

액티브 네트워크 기술을 적용한 이동 콘텐츠 적응형 네트워크의 구현

이준호, 전해조, 임경식
경북대학교 컴퓨터과학과
e-mail : {jhlee , hjjun, kslim}@ccmc.knu.ac.kr

Implementation of Mobile Contents Adaptation Network using Active Network Technology

Junho Lee, Haejo Jeon, Kyungshik Lim
Dept. of Computer Science, Kyungpook National University

요 약

현재 무선 인터넷 환경은 이동 단말에 콘텐츠 서비스를 제공할 경우, 서비스 사용자가 소유한 단말 환경의 다양성을 고려한 콘텐츠 최적화 작업을 필요로 한다. 기존 환경에서는 이런 기능을 위한 서버를 따로 설치하여 관리함으로써 사용자가 늘어나게 되면 서버에 부하가 집중되어 제공되는 서비스의 질이 저하되는 문제가 발생한다. 본 논문에서는 이런 콘텐츠 최적화 기능을 망에서 제공하여 서버의 부하 집중 문제를 해결하는 이동 콘텐츠 적응형 네트워크(Mobile Contents Adaptation: MobiCAN)를 제안한다. MobiCAN 시스템은 ABone(Active Network Backbone) 데몬과 ANTS(Active Network Transfer System) 실행환경, 콘텐츠 최적화를 위한 액티브 응용으로 구성된다. 본 연구에서는 위와 같이 구성된 MobiCAN 시스템을 실제 무선 인터넷 망과 연동시킴으로써 해서 액티브 네트워크의 무선 인터넷 적용 가능성을 확인하였다.

1. 서론

최근 무선 인터넷에 대한 관심이 증가하면서 이동 단말의 보급률이 높아지고 무선 인터넷 서비스에 대한 요구가 증가하고 있다. 그러나 현재 무선 인터넷 환경은 다양한 단말 환경과 제공되는 서비스 사이의 호환성 제공 방법에 문제점을 갖고 있다. 예를 들면, 다양한 이동 단말기의 성능 및 환경에 따라 동일한 콘텐츠 서비스를 제공하기 위해서는 단말에 맞는 콘텐츠 최적화 작업이 필요하다. 이를 위해서 기존 무선 인터넷 환경은 콘텐츠 변환을 위한 전용 서버를 두어서 서비스를 제공하는데 이 경우 사용자가 증가하면 서버에 부하가 집중되는 문제가 발생한다. 이런 부하 집중 문제는 최적화 작업을 여러 노드로 분산시킴으로써 해서 해결이 가능하다.

DARPA 에서 개발한 액티브 네트워크는 기존의 수

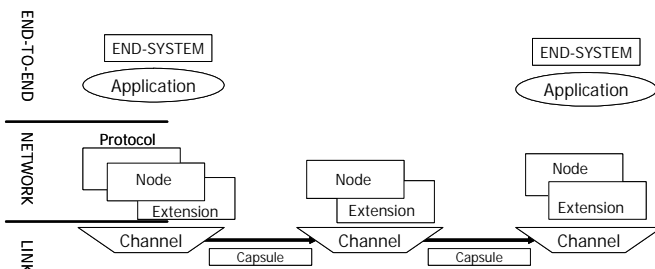
동적인 망 하부 구조를 능동적으로 바꾸고, 망의 기능 수행을 액티브 패킷(Active Packet)을 기반으로 라우터에서 가능하게 한다. 따라서 이 기술을 이용하면 기존의 무선 인터넷 서비스 환경에서 발생하는 전용 서버의 부하 집중 문제를 효과적으로 해결할 수 있다. 액티브 네트워크에 대한 연구는 주로 ATM 과 IP 망을 중심으로 진행되었고, 최근 이동 통신망의 물리적 특성의 제약을 최소화하기 위한 분야에 대한 연구도 시작되었으나, 무선 인터넷 환경에 대한 연구는 본격적으로 이루어지고 있지 않은 상황이다.

본 논문에서는 무선 인터넷 환경에서 콘텐츠 서비스를 제공할 때 발생하는 전용 서버의 부하 집중 문제를 해결하기 위해 액티브 네트워크 기술을 적용한 이동 콘텐츠 적응형 네트워크(Mobile Contents Adaptation: MobiCAN)를 설계 및 구현하였다.

2. 관련 연구

MIT 에서 개발한 ANTS(Active Network Transfer System)는 액티브 네트워크의 구성 노드에게 실행 환경과 네트워크 프로그래밍 모델을 제공한다. 또한 ANTS 는 액티브 패킷인 캡슐을 수신하는 채널과 그 캡슐을 처리하기 위해 필요한 코드를 받는 채널이 분리되어 있다. 다시 말해서 액티브 노드에서 캡슐을 받았을 때 이것을 처리할 수 있는 코드가 없는 경우 코드 분산 메커니즘에 따라 다른 노드로부터 코드를 전송 받아서 사용한다. 코드 분산 메커니즘에 따라 캡슐에는 타입과 프로토콜을 명시하며, 캡슐이 도착하면 노드는 프로토콜의 캐시를 검사한다. 만약 필요한 코드가 없을 경우에는 필요로 하는 코드에 대한 요청이 이전의 이웃 노드에 전달되고, 프로그램의 수행 상태는 휴면 상태가 된다. 그리고 필요한 코드 세그먼트가 다운로드 되어 수행이 가능해지면 휴면된 프로그램을 다시 수행시킨다. 만약 프로그램에 필요한 코드가 다 수집되지 않으면 그 캡슐은 버려진다.

아래 (그림 1)은 ANTS 실행환경의 구조를 설명하기 위한 클래스 구조도 이다.



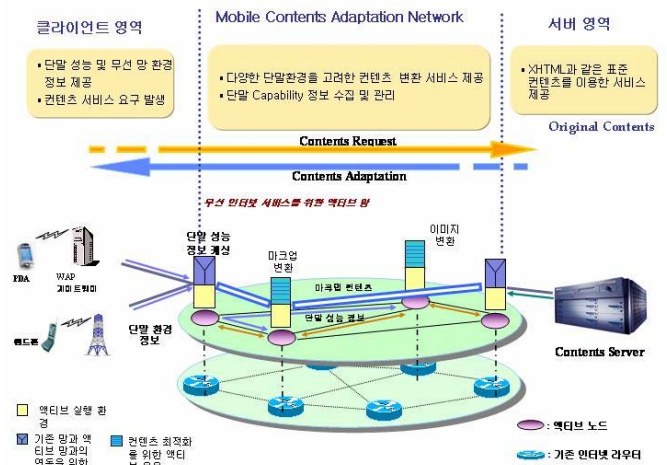
(그림 1) ANTS 실행환경 클래스 구조도

Node 는 각 노드의 실행을 관장하는 클래스이다. 이 클래스는 노드에 특정한 정보를 일시적으로 저장하기 위한 소프트 저장(soft store) 방식과 코드 분산 메커니즘에 따라 코드를 분배한다. 채널 클래스는 각각의 노드가 단대단 또는 매체 공유 기술로 연결된 링크 계층 인터페이스를 제공하여 사용된다. ANTS 에서 새로운 서비스는 Capsule 클래스와 Protocol 클래스, Application 클래스를 각각 상속하여 구현함으로써 제공된다. Capsule 클래스는 네트워크에서 패킷 처리를 제어하기 위하여 사용되며, 캡슐 포워딩과 관련된 코드는 코드 분배 프로토콜에 따라 네트워크 상에서 전달된다. Protocol 클래스는 캡슐의 구조를 정의하는데 사용되며 Extension 클래스는 각 노드에 특화된 서비스를 제공하기 위하여 사용된다. Application 클래스는 캡슐을 송수신하는 역할을 하며, 로컬에서만 수행되고 코드 분배 시스템에 의하여 이동되지 않는다.

3. MobiCAN 시스템

MobiCAN 시스템은 콘텐츠가 서버로부터 이동 단말까지 전달되는 과정에서 이동 단말의 환경에 적합한 형태로 동적으로 변환되는 서비스를 제공하며, 이런 기능을 위한 여러 노드들로 구성된다. 이 시스템은 콘텐츠 서비스를 위한 주요 기능들을 각 노드로 분산하

여 특정 노드로 부하가 집중되는 현상을 막을 수 있다. 시스템의 각 노드는 액티브 노드로 구성되며 노드 별로 서비스에 필요한 액티브 응용을 구동한다.



(그림 2) MobiCAN 시스템의 개략도

(그림 2)는 MobiCAN 시스템을 간략히 나타낸 것이다. 왼쪽이 무선 이동 단말들이 있는 영역이고 가운데 영역은 단말에 최적화된 콘텐츠를 제공하기 위한 변환 작업을 수행하는 액티브 노드들이 있는 MobiCAN 영역이며 오른쪽 영역에는 콘텐츠 서버가 위치한다. 이동 단말은 (그림 2)과 같이 액티브 노드 구간을 거쳐서 콘텐츠를 요청하고 그것의 응답 메시지인 콘텐츠가 MobiCAN 영역을 통과한 후 단말에 전달된다. 사용자의 요청에 의한 콘텐츠가 MobiCAN 영역의 콘텐츠 변환 노드에 도착하게 되면, 그 노드는 단말 환경 정보 저장소에 해당 단말의 환경 정보를 요청하게 된다. 요청된 환경 정보가 도착하면 콘텐츠 변환 노드는 그 정보를 바탕으로 콘텐츠 최적화 작업을 수행한다. 이때 단말 환경 정보 저장소의 기능을 클라이언트 영역과 MobiCAN 영역의 접점에 해당하는 액티브 노드에서 수행한다.

MobiCAN의 액티브 노드는 Anetd 데몬과 ANTS 실행 환경, 액티브 응용이 연동하여 동작한다. Anetd 데몬은 ABone(Active Network Backbone) 망을 구축하기 위한 데몬이며 ABone 은 액티브 네트워크 연구 프로그램을 지원하기 위하여 DARPA 에서 개발한 테스트 베드로써, 본 논문에서 MobiCAN 시스템을 구현하고 테스트하기 위해서 사용했다. ANTS 실행환경은 동적 프로그램 배포 및 실행 기능을 상위의 액티브 응용에게 제공한다. MobiCAN 은, ABone 상위에서 ANTS 실행환경을 구축하고 액티브 망을 구성한 다음 콘텐츠 최적화 기능을 구현한 액티브 응용을 배포하여 무선 인터넷 서비스를 제공한다.

3.1 ABone 망 구축

ABone 망은 ABone 노드, Anetd 클라이언트, ABOCC(ABone Coordination Center)로 구성된다. ABone 노드는 Anetd 데몬이 동작하여 실행환경이나 관련 프로그램을 설치 배포할 수 있는 노드이다. Anetd 클라

이언트는 그 ABone 노드에 프로그램 배포 및 실행 명령을 내리는 기능을 하는 노드이다. ABOCC는 코어 노드 운영 센터로서 액티브 노드의 상태를 모니터링 하며, 문제를 진단하고 관리하는 역할을 한다. ABone 노드를 구축하여 기존 ABOCC 서버에 등록하게 되면 기존 ABone의 일부로서 동작하며, 별도의 ABOCC를 만들어 이곳에 등록하게 되면 독립적인 ABone 망으로 동작한다. 본 논문에서는 후자를 선택하였다.

독립적인 ABone 망은 아래와 같은 방법으로 구축하였다.

● ABOCC 구성

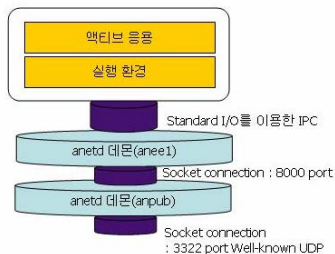
아파치 웹 서버를 수정하여 설정 파일이나 실행환경 등의 코드를 다운 받게 하여 ABOCC를 구성한다.

● ABone 노드 구성 및 ABOCC 등록

노드의 'abocc' 계정을 생성하여 Anetd 데몬을 설치한다. 그 후 Anetd 마스터 컨피규레이션(master configuration) 파일에 ABOCC 서버 항목을 새로 구성된 ABOCC로 설정한다.

● ACL, TCL 파일 설정

ACL(Access Control List)은 ABone 노드에 명령을 내릴 수 있는 권한을 가진 노드들에 대한 목록으로 이 목록에 없는 노드가 명령을 내릴 경우 실행을 막는 방법으로 접근을 제어한다. 목록에는 권한을 가진 노드들의 공개키로 채워진다. TCL(Trust Code Server List) 파일은 ABone 노드가 안전하게 프로그램을 다운로드 받을 수 있는 신뢰할 수 있는 코드서버들에 대한 목록파일로서 이 목록에 없는 서버로부터 코드를 다운 받는 것을 막아 악의적인 코드가 설치되는 위험을 제거한다.



(그림 3) MobiCAN 노드 구조

(그림 3)는 ABone 망으로 구축된 MobiCAN 시스템의 노드 구조를 나타낸 것이다. ABone 노드에는 위 그림과 같이 두 종류의 Anetd 데몬이 수행되는데 이들을 실행시키는 계정에 따라 구분한다. Anpub 계정으로 실행되는 데몬은 노드에 설치된 실행환경의 수가 복수일 경우 노드로 들어오는 액티브 패킷을 실행 환경에 따라 분배해주는 기능을 한다. aneeX 계정으로 실행되는 데몬은 실행 환경을 로드하고 실행시키는 역할을 수행한다.

3.2 ANTS의 설치 및 ABone과의 연동

ANTS 실행 환경은 Java로 된 공개 소프트웨어로서 소스코드를 다운 받아 컴파일 작업을 통해서 사용하였다. ANTS를 컴파일 하면 ANTS 실행환경과 액티브 응용 프로그램의 Jar(Java Archive) 파일이 생성된다. 액

티브 응용은 ANTS 실행환경 클래스 중 Application class와 DataCapsule class, Protocol class를 상속하여 구현한다. 본 논문에서 사용한 ANTS 실행환경 2.0.0 버전은 JavaNodeOS를 기반으로 동작하도록 설계되었기 때문에 노드 시스템의 클래스 경로에 JavaNodeOS 라이브러리를 추가하였다.

ANTS 실행 환경에 아래와 같이 세 개의 스크립트 파일을 설정해주면 설치가 완료된다.

- .start : Configuration Manger Class를 실행시키는 스크립트로서 .config, .route 등의 ANTS 환경 설정 파일을 설정해준다.
- .config : 네트워크 토폴로지(network topology)를 설정하고 실행 시에 필요한 응용 클래스를 저장한다.
- .route : 네트워크의 라우팅 테이블 정보를 나타내며 'makeroutes' 프로그램을 이용해서 생성한다.

ABone 망과의 연동을 위해서는 config 파일을 일반 설정과 다르게 해준다. 채널 정보 설정 시에 출력은 표준 출력으로 하며 프록시 설정은 anpub Anetd 데몬으로 지정한다. 포트 설정은 전송 지점을 나타내며 anpub Anetd 데몬의 입력 창구인 3322번 포트 번호를 사용한다.

3.3 액티브 응용 개발

이동 단말의 환경에 따른 콘텐츠 변환 서비스를 제공하기 위해서 아래와 같은 기술을 개발하여 MobiCAN 응용에 사용하였다.

● Internetworking 기술

기존 유무선 망과 MobiCAN 망과의 연동을 위한 기술이다. 기존 인터넷 망의 HTTP 메시지와 MobiCAN의 액티브 패킷간의 상호 변환 작업을 수행한다.

● 콘텐츠 변환 기술

컨텐츠를 요구되는 환경에 맞게 변환 시키는 기술이다. GIF, JPEG, WBMP 형태의 이미지 포맷간의 상호 변환이 가능하며, 이미지 크기 및 컬러 변환도 지원한다. 마크업 변환 기술로는 XHTML을 WML이나 MHTML, CHTML 등으로 변환하는 기능을 제공한다.

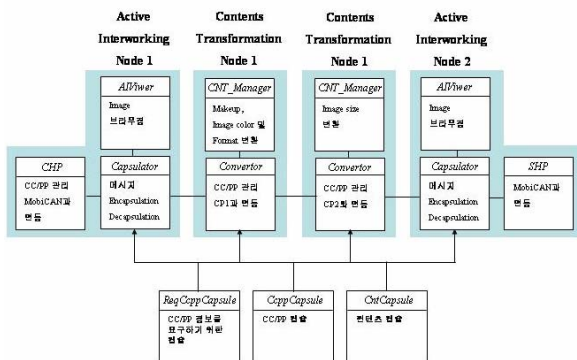
● 단말 정보 관리 기술

최적화된 콘텐츠 변환 서비스를 위해 이동 단말의 환경 정보를 관리하는 기술로써 CC/PP(Composite Capability/Preference Profile)의 리소스 기술 기법을 사용하여 관리한다.

MobiCAN의 액티브 노드는 액티브 응용의 기능에 따라 크게 액티브 인터넷워킹 노드(Active Internet working Node: AIN)와 콘텐츠 변환 노드(Contents Translation Node: CTN)로 구분된다. AIN은 Internet working 기술을 이용하여 기존의 망과 MobiCAN의 액티브 망을 연결시켜 서로 간의 연동을 가능하게 한다. 예를 들면 이동 단말로부터 콘텐츠 요청 메시지가 HTTP 형태로 전달되면 AIN에서 이것을 액티브 망에서 사용할 수 있는 캡슐로 만든 후 망 내부로 전달한다. 또한 AIN은 여러 이동 단말들의 환경 정보를 갖고, 컨

텐츠 변환 노드가 변환 작업에 필요한 단말 정보를 요청할 경우에 그 정보를 단말 정보 관리 기술을 이용하여 CC/PP 캡슐로 만들어 제공하기도 한다.

CTN 은 AIN 으로부터 받은 CC/PP 정보를 토대로 컨텐츠를 단말에 최적화한다. 즉 CTN 에 컨텐츠가 도착하게 되면 최적화를 위한 단말 환경 정보를 AIN 에게 요청하여 CC/PP 캡슐을 수신하고 캡슐로부터 단말 환경 정보를 추출한 다음 컨텐츠 최적화 작업을 수행한다. 아래 (그림 4)는 AIN 과 CTN 으로 구성된 MobiCAN 시스템의 노드들을 기능별 모듈로 나누어 표현한 것이다.



(그림 4) MobiCAN 기능별 모듈

3.4 현재 무선 망과의 연동

본 연구의 MobiCAN 시스템은 클라이언트 영역으로 WAP 브라우저를 내장한 COMPAQ PDA 와 KTF-X2000 핸드폰을 사용하였고, MobiCAN 영역으로는 2개의 AIN 노드와 CTN 노드를 구축하여 사용했다. 서버 영역에서는 아파치 웹 서버를 두어 XHTML 마크업 문서와 GIF, BMP, JPEG 와 같은 이미지 문서를 컨텐츠로 제공하였다. PDA 는 로컬 무선 망에서 WAP 게이트웨이를 통해 AIN 노드로 컨텐츠를 요청하도록 하였으며, 핸드폰은 KTF 망을 통하여 AIN 노드로 접속한다. KTF 단말기는 프락시 설정을 MobiCAN 의 AIN 노드로 설정함으로써 해서 쉽게 MobiCAN 과 연동할 수 있다.

3.5 기존 실행 환경의 제약

ANTS 실행환경은 보안상의 이유로 Java 라이브러리 중 상당부분의 사용을 제약하기 때문에 응용 개발에 한계가 있다. 특히 GUI (Graphic User Interface) 라이브러리를 비롯하여 socket 라이브러리는 그 사용이 원천적으로 봉쇄되어서 응용 개발에 많은 어려움을 겪었다. 결국 라이브러리 사용을 위하여 실행환경 밖에서 동작하는 프로세스를 생성하여 액티브 응용과 파일 I/O 를 사용한 IPC(interprocess communication)로 연동시켜 컨텐츠 최적화 응용을 구현하였다. 또한 실행환경에 설치된 응용을 다른 응용에서 연동하여 사용할 수 없기 때문에 코드 재사용 측면에서 비효율적이다. 그리고 코드의 동적 다운로드 기능이 Capsule class 와 Protocol class 로 제한되어 제공되기 때문에 실제

중요 기능을 위한 코드는 동적 배포 및 실행이 불가능하다.

4. 결론 및 향후 과제

본 논문에서는 액티브 네트워크를 사용한 이동 컨텐츠 최적화 서비스를 제공하기 위해서 액티브 망을 구성하고 응용 프로그램을 개발하였다. 액티브 망은 ABone 과 ANTS 실행환경을 사용하여 구축하였고 컨텐츠 서비스를 위한 액티브 응용 기술로써 인터넷 워킹(Internetworking) 기술, 컨텐츠 변환 기술, 단말 환경 정보 관리 기술을 개발하여 적용하였다. 또한 KTF 망을 통한 실제 무선 인터넷 망과의 연동으로 액티브 네트워크의 무선 인터넷 적용 가능성을 확인하였다. 그러나 기존 ANTS 실행 환경의 제약으로 인해 응용 개발 및 관리에 어려움을 겪었다. 향후 기존 실행환경의 단점을 보완하여 라이브러리 사용과 동적 프로그램 다운로드 기능의 제약 문제를 해결하고 코드 재사용성을 높인 실행환경을 개발하여 MobiCAN 에 적용하면 시스템 성능 개선에 큰 효과가 있을 것으로 기대된다.

참고문헌

- [1]이영희, Active Network 구조: 문제점 및 접근 방법, Sigcomm Review Vol. 1, No.1 December 2000, p109 - 126
- [2]이남희, Active 네트워크 기술 개발 현황, Sigcomm Review Vol.1, No.1 DEcember 2000, p127 - 142
- [3]Konstantinos psounis, Active Networks: Applications, Security, Safety, and Architecture, IEEE Communications Surveys, 1999, <http://www.comsoc.org/pubs/surveys>
- [4]Stephen F. Bush, Amit B. Kulkarni, Active Netowrks and Active Network Management, KAPP, ISBN 0-306-46560-4, 2001
- [5]“Wireless Markup Language,” WAP Forum, November 8, 1999. <http://www.wapforum.org/>
- [6]Vu Ahn Pham, Ahmed Karmouch, Mobile Software Agents : Overview, IEEE Communication Magazine, p26-36, July 1998
- [7]Oguz Anging, Andrew T. Campbell, Michael E. Kounavis, Raymond R.F., Mobiware Toolkit: Programmable Support for Adaptive Mobile Network, IEEE Personal Communications, p32-43, August 1998
- [8]Eric A. Brewer, ProActive Infrastructure, March 1998, <http://citeseer.nj.nec.com/brewer98proposal.html>
- [9]David Wetherall, Developing Network Protocols with the ANTS Toolkit, 1997, <http://www.sds.lcs.mit.edu/>
- [10]David J. Wetherall, Service Introduction in an Active Network, February, 1999, <http://www.lcs.mit.edu/publications/pubs/ps/MIT-LCS-TR-773.ps>
- [11]Graham Klyne, Franklin Reynolds, Chris Woodrow, Hidetaka Ohto, Composite Capability/Preference Profile(CC/PP): Structure and Vocabularies, March 15, 2001, <http://www.w3.org/TR/CCPP-struct-vocab/>