

Mobile CORBA 환경에서 게이트웨이간의 경로최적화 핸드오프

(Route-optimized Handoff in Mobile CORBA Environment)

신혜령[†] 이형우^{**} 김주호^{***}
(Hyeryung Shin) (Hyungwoo Lee) (Juho Kim)

요약 본 논문에서는 ALICE(Architecture for Location Independent CORBA Environment) 구조에 기반한 Mobile CORBA 환경에서 이동 게이트웨이(MG)간의 핸드오프에 따른 문제점을 해결하기 위한 경로 최적화 핸드오프를 제안한다. 이동 호스트가 빠르게 움직이고, 처리시간이 긴 작업의 경우에 핸드오프로 인한 MG간의 긴 터널링 체인이 발생할 수 있고, 이로 인해 메시지들이 불필요한 몇 단계의 라우팅과 재전송 과정을 거치게 된다. 이렇게 비효율적인 경로를 통해서 메시지 전송이 이루어지는 것은 여러 가지 제약조건을 가지는 무선망의 환경에서는 커다란 장애요소가 될 수 있다. 본 논문에서 제안한 서로운 핸드오프 알고리즘은 핸드오프시 관련된 MG들이 이동성에 관한 정보를 메시지(binding update message) 형태로 받아서 캐쉬로 관리하도록 한다. 그 결과 핸드오프 이후의 관련 MG는 캐쉬를 이용하여 이동 호스트와 최적의 경로로 통신이 가능하다.

키워드 : 모바일, 코바, 핸드오프, 경로최적화, 게이트웨이

Abstract The routing protocols designed for wired networks can hardly be used for mobile ad-hoc networks due to limited bandwidth of wireless transmission and unpredictable topological change. Recently, several routing protocols for mobile ad-hoc networks have been proposed. However, when these protocols are applied to support real time services like multimedia transmission, they still have problems in ad-hoc networks, where the topology changes drastically. In this paper, we propose a new route selection algorithm which selects the most reliable route that is impervious to route failures by topological changes by mobile hosts. For reliable route selection, the concept of virtual zone (stable zone and caution zone) is proposed. The zone is located in a mobile node's transmission range and determined by mobile node's mobility information received by Global Positioning System (GPS). The proposed algorithm is applied to the route discovery procedure of the existing on-demand routing protocol, AODV, and evaluated by simulation in various traffic conditions and mobility patterns.

Key words : wireless mobile network, ad-hoc network, route selection, GPS

1. 서론

최근 무선 통신의 대중화가 급격히 이루어지면서, 무선 컴퓨팅에 대한 관심이 높아지고 있다. 또한, 이질적이고 자율적으로 분산되어 있는 분산환경에서 어플리케이션을 구현하는데 기반이 되는 프레임워크로서 CORBA

(Common Object Request Broker Architecture) 기술도 급격히 성장하고 있다. OMG(Object Management Group)에서 정의한 CORBA는 이질적인 분산환경에서 객체지향 응용을 구현하기 위한 기반구조이다[1]. CORBA 기반을 이용하면, 서로 다른 환경의 응용들을 마치 하나의 컴퓨터에 있는 것처럼 투명하게 사용할 수 있게 된다. 무선 통신 시장의 대중화에 따라 이러한 CORBA 응용들을 컴퓨터에서만 아니라, PDA, 핸드폰 등과 같은 이동 단말에서도 사용하고자 하는 요구사항이 증가할 것이다.

Mobile CORBA의 한 구조인 ALICE[3](Architecture for Location Independent CORBA Environment)

[†] 비회원 : ACCENTURE

ute-nirv@hanmail.net

^{**} 비회원 : 서강대학교 컴퓨터학과

feelg@sogang.ac.kr

^{***} 종신회원 : 서강대학교 컴퓨터학과 교수

jhkim@ccs.sogang.ac.kr

논문접수 : 2000년 9월 21일

심사완료 : 2002년 3월 29일

구조를 통해서 이동 호스트 상에서 동작하는 CORBA 객체들이 표준 CORBA 응용들과 IIOP(Internet Inter-ORB Protocol)를 통해서 통신할 수 있다. ALICE는 CORBA상의 다양한 분산환경의 모든 부분에 상위 응용 계층과 연계한 세션 계층을 이용하여 구현하는 접근방식으로 이동환경의 문제를 해결한다. 이동 호스트와 고정된 호스트사이의 이동성에 따른 문제를 중계해주는 역할을 MG가 수행하고, 이동호스트와 MG에는 각각 이동 계층이 존재해서 세션 계층으로서 이동성의 문제를 해결해 주게 된다. 그런데, 이동 호스트가 움직임에 따라, MG가 변경되고, 기존 MG와 새로운 MG간에는 상태정보와 캐쉬 데이터 정보를 교환하는 핸드오프를 위한 절차가 수행되게 된다. ALICE 구조에서의 핸드오프 과정은 이동 호스트가 빠르게 움직이고, 처리시간이 긴 작업의 경우에 핸드오프로 인해 MG간 터널링의 경로가 길게 이어질 수 있고, 핸드오프 이후 이동호스트의 응답 지연이 심화될 수 있는 단점을 가지고 있다. 이러한 응답지연은 mobile CORBA 응용 서비스를 구현하는데 있어 장애요소가 될 수 있다. ALICE 구조에서 핸드오프로 인한 비효율적인 메시지 전송경로를 최적화 할 수 있다면 mobile CORBA 구조가 더욱 완전해 질 수 있을 것이다.

그간 핸드오프와 관련해서는 무선망에서 이동 호스트와 기지국간의 핸드오프에 대한 많은 연구가 있어 왔다 [5, 6, 7, 8, 9]. 그렇지만, 이러한 연구들은 mobile CORBA 구조에서 MG간의 핸드오프와는 관점이 상이하다. 트랜스포트 계층에서 이루어지는 작업들로, 핸드오프 그 자체의 효율성을 추구하는 하위 레벨의 연구들이며, 또한 CORBA IIOP 통신에 대해서 언급하고 있지 않다.

이에 본 논문에서는 ALICE 구조에 기반한 mobile CORBA환경에서 MG간의 핸드오프를 효율적으로 수행할 수 있는 방법을 연구하고자 한다. mobile CORBA환경에서 핸드오프 발생시 기존 핸드오프 과정이 유발할 수 있는 긴 터널링의 문제와 비효율적인 여러 차례의 Redirect의 발생을 개선할 수 있도록 한다. 이를 위해 Mobile IP에서 Perkins가 제안한 경로 최적화 기법[10]을 응용하여, 경로 최적화 핸드오프(ROH : Route Optimization Handoff)를 제안한다.

본 논문은 다음의 순서로 구성되어 있다. 2장에서는 연구의 기반이 되는 mobile CORBA 구조와 핸드오프와 관련된 기존의 연구 결과들에 대해서 알아본다. 3장에서는 Mobile CORBA 환경에서 핸드오프로 인한 문제점을 해결하기 위하여 본 논문이 제안한 MG간의 새

로운 경로 최적화 핸드오프 알고리즘을 소개한다. 4장에서는 제안한 알고리즘을 성능 평가하기 위해 실험 환경을 설정하고, 제안한 알고리즘의 성능을 분석한다. 5장에서는 본 논문의 결과를 종합하여 기술한다.

2. 연구배경

2.1 Mobile CORBA 구조

CORBA ORB는 일반적으로 IIOP를 사용하여 통신한다[1]. IIOP는 ORB간의 통신을 위한 일반 규약인 GIOP(General Inter-ORB Protocol)를 TCP/IP에서 동작하도록 정의한 것이다. 즉, TCP/IP 네트워크 상에 연결되어 있는 ORB간의 통신을 위한 규약이다. CORBA 표준에서는 IIOP를 필수적으로 지원하도록 규정하고 있다. ORB간의 연동은 IIOP만을 지원하더라도 큰 무리는 없다. 이러한 IIOP를 이동 환경에서 수행 가능하도록 확장한다면, mobile CORBA 구현이 가능하게 될 것이다.

그렇지만, 현재 IIOP를 포함한 표준 CORBA 기술은 이동 컴퓨팅 환경에서 사용되도록 설계되지 않았다. 이동 컴퓨팅 환경에서 CORBA를 사용하는 것은 하드웨어 이동성과 무선망의 특성에 의해 몇 가지 문제를 일으킬 수 있다. 이러한 문제를 해결하기 위한 구조로 ALICE를 비롯하여 Mobile CORBA와 관련된 구조가 제시되었다[2, 3, 4]. 그 중 ALICE 구조는 응용계층과 연계한 세션 계층에서 이동성에 따른 문제를 해결하도록 하고, 기존 IIOP 표준을 수용하면서, 이동성에 따른 문제를 해결하는 별도의 계층을 제안하였다. 따라서 기존의 TCP나 IIOP의 별다른 수정 없이 Mobile CORBA환경을 구현할 수 있는 장점을 가진다. 이 장에서는 Mobile CORBA구조인 ALICE의 구조에 대해서 살펴본다.

2.1.1 ALICE 구조

ALICE는 계층적인 접근법을 택하여 기존의 TCP/IP 구조와 표준 IIOP 메시지 구조를 유지하면서 별도의 세션 계층으로 이동 환경에 따른 여러 가지 문제점을 해결하도록 하고 있다. 또한, 유선 네트워크와 무선 호스트 사이에 MG가 마치 유선 망과 무선 망 사이의 브릿지와 같은 역할을 하도록 하여 신뢰할 수 없고 낮은 대역폭을 가지는 무선망의 문제들을 해결한다. ALICE의 구조는 클라이언트나 서버가 존재하는 고정된 네트워크와 이동 호스트 사이에 MG를 두도록 하는 것이다. 여기에서 MG의 중요한 역할은 이동 호스트에 대한 프록시로써 동작하여 통신 요청을 받아서 이동 호스트에게 적절하게 전달해 주는 것이다. 또 다른 역할은 상위 계층에 대해서 주소 번역과 재전송과 같은 역할을 하게 된다.

ALICE의 각 계층 구조를 살펴보자. 이동 호스트 상에는 IIOP계층이 가장 상위 계층으로 존재하고, 하위 계층인 TCP/IP 계층과 IIOP계층 사이에 S/IIOP 계층과 이동성 계층이 있다. MG상에는 S/IIOP계층이 있어 이동 호스트의 S/IIOP계층과 논리적 연결을 맺고, 이동성 계층과 TCP/IP계층 또한 이동 호스트상의 이동성 계층, TCP/IP 계층과 각각 논리적 연결을 맺는다. 고정 호스트는 일반적인 유선 망에서와 마찬가지로 IIOP계층과 TCP/IP계층을 가지고 있고, TCP/IP계층은 MG와, IIOP계층은 이동 호스트와 논리적 연결을 맺게 된다. 각 계층이 담당하는 역할을 간략히 살펴보도록 한다. 먼저, 이동성 계층은 CORBA나 IIOP와는 독립적으로 이동성을 지원하는 역할을 담당한다. 끊어진 TCP 연결에 대해서 투명하게 재 연결하는 역할을 하여 상위 계층에서는 이동성에 따른 사항을 고려하지 않고 표준의 IIOP로 통신하도록 한다. 이동 호스트 상에서의 이동성 계층의 역할은 연결 요청에 대해서 MG상의 TCP/IP 포트를 하나 할당하는 것이다. 또한, MG간의 핸드오프를 담당하며, 상위 계층에 현재의 네트워크 연결 포인터를 알려준다. 두 번째, S/IIOP 계층은 이 정보를 이용하여 오브젝트 레퍼런스에 대한 변환작업을 수행하게 된다. S/IIOP 계층에서는 오브젝트 레퍼런스의 발행과 인코딩과 같이 IIOP 프로토콜에서 이동성과 관계 있는 부분에 대한 처리를 해준다. 일반적인 TCP/IP 상의 CORBA의 IOR을 변환하여, 이동성을 고려한 "Swizzled IOR"을 만든다. 또한 MG 상의 S/IIOP 계층에서는 SIOR(Swizzled IOR)을 가지고 들어오는 요청을 적절한 이동 호스트에 전송하게 된다. SIOR은 IOR이 이동 호스트 자체의 호스트명 대신에 이동호스트가 오브젝트 레퍼런스를 생성하는 시점에 연결된 MG의 (호스트명, 포트번호) 프로파일을 가지는 것을 말한다. 마지막으로, IIOP 계층은 이동성과 관계없이 CORBA 표준 IIOP 프로토콜을 구현한 계층이다. 이 계층은 응용 계층에서 다른 CORBA 응용이나 IIOP로 적용된 오브젝트들과 서로 통신할 수 있도록 한다. 이것은 TCP/IP나 이동성 계층, S/IIOP 계층 위에 직접 제공될 수 있도록 하는 표준 소켓 인터페이스로 구현되어야 한다.

IIOP 프로토콜은 클라이언트/서버에 기초해서 만들 수 있는 소켓 프로그램으로 구현된다. 클라이언트가 연결을 요청하여 Request를 보내고 서버로부터 Reply를 받는다. 먼저 각 IIOP API를 IDL로 작성한 후 C++ 클래스로 매핑 한다. 이 API는 메시지와 끝점의 개념을 기반으로 한다. 설계는 메시지 표현, 데이터 마샬링, 전송 클래스, 통신 끝점의 4개 부분에 대해서 이루어진다.

IIOP의 8개 메시지를 클래스로 표현하는데, 각각 특성 별로 클라이언트나 서버 메시지를 상속받는다. CDR 클래스로부터 마샬링 연산을 상속받아 데이터 마샬링을 구현한다. 이는 모두 표준 CORBA 정의에 의한 것이다.

이동성 계층을 구현하기 위해서 일반 소켓 프로그램에 약간의 변경을 가하게 된다. 상위 계층에 이동성 정보를 제공하기 위해서 두 가지 종류의 callback 함수를 도입한다. 이동 호스트가 MG에 연결될 때 새로운 연결되는 게이트웨이 주소를 등록하게 된다.

MG가 변경될 때 이 callback 함수를 실행하게 될 것이다. 이 정보를 이용해서 S/IIOP 계층은 현재 이동 호스트의 연결 지점을 알 수 있다.

이동 호스트 상의 S/IIOP 계층은 현재 네트워크 연결 정보를 이용하여 SIOR을 만든다. MG에는 디폴트 포트가 있어 새로운 오브젝트가 생성되는지 항상 listen()하고 있다. 이동 호스트는 이 포트를 알고 있어서 새로운 오브젝트 레퍼런스를 생성할 때 디폴트 포트를 통해서 "Swizzling"을 하게 된다. 이동 호스트에 있는 오브젝트를 호출하기 위해서는 클라이언트는 해당 오브젝트에 대한 SIOR을 가지고 있어야 한다. 클라이언트가 SIOR 값을 가지고 이동 서버의 오브젝트를 호출하면 다음의 3단계를 거쳐서 오브젝트가 호출된다.

1. 먼저 SIOR내의 (주소, 포트 번호)를 가지고 해당 MG에 연결하는데, 이는 생성시점의 MG의 주소이다. 이동 호스트가 아직 연결 중이라면 연결한 MG가 IIOP 메시지를 이동 서버에 전달해준다.
2. 핸드오프가 일어난 후라면 클라이언트가 새로운 MG로 요청을 보낼 수 있도록 IIOP LOCATION_FORWARD 응답을 보낸다.
3. 이동 장비가 아직 재 연결되지 않은 상태라면 S/IIOP 계층에서 요청사항을 이동성 계층으로 보내고, 이동성 계층은 이동 호스트가 다시 연결 될 때까지 대기한다.

2.1.2 MG간 핸드오프

ALICE 구조에서 이동 호스트와 MG간의 연결이 끊어졌을 때, 이동 호스트가 같은 MG와 재 연결한다면 이동 호스트상의 ML은 재 연결 메시지를 보내게 된다. 그렇지만, 새로운 MG와 연결을 하게 된다면, 이동 호스트상의 ML은 핸드오프를 요청하는 메시지를 보낸다. 이전 MG와 새로운 MG간에는 핸드오프가 발생하게 되는데, 그 과정은 다음과 같다.

1. 이동 호스트상의 ML에서 새로운 MG에 대해 핸드오프 요청메시지를 보낸다. 여기에는 이전 MG의 주소와 현재 이동 호스트가 서비스 받고 있는

- LCID(Logical Connection Identifier)가 포함된다.
2. 새로운 MG의 ML은 이동 호스트의 ML에 대해서 확인응답을 보내고, 핸드오프 요청 메시지에 포함된 정보를 이용해서 이전 MG와 TCP 연결을 맺는다.
 3. 새로운 MG는 이전 MG의 ML에 대해서 TCP 연결을 이용하여 핸드오프를 요청한다.
 4. 이전 MG의 ML은 새로운 MG에게 아직 확인응답을 받지 못한 모든 데이터(LCID포함)와 이동 호스트에 대해서 보내지 못한 데이터를 보낸다.
 5. 관련 데이터의 전송과 새로운 MG의 ML 캐쉬 업데이트가 완료되면 이전 MG는 자신의 캐쉬 정보를 삭제하고, 핸드오프 완료 메시지를 보낸다.
 6. 이전 MG는 callback 함수를 이용해서 S/IOP 계층의 정보를 수정하여 새로운 MG의 주소가 정의 되도록 한다. 핸드오프 완료 메시지를 받은 새로운 MG는 이동 호스트의 이동성 계층에 대해서 핸드오프 완료 메시지를 전송한다.

2.2 핸드오프를 고려한 기존의 연구

핸드오프와 관련해서 무선망에서 핸드오프로 인한 패킷 손실을 최소화하기 위한 여러 가지 프로토콜과 방법들이 제안되었다[5, 6, 7, 8, 9, 10]. 핸드오프를 고려한 연구 중 경로 최적화와 관련된 Mobile IP에서 Charles E. Perkins가 제안한 경로 최적화[10]과 "Smooth handoff" 관련 연구에 대해서 살펴보도록 한다.

Charles E. Perkins는 Mobile IP의 프로토콜이 갖는 문제점을 제시하고, 경로를 최적 화할 수 있는 프로토콜을 제안하였다. 기본적인 Mobile IP 프로토콜은 이동 노드가 인터넷상에서 접속 지점을 바꾸면서, 홈 IP 주소를 계속 식별할 수 있도록 한다. 이동 노드에게 데이터그램을 보내는 호스트는 다른 방식과 마찬가지로의 방법으로 HA¹⁾에도 보여

주게 된다. 이 구조를 통해서 이동 노드와 correspondent node(CN)²⁾ 사이에 투명한 상호호연성이 제공될 수 있지만, 이동 노드에 대한 모든 데이터그램이 HA를 통해서 라우트 되는 측면이 있다. 따라서, 이동 노드에 대한 데이터그램은 최적의 경로보다 훨씬 긴 경로를 통해서 전달될 수 있다. 이에 Perkins는 경로 최적화를 위한 프로토콜을 제시한다. 경로 최적화 확장에서는 노드들이 이동 노드의 바인딩(binding)을 저장하

여, 그 바인딩 정보에 포함된 care of address(COA)³⁾ 정보를 가지고 HA를 통하는 긴 경로를 통하지 않고, 직접 데이터그램을 터널링할 수 있도록 하는 수단을 제공한다. 또한, 이동 노드가 움직였을 때나, 이전 주소로 보내졌을 때 생기는 이동 중인 데이터그램이 저장되어 직접 이동 노드의 새로운 COA로 보내질 수 있도록 한다.

경로 최적화에서는 노드가 하나 이상의 이동 노드의 COA를 포함하는 바인딩 캐쉬를 관리할 수 있도록 한다. 이동 노드의 HA가 데이터그램을 중간에 받아서, 이동 노드에게 전달해준다면, HA는 데이터그램을 보낸 노드에 바인딩 캐쉬 정보가 없다는 것을 추론할 수 있다. HA는 이 노드에게 binding update message를 보내서, 현재 이동 노드의 이동성 바인딩 정보를 알려준다. 여기에 대해서는 확인응답은 필요 없는데, 왜냐하면 바인딩이 실패했다면, 다음에 해당 노드가 다시 HA에 전달을 요청할 것이고, HA는 다시 binding update message를 보낼 수 있을 것이기 때문이다. 이와 유사하게, 어떤 노드가 데이터그램을 받았는데, 현재 이동 노드가 자신에게 속하지 않고, 이동 노드에 대한 바인딩 캐쉬 정보를 가지고 있다면, 이 노드는 데이터그램을 보낸 노드가 잘못된 바인딩 캐쉬 정보를 가지고 있음을 추론할 수 있다. 이 경우 이 노드는 이동 노드의 HA에게 binding warning message를 보내서, HA가 데이터그램을 보낸 노드에게 binding update message를 보내도록 해야한다. 이동 노드의 HA의 주소는 바인딩 캐쉬 정보를 보면 알 수 있고, 데이터그램을 보낸 노드의 주소는 데이터그램의 헤더 정보를 이용하면 된다. binding update에서와 마찬가지로 이 경우에도 확인응답은 필요가 없다. binding update가 일어나면 노드는 정보를 update 메시지상의 주소로 보내게 되므로, binding update 메시지에 대해서는 인증과정이 필수적이다. Binding cache정보를 이용하여 모든 노드는 이동 노드에게 최적화된 경로를 통해서 데이터그램을 전송할 수 있게 된다.

3. Mobile CORBA 환경에서의 ROH 알고리즘

이 장에서는 기존의 핸드오프 알고리즘에 따른 문제점을 분석한 후, MG간의 새로운 경로 최적화 핸드오프(ROH) 알고리즘을 제안한다.

3.1 기존 핸드오프 방식

- 3) COA 는 이동 노드의 현재의 네트워크 연결점에 관한 정보를 제공하며, 캡슐화 방법을 위한 중간 어드레스로 사용된다.

1) HA(Home Agent) 모빌 노드를 위해 현재의 이동성 결합(mobility bindings)의 레지스트리를 유지하고, 프록시 할라 이언트처럼 작동하는 가상의 라우터.

2) CN는 현재 이동 노드와 정보를 주고받는 node를 의미한다.

먼저 클라이언트로서의 이동 호스트가 관련된 경우 핸드오프에 따른 문제점은 만약 처리시간이 긴 작업을 요청하였고, 빠르게 이동하거나, 이동 횟수가 많은 이동 호스트는 매우 긴 터널링 과정을 거쳐서 응답 메시지를 받을 수 있다는 것이다. 그 문제점은 그림 1로 나타내었다. 두 번째 서버로써 이동 호스트가 관련된 경우의 문제점은 역시 이동 횟수가 많은 이동 호스트일 경우 클라이언트가 LOCATION_FORWARD Reply를 받는 과정을 반복하여 최종적으로 현재 이동 호스트가 연결되어 있는 MG를 찾아 서비스를 받게 된다는 점이다. 이러한 문제점을 그림 2로 나타내었다. Client는 n번째 request 메시지를 보낸 후에야 비로소 이동 서버로부터 Reply를 받을 수 있게 된다.

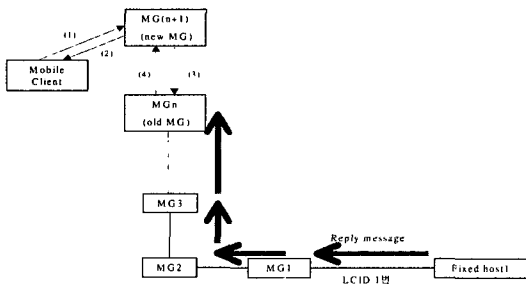


그림 1 Mobile CORBA의 핸드오프 문제점 : 클라이언트

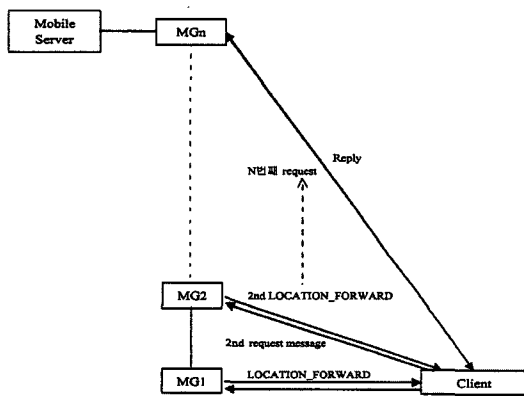


그림 2 Mobile CORBA의 핸드오프 문제점 : 서버

3.2 제안된 ROH 알고리즘

본 논문에서 제안하는 ROH 알고리즘에서 추가되는 과정은 다음과 같다. 새로운 MG가 이동 호스트로부터 받는 정보와 이전 MG로부터 받는 정보를 이용하여, 이동 호스트에 현재까지 서비스를 제공하거나, 이동 호스

트가 서버로 IOR을 등록한 MG에게 현재의 이동성 정보를 Bind Update 메시지를 통해 전달하고, Bind Update 메시지를 받은 MG는 자신의 call back 함수를 이용하여 등록 정보를 수정한다.

Bind Update 패킷에는 이동 호스트가 연결될 새로운 MG의 주소 정보가 포함된다. Bind Update 패킷은 이동 호스트와 새로운 MG의 정보를 가지고 관련 MG에 전송된다. Bind Update 패킷을 보내는 MG는 이동 호스트가 클라이언트인 경우와 서버인 경우에 따라서 다르다. 서버로써 이동 호스트가 관련된 경우에는 SIOR을 발행한 MG가 이동 호스트의 현재 MG 주소인 MGn의 주소 값을 가지고 있다면, 여러 번의 Redirect 과정이 필요 없이 한번의 LOCATION_FORWARD 메시지를 보내서, 곧바로 현재 이동 호스트로 보내 줄 수 있을 것이다. 이 구조는 모든 패킷이 일단 HA로 전송되고, HA는 항상 이동 노드의 현재의 주소(COA)를 알아서 FA로 보내 줄 수 있는 Mobile IP의 구조와 유사하다. 그렇지만 Mobile CORBA 구조에서는 모든 MG(MG)가 동등한 조건이므로, MG(MG) 중 HA의 역할을 하게 될 주소를 Main MG로 정의하고, 추가적으로 MG에서 캐쉬로 관리해야 한다. 그림 3은 추가되어야 할 캐쉬 구조이다.

Server Active Flag	주소
1	주소 of Main MG

그림 3 ROH 알고리즘에서의 서버관련 캐쉬

MG의 S/IOP 계층에서 수행되는 Swizzling 관련 함수에 약간의 수정을 가하여 Swizzling시 이동성 계층의 이동 호스트 관련 정보에 서버 활성화 플래그를 '1'로 세트하고, 현재 자신의 주소를 저장한다. 핸드오프 발생시 기존 MG에서 캐쉬 정보를 받은 새로운 MG는 서버 활성화 플래그가 '1'이고, Main MG 주소가 기존 MG의 주소와 다를 경우, Main MG 주소에 대해 Bind Update 메시지를 보내어 현재 이동 서버의 MG 정보를 Main MG에서 등록할 수 있도록 한다. 즉 기존 핸드오프 과정에서 서버 활성화 플래그를 확인하여, Main MG에 Bind Update 메시지를 보내는 과정이 추가된다. 기존 구조에서는 핸드오프 발생시 항상 Reswizzling을 하도록 하고 있는데, ROH 알고리즘에서는 IOR 발생시점의 SIOR을 그대로 가지고, IOR 발행시점의 MG를 Mobile IP의 HA와 유사한 역할을 하도록 하여, 모든 클라이언트 요청은 일단 Main Mobility Gateway(Main MG)에서 현재의 MG로 연결되도록 한다.

클라이언트로서의 이동 호스트에 대한 핸드오프에서는 서버와 논리적 연결을 맺고 있는 MG의 정보를 현재의 이동성 정보로 수정하도록 한다. 핸드오프시 데이터를 주고받고 있던 LCID를 갖고 있는 MG의 주소를 저장해서, 해당 주소로 Bind Update 메시지를 보낸다.

LCID	MG 주소
------	-------

그림 4 ROH 알고리즘에서의 클라이언트 관련 캐쉬

위의 구조와 같은 정보를 추가로 이동성 계층에서 관리한다. 핸드오프시 이전 MG로부터 그림 4와 같은 형태의 정보를 받은 후, 현재 이동 클라이언트가 실제로 사용하고 있는 LCID중에서 MG 주소가 이전 MG가 아닌 경우에 대해서 Bind Update 메시지를 보낸다. 이동 호스트로부터 현재 이동성 계층에서 저장하고 있는 LCID 정보를 받아 현재 이동 클라이언트가 실제로 사용하고 있는 LCID를 판별한다. 이렇게 하여 한 개 혹은 그 이상의 MG에 대해서 Bind Update 메시지를 보낼 수 있다. ROH 알고리즘은 다음의 표 1과 같다.

표 1 ROH 알고리즘

```

▶ 새로운 MG의 동작
IF ( ServerFlag = '1' and MainMGAddr ≠ OldMGAddr)
  BindUpdate( NewMGAddr, MainMGAddr );
  READ LCID_CACHE_FROM_MH UNTIL THE CACHE ENDS
  {
    IF ( LCIDAddr ≠ OldMGAddr)
      BindUpdate( NewMGAddr, LCIDAddr );
  }
  BindUpdate( NewAddr, ToAddr )
  {
    make BindUpdate packet /*
BindUpdate 패킷을 생성 */
    BindUpdate packet { NewAddr, ToAddr }
    send (BindUpdate packet) /* BindUpdate
패킷을 전송 */
  }

▶ 관련 MG의 동작
IF (BindUpdate packet receive)
  SHOP.callback(NewAddr)
    
```

Bind Update는 "핸드오프 완료" 이후에 한 과정으로 추가하여 핸드오프 지연이 발생하는 것을 방지한다. 부가적인 작업인 Bind Update의 성공적 수행여부와 관계

없이 이동 호스트는 새로운 MG의 서비스를 받아 CORBA 메시지들을 주고받을 수 있게 된다. 그것은, 아직 Bind Update가 수행되기 전의 MG들은 이전 MG로 메시지를 보내게 되고, 이전 MG는 포워딩(Forwarding) 포인터로서 메시지들을 새로운 MG에게 전송할 수 있기 때문이다.

3.3 제안된 알고리즘의 특징

제안한 ROH 알고리즘은 Perkins가 제안한 Route Optimization 기법들 중 Bind Cache의 개념과 Bind Update의 개념을 응용하였다.

ROH 알고리즘의 장/단점을 살펴보면 다음과 같다. 단점은 핸드오프시 Bind Update를 하기 위한 추가적인 캐쉬 공간이 요구되어 오버헤드를 가진다는 것이다. 반면, 장점은 고정 호스트와 연결된 논리적 연결의 생명주기가 길고, 이동 호스트가 자주 이동하는 경우에 발생하는 MG간의 긴 체인을 방지하여 고정 호스트와 이동 호스트간의 최적의 경로로 통신이 가능하다. 또한, 이동 서버의 핸드오프로 인해 발생할 수 있는 호출시 Redirect 회수를 1회로 한정할 수 있어, 이동서버 호출시간 지연을 개선한다. 마지막으로, 핸드오프 발생시 Reswizzling 과정이 불필요하며, 오브젝트 생성시점의 SJOR을 가지고 계속 서비스가 가능하다.

4. 실험 결과

본 논문에서는 제안된 알고리즘의 성능을 평가하기 위해 확장된 ns⁴[11]라는 시뮬레이터를 기반으로 하여 결과를 도출해 낸다. ns는 생성된 패킷으로 사건이 유도되는 시뮬레이터이다. 무선망의 구현은 CMU's Monarch group에서 적용시킨 ns에 대한 이동성 확장을 사용하였으며, ns에서 제공하는 이동 노드와 기지국 노드 객체를 각각 이동 호스트와 MG로 매핑하고 실험을 진행하였다. 구현 환경은 오픈 리눅스 2.3 환경에서 ns를 설치하고, C++과 Tcl Script 언어를 사용하였다. 무선망에서 핸드오프를 구현하고, Mobile CORBA 통신을 구현하기 위해서 기존의 C++로 작성된 Tahoe TCP Agent를 수정한 Agent를 작성하였으며, 실험은 Tcl Script로 작성하였으며, 결과는 trace file로 기록하여 xgraph의 형태로 나타내었다.

4.1 네트워크 모델

네트워크 환경은 제안된 알고리즘의 성능 변화를 이해하는데 쉽게 하기 위해서 간단하고 매우 단순하게 구성하였다. 먼저, 이동 서버 및 이동 클라이언트의 랜

4) ns는: berkeley에서 제공하는 network simulator이다.

드오프와 이에 따른 성능변화를 실험하기 위해서 그림 5와 같은 네트워크 환경을 구성하였다. 이동 호스트, 고정 호스트, 그리고 관련된 4개의 기지국, 이렇게 6개의 노드로 구성하였다. 네트워크 링크는 고정 호스트와 기지국사이에 56Kbps의 유선링크가 있고, 이동 호스트와 기지국 사이에는 ad hoc 네트워크 상에서 패킷이 전송될 수 있는 DSDV routing algorithm을 사용하여 무선상의 패킷 전송이 이루어진다. 고정 호스트에 있는 TCP는 슬로우 스타트, 혼잡 회피, 빠른 재전송 알고리즘이 내장된 Tahoe TCP를 사용한다. 최대 윈도우의 크기는 32 로 설정하고, 패킷의 크기는 128 Byte이다. 무선환경을 위해서 채널, 네트워크 인터페이스, 라디오 전파 모델, MAC 프로토콜, 인터페이스 큐, 링크 계층과 ARP(Address Resolution Protocol)로 구성된 CMU 모델을 이용하였다.

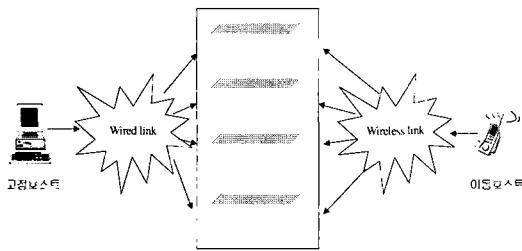


그림 5 네트워크 모델

4.2 구현

위의 그림과 같은 네트워크 모델에서 Mobile CORBA 환경에 대한 실험을 위해 핸드오프와 관련이 있는 Mobile CORBA의 일부를 간단히 구현하였다. 이동 호스트가 서버로써 하나의 오브젝트를 CORBA 객체로 등록하고자 할 때, MG에게 오브젝트 등록을 요청하면 SIOR이 파일로 만들어지도록 Agent를 수정하였다. 클라이언트의 요청 메시지는 IIOP 형식 중에서 request body중 프로파일의 주소와 오브젝트 키 관련된 부분만을 실험을 위해 추가하고, 나머지는 bulk data로 가정하였다. 서버의 reply나 location forward 메시지에서 프로파일 정보를 제외한 나머지 부분은 bulk data로 한다. 이동 호스트(MH)가 핸드오프 요청 메시지를 새로운 MG에 보내면, 새로운 MG(MG1)에서 이전 MG(MG0)와 Tahoe TCP 연결을 설정하여, 이전 MG(MG0)로 핸드오프 요청 메시지를 보낸다. 핸드오프 요청 메시지를 받은 이전 MG(MG0)에서는 add_callback(int mh,int new_mg) 함수를 호출하여 자신의 등록 정보를 수정하

고, 캐쉬 정보를 새로운 MG(MG1)으로 보내도록 하였다. 캐쉬 정보는 ROH 알고리즘에 관련된 캐쉬 정보만 보내고, 나머지 정보들은 Bulk Data로 가정한다. MG0로부터 캐쉬 정보를 받은 MG1은 서버 관련 정보를 읽어, Main MG 주소가 MG0의 주소와 다르면 Main MG로 Bind Update message를 보내도록 한다. 클라이언트와 관련된 경우에 대해서는 LCID 정보를 읽어, 해당 MG를 찾아서 Bind Update message를 보낸다. 실험에서는 MG0에서 클라이언트 연결이 이루어지는 것으로 하였다.

4.3 결과

본 실험에서 사용되는 기초 성능 지표는 수행시간이다. 이동 서버와 관련된 경우에 대한 실험에서는 호출완료시간을 성능평가 지표로 사용하였으며, 이동 클라이언트가 관련된 경우에 대해서는 응답완료시간을 성능지표로 하였다. 서버호출완료시간은 이동 서버의 핸드오프 과정 중 고정 호스트인 클라이언트가 오브젝트를 호출하여 응답(Reply)을 받는데 소요되는 평균시간으로 정의된다. 이것은 클라이언트가 이동 호스트상의 오브젝트를 요청할 때의 시간과 이동 호스트로부터 오브젝트 호출완료 메시지를 수신할 때의 시간의 차이로 측정된다. 응답완료시간은 이동 클라이언트의 핸드오프 과정 중 이동 클라이언트가 고정 호스트로부터 응답(Reply)을 받는 프로세스를 수행하기 위한 평균시간으로 정의된다. 고정 호스트가 이동 클라이언트에게 응답(Reply) 메시지를 보낼 때의 시간과 이동 클라이언트가 응답(Reply) 메시지를 수신할 때의 시간 차이를 측정한다.

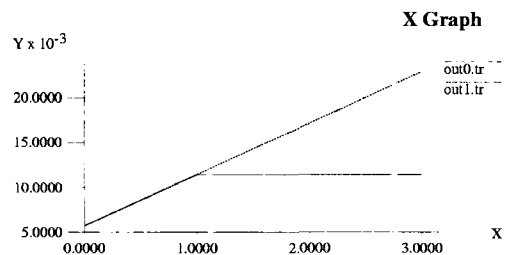


그림 6 서버 호출 완료시간에 따른 비교

그림 6과 7은 실험결과의 그래프이다. 그림 6은 핸드오프 횟수에 따른 클라이언트의 서버호출완료시간의 그래프이고, 그림 7은 핸드오프 횟수에 따른 응답완료시간의 그래프이다. 그래프 상에서 X축이 핸드오프 횟수이고 Y축이 서버 호출완료시간 및 응답완료시간을 나타낸다. 그림 6과 7에서 점선 out0.tr과 out-cl.tr이 기준

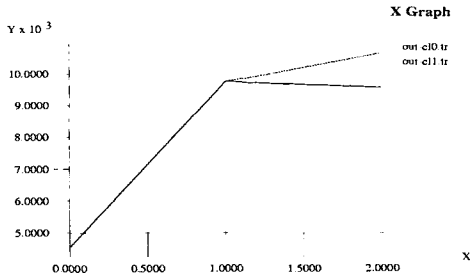


그림 7 응답 완료시간에 따른 비교

방법의 핸드오프가 일어난 과정에서의 서버호출시간 그래프이고, 실선 out1.tr과 out.c11.tr이 ROH 방법을 적용한 경우이다.

기존의 핸드오프 방식과 ROH 알고리즘을 적용한 경우를 비교해 보면, 기존 방식에 따른 결과는 서버호출 완료시간이나 응답완료시간이 핸드오프 횟수에 비례하여 증가하는 그래프를 나타내는 반면, ROH 알고리즘을 적용한 경우에는 비교적 일정한 호출완료시간을 나타내었다. 실험 결과에서 이동 서버가 관련된 경우에 대해서 기존 알고리즘이 더욱 현저히 성능 저하를 보이고 있음을 결과로부터 알 수 있다. 이것은 이동 클라이언트와 관련된 터널링과정이 MG들간의 재전송 과정이므로, MG와 고정호스트가 관련된 location forward 과정보다는 그나마 효율적일 수 있음을 의미한다. 두 실험 결과에서 모두 기존 알고리즘은 핸드오프 횟수에 비례하여 시간지연이 증가하는 결과를 보였다. 이에 반해 ROH 알고리즘을 적용한 경우에는 핸드오프 횟수가 증가해도 거의 일정한 시간지연을 나타내고 있음을 확인할 수 있다.

5. 결 론

본 논문에서는 ALICE 구조에 기반한 mobile CORBA 환경에서 MG간의 핸드오프로 인해 발생하는 긴 터널링의 문제와 비효율적인 여러 차례의 Redirect의 발생을 개선할 수 있는 경로 최적화 핸드오프(ROH)를 제안하였다. 이를 위해 핸드오프를 고려한 연구결과들을 살펴보고, Mobile IP의 Perkins가 제안한 Route Optimization 기법들을 mobile CORBA 환경에서 응용하였다.

ROH 알고리즘은 핸드오프로 인한 비효율적인 터널링을 최대한 개선할 수 있도록 하여, 핸드오프 이후의 Mobile CORBA 서비스의 성능향상을 목표로 하였다. ROH 알고리즘의 주요 내용은 핸드오프시 Bind Update 메시지를 이동 호스트와 관련된 MG로 전송하도록 하는

것이다. Bind Update 메시지를 받은 MG는 자신의 등록 정보를 수정하여, 핸드오프 이후의 CORBA 통신을 할 때 최적의 경로로 통신이 가능하다. 또한 이동 서버의 핸드오프로 인해 발생할 수 있는 호출시 Redirect 회수를 1회로 한정할 수 있어, 이동서버 호출지연을 개선한다. 그렇지만, 핸드오프시 Bind Update를 하기 위한 추가적인 캐쉬 공간이 요구되어 오버헤드를 가지는 단점도 있다. 실험결과 핸드오프의 횟수가 증가함에 따라서, 기존 방식은 서버호출시간이 급격히 지연됨에 비해 ROH 알고리즘은 평균화된 서버호출시간을 나타냄으로서, ROH 알고리즘이 핸드오프에 따른 CORBA 서비스지연을 개선할 수 있음을 확인할 수 있었다.

이러한 방법은 불필요한 메시지 전송이나, 라우팅에 따른 네트워크 성능감소를 미연에 방지할 수 있으며, 핸드오프 횟수가 증가하고, 긴 생명주기를 가지는 CORBA 응용을 실행할 경우에 대해서 더 큰 성능향상을 기대할 수 있다.

참 고 문 헌

- [1] Object Management Group, The Common object Request Broker : Architecture and Specification in V2.2 Object Management Group, February 1998.
- [2] Mads Haahr, Raymond Cunningham and Vinny Cahill, Distributed Systems Group. Supporting CORBA Applications in a MOBILE Environment in Mobicom '99 Seattle Washington USA.
- [3] Raymond Cunningham, Architecture for Location Independent CORBA Environments, ALICE, September 1998.
- [4] PeterKemp et. al. Design of MASE V2 on the Move Project, <http://www.sics.se/>, 1997.
- [5] R.Caceres, L.Iftode, Improving the Performance of Reliable Transport Protocols in Mobile Compute Environment in IEEE Journal on Selected Areas in Communications,13(5),pp. 850-857, June 1994.
- [6] A.Badre, B.R. Badrinath, I-TCP : Indirect TCP for Mobile Hosts Technical Report DCS-TR-314, 15th International Conference on Distributed Computing Systems Rutgers University, October 1994.
- [7] R.Caceres,V.N Padmanabhan, Fast and Scalable Handoffs for Wireless Internetworks in Proceeding of the ACM MobiCom '96,November 1996.
- [8] H.Balakrishnan,S.Seshan,R.H.Katz, Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks in ACM Wireless Networks, Vol.1, Issue4 , pp.469-481,December 1995.

- [9] 김효선, 유 혁, 무선망에서 TCP 패킷 손실 최소화를 위한 핸드오프, 한국통신학회통신정보융합기술대회집, 99년 4월.
- [10] Charles E.Perkins. Mobile IP : Design Principles and Practices in Addison-Wesley, pp. 275, 1998.
- [11] Marc Greis, Tutorial for the UCB/LBNL/VINT Network Simulator "ns" on technical report.
- [12] P.Reynolds, R.Brangleon, Service Machine Development for and Open Long-term Mobile and Fixed Network Environment. Project deliverable, DOLMEN Consortium, December 1996.
- [13] IONA. IIOP Engine Programming Guide, IONA Technologies plc, 1997.
- [14] Object Management Group,Telecom Domain Task Force Request for Information Supporting Wireless Access and Mobility in CORBA, Request For Proposal, OMG Document telecom/98-06-04, June 1998.



신혜령

1995년 2월 서강대학교 화학과 학사.
2000년 2월 서강대학교 정보통신대학원 석사. 현재 ACENTURE 근무. 관심분야는 정보통신 등



이형우

1999년 2월 서강대학교 컴퓨터학과 학사.
2001년 2월 서강대학교 컴퓨터학과 대학원 석사. 현재 서강대학교 컴퓨터학과 대학원 박사과정. 관심분야는 CAD 및 하드웨어 시스템 설계



김주호

1987년 University of Minnesota at Minneapolis 전산학 학사. 1995년 University of Minnesota at Minneapolis 전산학 박사. 1995년 ~ 1996년 Cadence Design Systems Inc. San Jose, California, Senior Member of Technical Staff. 1997년 3월 ~ 현재 서강대학교 컴퓨터학과 교수. 관심분야는 CAD 및 하드웨어 시스템 설계 등