개선 효과를 얻을 수 있었다.

제 6 장 결론 및 향후과제

한국뿐만이 아니라 세계 여러 나라들의 모바일 데이터 트래픽 증가는 기술의 발전과 더불어 해마다 폭발적으로 증가하고 있다. 회선망이 증설되고 전송량이 늘어날수록 고용량의 데이터의 전송에 대한 진입 장벽이 점차 사라져 데이터 트래픽 초과에 대한 망 부하는 계속될 것으로 전망된다. 이미 국내 통신3사는 천문학적인 비용을 망증설을 통한 데이터 트래픽 처리에 지불하고 있다.

본 논문에서 MACS를 통하여 모바일 환경에서 캐시 와 풀링 시스템에 대해 제안하였다. 불필요한 트래픽을 사용자 측에서 걸러내어 MACS를 적용한 BISB애플리케이션의 결과에서 보듯 서버에 대한 데이터 요청 등을 감소시켜 이로 인해 모바일 데이터 트래픽 발생 양을 최소화 및 배터리 사용량을 증가 시키는 등의 효과를 얻었다.

향후에는 사용자 행동 패턴을 연구하며 트래픽을 좀 더 세분화시켜 사용자가 일일이 TTL을 설정하는 불편함을 최소화 시키는 연구가 진행되어야 할 것이다.

참고문헌

- [1] Cisco Systems, Visual networking Index: Global Mobile Data Traffic Forecast Update 2010-2015, Cisco Research Center, 2011.2.
- [2] Morgan Stanley & Co, The mobile internet report, Morgan Stanley Research, 2009.12.
- [3] R. Beale, Supporting social interaction with smart phones, IEEE Pervasive Computing, pp35-41, 2005.4.
- [4] Jennifer G Sheridan, Michael Rohs, Rafael Ballagas, The smart phone: a ubiquitous input device, IEEE pervasive Computing. pp.70-77, 2006.1.
- [5] 김학영, 남궁한, 민옥기, 모바일 클라우드 기술 동향, 전자통신 동향분석 제25권 제3호, 2010.6.
- [6] 강부미, 구경철, 박정환, 백종현, 장종표, 전덕중, 전철기, 진병문, 진수경, 정보통신표준기술력향상사업 ICT 중점기술 표준화전략 맵 종합보고서 5, 한국정보통신기술협회, 2011.1.
- [7] <http://www.hatena.co.kr/1061>
- [8] Chetan sharma consulting(2010), Managing growth and profits in the yotta-byte era, 2010.6

- [9] 임동민, 전수연, 모바일 트래픽 증가에 대한 이동통신사업자의 대응동향, 정보통신정책연구원, 2010.9
- [10] http://www.appleinsider.com/articles/09/11/05/inside_googles_android_and_apples_iphone_os_as_core_platforms.html
- [11] J. Wang, A Survey of web caching schemes for the Internet, ACM SIGCOMM Computer Communication Review, 1999.
- [12] J. Gwertzman, M. Seltzer, World Wide Web Cache Consistency, USENIX, 1996.1.
- [13] <http://joridari.tistory.com/18>
- [14] D. Andresen, K. Devaram, SOAP optimization via client side caching, Proceedings of the ICWS 2003.
- [15] D. Erickson, M. Modjeski, S. Lauzon, WebSphere connection pooling, IBM WebSphere Application Server User Guide, 2001.
- [16] C. K. Toh, Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks, IEEE Communications Magazine, pp.138-147, June, 2001.
- [17] J. M. Rulnick, N. Bambos, Mobile power management for maximum battery life in wireless communication networks, INFOCOM 1996, Fifteenth Annual Joint Conference of the IEEE Computer Societies. vol.2, pp.443-450, 1996.

- [18] A. Gupta, A. Sata, A. Zimmermann, CK Lu, C. Milanesi, H. J. Vergne, R. Cozza, T. H. Nguyen, Market Share Analysis: Mobile Devices, Worldwide, 1Q11, Gartner, 2011.5.
- [19] Informa UK Ltd, Mobile Internet Traffic: Analysing Global Usage Trends, Informa telecoms & media, 2010.8.
- [20] Veronika Peralta, Data Freshness and Data Accuracy: A State of the Art, CSIC ,2006.3.
- [21] <http://developer.android.com/guide/practices/design/performance.html>
- [22] J.W. Park, M.S. Kim, Design and Implementation of Client-Side Caching Service for Mobile Device, ICIC International 2011, vol2, pp301-306, 2011.4.
- [23] E.S. Yoo, H.K. Choi, H.K. Lee, A Study on the BIS(Bus Information System) choice behaviors using stated preference data, Proceedings of the Eastern Asia Society for Transportation Studies, vol.5, pp.325-338, 2005.

Design and Implementation of Client-Side Caching System for Mobile Traffic Environment

Jiwon Park

Department of System Management Engineering, Graduate
School, Pukyong National University

ABSTRACT

Mobile devices are nowadays increasingly used to access the Internet as well as wired devices. It is partly because the use of mobile device makes it easy to access the Internet from anywhere. Now the traffic data caused by mobile devices are not so negligible compared to that of wired connection. Increased traffic of mobile access causes traffic congestion and extra costs for service provider to maintain proper service quality. Service provider cannot always increase the Internet bandwidth and afford additional servers to meet high-level of user demand. So there should be a way to control user traffics while minimizing the impact on service quality. To handle this problem, we utilized users' repetitive access pattern for the same web

resources. If we can provide a reuse mechanism for the same resources at the client-side, then we can diminish traffics on the server by reducing client requests and can obtain faster responsiveness of client device as well. In this paper, we design a client-side caching and pooling module which is called MACS (Mobile Architecture for Client-side caching Service). By applying our MACS module to BISB (Bus Information System for Busan), we can reduce server-side traffics, and obtain high client-side responsiveness without degrading service quality.

keyword : Mobile Caching, Data Traffic, Client-side



감사의 글

먼저 대학원에 진학하여 공부하는 동안 항상 격려해주신 부모님께 감사드립니다. 또한 학부생활 및 대학원 생활동안 많은 지도를 해주신 여러 교수님들께 감사드립니다. 부족한게 정말 많은 저인데도 불구하고 많은 지도와 격려를 해주신 여러 교수님들 덕택에 무사히 논문을 작성하게 되었습니다.

논문을 작성하면서 가장 많은 도움을 주신 김민수 교수님 그리고 미국에 있으신 김민수 교수님을 대신해 논문을 전반적으로 다듬어 주신 옥영석 교수님과 논문의 부족한 부분을 보완하도록 지도해주신 김영진 교수님, 구평희 교수님 많은 교수님들 덕택에 논문이 완성되었습니다. 논문에 대한 직접적인 도움은 아니지만 여러 프로젝트의 경험을 쌓게 해주신 이운식 교수님에게도 감사를 드립니다. 격려의 말로서 격려를 해주신 박병무 교수님과 권혁무 교수님에게도 감사를 드립니다.

입학한지가 며칠 되지도 않은 것 같은데 벌써 졸업이라니 즐거움 보다 사회에 대한 두려움이 앞서기만 하지만 저만의 노력이 아닌 여러 사람들의 도움과 격려로 졸업을 하게 된 은혜에 보답하기 위해 훌륭한 사람이 되도록 노력하겠습니다.

박지원 드림