

---

# 모바일 환경에서 필드 업데이트를 지원하는 시공간 데이터의 양방향 동기화 시스템의 설계 및 구현

김홍기\* · 김동현\*\* · 조대수\*\*\*

A Design and Implementation of a Two-Way Synchronization System of Spatio-Temporal Data Supporting Field Update in Mobile Environment

Hong-Ki Kim\* · Dogn Hyun Kim\*\* · Dae-Soo Cho\*\*\*

---

본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C05)에 의해 수행되었음

---

## 요 약

유비쿼터스 컴퓨팅 환경에서 GIS 서비스는 모바일 기기를 이용하여 언제, 어디서나 시공간 데이터를 이용할 수 있는 특징이 있다. 또한 최신의 공간 데이터를 무선 네트워크를 이용하여 서버로부터 클라이언트는 업데이트를 제공한다. 하지만 기존의 시스템은 실세계의 변화를 서버에서 일정 주기로 수집하기 때문에 사용자에게 최신 정보를 제공하는데 오랜 시간이 소요되는 단점이 있다. 이 논문에서는 기존의 문제점을 해결하기 위해 필드 업데이트를 지원하는 양방향 동기화 시스템을 제안한다. 제안하는 시스템은 모바일 기기를 이용하여 클라이언트에서 실세계의 변경된 사항을 수집하고, 수집된 데이터를 서버에 전송한다.

## ABSTRACT

In ubiquitous GIS services is possible to use the spatio-temporal data using a mobile device at anytime. Also, client is transmitted latest spatio-temporal data from server. But traditional systems have a problem that the time of transmitting latest information from server to client takes long time because of collecting data periodically. In this paper, we proposed Two-way Synchronization system supporting field update to solve the existing problem. This system uses mobile device for collecting changed data in the real world and sending collected data to server.

## 키워드

필드 업데이트, 양방향 동기화 프로토콜, 시공간 데이터, 모바일

## Key word

Field Update, Two-Way Synchronization Protocol, Spatio-Temporal Data, Mobile

---

\* 동서대학교 유비쿼터스 IT학과 박사과정  
\*\* 동서대학교 컴퓨터정보공학부 부교수  
\*\*\* 동서대학교 컴퓨터정보공학부 조교수(교신저자)

접수일자 : 2009. 12. 21  
심사완료일자 : 2010. 01. 15

## I. 서 론

유비쿼터스 컴퓨팅 환경에서 GIS(Geographic Information System) 서비스는 모바일 기기를 이용하여 언제, 어디서나 이용할 수 있는 특징이 있다. 또한 무선 네트워크를 이용하여 사용자에게 필요한 공간데이터를 신속하게 제공할 수 있는 장점이 있다[1]. GIS 서비스는 최신 공간데이터를 사용자에게 제공하기 위해 지속적으로 업데이트를 해야 한다. 기존의 시스템[2]은 최신 공간데이터를 사용자에게 빠르게 적용시키기 위해 무선 네트워크를 통해 실시간 업데이트를 제공한다[3]. 그러나 실제세계의 변경사항이 사용자들에게 적용되기까지 오랜 시간이 소요된다. 왜냐하면 실제세계 변경사항을 서버 측에서 수집하고, 수집된 내용을 클라이언트로 전송하는 방식으로 업데이트를 지원하기 때문이다. 무선 네트워크를 통한 업데이트[1][2]는 클라이언트의 요청이 있을 때 최신 정보를 제공할 수 있지만 서버 측에서 실제세계의 변경사항을 수집하는 시기가 한정되어 있기 때문에 실제세계의 변경사항이 빠르게 적용되기는 어렵다. 따라서 실제세계의 변경사항을 빨리 수집할 수 있는 기술이 요구된다.

이 논문에서는 최신의 공간데이터를 신속하게 제공하기 위해 모바일 환경에서 필드 업데이트를 지원하는 양방향 동기화 시스템을 제안한다. 제안하는 시스템은 실제세계의 변경된 정보를 서버에서 수집하여 클라이언트와 동기화하는 방법뿐 아니라 변경된 정보를 클라이언트에서 수집하여 서버와 동기화하는 방법을 모두 지원하는 양방향 동기화 시스템이다. 따라서 실제세계의 변경사항을 서버 측이 일정 주기로 수집하는 것이 아니라 클라이언트에서 변경사항을 즉시 수정하여 서버와 동기화함으로써, 다른 클라이언트들이 빠르게 최신의 공간 데이터를 업데이트 받을 수 있다.

이 논문의 구성은 다음과 같다. 2장에서는 관련연구를 기술하고, 3장에서는 부분 동기화와 필드 업데이트를 통한 양방향 동기화 프로토콜에 대해서 설명한다. 4장에서는 제안하는 시스템에 대해서 기술하고 5장에서는 결론 및 향후 해결해야 될 과제에 대해서 설명한다.

## II. 관련연구

ActMap[3] 시스템은 유럽 자동차 회사들과 디지털 지도 제작 업체와의 프로젝트 팀에 의해 2007년도에 제안된 내비게이션 지원 시스템으로 자동차에 탑재된 내비게이션 기기에 무선 네트워크를 이용하여 지도의 최신 변경사항을 반영할 수 있도록 지원하는 시스템이다. ActMap은 유선에 비해 대역폭의 제한을 많이 받는 무선 네트워크상에서 서버와 모바일 단말기 사이에 고용량 지리정보의 효율적인 동기화를 위하여 부분 점진적 갱신(Partial Incremental Update)을 지원하며, 부분 점진적 갱신을 위한 방법을 제시한다.

첫째, 계층(Layer)과 분할(Partition)을 이용하여 갱신을 제공한다. 이 방법은 갱신 발생 시 네트워크의 부하를 줄이기 위해 갱신이 발생한 계층 또는 분할의 갱신 정보만을 모바일 단말기로 전송하는 방법이다.

둘째, 필터를 이용하여 갱신을 제공한다. 모바일 단말기 사용자가 직접 원하는 정보를 등록함으로써 등록된 정보에 대해서만 필터링을 하여 갱신을 하는 방법이다. 특정 영역, 계층 또는 시간, 우선순위 등이 해당된다[1].

셋째, 버전관리를 이용하여 갱신을 제공한다. 서버와 모바일 단말기의 버전을 이용하여 버전이 일치하지 않는 지리정보만을 갱신한다.

넷째, 갱신 집합을 이용하여 갱신을 제공한다. 하나의 같은 지리정보에 대해서 갱신을 제공받기 전까지 여러 번 변경이 이루어 졌을 때, 가장 최근에 변경된 정보만을 제공하는 방법이다.

다섯째, 기준지도를 이용하여 갱신을 제공한다. 지리정보 서비스를 사용하기 위해 최초로 제공받은 전체 지리정보를 기준으로 하여, 추후에 갱신된 정보에 대해서만 모바일 단말기에 갱신을 제공하는 방법이다.

위의 다섯 가지 방법을 통하여 ActMAP 시스템은 제한적인 대역폭을 가지는 모바일 환경에서 지리정보의 크기를 최소화하여 효율적인 동기화를 지원한다. 하지만 모바일 단말기를 통해 현장에서 수집, 변경된 정보의 갱신은 ActMAP 시스템은 지원하지 않는다. 따라서 유비쿼터스 환경에서의 GIS 서비스는 현장에서 모바일 단말기를 통한 수집, 변경된 정보를 서버와 동기화할 수 있는 양방향 동기화 프로토콜[1][3]이 필요하다.

### III. 양방향 동기화 프로토콜

양방향 동기화는 서버에서 클라이언트로의 단방향 동기화와 클라이언트에서 서버로의 단방향 동기화로 구성될 수 있다.

#### 3.1 용어정리

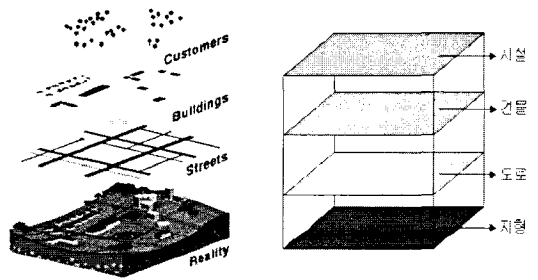
이 절에서 사용되는 용어는 다음과 같다.

- 계층(Layer) : 전체 데이터를 주제별로 분류하여 생성한 계층
- 분할(Partition) : 각각의 계층을 지리적으로 분할한 영역
- DO(Delta Object) : 최종 동기화 시간(LST) 이후 서버에서 변경된 데이터
- DOS(Delta Object Set) : DO들의 집합
- TDO(Temporary Delta Object) : 최종 동기화 시간 이후 클라이언트에서 변경된 데이터로서 동기화를 통해 서버에 저장되기 전까지는 임시 데이터의 특성을 가짐
- TDOS(Temporary Delta Objects Set) : TDO들의 집합
- CR(Copy Region) : 클라이언트가 복사해서 가지고 있는 데이터영역 중 동기화를 요청하는 계층, 분할의 집합
- LST(Last Sync. Time) : 클라이언트가 서버와 최종 동기화를 한 시간으로 계층, 분할 별로 관리
- LUT(Last Update Time) : 서버의 데이터가 최종적으로 갱신된 시간으로 계층, 분할 별로 관리

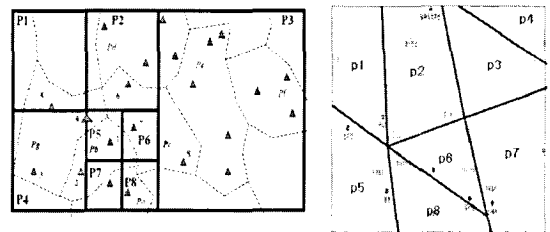
#### 3.2 서버에서 클라이언트로의 단방향 동기화

클라이언트는 서버에서 관리하는 전체 데이터 중에서 일부의 데이터에 대해서 관심을 가질 수 있다. 따라서 전체 데이터 셋은 분할해서 관리할 필요가 있다. 이 논문에서는 계층과 분할을 사용하여 전체 데이터를 구조적으로 관리한다. 전체 데이터를 주제별로 분류한 것을 계층이라고 한다. 예를 들면, 도로 계층, 건물 계층 등이 있다. 각각의 계층을 지리적으로 나눈 것을 분할이라 한다. 분할은 구, 군 등의 행정 경계에 따라 나눌 수도 있으며, 고정된 크기의 격자로 나눌 수 있다.

그림 1에는 계층과 분할의 예를 보이고 있다.



(a)



(b)

그림 1. 계층과 분할의 예  
(a) 계층의 예 (b) 분할의 예

Fig 1. Example of Layer and Partition  
(a) Example of Layer (a) Example of Partition

서버는  $n$ 개의 계층으로 구성된 데이터 셋이 있으며, 각각의 계층  $L_i$ 는  $k$ 개의 분할로 구성되므로 서버가 관리하는 데이터 셋(SDS : Server Data Set)은 다음과 같이 표현된다.

$$SDS = \bigcup_{i=1}^n L_i, L_i = \{P_1, P_2, P_3, \dots, P_k\} \text{ (단, } n, k$$

는 양의정수). 각 클라이언트는 SDS 중 일부의 데이터를 다운받아 관리하게 되므로, 임의의 클라이언트  $C_i$ 가 관리하는 데이터 셋(CDS : Client Data Set)은  $CDS \subset SDS$ 로 표현된다. 클라이언트  $C_i$ 는 자신이 관리하는 CDS 중에서 서버와 동기화를 원하는 영역을  $CR \subset CDS$ 로 표현한다.

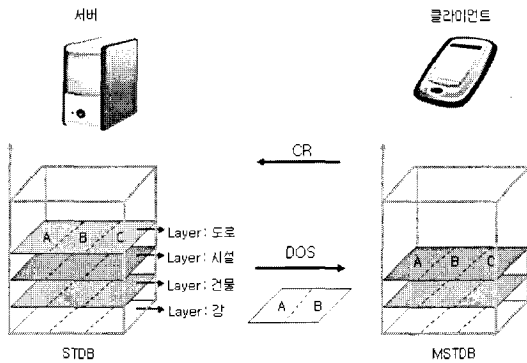


그림 2. 클라이언트 요청에 의한 부분 동기화  
Fig 2. Partial synchronization about client request

그림 2를 예를 들어 설명하면, 서버에는 도로, 시설 건물, 강 4개의 계층이 존재하고, 각각의 계층은 A, B, C 3개의 분할로 관리되므로  $SDS = \{ \text{도로A}, \text{도로B}, \text{도로C} \} \cup \{ \text{시설A}, \text{시설B}, \text{시설C} \} \cup \{ \text{건물A}, \text{건물B}, \text{건물C} \}$ 이다. 클라이언트  $C_i$ 는 서버의 데이터 중에서 시설, 건물, 강 계층의 A, B 분할을 다운받아 사용하므로  $C_i.CDS = \{ \text{도로A}, \text{도로B} \} \cup \{ \text{시설A}, \text{시설B} \} \cup \{ \text{건물A}, \text{건물B} \}$ 이다. 즉, 서버의 데이터 중에서 시설, 건물, 강 계층의 데이터만 관심의 대상이고, 그 중에서도 분할 A, B 영역만을 관리한다. 클라이언트  $C_i$ 는 자신이 관리하는 모든 CDS 중에서 동기화가 필요한 영역을 일부만 선택하여, CR을 통해 서버에 동기화를 요청할 수 있다. 그림 2에서는 클라이언트  $C_i$ 가  $C4.CR = \{ \text{강A}, \text{강B} \}$ 에 대해 서버에게 동기화를 요청하고, 서버는 해당 CR 영역의 데이터를 클라이언트  $C_i$ 에게 전송하는 부분 동기화 진행 과정이다. 서버와 클라이언트가 CR 영역의 데이터에 대해서 동기화를 진행할 때, CR 영역 내의 모든 데이터를 동기화 하는 것은 아니다. 이 논문에서는 최종 동기화 시점 이후에 변경된 데이터에 대해서만 동기화를 수행함으로써 부분 동기화를 지원한다. 따라서 각각의 클라이언트  $C_i$ 는 자신이 최종적으로 동기화한 시간인 LST를 분할별로 관리해야한다.

서버에서 클라이언트로의 단방향 동기화만을 고려한다면 즉, 데이터의 갱신이 서버에서만 발생할 경우에는 부분 동기화 알고리즘은 매우 간단하다. 서버는 동기화를 요청한 클라이언트  $C_i$ 의 LST와 CR에 대해서 CR 내에 포함된 객체들 중에서 LST 보다 늦게 갱신된 데이터만을 클라이언트로 전송하면 된다. 이와 같이

LST 보다 늦게 갱신된 데이터를 DOS(Delta object Set)이라 표현한다.

클라이언트에서 서버로의 동기화를 포함한 양방향 동기화를 지원하기 위해서는 동일한 데이터를 서버와 하나 이상의 클라이언트에서 동시에 갱신함으로써 발생하는 변경충돌 문제를 고려해야한다.

### 3.3 필드 업데이트 : 클라이언트에서 서버로의 단방향 동기화

이 논문에서 클라이언트는 현장에서의 변경사항을 객체 단위로 수집 한다. 클라이언트  $C_i$ 가 관리하는 CDS의 계층  $L_i$ , 계층  $P_k$ 는  $n$ 개의 객체를 가지고 있을 때,  $L_j.P_k = \{ O_1, O_2, O_3, \dots, O_n \}$ 으로 표현되며, 각각의  $O_i$ 는 객체가 실제 변경된 시간과 데이터베이스에 갱신된 시간을 유지하고 있다.  $O_i$ 가 가지는 실제 변경된 시간은 클라이언트가 현장에서 변경 및 수집한 시간을 의미한다. 클라이언트는 LST 이후에 변경된 모든 객체들을 TDOS(Temporary Delta Object Set)라 표현한다. 클라이언트에서 변경 및 수집한 데이터는 서버와 동기화 이전에는 신뢰할 수 있는 데이터가 아니기 때문에 임시로 변경된 데이터로 표현된다.

클라이언트에서 서버로의 단방향 동기화는 다음의 절차에 따라 수행된다. 클라이언트는 서버에서 동기화 수락 이후 TDOS를 전송하고, 서버에서는 클라이언트로부터 받은 TDOS를 서버 SDS에 갱신시킨다. 이 때, 서버 데이터베이스의 최종 갱신 시간인 LUT도 함께 갱신된다. 서버는 클라이언트의 TDOS가 SDS에 갱신될 때 데이터 일관성 유지를 위해 검사를 수행해야 한다. 왜냐하면 클라이언트에서 변경한 TDOS에 존재하는 객체를 서버 또는 다른 클라이언트가 중복해서 변경할 수 있으며, 동일한 객체에 대한 변경은 서로 충돌을 발생시킬 수 있기 때문이다.

이 논문에서는 클라이언트의 LST와 서버의 LUT를 기준으로 변경충돌[4][5][6] 검사를 수행한다. 클라이언트의 LST가 서버의 LUT와 같을 경우는 최종 동기화 이후 서버 SDS의 정보가 갱신된 것이 없다는 것을 의미하므로 클라이언트의 TDOS에 대해 충돌이 발생하지 않는다. 반면에 클라이언트의 LST가 서버의 LUT 보다 작을 경우 변경충돌이 발생할 가능성이 있다. 즉, 최종 동기화 이후에 서버 SDS의 정보가 갱신되었지만 클라이언트에서는 이와 같은 사실을 모르고 변경이 있었기

때문이다. 클라이언트에서 변경한 객체가 서버 SDS에 갱신된 정보와 동일하다면 변경충돌이 발생한다. 변경충돌이 일어나면 클라이언트로부터의 TDOS 갱신 요청은 취소되고, 클라이언트에서는 해당 TDOS 데이터를 삭제한다.

그림 3은 클라이언트의 TDOS를 서버 SDS에 갱신할 때, 변경충돌을 검사하는 과정을 예로 보이고 있다. 서버에서는 각각의 계층과 분할에 대해 데이터가 최종 갱신되는 시간으로서 LUT를 관리한다. 클라이언트는 서버로부터 다운받아 관리하는 CDS의 모든 계층, 분할에 대해 서버와의 최종동기화 시간을 관리하는 LST가 존재한다. 그림 3에서 서버에는 강A.LUT=0, 클라이언트에는 강A.LST=0과 같이 LUT, LST가 관리되는 것을 보이고 있다.

그림 3에서 클라이언트는 최종 동기화 시점  $t_0$  이후  $t_1$  시간에 강A.O<sub>i</sub>를 변경하였다. 클라이언트는  $t_1$  시간에 변경된 데이터를 서버에 갱신시키기 위해서  $t_2$  시간에 동기화를 요청한다.  $t_3$  시간에 서버는 동기화를 수락하였다. 클라이언트는 변경된 데이터 TDOS={강A.O<sub>i</sub>}을  $t_4$  시간에 서버로 전송한다. 서버는 클라이언트로부터 전송받은 TDOS를 SDS에 갱신시키기 전에 변경충돌 검사를 수행한다. 변경충돌은 클라이언트에 존재하

는 강A.LST와 서버에 존재하는 강A.LUT를 비교하여 판단한다. 강A.LST가 강A.LUT보다 작고,  $TDOS \cap DOS \neq \emptyset$ 인 경우에만 변경충돌이 발생한다. 강A.LST가 강A.LUT보다 작다는 것은 클라이언트에서 서버와 최종동기화 이후에 서버 SDS에 강A 영역내의 데이터가 갱신된 것을 의미한다. 서버에서 갱신된 데이터는 DOS에 포함되므로,  $TDOS \cap DOS \neq \emptyset$ 는 강A.LST 시간 이후에 서버에 갱신된 데이터와 클라이언트에서 변경한 데이터가 동일하다는 것을 의미한다. 따라서 동일한 객체에 대해서 클라이언트와 서버 모두 변경이 있었기 때문에 데이터 일관성을 유지할 수가 없다. 이 논문에서는 데이터 일관성을 유지하기 위해서 서버에 먼저 갱신된 데이터를 유지하며, 클라이언트에서 변경된 데이터는 무시한다.

그림 4는 필드 업데이트를 지원하는 양방향 동기화 프로토콜을 보여주고 있다. 양방향 동기화 프로토콜 진행 순서는 다음과 같다. 클라이언트에서 서버로 동기화를 요청한다. 요청하는 형태는 클라이언트에서 변경사항이 있는 경우와 없는 경우 2가지로 구분할 수 있다. 클라이언트의 변경사항이 없는 경우는 동기화 프로토콜이 아주 간단하다. 클라이언트의 관심 있는 CR영역을 서버에 요청하고 서버에서는 요청한 CR영역에

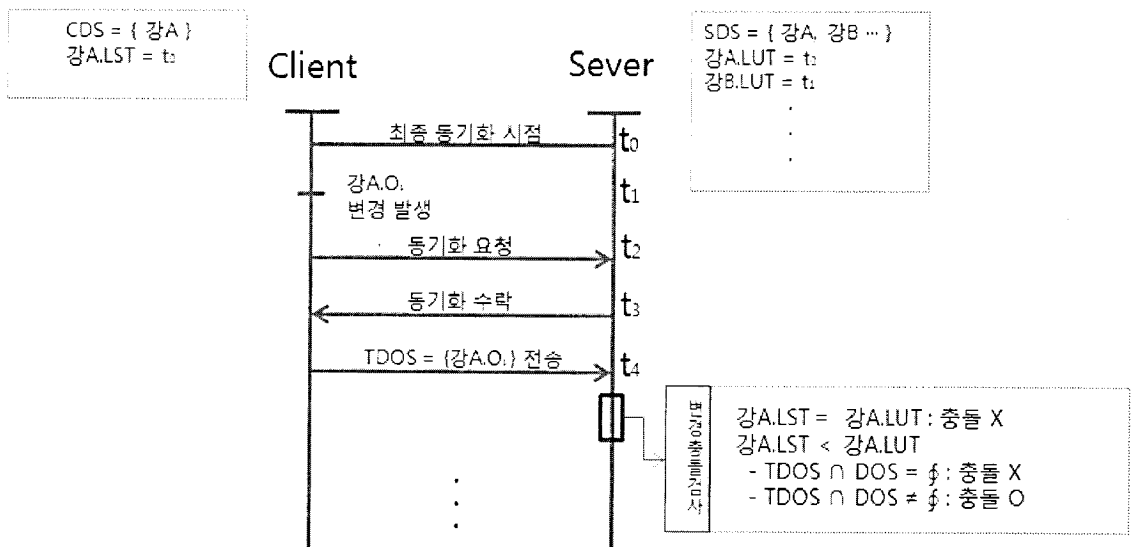


그림 3. 클라이언트의 변경요청에 의한 변경충돌 검사  
Fig 3. An Update conflict Inspection about client Update request

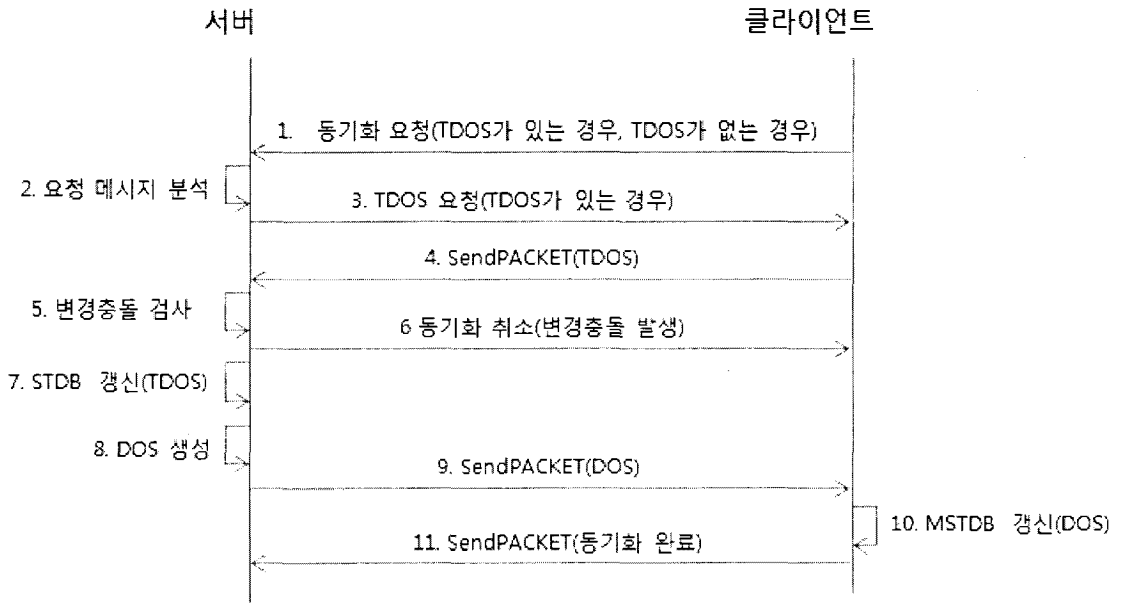


그림 4. 필드 업데이트를 지원하는 양방향 동기화 프로토콜  
Fig 4. A two-way synchronization protocol supporting field update

데이터를 클라이언트로 전송한다. 클라이언트에서 변경사항이 있는 경우 서버에서 변경충돌검사 작업을 수행한다. 서버는 클라이언트로부터 전송받은 TDOS가 변경충돌이 없으면 서버 STDB에 갱신하고, 갱신된 영역에 DOS를 생성하고 클라이언트에게 전송한다. 이때, 변경충돌이 있으면 클라이언트로부터 전송받은 TDOS에 대해서 갱신하지 않는다. 클라이언트에서 변경한 임시데이터 TDOS는 서버와 동기화작업을 수행함으로써 임시 데이터가 아닌 정식데이터로 갱신된 것이다.

트 동기화 시스템은 통신\_매니저, 동기화\_매니저, 동기화\_어댑터, MSTDB로 구성되어 있다.

#### IV. 구현된 시스템

##### 4.1 시스템 구조

구현된 시스템은 서버 동기화 시스템과 모바일 클라이언트 동기화 시스템으로 구성된다. 서버 동기화시스템에는 통신\_매니저, 동기화\_매니저, 변경\_충돌\_매니저, 동기화\_어댑터, STDB로 구성되어 있으며, 클라이언

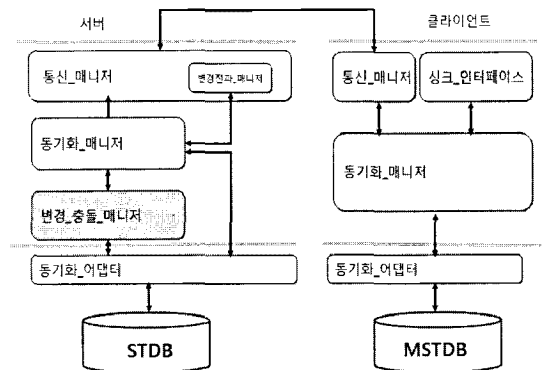


그림 5. 시스템 전체 구성도  
Fig 5. The System configuration

그림 5는 시스템 전체 구성도를 보여주고 있다. 시스템의 각 구성 요소의 역할은 다음과 같다. 통신\_매니저(클라이언트)는 서버의 통신\_매니저와 통신을 수행

한다. 동기화\_매니저로부터 받은 데이터를 전송하거나 서버로부터 받은 데이터를 동기화\_매니저로 전송한다.

동기화\_매니저(클라이언트)는 동기화 작업 수행에 핵심을 담당한다. 동기화\_어댑터에게 동기화할 CR영역을 요청하고 받은 데이터를 통신\_매니저에게 알리고, 통신\_매니저로부터 받은 서버의 동기화 데이터를 동기화\_어댑터에게 전송하여 동기화를 수행하는 역할을 한다.

통신\_매니저(서버)는 클라이언트의 통신\_매니저와 통신을 수행한다. 동기화\_매니저로부터 받은 데이터를 전송하거나 클라이언트로부터 받은 데이터를 동기화\_매니저로 전송한다.

동기화\_매니저(서버)는 동기화 요청 메시지를 받았을 때 동기화할 CR을 동기화\_어댑터에게 요청을 하고 받은 CR 영역을 패킷화 시켜서 통신\_매니저로 전송하는 역할, 통신\_매니저로부터 받은 클라이언트 측의 동기화 데이터를 변경\_충돌\_매니저로 전송하는 역할을 한다.

변경\_충돌\_매니저(서버)는 동기화\_매니저로부터 받은 클라이언트의 TDOS 변경충돌 검사를 수행한다. 변경충돌이 없을 경우에는 동기화\_어댑터로 TDOS를 전송을 한다. 변경충돌이 발생하게 되면 동기화 작업이 취소된다.

동기화\_어댑터는 제안하는 시스템이 DB의 종류에 독립적으로 구현될 수 있도록 한다. 즉 동기화\_어댑터는 특정 DB에 의존적인 입출력을 담당하고, 동기화 시스템에서는 동기화\_어댑터의 입출력 API를 이용하여 DB와의 입출력을 처리하므로, DB의 종류에 독립적인 동기화 시스템이 가능하다.

#### 4.2 구현 결과

구현된 시스템의 서버환경은 Dual Core 3.00GHz, 2.00GB RAM 200GB S-ATA hard Disk이고, 클라이언트 환경은 Intel PXA270 624MHZ, 256MB RAM으로 되어 있다. 구현된 결과를 제안한 프로토콜을 평가하기 위해서 인터넷 설치 기사의 작업을 시나리오로 구현하였다. 인터넷 설치기사 A는 오전, B는 오후로 나누어 같은 구역 내에서 설치 작업을 진행한다. A가 진행한 작업은 모바일 단말기를 통해 현장에서 즉시 서버와 동기화 된다. B는 작업을 시작하기 전 서버와 동기화를 통

해 오전에 A기사가 작업한 정보를 얻고 필요한 작업을 수행한다.

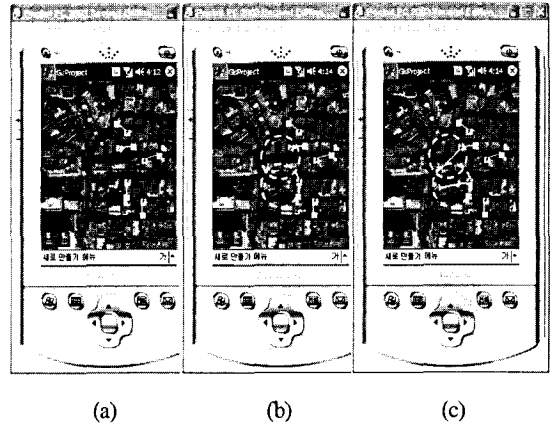


그림 6. 동기화 시나리오의 구현  
(a)동기화 이전 (b)데이터 수정 (c)동기화 완료  
Fig 6. An implementation of synchronization scenario  
(a) Before Sync (b) Data Update (c) After Sync

그림 6은 설치기사 A가 작업을 수행하는 단계를 나타내고 있다. 그림 6.(a)는 A기사가 작업을 하기 위해 최초로 서버와 동기화를 수행하여 전송받은 데이터이다. 그림 6.(b)는 설치기사 A가 가정집 2곳에 인터넷을 설치하고, 서버와 동기화를 수행하기 전에 임시데이터 형식으로 연결선이 빨간색으로 표시되고 있다. 원으로 표시되는 영역이 인터넷 설치된 부분이다. 그림 6.(c)는 A기사가 설치한 곳에 대해서 서버와 동기화를 수행하여 설치를 완료한 모습이다. 그림 6.(b)에서 임시로 표시된 영역이 서버와 동기화를 진행하면서 연결선이 빨간색에서 청록색으로 표시되며 서버와 동기화가 되었다. 설치기사 B는 오후에 설치기사 A가 작업한 영역 그림 6.(c)를 다운 받아 오후 작업을 수행한다.

## V. 결 론

기존의 모바일 GIS 서비스는 최신의 공간 데이터를 빠르게 제공하기 위해 무선 네트워크를 통하여 업데이트를 수행 하였다. 기존의 방법은 서버에서 클라이언트로 업데이트하는 시간을 줄였다. 하지만 이 방법은 실제계의 변경사항이 수집하는 시기를 고려하지 않아 결과

적으로 사용자에게 실세계의 변경사항이 업데이트 되는 시간이 오래 소요된다. 따라서 이 논문에서는 실세계의 변경사항을 모바일 기기를 이용하여 수집하고, 무선 네트워크를 이용하여 클라이언트에서 서버와 동기화하는 시스템을 제안하였다. 이 시스템은 클라이언트에서 변경된 데이터를 바로 수집하고 서버와 동기화하기 때문에 클라이언트에게 최신의 공간 데이터를 빠르게 제공할 수 있다. 향후 동일한 공간 데이터에 대해 다수의 클라이언트의 변경을 효율적으로 처리할 수 있는 연구가 필요하다[7].

### 참고문헌

- [1] 두용재, 진성일, “텔레매틱스 환경에서 모바일 단말과 중앙 서버간 파라미터를 이용한 동기화 기법 연구,” 한국정보과학회, 정보과학회 논문지: 데이터베이스 제32권 제4호, 2005. 8, pp.416~428.
- [2] 이혜진, 김진석, “모바일 환경에서 공간데이터 동기화 시스템 설계,” 한국정보과학회, 2004년 가을 학술 발표논문집 제31권 제 2호, 2004. 10, pp. 184~186.
- [3] “ActMAP White Paper and interfaces to the FeedMAP framework,” white paper, 2007.
- [4] 최진오, 홍봉희, “중복된 공간 데이터 관리를 위한 공동작업 트랜잭션 모델,” 한국정보과학회, 한국정보과학회 학술발표논문집, 한국정보과학회 1997년 봄 학술발표논문집 제 24권 제1호, 1997. 4, pp. 88~92.
- [5] Shuai hao, Hock Beng Lim, “Data Synchronization in Distributed and Constrained Mobile Sensor Networks,”UIC2007, LNCS 4611, pp.673~683, 2007
- [6] Mi-seon Choi, Young-Kuk Kim, Juno Chang, “Transaction-Centric Split Synchronization Mechanism for Mobile E-Business Applications,” IEEE,2005
- [7] 김홍기, 김동현, “모바일 환경에서 시공간 데이터의 동시 양방향 동기화 기법”, 한국해양정보통신학회 2008 춘계종합학술대회 논문집, Vol12, No.1, pp.171-174, 2008

### 저자소개

김홍기(Hong-Ki Kim)

한국해양정보통신학회 논문지  
제12권 제12호 참조

김동현(Dong Hyun Kim)

한국해양정보통신학회 논문지  
제13권 제11호 참조

조대수(Dae-Soo Cho)

한국해양정보통신학회 논문지  
제13권 제11호 참조