http://dx.doi.org/10.5392/JKCA.2012.12.06.048

콘텐츠추적정보 관리 시뮬레이터

Simulator for Management of Tracking Information of Digital Content

이승원, 최 훈
충남대학교 컴퓨터공학과

Seung-Won Lee(slsw8406@naver.com), Hoon Choi(hc@cnu.ac.kr)

요약

스마트폰이나 태블릿 PC와 같은 모바일 디바이스에 이용되는 디지털 콘텐츠 수가 IT 산업 발달과 함께 급속도로 증가하고 있다. 이와 함께 디지털 콘텐츠 관리 방법에 대한 연구도 활발히 진행되고 있다. 기존 연구에서 인터넷 불가능지역에서 모바일 디바이스간에 디지털 콘텐츠 활용 정보를 효율적으로 관리할 수 있는 CTI가 정의되었으며, 콘텐츠가 전달될 때 마다 생성되는 CTI에 대한 관리, 오버헤드 감소, 가능한 빠른 시간에 많은 CTI를 수집할 수 있는 기술 등과 같은 콘텐츠추적정보 관리 방법을 제안하였다. 본 논문는 콘텐츠추적정보 관리 방법을 검증하고, 성능을 분석할 수 있는 시뮬레이터를 설계, 구현하였다. 이 시뮬레이터는 인터넷 접근이 안되는 환경에서 모바일 디바이스의 이동에 따른 콘텐츠 이동을 가상으로 시뮬레이션하고, 콘텐츠추적정보 관리 방법에 대한 효율적인 동기화 오버헤드 감소와 여러 이점들을 검증하였다.

■ 중심어 : | 디지털콘텐츠 | 콘텐츠추적정보 | 동기화 | 인터넷 불가능지역 |

Abstract

The number of digital content used in mobile devices such as smartphones or tablet PCs has been rapidly increasing along with the development of IT industry. At the same time, studies on digital content management are actively conducted. One of the previous studies defined CTI(Content Tracking Information) for efficient management of the information on digital content usage between mobile devices in an area in which the internet is inaccessible, and suggested ways to deal with CTI management, reduction of the overhead and a technique of collecting as many CTIs as possible within a short time. This paper presents the design and implementation of a simulator capable of verifying the logic and the performance of the CTI management method. This tool simulates a mobile environment where devices move around in the internet inaccessible area and propagates digital contents each other. It shows several advantages including the efficient synchronization of CTI management and overhead reduction.

■ keyword : | Digital Content | Content Tracking Information | Synchronization | Internet-inaccessible Area |

Ⅰ. 서 론

최근 모바일 디바이스와 데스크톱 PC 등에서 이용 가능한 디지털 콘텐츠 수가 기하 급수적으로 증가하고 있다[1][2]. 이에 따라, 이러한 디지털 콘텐츠의 전달 경로, 전달 방법 및 사용 패턴 등과 같은 디지털 콘텐츠의

* 이 연구는 2010년도 충남대학교 학술연구비에 의해 지원되었음
접수번호 : #120320-002
접수일자 : 2012년 03월 20일
심사완료일 : 2012년 04월 10일
교신저자 : 최훈, e-mail : hc@cnu.ac.kr

활용 정보를 효율적으로 관리함으로써, 디지털 콘텐츠 제공업자뿐만 아니라 소비자에게 보다 질 높은 서비스를 제공하기 위해 디지털 콘텐츠 관리에 대한 연구가 활발히 진행되고 있다[3-5]. 이러한 디지털 콘텐츠 관리에 대한 연구로서, 메타 데이터를 이용하는 방법들이 널리 쓰이고 있다[6][7]. 이러한 메타 데이터는 디지털 콘텐츠의 효율적인 관리를 위해서는 매우 유용하지만, 관리의 오버헤드가 발생하기 때문에 경량화 및 표준화에 대해서도 연구되고 있다[8].

지금까지의 보고된 연구들에는 인터넷 환경에서 사용자 컴퓨터들이 서버 컴퓨터에서 디지털 콘텐츠를 다운로드 받는 경우에 서버 컴퓨터에 적용되는 관리 기법들이 주로 연구되었으며, 모바일 환경에서의 사용자 컴퓨터들끼리 직접 디지털 콘텐츠를 배포하는 경우의 관리 기법에 대한 연구는 많지 않다[9][10]. 대표적인 연구로서 MANET(Mobile Ad-hoc Network)환경에서의 디지털 콘텐츠의 효율적인 관리 기법이 제안된 바 있다[11]. 이 연구에서는 인터넷 접속 가능지역과 인터넷 접속 불가능 지역으로 정의하였다[그림 1].

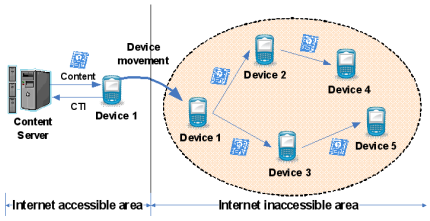

그림 1. 모바일 환경 정의 콘텐츠 전달 과정

[그림 1]에서 인터넷 불가능 지역이란 오지, 산간 지역 같이 무선 인터넷 인프라가 갖춰져 있지 않은 지역을 말하고 있다. 인터넷 불가능 지역에서는 모바일 디바이스들에게 콘텐츠를 전달할 수 있는 방법과 디바이스간에 콘텐츠를 공유하는 메커니즘이 필요하다. 그러기 위해서는 Ad-hoc 네트워크를 이용한 peer-to-peer 기술이 필요하다[11].

연구에서는 이런 환경속에서 모바일 디바이스간에 디지털 콘텐츠 활용 정보를 효율적으로 관리할 수 있는 CTI(Content Tracking Information)를 정의하고, CTI 생성 방법과 CTI 동기화 기법을 제안하였으며, CS (Content Server)가 각 모바일 디바이스가 가지고 있는 CTI를 가장 짧은 시간에 수집할 수 있는 최적의 메커니즘을 제안하였다. 본 논문에서는 제안된 메커니즘을 객관적으로 검증할 수 있는 시뮬레이터를 설계 및 구현하고 시뮬레이션 결과를 기술한다.

본 논문의 구성은 기존 연구에서 제안된 콘텐츠추적정보(CTI) 관리 방법에 대해서 Ⅱ장에서 기술하고, Ⅲ장에서는 제안된 방법을 검증하기위한 시뮬레이터 설계 및 구현에 대해 기술한다. Ⅳ장에서는 성능평가 및 검증을 기술한다. 마지막으로, Ⅴ장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

Ⅱ. 콘텐츠추적정보 관리 방법

콘텐츠추적정보 관리란 모바일 디바이스들이 CS로부터 콘텐츠를 다운로드하면, CS는 CTI를 생성하여 콘텐츠와 함께 모바일 디바이스에 전달한다. 해당 모바일 디바이스는 다시 인터넷 불가능 지역으로 접근하여 주변 모바일 디바이스에 콘텐츠를 공유하고 CTI를 동기화하면서 여러 디바이스들에게 전파시킨다. 하지만 이러한 과정이 지속되면, 각 모바일 디바이스마다 빈번한 동기화로 인해 오버헤드가 발생하고, 필요하지 않는 CTI로 인한 메모리 공간 소비가 발생한다. 뿐만 아니라 각 모바일 디바이스들이 CS에 쉽게 접근하기 어려운 가운데, CTI정보를 필요로 하는 CS는 가능한 빠른 시간에 많은 CTI를 수집할 필요가 있다.

2.1 CTI element format

CTI은 어느 콘텐츠의 근원지와 이동경로를 표시하는 정보이며, 해당 콘텐츠와 함께 디바이스에 저장된다. 다음 [그림 2]는 CTI 구조이다.

```
struct CTI {
    int     from;
    int     to;
    int     sequence;
    boolean  flag;
}
```

그림 2. CTI 구조체 정보

[그림 2]에서, 'from' 속성은 콘텐츠를 제공한 디바이스의 식별자이고, 'to' 속성은 콘텐츠를 수신한 디바이스의 식별자이며, 'sequence' 속성은 'from' 디바이스가 서로 다른 디바이스들에게 같은 콘텐츠를 전달한 횟수이다. 'flag' 속성은 CTI가 인터넷 가능한 지역에 위치하고 있는 CS와 접속하여 CTI가 업로드 되었음을 나타낸다.

2.2 CTI list 생성

CTI list는 CTI element의 집합으로서, 각 디바이스에 콘텐츠별로 하나씩 존재한다. 하나의 디바이스가 하나의 콘텐츠를 주변 디바이스에게 전달할 때마다, 제공하는 디바이스와 제공받는 디바이스간에 새로 생성된 CTI element를 각 CTI list에 추가한다. 예를 들어, [그림 3]에서 device1이 device2로 콘텐츠를 전달할 때, 1-2-1-f의 값을 갖는 CTI element가 생성되어, device1의 CTI list에 추가하고, 해당 CTI list가 콘텐츠와 함께 device2에 제공된다. 계속해서, 같은 콘텐츠가 device2에서 device3으로 전달할 때, 2-3-1-f의 값을 갖는 CTI element가 device2의 CTI list에 추가되고, 콘텐츠와 함께 device3에게 전달된다. sequence값은 각 디바이스별로 부여하는 integer counter value로서, 0에서 시작하여 주변 디바이스에게 콘텐츠를 전달 될 때마다 1씩 증가한다.

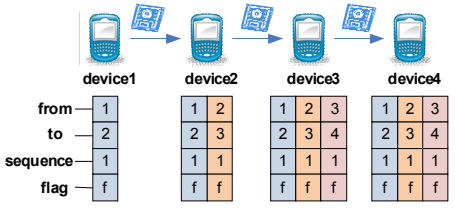

그림 3. CTI list 예제 1

2.3 CTI 동기화

CTI 동기화의 핵심적인 알고리즘은 다음과 같이 3가지 방법으로 나누어지며 참고문헌 [11]에 상세히 기술되어 있다.

① 디바이스 간 CTI list 동기화

같은 콘텐츠를 보유하고 있는 모바일 디바이스가 무선 신호범위 안에 들면, 서로 보유하지 않은 CTI를 동기화한다. 이는 인터넷 접속이 원활하지 못한 환경에서, 각 디바이스들이 가능한 많은 CTI를 확보하도록 하여, 어느 한 모바일 디바이스가 인터넷에 접속하는 경우 다른 디바이스를 대신하여 CS에 CTI를 업로드하기 위한 방법이다. 이때, CTI list는 from 항목과 sequence 항목을 기준으로 정렬하게 된다.

② CS와 CTI 동기화

하나의 모바일 디바이스가 인터넷 접속가능 지역으로 이동하여, CS와 접속되면 해당 모바일 디바이스가 보유한 CTI는 CS에 업로드 된다. 이후, CS에 CTI를 업로드한 디바이스는 CTI list에서 최소의 CTI 정보를 제외한 나머지 CTI를 삭제함으로써, 해당 디바이스에서 CTI list로 인한 메모리 공간을 절약하게 한다. 이때 flag필드는 t(true)가 된다.

③ CS와 동기화한 디바이스와 일반 디바이스 간 CTI 동기화

어느 디바이스가 CS와 동기화 후, 인터넷 불가능한 지역으로 다시 진입하면 주변 디바이스에게 자신이 CS와 동기화 되었음을 알려주고, CS에 업로드 했던 CTI 정보를 기반으로 최소의 CTI를 제외한 나머지 CTI를 삭제하도록 만든다.

Ⅲ. 검증용 시뮬레이터 설계 및 구현

3.1 시뮬레이션 모델

검증용 시뮬레이터는 인터넷 불가능한 지역에서 MANET 환경을 모델링하며, 제안하는 콘텐츠추적정보 관리 방법의 검증을 자동으로 시험할 수 있는 형태이어야 한다. 따라서, 다음과 같은 기능과 가정을 기반으로 설계되어야 한다.

- 인터넷 접속 가능 지역과 접속 불가능 지역을 구분

해야 함

- 모바일 디바이스의 무선통신 신호가 닿는 지역을 시뮬레이션 모델에서 하나의 셀(cell)로 나타냄
- 두 개 이상의 디바이스가 같은 셀에 위치하게 되면, 서로 ad-hoc 네트워크를 형성하고 peer-to-peer 방식으로 상호 통신함
- Discrete time 시뮬레이션을 적용함. 시뮬레이션 시간의 한 step 마다 디바이스가 원래 있던 셀에 그대로 머물거나, 또는 인접한 셀로 이동함. [그림 4]와 같이 하나의 디바이스가 이동 가능한 경우 수는 현재 위치한 셀을 포함하여 모두 9가지임. 각 셀로 이동 확률은 동일하게 1/9로 가정함

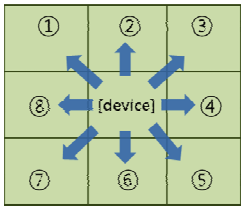

그림 4. 모바일 디바이스의 이동 경우 수

- 시뮬레이션 step 수를 시간 값으로 측정함
- 디바이스가 인터넷 불가능 지역의 셀에서 인터넷 가능지역의 셀로 이동하는 확률은 1/50 로 가정함. 예를 들어 아래 [그림 5]에서 CS가 위치한 셀이 인터넷 가능지역이고 그 주변 셀들은 인터넷 불가능 지역일 때, 디바이스가 인터넷 불가능 지역의 셀에 있다가 다음 시뮬레이션 step때 CS가 있는 인터넷 가능 지역 셀로 이동할 확률을 의미함

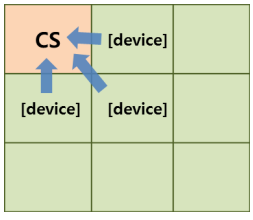

그림 5. 디바이스와 CS와 접속 확률

본 연구가 대상으로 하는 지역인 오지에서는, 디바이스 소유자가 대부분의 시간을 인터넷 불가능지역에서 머물게 된다. 자신의 생활 주변에서 디바이스를 소지한 채 부근을 주로 이동하다가 도시와 같은 인터넷 가능 지역으로는 가끔 이동하기 때문에 확률을 1/50 이라는 적은 값으로 설정하였다. 이 값은 변경 가능하며, 값의 대소는 CS가 CTI 정보를 수집하는데 걸리는 시간에만 영향을 미칠 뿐, 동기화 방식 알고리즘 자체와는 상관이 없다.

3.2 시뮬레이터 프레임워크

앞에서의 모델에 입각해 인터넷 불가능지역 환경이 구축되어 있는 모듈과 자유로운 좌표이동과 peer-to-peer 방식이 가능한 디바이스 모듈, 콘텐츠를 저장하기 위한 모듈로 구성된 시뮬레이터 프레임워크를 개발하였다[그림 6].

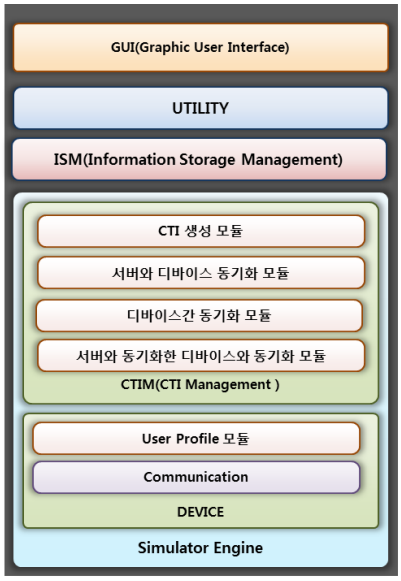

그림 6. 시뮬레이터 프레임워크 구조

- GUI 모듈 : 디바이스와 CS 위치, CS에서 제공되는 콘텐츠, MANET의 크기 면적을 설정할 수 있는 입력기능, 그리고 시뮬레이션 상황을 실시간으로 모니터링하기 위한 출력부 제공
- ISM(Information Storage Management) 모듈 : CS

로부터 제공되는 콘텐츠와 디바이스로부터 생성된 CTI list 정보 저장 및 관리

- CTIM(CTI Management) 모듈 : 디바이스간에 동기화 하면서, 필요한 프로파일 정보를 분석하고, 콘텐츠추적정보 관리 모듈 알고리즘별 실행
- DEVICE 모듈 : 디바이스와 CS, 디바이스와 디바이스간에 인증(authentication)과 데이터 송수신 관리

3.3 구현환경

검증용 시뮬레이터를 구현하기 위한 개발환경은 [표 1]과 같다.

표 1. 시뮬레이터 개발 환경

항목	내용
개발 언어	Java
JDK	JDK1.6.0_23
개발 Tool	Eclipse
운영체제	Windows 7 Professional k

3.4 구현결과

[그림 7]은 콘텐츠추적정보 관리 시뮬레이션을 위한 GUI이며, 각 기능은 다음과 같이 구성된다.

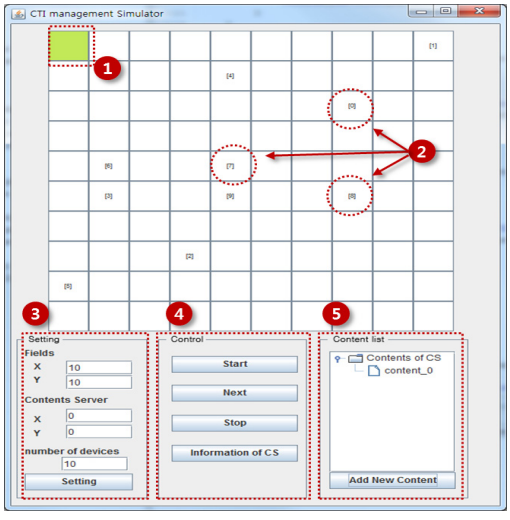

그림 7. CTI 시뮬레이터 관리 GUI

① CS 배치 상태
② 모바일 디바이스 이동 상태
③ 모바일 디바이스, MANET 면적넓이, CS 위치 값을 설정하는 입력박스
④ 시뮬레이터 응용 동작버튼
⑤ CS가 제공하는 콘텐츠 리스트

[그림 7]과 같이 검증용 시뮬레이터는 파라미터와 그리드 섹션을 보여주는 사용자 그래픽 인터페이스를 제공한다. 시뮬레이션이 시작할 때, 여러 작은 정사각형안에 모바일 디바이스가 임의 위치로(random) 배치된다. 작은 정사각형은 3.1절에서 설명한 셀을 나타낸다. 각 디바이스는 셀에서 한 칸씩 이동하여, 인접한 셀로 이동을 시도한다. 또한, CS가 가지고 있는 콘텐츠 리스트와 각 디바이스마다 가지고 있는 콘텐츠, 그리고 CTI 상태정보를 로그파일에 출력하게 된다.

다음 [그림 8]는 시뮬레이터를 통해 CS가 시뮬레이션하는 도중에 CTI를 수집한 로그 기록의 일부분이다. device9의 CTI 정보를 따르면, 현재 device9는 device5로부터 Content_0을 전달받았고, device5는 deivce2로 전달받았으며, device2는 device4로부터 전달 받았음을 알 수 있다.

```
Information of device[3]
Content_0=> (4, 2, 1, false) (2, 5, 1, false) (2, 8, 2, false)
(2, 6, 3, false) (6, 1, 1, false) (6, 3, 2, false)
Information of device[4]
Content_0=> (4, 2, 1, false)
Information of device[5]
Content_0=> (4, 2, 1, false) (2, 5, 1, false) (5, 9, 1, false)
(5, 7, 2, false)
Information of device[6]
Content_0=> (4, 2, 1, false) (2, 5, 1, false) (2, 8, 2, false)
(2, 6, 3, false) (6, 1, 1, false) (6, 3, 2, false)
Information of device[7]
Content_0=> (4, 2, 1, false) (2, 5, 1, false) (5, 9, 1, false)
(5, 7, 2, false)
Information of device[8]
Content_0=> (4, 2, 1, false) (2, 5, 1, false) (2, 8, 2, false)
Information of device[9]
Content_0=> (4, 2, 1, false) (2, 5, 1, false) (5, 9, 1,
false)
```

그림 8. 수집된 CTI list 로그 내역

Ⅳ. 성능평가 및 검증

본 장은 시뮬레이터를 통해 콘텐츠추적정보관리 방법의 정확성 검증과 그 성능에 대한 실험결과를 보여준다. 실험방법은 다음과 같은 시나리오를 가정한다.

① 인터넷 가능 지역에 있는 CS가 하나의 콘텐츠 제공
② 인터넷 불가능 지역에 위치한 모든 디바이스가 해당 콘텐츠를 수신하려고 함
③ CS는 모든 디바이스로 부터 CTI를 수집하기까지, 소요된 총 step 수를 측정
④ 검증용 시뮬레이터에서는 다음과 같은 3개의 실험을 가정
   A. CTIM 실험 : [11]의 콘텐츠추적정보 관리 방법 알고리즘 적용
   B. Test A 실험 : 콘텐츠추적정보 관리 방법 알고리즘 중 디바이스간 CTI 동기화 알고리즘 제외
   C. Test B 실험 : 모든 디바이스는 CTI list를 생성하지 않고 직접 CS에 접근하여 자신의 ID를 업로드 하여 보고

위와 같이 3가지 실험 방법을 통하여, 검증용 시뮬레이터는 하나의 콘텐츠를 대상으로 400개의 필드 안에, 각 디바이스의 수 10개, 20개, 30개, 40개로 설정하고, CTIM, Test A, Test B 실험 방법대로 각 30회 시뮬레이션을 실행하여, 디바이스의 step의 누적값 평균을 구하였다.

[그림 9]은 실험에 참여하는 디바이스 개수의 80%에 해당하는 디바이스에 대한 CTI를 CS가 수집하기까지의 시뮬레이션 step 수를 나타낸다, 즉 디바이스가 20개라면 16개 디바이스가 콘텐츠를 받았음을 CS가 알게 되는 시점까지의 평균 시간이다. [그림 10][그림 11]은 각각 90%, 100% 디바이스들에 대한 CTI를 CS가 수집하기까지의 실험결과를 나타낸다.

[그림 11]의 실험결과는 디바이스가 증가할수록, Test A는 디바이스 간에 CTI를 공유하지 않고 자신이 보유하고 있는 CTI만을 CS에 업로드하는 방식이기 때문에 CTIM보다 2배에서 12배까지 시간이 경과되는 결과를 볼 수 있다.

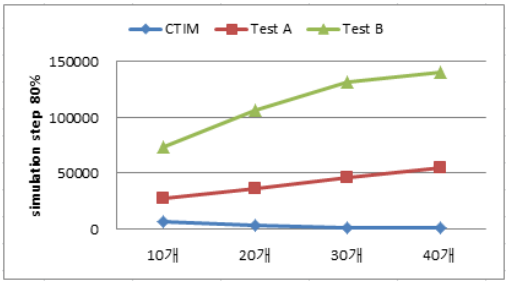

그림 9. CTI 80% 수집 시 경과되는 시간

그림 10. CTI 90% 수집 시 경과되는 시간

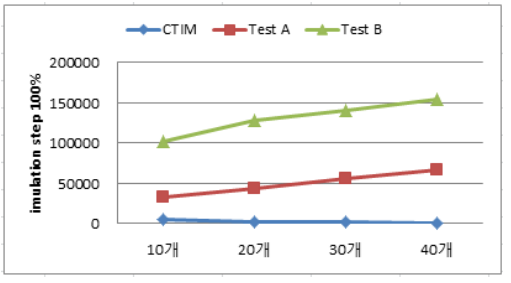

그림 11. CTI 100% 수집 시 경과되는 시간

Test B는 각 디바이스가 개별적으로 CS에 직접 접근하여 자신이 콘텐츠를 받았음을 신고하는 방식이기 때문에 CTIM보다 2배에서 20배까지 시간이 경과되는 결과를 볼 수 있다.

Test A, B 모두 디바이스 개수가 늘어날수록 CS가 원하는 정보를 수집하기 까지 시간이 더 걸리지만, CTIM 실험에서는 디바이스들 개수가 많아질수록 어떤 것이든 CS와 일찍 동기화할 가능성이 높아지게 된다.

디바이스들끼리 다른 디바이스 CTI 정보를 공유하고 있기 때문에 더 많은 CTI 정보를 CS에 일찍 전달하게 되고, 결국 시간이 짧아지는 것이다.

디바이스들끼리 CTI를 공유하게 되면, CTI정보 저장에 따라 메모리 공간이 소비되기 때문에, CS와 동기화가 된 CTI는 디바이스에서 삭제하게 된다. 시뮬레이터는 이런 기능도 구현하였다. 불필요한 CTI를 삭제시켜 최소 CTI 길이로 축소시키는 실험 결과를 [그림 12]에 보였다.

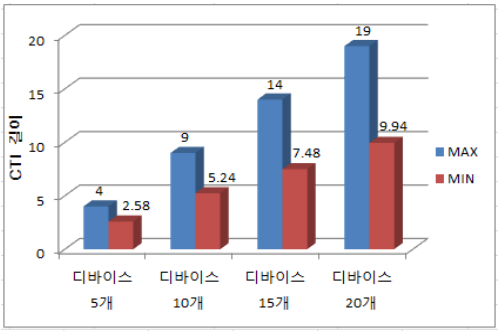

그림 12. 각 디바이스의 CTI 압축률

다음 [표 2]는 시뮬레이터의 확정성(Scale)을 평가하기 위한 실험조건이다. [표 2]는 시뮬레이터에서 설정된 400개 셀 안에서 1개 디바이스부터 최대 800개 디바이스까지 참여시켜 실험하였으며, CS로부터 모든 디바이스의 CTI를 수집할 때까지 경과된 디바이스의 step을 측정하고, 디바이스가 증가할 때마다 시뮬레이션 시간과의 상반관계를 측정하였다.

표 2. 시뮬레이션 Parameters

콘텐츠	셀(cell) 수	디바이스
1개	400개	1개
1개	400개	50개
1개	400개	100개
1개	400개	200개
1개	400개	400개
1개	400개	800개

다음 [그림 13]와 [표 3]과같이, 400개 필드크기 안에서 참여하는 디바이스가 증가할수록, 디바이스 간에 CTI를 공유하면서 더 많은 CTI를 빠르게 확보할 수 있었으며, CS로도 더 자주 동기화되기 때문에 시뮬레이션 step이 대폭 감소됨을 확인하였다.

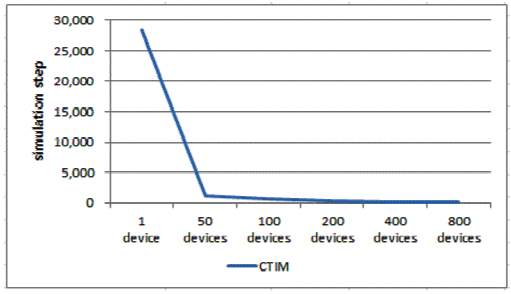

그림 13. Scale 검증 결과 그래프

표 3. Scale 검증 결과 수치

디바이스	1	50	100	200	400	800
총 Step	28,341	1,087	659	261	150	94

V. 결론

본 논문에서는 인터넷 가능지역과 불가능지역을 포함하는 모바일 환경을 위한 효율적인 콘텐츠추적정보 관리 방법을, 현실 모델과 비슷한 시뮬레이터를 개발하여 콘텐츠추적정보 관리 방법에 대한 객관적인 성능 평가 및 검증을 수행하였다.

시뮬레이터에서는 먼저 인터넷 불가능지역을 설정하였다. 이후, 시뮬레이터 상에서, 모바일 디바이스 간의 peer-to-peer 통신을 통한 콘텐츠 공유 및 콘텐츠추적 관리 방법을 검증하였다. 아울러, CS나 모바일 디바이스가 가능한 많은 CTI를 수집할 수 있다는 것과 CS가 모든 모바일 디바이스들이 소유하고 있는 CTI를 최단 시간에 수집할 수 있음을 검증하였다. 또한, 모바일 디바이스간에 CTI를 동기화하는 과정에서 메모리 오버헤드가 효과적으로 감소되는 것을 확인하였다.

참 고 문 헌

[1] http://help.adobe.com/en_US/Dreamweaver/10.0_Using/WSc78c5058ca073340dcda9110b1f693f21-79cca.html - Accessed Aug. 20, 2009.

[2] M. J. Katchabaw, H. L. Lutfiyya, and M. A. Bauer, "Usage-based Service Differentiation for End-To-End Quality of Service Management," 2003. Proceedings of the 2003 IEEE Int'l Performance Computing and Communications Conference, pp.513-520, 2003(4).

[3] J. Y. Hong, E. H. Suh, and J. Y. Kim, "Context-aware System for Proactive Personalized Service based on Context History," Expert Systems with Applications, Vol.36, No.4, pp.7448-7457, 2009(5).

[4] Y. Yamato, H. Ohnishi, and H. Sunaga, "Study of Service Processing Agent for Context-Aware Service Coordination," Services Computing, 2008. SCC '08. IEEE Int'l Conference on Services Computing Vol.1, 2008(7).

[5] Tracer, http://www.trendbird.co.kr/1885 - Accessed Feb. 10, 2010.

[6] 김재인, 김대인, 송명진, 한대영, 황부현, "디지털 콘텐츠 연관 정보 관리를 위한 메타데이터 설계 및 시스템 개발", 한국콘텐츠학회논문지, 제9권, 제7호, pp.27-36, 2009.

[7] 김형도, "ebXML 등록저장소를 이용한 이러닝 객체 메타데이터의 표현과 관리", 한국콘텐츠학회 논문지, 제6권, 제11호, pp.249-257, 2006.

[8] 박충범, 이영석, 최훈, "휴대폰을 통한 개인 최적화된 학습 서비스 시스템 설계", 한국차세대컴퓨팅학회 논문지, Vol.6 No.2, pp.21-30, 2010(4).

[9] Data Conversion Laboratory, DC Tracker (Digital Content Tracker), https://home.dclab.com/demo/dcdemo1.asp-Accessed Jan. 4, 2010.

[10] J. H. Hsiao, C. H. Li, C. Y. Chiu, J. H. Wang, C. S. Chen, and L. F. Chien, "Effective Content Tracking for Digital Rights Management in Digital Libraries," Lecture Notes in Computer Science, Vol.4172, pp.415-425, 2006.

[11] 박병성, 박충범, 최훈, "Contents Tracking Information 관리 기법", 한국정보처리학회, 제15권, 제2호, 2008(11).

저 자 소 개

이 승 원(Seung-Won Lee) 정회원

- 2010년 2월 : 배재대학교 컴퓨터공학과(공학사)
- 2012년 2월 : 충남대학교 컴퓨터공학과(공학석사)
- 2012년 3월 ~ 현재 : 충남대학교 컴퓨터공학과 박사과정

<관심분야> : 미들웨어, 모바일컴퓨팅, 분산시스템

최 훈(Hoon Choi) 정회원

- 1983년 : 서울대학교 전산기공학과 학사
- 1993년 : Duke University 전산학과 박사
- 1983년 ~ 1996년 : 한국전자통신연구원
- 1996년 3월 ~ 현재 : 충남대학교 컴퓨터공학과 교수

<관심분야> : 분산시스템, 미들웨어