편재형 공간에서 사용자 이동성을 지원하는 위치 인식 VOD 서비스의 구현

(Implementation of Location-Aware VOD Service supporting User Mobility in Ubiquitous Spaces)

최 태 욱 † 정 기 동 ††
(Tae Uk Choi) (Ki Dong Chung)

요 약 편재형 공간에서 사용자는 자유롭게 이동하면서 VOD(Video On Demand) 서비스를 제공받기 원한다. 그러나 기존의 VOD 시스템은 사용자의 위치에 종속적이기 때문에 한 세션이 종료될 때까지 서버는 한 클라이언트에게만 데이타를 전송한다. 그래서 만일 사용자가 이동한다면 이전 세션을 종료시키고 새로운 클라이언트에서 다시 세션을 연결해야 한다. 그러나 위치 인식 VOD 시스템은 세션의 연결과 종료를 자동으로 수행함으로써 사용자 이동성을 지원한다. 즉, 사용자의 위치 이동을 자동으로 인식하고 사용자 근처의 클라이언트로 데이타 전송 흐름을 바꾸어 연속적으로 서비스를 제공한다. 본 연구는 편재형 공간에서 이동 중인 사용자에게 연속적인 서비스를 제공하기 위하여 위치 인식 VOD 서비스 구조와 세션 핸드오프 기법을 제안하고 자바(Java)와 지니(Jini)를 기반으로 위치 인식 VOD 프로토타입 시스템을 구현한다. 그리고 실험에서 제안된 핸드오프 기법이 다른 기법에 비해 적은 핸드오프 시간을 가짐을 보인다.

키워드 : VOD, 사용자 이동성, 위치 인식, 편재형 공간

Abstract In ubiquitous spaces, a user wants to be provided with a VOD service while moving one space to another freely. Since the traditional VOD system is dependent on user location, the VOD server transmits video data to a single client during a session. If the user moves to another space, he/she should close the old session and make a new request for the same video. However, the location-aware VOD system supports user mobility by closing and opening the session automatically. That is, the VOD system automatically perceives the movement of a user and allows the server to change the data flow so that a client near the user can receive the video data. This paper proposes a location-aware VOD service architecture and a session handoff scheme to provide a mobile user with continuous video delivery and implements a location-aware VOD prototype system based on Jini and Java. In experiment, we show that the proposed handoff scheme has a smaller handoff delay than the other handoff scheme.

Key words : VOD, User Mobility, Location Awareness, Ubiquitous Spaces

1. 서 론

종래의 VOD 시스템은 사용자 위치에 종속적이다. 즉, 사용자가 직접 컴퓨터로 가서 클라이언트를 통해 비디오를 요구하고 서버는 세션을 생성하고 비디오 데이타를 전송한다. 만일 사용자가 다른 위치로 이동하여 그 비디오를 계속 보고싶다면 사용자는 현재 컴퓨터에서 비디오 재생을 멈추고 새로운 컴퓨터로 이동한 후 다시 비디오를 요구해야 할 것이다. 또한 계속해서 비디오를 시청하기 위해서 사용자는 이전 위치에서 마지막으로 재생된 프레임 위치를 탐색해야 한다. 그러나 위치 인식 VOD 시스템은 사용자 이동성(User Mobility)[1-7]을 지원한다. 즉, 사용자의 위치를 자동으로 인식하여 사용자 근처의 클라이언트로 데이타 전송 흐름을 변경 할 수 있다. 편재형 공간에서 사용자 이동성은 모바일 환경에서의 호스트 이동성(Host Mobility)[8-12]과는 개념이 다르다. 모바일 환경에서는 사용자가 휴대하는 무선 단말에게 끊김없이 데이타를 전송하는 것이 중요하다. 그러나 편재형 환경에서는 무선 단말에게 서비스를 제공하는 것이 아니라 도처에 편재된 컴퓨터들이 공동으

† 비 회 원 : 부산대학교 전자계산학과
tuchoi@pusan.ac.kr
†† 종신회원 : 부산대학교 전자전기정보컴퓨터공학부 교수
kdchung@melon.cs.pusan.ac.kr
논문접수 : 2004년 2월 24일
심사완료 : 2004년 11월 3일

로 협력하여 한명의 이동 사용자에게 서비스를 제공한다. 즉, 호스트 이동성에서 통신의 주체는 이동호스트(컴퓨터)이지만 사용자 이동성에서 통신의 주체는 사용자(사람)이라는 것이다. 궁극적으로 사용자 이동성은 호스트 이동성을 포함한다[3].

그림 1은 편재형 환경에서 VOD 시스템이 고려해야 할 사용자 이동 시나리오를 보여준다. 사용자는 편재형 공간 A에서 비디오 시청을 시작한다. 이 후 사용자는 공간 A에서 공간 B로 이동한다. 이동 중에 사용자는 PDA를 통해서 비디오 시청을 계속한다. 물론 비디오 화질이 매우 떨어지거나 음성만을 수신할 수도 있다. 공간 B로 이동한 사용자는 공간 B에 있는 컴퓨터를 통해서 비디오 시청을 계속해서 수행한다.

이러한 시나리오는 VOD 시스템에게 새로운 의미를 부가한다. 다시 말해, VOD 시스템은 사용자의 이동을 감지할 수 있어야 하고 사용자의 개입 없이 사용자가 이동한 호스트로 데이타 흐름을 바꿀 수 있어야 한다. 그러나 기존 연구들[1-3]은 한 사용자가 한 호스트에서만 인식된다는 가정을 하기 때문에 이러한 이동 시나리오를 완벽히 지원해줄 수 없다. 본 논문은 사용자 위치가 여러 컴퓨터에서 동시에 인지될 수 있다는 것을 고려하여 이동 사용자에게 연속적인 비디오 스트리밍을 제공할 수 있는 위치 인식 VOD 서비스를 구현한다. 이를 위해서 위치 인식 VOD 서비스 구조와 세션 핸드오프 기법을 제안한다. 또한, Air ID[13] 장비를 이용하여 사용자 위치 인식 모듈을 구현한다.

논문의 구성은 다음과 같다. 2장에서는 위치 인식 VOD 서비스를 위한 시스템 구조를 설명하고 3장에서는 사용자 이동성을 지원하기 위한 세션 핸드오프 프로토콜 기법을 제안한다. 4장에서는 위치 인식 VOD 시스템의 구현 사항을 기술하고, 5장에서는 구현된 시스템에서 핸드오프 시간을 측정한다. 마지막으로 6장에서 결론을 맺는다.

2. 위치 인식 VOD 서비스 구조

그림 2는 제안된 위치 인식 VOD 시스템의 구조를 보여준다. 이 시스템은 VOD Service, VOD Client, VOD Sender로 구성되며 두 개의 통신 인터페이스(지

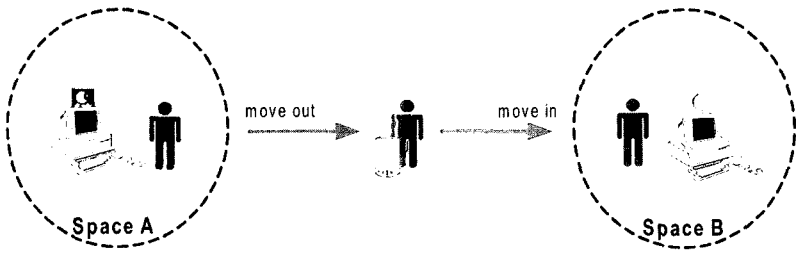


그림 1 편재형 환경에서 사용자 이동 시나리오

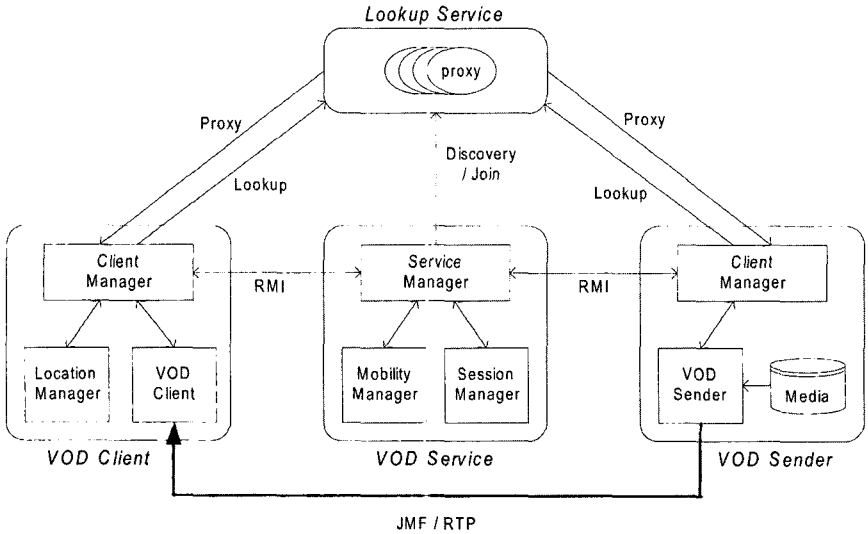


그림 2 위치 인식 VOD 서비스 구조

니[14,15]와 JMF[16])를 가진다. 지니(Jini)는 VOD 시스템의 제어 정보 통신을 위해서 사용되고 JMF(Java Media Framework)는 미디어 데이타의 전송과 재생을 위해 사용된다.

지니는 썬마이크로시스템즈(Sun Microsystems)에서 개발된 분산환경 플랫폼으로 분산 네트워크 환경에서 동적인 서비스의 생성과 구성을 용이하게 한다. 지니 응용은 크게 조회서비스(Lookup Service), 지니 서비스, 지니 클라이언트, 프락시(proxy)로 구성된다. 조회서비스는 일반 지니 서비스의 프락시 객체를 저장 관리하는 서비스이다. 지니 서비스는 특정한 기능을 수행하는 실체(entity)이며 사용자, 프로그램 또는 다른 서비스에 의해 사용된다. 이러한 서비스들은 발견/참여(Discovery/Join) 과정을 통하여 조회서비스를 찾고 자신의 프락시 객체를 조회서비스에 등록한다. 또한 지니 클라이언트는 조회서비스로부터 특정 서비스의 프락시를 다운로드 받아서 해당 서비스의 기능을 이용한다. 프락시 객체는 직렬화된 객체로서 지니 서비스에 대한 통신 인터페이스를 제공한다.

2.1 VOD Service

VOD Service는 VOD 시스템의 제어를 담당하는 지니 서비스이다. 즉, VOD 세션을 관리하고 사용자 이동성을 지원하며 VOD Client와 VOD Sender를 제어한다. Service Manager는 지니 서비스를 생성하고 자신의 프락시를 조회서비스에 등록한다. Mobility Manager는 VOD 클라이언트로부터 사용자 위치 정보를 전송받아 내부적으로 저장 관리한다. 저장되는 정보는 사용자의 위치, 각 클라이언트의 위치, 사용자 주위에 있는 호스트 정보 등을 포함한다. Session Manager는 각 사용자가 요구한 비디오 세션 정보를 관리한다. 이 세션 정보는 세션의 QoS 정보와 재생중인 비디오의 프레임 위치 정보를 포함한다. 핸드오프가 일어날 때 Session Manager는 마지막 재생된 프레임 위치를 기억하고 있다가 핸드오프가 끝나면 기억된 프레임 위치부터 다시 재개하도록 VOD Client에게 정보를 제공한다.

2.2 VOD Sender

VOD Sender는 VOD Client에게 비디오를 전송하는 역할을 담당한다. Client Manager는 조회서비스로부터 VOD 서비스의 프락시를 다운로드 받은 후 이 프락시를 통해 VOD Service와 통신한다. VOD Sender는 사용자가 요구한 비디오 데이타를 사용자 근처의 VOD Client에게 전송한다. 핸드오프가 일어날 때 VOD Sender는 VOD Client들의 요청에 따라 데이타 전송을 멈추고 다시 전송을 재개한다. 데이타 전송을 위해 VOD Sender는 JMF의 RTP를 이용한다.

2.3 VOD Client

VOD Client는 이동하는 사용자에게 비디오를 연속적으로 보여주는 기능을 수행한다. Client Manager는 VOD Sender의 Client Manager와 기능이 동일하다. 즉, Jini 기반 통신을 위한 인터페이스 역할을 한다. VOD Player는 실제 비디오를 화면에 재생하는 모듈로서 JMF를 이용해서 구현된다. Location Manager는 사용자 위치 정보를 수집하고 이를 VOD Service에게 전송한다. 본 연구에서는 Air ID[13] 장비를 이용하여 이 모듈을 구현한다.

3. 사용자 이동성 지원

3.1 주호스트(Primary Host)의 결정

편재형 공간에서 사용자는 여러 호스트 사이를 이동한다. 이 때 이동하는 사용자를 서비스하는 호스트는 사용자 근처의 호스트들 중에서 결정되어야 할 것이다. 우리는 사용자에게 서비스가 가능한 호스트들을 후보 호스트(Candidate Host)라고 정의하고 사용자에게 주된 서비스를 제공하는 호스트를 주 호스트(Primary Host)라고 정의한다. 그림 3은 위치인식기가 탑재된 3개의 호스트에서 한 사용자의 위치정보 획득에 대한 예를 보인다. 현재 RF 범위에서 사용자를 인식할 수 있는 호스트는 TV와 노트북(Notebook)이다. 따라서 후보 호스트 리스트는 CHL={TV, Notebook}로 표현될 수 있다.

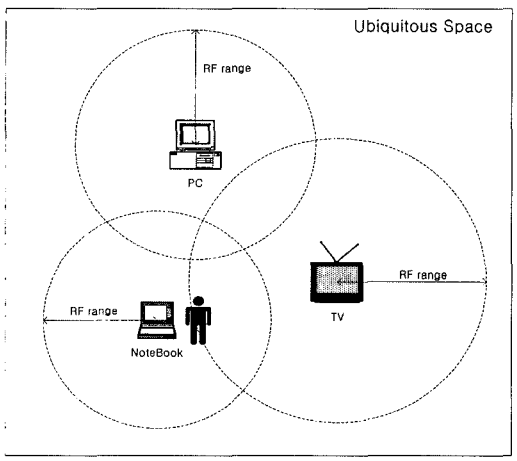


그림 3 편재형 공간에서 후보 호스트와 사용자 위치의 예

위의 예에서처럼 후보 호스트가 2개 이상일 경우 하나의 주 호스트를 선택해야 한다. 간단히 사용자와 가장 가까운 호스트를 주 호스트로 할 수 있고 성능이 가장 높은 호스트를 주 호스트로 결정할 수 있다. 어떤 사용자는 화면크기가 작은 PDA나 노트북보다는 거리는 좀 멀지만 화면이 크고 음향이 좋은 HDTV를 선호할 수

있고 어떤 사용자는 화면은 작지만 가까이서 볼 수 있는 노트북을 선호할 수 있다. 따라서 상황에 따라 사용자에게 가장 높은 만족도를 제공해줄 수 있는 호스트가 주 호스트가 되어야 한다. 본 연구는 사용자에게 최대의 QoS 만족도를 제공할 수 있는 호스트를 결정하기 위해서 평가함수(Evaluation Function)를 정의한다. 이 함수의 결과값은 시스템 사양, 사용자 선호도에 비례하고 사용자와의 거리에 반비례한다. 본 연구는 호스트의 사양과 사용자 선호도가 다음과 같을 때,

호스트 사양: $C^i = (c_1, c_2, c_3, ...) = (c^i_{Monitor}, c^i_{CPU}, c^i_{Bandwidth}, ...)$

사용자 선호도: $W = (w_1, w_2, w_3, ...) = (w_{Monitor}, w_{CPU}, w_{Bandwidth}, ...)$, $\sum w_i = 1$

호스트 H_i에 대한 평가 함수를 식 (1)과 같이 정의한다. N은 후보 호스트의 개수이다. l는 사용자와의 거리를 나타내며 $c_j^{\max}$는 리소스 c_j의 최대값으로 편재형 공간 안에 존재하는 호스트들 중에서 가장 큰 값을 의미한다.

$$E(H_i) = \frac{\sum_{j=1}^{N} (\frac{c_j}{c_j^{\max}} \times w_j)}{\sqrt{l}} \quad (1)$$

예를 들어 어떤 편재형 공간에서 후보호스트 리스트가 {PC, Notebook}이며 표 1과 같은 조건을 가진다고 가정할 때,

표 1 평가 함수 E의 값을 구하기 위한 조건의 예

조건	값
최대 사양 값	$C^{\max} = (c^{\max}_{Monitor}, c^{\max}_{CPU}, c^{\max}_{Bandwidth})$ = (30inch, 2.0GHz, 100Mbps)
사용자 선호도	$W = (w_{Monitor}, w_{CPU}, w_{Bandwidth})$ = (0.5,0.3,0.2)
PC의 사양	$C^{PC} = (c^{PC}_{Monitor}, c^{PC}_{CPU}, c^{PC}_{Bandwidth})$ = (19inch, 1.0GHz, 100Mbps)
PC와의 거리	1.0 m
Notebook의 사양	$C^{NB} = (c^{NB}_{Monitor}, c^{NB}_{CPU}, c^{NB}_{Bandwidth})$ =(15inch, 800MHz, 10Mbps)
Notebook과의 거리	0.5 m

각 호스트의 평가함수 E의 값은 다음과 같이 계산된다.

$E(PC) = \{(19/30)0.5 + (1024/2048)0.3 + (100/100)0.2\} / \sqrt{1.0} = 0.666$

$E(NB) = \{(15/30)\ 0.5 + (800/2048)\ 0.3 + (10/100)\ 0.2\} / \sqrt{0.5} = 0.547$

따라서 이 경우 노트북(NB)이 거리는 가깝지만 PC의 평가함수 값이 더 크므로 PC를 주 호스트로 결정한다.

3.2 세션 핸드오프

새로운 주호스트가 결정된 후에는 이전 주호스트(Primary Host)와 새 주호스트 사이에 세션의 이동 또는 핸드오프가 필요하다. 그림 1의 시나리오에서 볼 수 있듯이 사용자가 이동함에 따라 주호스트는 PC에서 PDA로 그리고 PDA에서 PC로 바뀐다. 이것은 VOD 세션의 수신자가 PC에서 PDA로 그리고 PDA에서 PC로 바뀜을 의미한다. 또한, 서비스 관점에서 볼 때 세션 핸드오프는 핸드오프 시점에서 서브 세션 (sub-session)이 새로 생성되고 소멸되는 과정으로 볼 수 있다. 그림 4는 그림 1의 시나리오에서 나타나는 3개의 서브세션을 보여준다.

세션 정보는 송신자와 수신자의 IP주소 그리고 데이타 파일로 구성되며 서브세션 정보는 송신자와 수신자의 IP 주소 그리고 미디어 파일 내의 마지막 재생 프레임의 위치를 포함한다. 본 연구는 세션과 서브세션을 그림 5와 같이 표현한다.

```
Session ( Sender, Receiver, Movie)
SubSession ( Sender, Receiver, Movie, Position)
```

그림 5 세션과 서브세션의 표현

세션 정보는 VOD Service의 Session Manager가 관리하고 서브세션은 VOD Client와 VOD Sender간에 설정되고 해제된다. 예를 들어 그림 4의 서브세션들은 다음과 같이 표현될 수 있다.

SubSession("164.125.165.129", "164.125.165.144", "test.mpg", 0)
SubSession("164.125.165.129", "164.125.165.145", "test.mpg", 1289)
SubSession("164.125.165.129" "164.125.165.146", "test.mpg", 2847)

세션 핸드오프 과정은 사용자 이동 패턴에 영향을 받는다. 따라서 편재형 공간에서 일어날 수 있는 사용자 이동 패턴을 아래와 같이 가정한다.

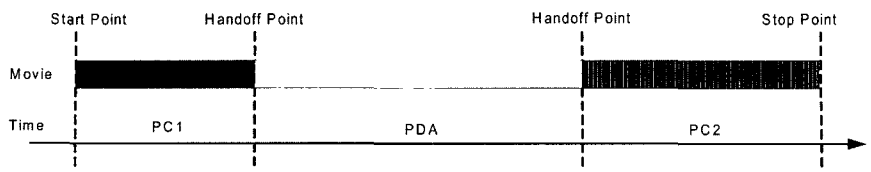


그림 4 핸드오프와 서브세션의 관계

- 이동패턴 I : 사용자가 편재형 공간 사이를 이동할 때 단말을 휴대하지 않고 이동하는 경우로 이동 중에 VOD 서비스를 받을 수 없다.
- 이동패턴 II : 사용자가 편재형 공간 사이를 이동할 때 무선 단말을 휴대하고 이동하는 경우로 이동 중에도 VOD 서비스를 계속해서 받을 수 있다.

3.2.1 하드(Hard) 핸드오프 기법

하드 핸드오프 기법은 사용자 이동패턴 I을 지원한다. 즉, 사용자가 하나의 공간에서 비디오를 시청하다가 다른 공간으로 이동한 후 다시 비디오를 시청하는 경우이다. 그림 6은 하드 핸드오프 과정을 보여준다. 먼저 사용자는 Client A에서 VOD Service에게 비디오를 요구하고(Step 1) VOD Service는 주호스트를 Client A로 결정한다(Step 2). Client A는 바로 서브세션 연결을 위한 메시지를 VOD Sender에게 보내고(Step 3) VOD Sender는 서브세션 연결 후 Client A에게 비디오 데이타를 전송한다(Step 4). 사용자가 Client A의 영역 밖으로 나갈 때 Client A는 VOD Service에게 'move out' 이벤트를 보낸다(Step 5). VOD Service는 새로 주호스트 결정하고 Client A에게 주호스트의 변경 사항을 알린다(Step 6). 주호스트가 존재하지 않을 경우 서비스는 중단된다. 이 경우 Client A는 VOD Sender에게 서브세션의 해제를 요청한다(Step 7). VOD Sender는 서브세션의 종료 후 마지막 전송 프레임의 위치를 VOD Service에게 반환한다. 사용자가 Client B의 영역 안으로 들어올 때 Client B는 'move in' 이벤트를 VOD Service에게 보낸다(Step 8). VOD Service는 새로 주호스트를 결정하고 Client B에게 주호스트 정보와 시작 프레임 위치를 보낸다(Step 9). Client B는 VOD Sender에게 서브세션 연결을 요청하고(Step 10) VOD Sender는 시작 프레임 위치를 찾아 데이타 전송을 재개한다(Step 11). 사용자가 비디오 시청을 끝낼 때 Client B는 VOD Service에게 전체 세션의 종료를 요구하고(Step 12) VOD Service는 VOD Sender에게 서브세션의 종료를 명한다(Step 13).

3.2.2 소프트(Soft) 핸드오프 기법

[1]-[3]의 연구들은 한 사용자가 한 호스트에서만 인식된다는 가정하에 하드 핸드오프 기법만을 기술하고 있다. 그러나 현실적으로 한 사용자는 여러 호스트에서 인식될 수 있으며 무선단말을 휴대하고 이동이 가능하다. 그래서 본 연구는 사용자 이동패턴 II를 지원할 수 있는 소프트 핸드오프를 제안한다. 그림 7은 사용자가 Client B(노트북)를 들고 Client A(데스크탑)로 이동 할 때 발생하는 소프트 핸드오프 과정을 보여준다. 사용자는 Client A를 통해 VOD Service에게 비디오를 요구하고 VOD Sender로부터 비디오 데이타를 전송 받는다(Step 1,2,3,4). 사용자가 새로운 공간으로 이동할 때 Client B가 사용자를 인식하고 'move in' 이벤트를 VOD Service에게 전송한다(Step 5). VOD Service는 주호스트를 Client B로 결정하고 Client A에게 주호스트 변경 정보를 알린다(Step 6). Client A는 바로 VOD Sender에게 서브세션의 중지를 요청한다(Step 7). VOD Sender로부터 마지막 프레임 위치가 반환되면 VOD Service는 Client B에게 주호스트 결정 정보를 알리고(Step 8) Client B는 VOD Sender에게 새로운 서브세션 연결을 요청한다(Step 9). VOD Sender는 시작 프레임의 위치를 찾아 데이타 전송을 시작한다(Step 10). 이후의 과정은 하드 핸드오프와 동일하다. 하드 핸드오

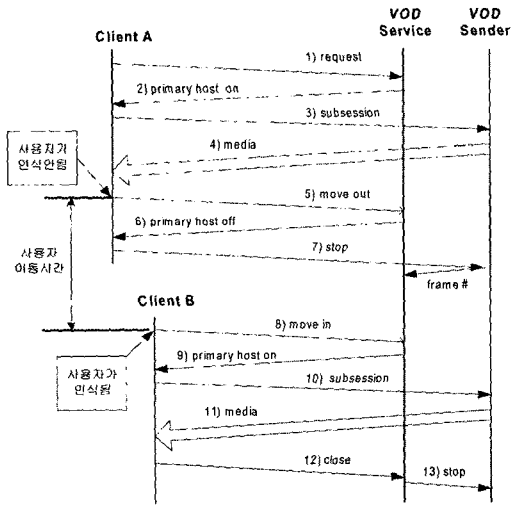


그림 6 하드 핸드오프 (Hard Handoff) 기법

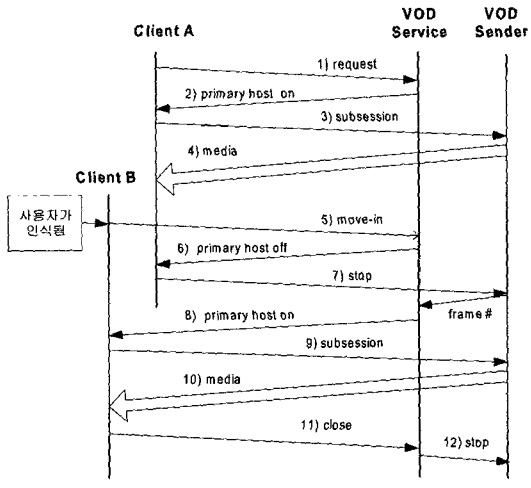


그림 7 소프트 핸드오프(Soft Handoff) 기법

프는 사용자가 하나의 공간을 벗어나거나 들어오는 시점에 발생하며 상대적으로 멀리 떨어진 두 호스트간에 일어난다. 반면 소프트 핸드오프는 후보호스트 목록이 변경되는 시점에 근접한 두 호스트간에 일어난다. 따라서 소프트 핸드오프의 지연시간은 하드 핸드오프의 지연시간보다 적다.

4. 위치 인식 VOD 시스템의 구현

4.1 클래스 구조

본 연구는 실제로 지니와 자바를 기반으로 위치 인식 VOD 시스템을 구현하였다. 그림 8은 VOD Service의 클래스 다이어그램을 보여준다. LAVODService는 DiscoveryListener 객체를 이용하여 Jini의 조회서비스(Lookup Service)를 찾고 ServiceRegistra 객체를 통해 LAVODProxy 객체를 조회서비스에 등록한다. LAVODServiceManager는 VOD Service의 핵심 제어 모듈로써 사용자의 위치를 추적 관리하는 MoblityMgr과 사용자에게 서비스중인 비디오 세션을 관리하는 SessionMgr을 포함한다.

그림 9는 VOD Client의 클래스 다이어그램을 나타낸다. LAVODClient는 DiscoveryListener를 통해 지니 조회서비스를 찾고 LAVODProxy 객체를 다운로드 받는다. 실제로 LAVODClient는 프락시 객체를 통해 LAVODService에게 이벤트 메시지를 전송하고 ServiceMsgListener를 통해 LAVODService로부터 이벤트 메시지를 수신한다. AVReceiver는 서브세션을 연결 및 해제하고 RTPManager를 통해 데이타를 수신하며 PlayerWindow를 통해 수신된 데이타를 재생한다.

그림 10은 VOD Sender의 클래스 다이어그램을 보여준다. 기본적인 구조는 VOD Client와 유사하다.

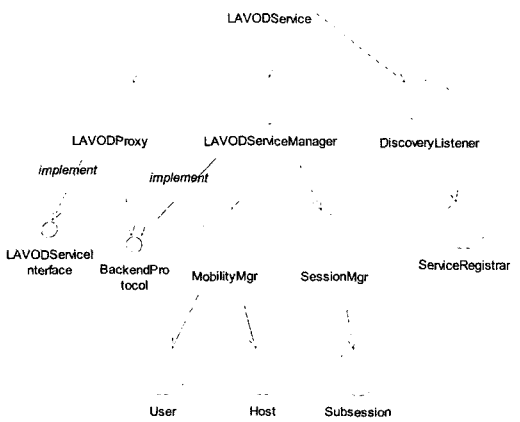


그림 8 VOD Service의 클래스 다이어그램

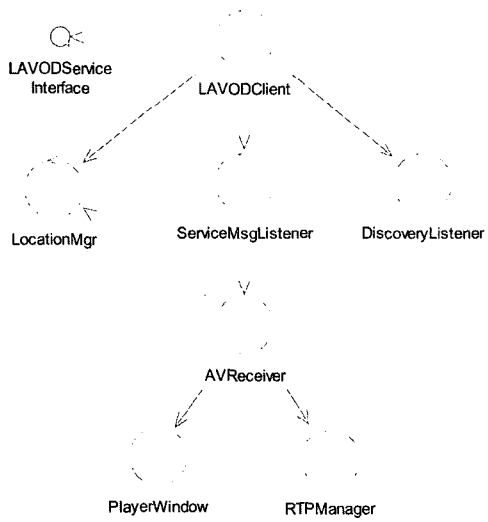


그림 9 VOD Client의 클래스 다이어그램

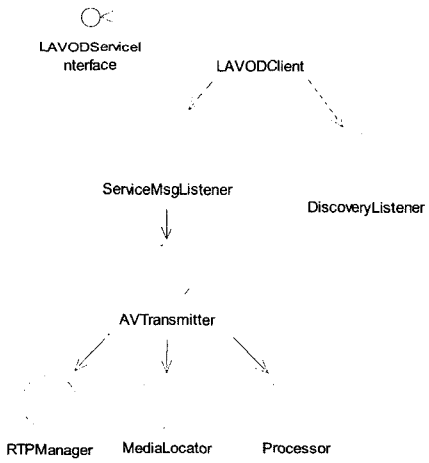


그림 10 VOD Sender의 클래스 다이어그램

LAVODSender는 프락시 객체를 다운로드하기 위한 DiscoveryListener 객체와 LAVODService에서 이벤트를 수신하기 위한 ServiceMsgListener 객체를 포함한다. ServiceMsgListener를 통해 서브세션 연결 이벤트가 수신되었을 때 AVTransmitter는 서브세션을 연결하고 RTPManager를 통해 데이타를 전송한다.

4.2 위치 인식 모듈의 구현

VOD Client의 LocationMgr 객체는 AIR ID LT Software Development Kit[13]를 이용하여 구현되었다. 그림 11에서 볼 수 있듯이 AIR ID 장치는 배지와 인식기로 구성되며 인식기는 일반 PC의 시리얼 포트에 연결된다. 사용자가 배지를 부착하고 인식기 근처로 다가오면 인식기는 자동으로 배지 ID를 인식하여 PC에

그림 11 AIR ID LT Software Development Kit

전송한다. 본 연구는 제공되는 C++ API를 이용하여 배지로부터 사용자 ID를 추출해낼 수 있는 프로그램(ExtractID.exe)을 구현하였다. 이 프로그램은 0.5초 간격으로 사용자 ID를 추출하여 로그파일(location.log)에 저장한다.

VOD Client의 Location Manager는 "ExtractID.exe"가 생성하는 로그파일로부터 사용자 ID 정보를 검색하는 함수를 구현한다. 다음은 Location Manager 클래스의 getID() 함수를 보여준다.

```
Public static String getID() {
  FileInputStream fis = new FileInputStream("location. log");
  BufferedReader ds =
    new BufferedReader(new InputStreamReader (fis));
  String ID = ds.readLine();
  Fis.close();
  Return ID;
}
```

Location Manager 클래스는 독립된 쓰레드로 동작하며 0.5간격으로 getID()를 반복적으로 호출하고 검색된 ID 정보를 통해 사용자 이동을 판단하는 프로그램이다. 다음은 이 쓰레드의 Run() 함수를 보인다.

```
  Public void Run() {
    While (!_finished) {
      Thread.sleep(500);
      Id = getID();
      if ((LastId == NULL) && (Id != NULL))
        SendMoveInEvent(Id);
      else if ((LastId != NULL) && (Id == NULL))
        SendMoveOutEvent(LastId);
      else if ( (LastId != NULL) && (Id != NULL)&&
        (!Id.equals(LastId)) )
        ArrivedNewUser(Id);
      LastId = Id;
    }
  }
```

사용자의 이동에 따라 반환되는 ID값은 달라진다. Run() 함수는 ID 값의 변화를 보고 사용자 이동성을 판단한다. 만일 ID가 NULL에서 유효한 값으로 바뀌었다면 "move-in" 이벤트를 발생시킨다. 또한 ID가 유효한 값에서 NULL로 바뀌었다면 "move-out" 이벤트를 발생시킨다. 그리고 ID와 LastID가 NULL이 아니고 서로 다르다면 새로운 사용자가 인식되었음을 VOD Service에게 알린다.

4.3 프레임의 위치의 추출과 탐색

VOD Sender의 AVTransmitter는 비디오 데이타 내에서 특정 프레임에 대한 임의접근이 가능해야 한다. 즉, 새로운 서브세션이 시작될 때 시작 프레임의 위치를 찾을 수 있어야 하고 서브세션이 끝날 때 마지막 재생 프레임의 위치를 추출할 수 있어야 한다. JMF에서 프레임 위치정보를 얻기 위해서는 재생기가 시작되기 전에 FramePositioningControl에 대한 레퍼런스를 얻어야 한다.

```
FramePositioningControl fpc = (
  FramePositioningControl)player.getControl(
  "javax.media.media.control.framePositioningControl);
```

그리고 서브세션을 시작할 때 다음과 같이 마지막 프레임 위치를 검색한다.

```
  fpc.seek(startFrame);
```

미디어의 재생을 끝날 때 마지막 프레임의 위치는 다음과 같이 추출한다.

```
  int lastFrame = fpc.mapTimeToFrame(player.getMediaTime());
```

5. 실 험

Jini Development Kit 1.2.1과 JDK 1.3을 기반으로 구현된 위치 인식 VOD 시스템을 실행하기 위해서는 지니 응용에 필요한 웹서버, RMI 활성화 데몬, 조회서비스들이 먼저 수행되어야 한다. 이러한 서비스들은 여러 대의 호스트에서 분산적으로 수행이 가능하다. 그림 12는 Windows XP가 탑재된 노트북 환경에서 VOD Client의 실행 화면을 보여준다. 사용자의 위치가 인식되면 비디오 재생기는 자동으로 재생을 시작한다. 사용자가 새로운 호스트로 이동할 때 이전 호스트의 비디오 재생기는 자동으로 재생을 멈추고 새 호스트의 비디오 재생기는 자동으로 재생을 시작한다. 사용자는 버튼을 눌러 세션을 생성하고 종료시킬 수 있다.

구현된 VOD 시스템에서 제안된 세션 핸드오프의 성능을 알아보기 위해서 핸드오프 시간을 측정하였다. 실험을 위해 그림 13과 같은 테스트베드를 구축하였다. 테

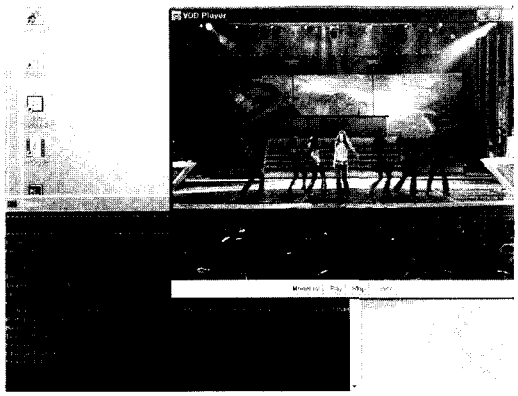


그림 12 VOD Client의 실행 화면

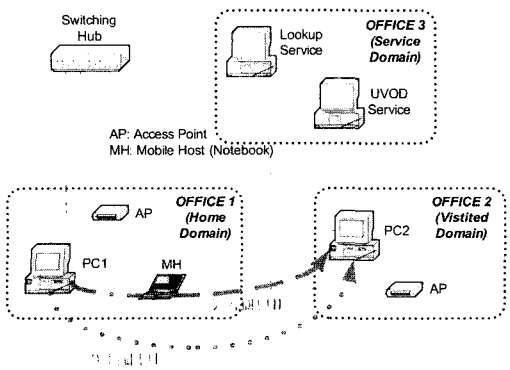
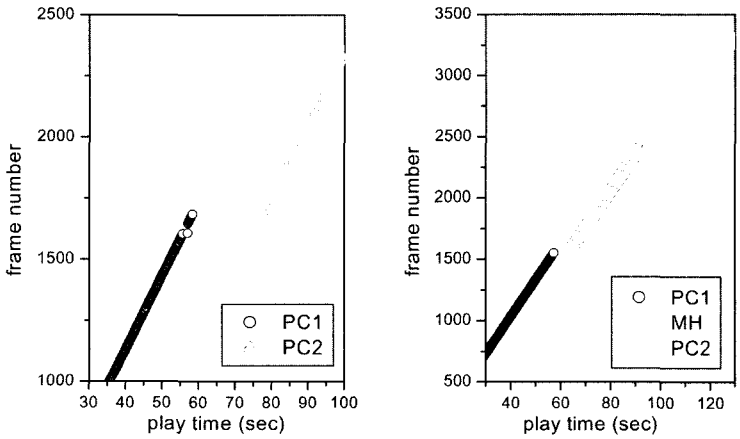


그림 13 핸드오프 성능을 측정하기 위한 실험환경

스트베드는 Home 영역(domain), Visited 영역, Service 영역으로 이루어진다. 사용자는 Home 영역에서 사용자 이동패턴에 따라 Visited 영역으로 이동한다. 이동패턴 I은 하드 핸드오프 기법을 위해 이동패턴 II는 소프트 핸드오프 기법의 측정을 위해 사용된다. Home 영역과 Visited 영역의 PC1, MH, PC2에는 VOD Client가 위치하고 Service 영역에는 VOD Service와 VOD Sender 그리고 지니의 조회서비스가 위치한다. PC1과 PC2의 사양은 동일하며(Pentium 1.3GHz) MH의 사양(Pentium 800MHz)보다 높으며 실험 데이타는 MPEG 4로 인코딩된 AVI 파일을 이용하였다.

핸드오프 시간은 이전 호스트에서 비디오 재생을 중지한 시각부터 새 호스트에서 비디오 재생을 재개한 시각의 차이를 의미한다. 그림 14는 VOD Client들에서 측정된 프레임 들의 재생시간을 보여준다. 핸드오프가 일어나는 시점에서 단절되는 프레임 재생간격이 하드 핸드오프 기법보다 소프트 핸드오프 기법이 훨씬 작음을 직관적으로 알 수 있다.

핸드오프시간을 자세히 분석하기 위해서 closing 시간, moving 시간, resuming 시간으로 나누어 측정하였다. Closing 시간은 이전 호스트에서 이전 서브세션을 종료하는데 걸리는 시간이고, moving 시간은 사용자가 하나의 호스트영역에서 벗어나 다른 공간의 호스트 영역으로 이동하는 시간을 의미한다. Resuming 시간은 새 호스트에서 비디오 재생을 재개하는 시간으로 이 시간은 서브세션을 연결하고 시작 프레임을 찾고 프레임을 버퍼링하는 시간을 포함한다. 표 1에서 볼 수 있듯이, closing 시간은 두 기법 모두 매우 작다. Resuming 시간은 데이타 버퍼링으로 인해 두 기법 모두 상당히 크지만 클라이언트 시스템 성능에 따라 약간 차이가 남을 보인다. 그러나 moving 시간은 소프트 핸드오프 기법에서는 필요 없지만 하드 핸드오프 기법에서는 매우



(a) 하드 핸드오프 기법 (b) 소프트 핸드오프 기법

그림 14 VOD Client에서 측정된 프레임 재생 시간

커서 전체 핸드오프시간이 상당히 크게 한다.

표 2 핸드오프 기법들의 핸드오프 시간

기법 / 시간	하드 핸드오프 (PC1->PC2)	소프트 핸드오프	
		PC1->MH	MH >PC2
Closing time	0.047	0.042	0.046
Moving time	10.953	0.0	0.0
Resuming time	6.156	7.998	6.547
Handoff time	17.156	8.040	6.593

6. 결 론

편재형 공간에서 VOD 시스템은 사용자 이동성을 지원할 수 있어야 한다. 본 논문에서는 편재형 공간에서 사용자 이동성을 지원할 수 있는 위치 인식 VOD 서비스 구조와 이를 위한 하드 핸드오프 기법과 소프트 핸드오프 기법을 제안하였다. 그리고 자바와 지니를 이용하여 위치 인식 VOD 프로토타입 시스템을 구현하였다. 그리고 실험에서 소프트 핸드오프 기법은 사용자 이동 시간을 없앰으로써 핸드오프 시간을 크게 줄일 수 있음을 보였다. 그러나 소프트 핸드오프도 상당한 지연시간을 수반하여 끊김 없는 비디오 서비스를 제공하기는 무리가 있다. 따라서 향후 끊김 없는 서비스 제공을 위해 핸드오프 시간을 더욱 줄일 수 있는 연구가 필요하다. 또한, VOD 서비스에 참여하는 호스트들의 사양과 자원 양에 따라 적응적으로 QoS를 조절할 수 있도록 핸드오프 기법을 보완할 필요가 있다.

참 고 문 헌

[1] Bo Zou, "Mobile ID Protocol: a Badge-Activated Application Level Handoff of a Multimedia Streaming to Support User Mobiltiy," Master's Thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaing, 2000.

[2] Long Wang, "MobAudio: Architecture of Distributed Audio on Demand Services with User Mobility Support," Master's Thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaing, 2001.

[3] Yi Cui, K. Nahrstedt, D. Xu, "Seamless User-level Handoff in Ubiquitous Multimedia Service Delivery," Multimedia Tools and Applications Journal, 2003.

[4] J. P. Sousa and D. Garlan, "Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments," IEEE/IFIP Conference on Software Architecture, Montreal, 2002.

[5] A. D. Stefano, C. Santoro, "NetChaser: Agent Support for Personal Mobility," IEEE Internet Computing 4(2): 74-79, 2000.

[6] H.J. Wang et al. "ICEBERG: An Internet-core Network Architecture for Integrated Communications," IEEE Personal Communications, Special Issue on IP-based Mobile Telecommunication Networks, 2000.

[7] G. Appenzeller et al. "The Mobile People Architecture," ACM Mobile Computing and Communication Review, Vol. 1, No. 2, 1999.

[8] Charles Perkins, "Ipv4 Mobility Support," Request for Comments 2002.

[9] David Johnson, "Scalable Support for Transparent Mobile Host Internetworking," Mobile Computing, 1996.

[10] Henning Schulzrinne and Elin Wedlund, "Application-Layer Mobility Using SIP," ACM Mobile Computing and Communication Review, vol. 1, no.2, 1999.

[11] D. A. Maltz and P. Bhagwat, "MSOCKS: An Architecture for Transport Layer Mobility," Proc IEEE Infocom '98, 1998.

[12] A. Snoeren and H. Balakrishnan, "An End-to-End Approach to Host Mobility," Proc. MobiCom '99. 1999.

[13] "Air ID System," in http://www.pcprox.com.

[14] Sun Microsystems, http://developer.java.sun.com/developer/products/jini

[15] Edwards, W. Keith, "Core Jini," Prentice Hall PTR, 1999.

[16] Sun Microsystems, http://java.sun.com/products/java-media/jmf

최 태 욱

1997년 동의대학교 전산통계학과 졸업(학사). 1999년 부산대학교 대학원 전자계산학과 졸업(이학석사). 2000년~현재 부산대학교 대학원 전자계산학과 박사과정. 관심분야는 유비쿼터스 컴퓨팅, 멀티미디어 통신

정 기 동

1973년 서울대학교 졸업(학사). 1975년 서울대학교 대학원 졸업(이학석사). 1986년 서울대학교 대학원 졸업(이학박사) 1990년~1991년 MIT대학 교환 교수 1995년~1997년 부산대학교 전자계산소 소장. 1999년~2001년 부산대학교 BK21 사업 단장. 1978년~현재 부산대학교 전자계산학과 교수. 관심분야는 병렬처리, 멀티미디어