

# 효율적인 응용 계층 멀티캐스트 서비스 개발을 위한 프레임워크

김성훈<sup>0</sup> 강경란 XuanTung Hoang 김문석 이동만  
한국정보통신대학원대학교 공학부  
{kimsh<sup>0</sup>, korykang, tung\_hx, kms1006, dlee}@icu.ac.kr

## A Framework for Efficient Service Development using Application Layer Multicast Protocols

Sunghoon Kim<sup>0</sup>, Kyungran Kang, XuanTung Hoang, Moonseok Kim, and Dongman Lee  
School of Engineering, Information and Communications University

### 요 약

응용 계층 멀티캐스트는 IP 멀티캐스트의 배치 문제를 효율적으로 해결하지만 각 응용 서비스의 다양한 요구 사항을 융통적으로 해결하지 못한다는 단점을 가진다. 본 연구에서는 응용 계층 멀티캐스트 서비스의 공통 기능과 응용 서비스의 다양한 요구 사항을 충족시킬 수 있는 단일화된 인터페이스를 제공함으로써 융통성이 있고 재사용이 가능한 응용 계층 멀티캐스트 프레임워크를 설계하고 구현한다. 본 프레임워크는 응용 서비스의 요구에 따라 다양한 응용 계층 멀티캐스트 기법을 선택적으로 재배치할 수 있으므로 쉽고 빠른 응용 서비스의 개발 및 구축을 가능하게 하고 기법 개발자에는 개별 기법을 응용 프로그램과 독립적으로 확장하거나 추가할 수 있도록 한다. 본 논문에서는 응용 계층 멀티캐스트 서비스의 요구 사항 분석을 통해 본 프레임워크의 설계 및 구현에 대해 기술하며 그 효율성과 실제 적용성을 검증하기 위해 TBCP 프로토타입을 구현한 결과를 기술한다.

### 1. 서론

네트워크의 보급 확대와 성능 향상으로 네트워크 게임, 오디오/비디오 컨퍼런싱, 멀티미디어 스트리밍, 데이터/컨텐츠 배포 등과 같이 네트워크 상에서 여러 사용자들 간에 데이터 교환이 이루어지는 다중 사용자 응용 프로그램들이 주목을 받고 있다. 다자간 데이터 교환을 지원하기 위한 대표적인 기술로 IP 멀티캐스트[1]가 고려되어 왔으나, 실제 인터넷 상에서 보편적인 서비스로서 사용되지 못하고 있다. 이러한 현실 상황을 고려하여, IP 멀티캐스트에 대한 대안으로 응용 계층 멀티캐스트 기법들이 제안되고 있는데, 이 기법들에서는 세션 내 참가자들이 응용 계층에서 IP 멀티캐스트 라우팅 트리와 유사하게 데이터 전달을 위한 트리를 구성하고 트리를 따라 유니캐스트를 사용하여 데이터를 전달한다.

지금까지 제안된 대표적인 응용 계층 멀티캐스트 기법으로는, Narada[2][3], ALMI[4], NICE[5], TBCP[6], RMX[7], ScatterCast[8] 등이 있는데, 각각의 기법은 특정 응용 서비스를 목적으로 하여 개발되었다. 예를 들어, Narada나 ALMI 등은 오디오/비디오 컨퍼런싱과 같은 응용 서비스에 적합하고, RMX는 신뢰적 전송이 필요한 응용 서비스에서 사용하기에 적합하다. 그러므로, 응용 프로그램 개발자는 개발하고자 하는 서비스의 요구 사항에 따라 해당 기법을 선택해서 구현해야 한다. 그러나, 응용 계층 멀티캐스트 기법에 접근할 수 있는 표준화된 인터페이스가 정의된다면, 정의된 인터페이스를 지원하는 범위 안에서 기 개발된 기법들에 대해서는 코드 재사용이 가능하므로 응용 프로그램 개발자가 매 개발마다 반복하여 개발해야 하는 부담을 줄일 수 있다. 그리고, 기법 개발자 입장에서 정의된 인터페이스를 제공하는 한, 개별 기법들을 응용 프로그램과 독립적으로 확장하거나 추가할 수 있고, 응용 계층 멀티캐스트 내 개별 기능 요소들을 개발하고자 할 시에도 자신이 관심을 갖고 있는 기능 외의 기능들에 대해서는 기존 개발된 코드를 재활용할 수 있게 된다.

본 논문에서는 응용 계층 멀티캐스트 기법들을 활용하여 개발할 수 있는 응용 서비스들의 요구 사항들을 공통적인 기능과 선택적인 기능으로 분류하여 분석한다. 이러한 분석을 기반으로 하여, Internet Engineering Task Force 내의 Reliable Multicast Transport 워킹 그룹에서 신뢰적 멀티캐스트 기법들을 위한 기능 요소들을 정의하고 이들 간의 관계를 정의하는 것[9]과 유사하게, 다중 사용자 응용 서비스

를 위해 응용 계층 멀티캐스트 기법 내의 기능 요소들을 정의한다. 이 기능 요소들을 포괄하는 프레임워크를 객체 지향적 기법으로 설계하고 구현함으로써 점진적으로 다양한 응용 계층 멀티캐스트 기법들을 추가하거나 확장할 수 있도록 하고, 개별 응용 프로그램 개발에 활용할 수 있도록 한다.

본 논문은 다음과 같이 구성된다. 2장에서 기존에 제안된 응용 계층 멀티캐스트 기법들을 응용 분야와 특징 면에서 소개하고 3장에서는 응용 계층 멀티캐스트 기법을 활용하는 응용 서비스들이 갖는 요구 사항들을 기술한다. 그리고, 4장에서는 프레임워크의 구조를 제안하고 이 프레임워크에 기존의 프로토콜을 적용한 구현 결과를 기술한다. 5장에서는 본 연구의 결론 및 향후 연구를 기술한다.

### 2. 응용 계층 멀티캐스트 기법

Narada, ALMI는 참가자의 수가 적고 넓은 지역에 산재해 있는 멀티캐스트 그룹을 가정하는 응용 서비스에 적합하다. 개별 참가자는 다른 모든 참가자들의 존재를 인지하고 상대적 거리를 이용하여 데이터 경로의 기반이 되는 메쉬(mesh)를 구성하고, 이를 이용하여 데이터 전달을 위한 스패닝 트리(spanning tree)를 형성한다. 따라서, 데이터의 생성자부터 수신자까지 최적의 지연 시간을 갖는 전송 경로를 만들어낼 수 있다. Scattercast와 RMX는 네트워크 상에 에이전트를 위치시키고 에이전트들 간에 메쉬를 형성한 후 데이터 경로가 되는 스패닝 트리를 구성한다. 이때 에이전트에서 신뢰적 전송을 위한 여러 복구 등의 기능을 수행한다.

TBCP는 멤버들이 참가자들 간에 부가적인 메쉬 대신 멀티캐스트 그룹에 가입하면서 바로 데이터 전송 트리를 생성한다. 새로운 참가자는 소스로부터 자신까지의 거리를 고려하고, 전체 참가자들 중 일부와의 메시지 교환을 통해서 자신의 위치를 결정한다. 이런 기법들은 실시간 방송 서비스나 데이터/컨텐츠 배포 등의 일대다 응용 서비스에 적합하다. 유사한 기법으로 HMTF[10]가 있다.

NICE는 지역적으로 가까이 위치한 멤버들을 묶어서 클러스터를 형성하고 각 클러스터에서 리더를 선정하여 계층 구조를 구성하고, 새로운 멤버는 재귀적인 절차로 자신의 클러스터와 레벨을 결정한다. 이렇게 생성된 위상을 이용하여 송신자 기반 트리 형식의 데이터 전달 경로를 생성한다. 이 기법은 채팅이나 네트워크 멀티 플레이어 게임과 같은 대화형 서비스에 적합하다.

Overcast[11]는 실시간 스트리밍 서비스를 위한 기법으로, 수 천의 참가자까지 지원할 수 있다. 세션 내의 참가자들의 생명 주기가 상대적으로 긴 응용 서비스에 적합하고, 네트워크 트래픽 상태 변화에 효율적으로 대처한다. 스트리밍 서비스를 위한 기법으로 PeerCast[12]도 제안되었다.

3. 요구 사항 분석

현재까지 제안된 응용 계층 멀티캐스트 기법들이 대상으로 하는 응용 서비스를 중심으로, 다중 사용자 응용 서비스에서 응용 계층 멀티캐스트 기법으로부터 요구하는 기능들을 공통적인 것과 선택적인 것으로 분류한다.

3.1 공통 요구 사항

첫째, 참가자가 동적으로 가입하고 탈퇴할 수 있어야 한다. 그리고 응용 계층 멀티캐스트 기법에서는 개별 참가자들이 데이터 전달 기능을 수행해야 하므로 자신의 이웃 참가자들에 대한 멤버십 정보를 동적으로 유지하고 관리해야 한다.

둘째, 응용 계층 멀티캐스트에서는 데이터 전달을 중단 시스템이 담당하는데, 중단 시스템은 IP 멀티캐스트에서 데이터 전달을 담당하는 라우터에 비해 오류가 발생할 확률이 크다. 즉, 세션 참가자들의 분할을 초래할 수 있으므로 응용 계층 멀티캐스트는 이러한 분할을 탐지하고 복구할 수 있는 기능을 지원해야 된다.

셋째, 응용 계층 멀티캐스트에서 데이터 전달 시에, TCP나 UDP 등과 같이 전송 계층에서 제공하는 프로토콜을 활용한다. 그러므로, 이용되는 응용 계층 멀티캐스트에서의 채널과 전송 계층에서의 채널을 대응시키는 기능이 필요하다. 또한, 이웃 참가자에게 데이터를 전달할 때 자신의 수신 속도와 이웃 참가자의 수신 속도 사이에 차이가 발생하는 경우 일시적으로 데이터를 보관하는 기능도 함께 제공되어야 한다.

3.2 응용 서비스에 종속적인 요구 사항

스트리밍 서비스, 오디오/비디오 컨퍼런싱과 같은 응용에서는 비교적 큰 규모의 데이터를 실시간에 전달하므로 과도한 트래픽을 발생시켜 혼잡을 초래할 수 있으므로 응용 계층 멀티캐스트에서 혼잡 제어 기법이 지원되어야 한다. 그리고, 콘텐츠 배포 서비스에서는 데이터가 손실없이 전달되는 것이 가장 중요하므로, 응용 계층 멀티캐스트에서 신뢰성 있는 전달 서비스가 지원되어야 한다. 이 외에도 네트워크 게임과 같은 응용 서비스도 멀티캐스트 기법을 사용하여 개발이 가능하다. 이러한 응용 서비스들에서는 데이터의 보안을 위해 암호화 기법을 적용할 수 있으며, 허락되지 않은 사용자의 데이터 접근을 방지하기 위해 응용 서비스 차원에서 뿐만 아니라 멀티캐스트 서비스 차원에서도 데이터 전달 혹은 세션 가입 과정에서 사용자 인증을 요구할 수 있다.

4. 응용 계층 멀티캐스트 프레임워크

4.1 전체 구조

본 논문에서 제안하는 응용 계층 멀티캐스트 프레임워크의 목적은 응용 서비스 개발자가 서비스 개발에 있어서 응용 계층 멀티캐스트 내 기능 요소들을 통합하여 서비스를 완성할 수 있도록 하는 것이다. 즉, 앞서 기술한 기능 요소들을 빌딩 블록의 형태로 제공하고, 빌딩 블록들 간의 인터페이스를 정의한다. 또한, 응용 계층 멀티캐스트 기법의 개발자 또한 정의된 인터페이스를 제공하기만 한다면, 자신들이 개발한 기법들을 빌딩 블록의 형태로 프레임워크 안에 포함시킬 수 있다.

제안하는 프레임워크는 그림 1이 보여 주듯이 TCP나 UDP와 같은 기존의 전송 계층을 기반으로 하며, '응용 계층'과 '오버레이 계층'으로 나뉘어진다. 응용 계층에는, 신뢰적 전송, 혼잡 제어, 사용자 인증, 암호화 등과 같이 응용 서비스의 필요에 따라 선택할 수 있는 기능 요소들이 위치한다. 각 기능 요소들 사이에는 상하 관계가 명시적으로 정의되지 않고, 응용 서비스 개발자가 필요한 기능 요소들을 선택하여, 상하 관계를 설정하여 상호 연결한다. 오버레이 계층은 '채널 관리'와 '위상 관리' 계층으로 이루어진다. 채널 관리에서는 하위의 전송 계층에 직접 접근하여, 데이터를 송수신하고 수신된 데

이터에 대해 상위 계층에 이를 전달한다. 데이터 전달, 멤버십 관리, 분할 관리 등의 기능은 위상 관리 계층에서 수행한다.

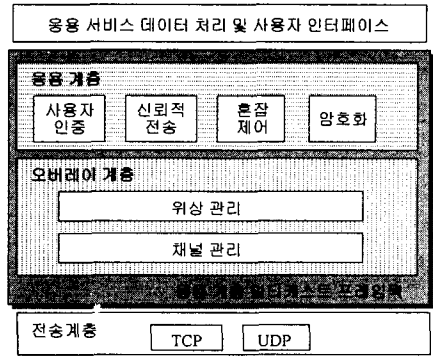


그림 1 프레임워크의 계층적 구조

4.2 기본 클래스 설계

그림 2는 프레임워크 내 클래스 계층을 보이고 있다. 실선 박스로 표현된 클래스는 개별 기능 요소들을 위한 추상 클래스를 나타내고, 점선으로 표시되는 클래스는 이러한 추상 클래스에서 정의하는 인터페이스를 제공하며 개별 기법들을 구현한 경우를 의미한다.

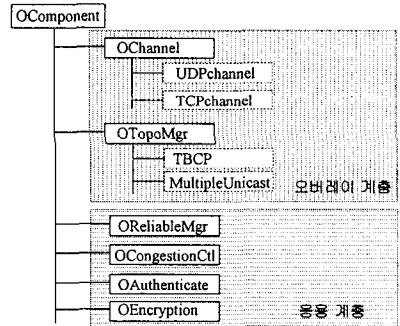


그림 2 프레임워크 내 클래스 계층

OComponent 클래스는 프레임워크의 기초가 되는 클래스로서, 모든 기능 요소가 제공하는 최소한의 공통된 인터페이스를 정의한다. OComponent 클래스는 멤버 변수로 lower 와 upper를 갖는데, 각각은 자신의 하위 기능 요소와 상위 기능 요소들에 대한 참조이다. 하위 기능 요소 참조를 정의하는 메소드로 setLowerComponent()를 정의하고, setUpperComponent()를 정의하여 상위 기능 요소 참조를 정의하는 메소드로 사용한다. 그리고, 상하 기능 요소들과의 통신을 위하여 다음의 두 가지 메소드를 정의한다.

- deliver(fromAddress, data) : 수신한 데이터를 상위 기능 요소에게 전달하는 기능을 수행한다. 메소드의 입력 변수로 송신자 정보(fromAddress)와 데이터(data)를 지정한다.
- send(toAddress, data) : 하위 기능 요소에게 데이터 전달을 요청하기 위한 메소드로서, 해당 데이터의 수신자 주소(toAddress)와 전송하고자 하는 데이터(data)를 입력 변수로 사용한다. toAddress가 0인 경우는 멀티캐스트 전송을 의미하고, 0이 아닌 경우는, 수신 신뢰성 관리 기능 요소에서 여러 복구를 위해 상위 멤버에게 재전송을 요구할 때와 같이, toAddress로의 유니캐스트 전송을 의미한다.

본 논문의 프레임워크에서는 이벤트 기반의 메시지 수신을 가정하기 때문에, 상위 기능 요소나 하위 기능 요소로부터 데이터를 기다리는 기능을 수행하는 메소드를 정의하지 않는다.

그림 3에서 *O*TopMgr 클래스와 *O*Component 클래스의 클래스 다이어그램을 보인다. *O*TopMgr 클래스는 위상 관리 기능 요소를 위한 추상 클래스이다. *O*TopMgr 클래스는 *O*Component 클래스를 상속하여 데이터 송수신에 관한 인터페이스를 상속 받고, 그룹 멤버의 가입 및 탈퇴를 위한 인터페이스를 추가로 정의한다. 위상 관리 기능 요소에서 제공해야 하는 기능들은 *O*TopMgr 클래스 내에서 구현하지 않고, 개별적으로 제안된 해당 응용 계층 멀티캐스트 기법들을 *O*TopMgr 클래스를 상속하여 구현하여, 사용자가 필요에 따라 해당 기법들을 선택할 수 있도록 한다. 만일, TBCP[6]에서 제안하고 있는 위상 관리 알고리즘을 프레임워크 안에 포함시키고자 한다면, 그 TBCP를 구현한 모듈 'TBCP 클래스'는 그림 2에서와 같이, *O*TopMgr 클래스를 상속하는 자식 클래스로 위치한다.

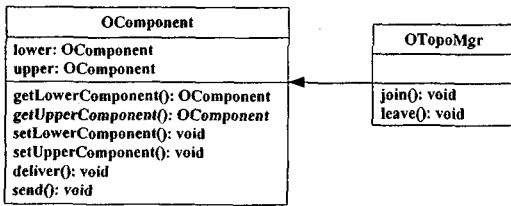


그림 3. 클래스 다이어그램: *O*Component와 *O*TopMgr 클래스

채널 계층은 응용 서비스 프로그램이 구동될 시스템에서 제공하는 TCP 혹은 UDP 등의 전송 계층을 감싸고 있다. 하나의 *O*Channel 클래스 객체가 다수의 TCP 또는 UDP 들을 관리한다. 그림 2에서도 보는 바와 같이 *O*Channel 클래스 또한 *O*Component 클래스를 상속한다.

4.3 응용 서비스의 구현

본 프레임워크의 효율성 및 실제 적용성을 확인하기 위해 본 프레임워크를 기반으로 하는 간단한 문자 중계 서비스를 구현하였다. 프로그램 개발 환경으로 Java SDK 1.3.1을 사용하였다. 그림 4는 구현된 응용 서비스에 이용된 기능 요소의 클래스들 간의 상속 관계와 상하 관계를 나타낸다. 응용 계층 멀티캐스트 기법 중 위상 관리의 전형적인 예로 언급되는 TBCP를 구현한 TBCP 클래스는 *O*TopMgr 클래스를 상속하여 구현하였다. 데이터 송수신을 위해 UDP를 사용하도록 *O*channel 클래스를 상속하여 *UDPChannel*을 구현하였다. UDP가 갖는 비신뢰적인 속성 때문에, *O*ReliableMgr 클래스를 상속하여 *ReliabilityMgr* 클래스를 구현하여 ACK 과 ACK 타이머로 이루어지는 여러 복구 기법(들)을 제공하도록 하였다.

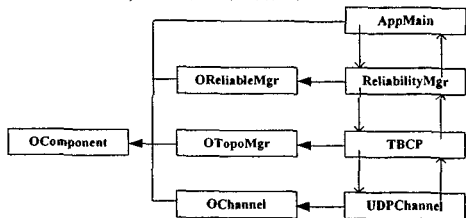


그림 4. 응용 서비스 내의 기능 요소간의 클래스 관계

그림 5의 코드는 구현한 서비스의 프로그램의 한 부분으로, 본문에서 설명하는 프레임워크를 이용하여 응용 서비스를 구현하는 방법의 예를 보인다. 위상 관리 기능 요소인 TBCP 클래스에 해당하는 객체(tmanager)와 신뢰성 관리 기능 요소인 ReliabilityMgr 클래스에 해당하는 객체(rmanager)를 생성한다. 응용 프로그램 내의 최상위 객체는 응용 프로그램 자신이 되고, rmanager가 그 하위 계층이 되며, tmanager가 그 하위에 위치하게 된다. 사용자가 데이터를 송신하고자 하는 경우, 자신의 하위 클래스의 send()를 호출하면, 앞서 정의한 객체들간의 상하 관계를 따라 데이터가 전달된다. 그리고, 최종적으로 그림 4에서 보이는 *UDPChannel* 을 사용하여 데이터가 네트워크를 통해 전송된다.

```

// Create Components
OComponent tmanager = new TBCP(RPADDRESS, TBCP.TBCP_SENDER);
OComponent rmanager = new ReliabilityMgr();

// Setting the upper and lower components of the topology manager,
tmanager.setUpperComponent(rmanager);

// Setting the upper and lower components of reliability manager,
rmanager.setLowerComponent(tmanager);
rmanager.setUpperComponent(this);

// Setting the upper and lower components of the application itself
this.setUpperComponent(null);
this.setLowerComponent(rmanager);

((TBCP)tmanager).join();

// Data Sending Part
while (true) {
    try {
        String line = reader.readLine();

        // Sending data to lower component
        this.getLowerComponent().send(null, line.getBytes());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
    
```

그림 5. 응용 프로그램 예제 코드

5. 결론

응용 계층 멀티캐스트는 IP 멀티캐스트가 안고 있는 배치 (deployment) 문제를 근본적으로 해결하지만 각 응용 서비스 개발 시 적당한 기법을 선택하여 개별적으로 구현해야 한다는 부담을 갖는다. 본 연구에서는 응용 계층 멀티캐스트를 사용하여 응용 서비스를 개발할 경우를 위해 효율적이고 재사용이 가능한 프레임워크를 설계 하고 구현하였다. 본 프레임워크는 응용 계층 멀티캐스트를 위한 기능 요소들을 정의하고 개별 기능 요소들을 빌딩 블록의 형태로 응용 서비스 개발 시 선택적으로 조합하여 개발할 수 있도록 한다. 이는 응용 서비스 개발을 용이하게 할 뿐만 아니라, 또한 응용 계층 멀티캐스트 프로토콜의 쉽고 독립적인 개발 및 시험을 가능하게 할 수 있다.

향후 연구로 프레임워크 내 응용 계층에 해당하는 기능 요소를 위한 추상 클래스들의 인터페이스를 정의하고, 보다 다양한 기능 요소들을 통합 개발하여 본 프레임워크의 효율성과 타당성을 입증하도록 할 것이다.

참고문헌

- [1] S. Deering, "Host Extensions for IP Multicasting," Internet RFC 1112, Aug. 1989.
- [2] Y. Chu, S. G. Rao, and H. Zhang, "A Case for End System Multicast," ACM SIGMETRICS'00, pp. 1-12, Jun. 2000.
- [3] Y. Chu, S. G. Rao, S. Seshan and H. Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture," ACM SIGCOMM'01, pp. 55-67, Aug. 2001.
- [4] D. Pendarakis, S. Shi, D. Verma and M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure," USITS 2001, pp. 49-60, Mar. 2001.
- [5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," ACM SIGCOMM'02, pp. 205-217, Aug. 2002.
- [6] L. Mathy, R. Canonicco, and D. Hutchison, "An Overlay Tree Building Control Protocol," NGC 2001, Nov. 2001.
- [7] Y. Chawathe, S. McCanne, and Eric A. Brewer, "RMX: Reliable Multicast for Heterogeneous Networks," IEEE INFOCOM'00, pp. 795-804, Jun. 2000.
- [8] Y. Chawathe, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service," Ph.D. Thesis, University of California, Berkeley, Dec. 2000.
- [9] R. Kermode and L. Vicisano, "Author Guidelines for Reliable Multicast Transport (RMT) Building Blocks and Protocol Instantiation documents," RFC 3269, IETF RMT Working Group, Apr. 2002.
- [10] B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast to End Users," IEEE INFOCOM'02, Jun. 2002.
- [11] J. Jannotti, D. K. Gifford, K. L. Johnson, M. Frans Kaashoek, and J. W. O'Toole Jr., "Overcast: Reliable Multicasting with an Overlay Network," USENIX Symposium on Operating Systems Design and Implementation, pp. 197-212, Oct. 2000.
- [12] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Transience of Peers and Streaming Media," HotNets-I (2002), Oct. 2002.