

# 분산 실시간 멀티캐스트 프로토콜에서 그룹 관리

노진홍<sup>o</sup> 나성국<sup>\*</sup> 홍영식<sup>\*</sup>  
동국대학교 컴퓨터공학과  
삼성전자주식회사<sup>\*</sup>  
{jhno<sup>o</sup>, nsg94, hongys}@dgu.ac.kr

## Group Management in the Distributed Real-time Multicast Protocol

Jin-Hong No<sup>o</sup> Sung-Guk Na<sup>\*</sup> Young-Sik Hong<sup>\*</sup>  
Dept of Computer Engineering, Dongguk Univ.  
Samsung Electronics Co.<sup>\*</sup>

### 요 약

인터넷 보급의 확산으로 네트워크 컴퓨팅이 급속히 발달하여 멀티캐스트 프로토콜의 형태도 실시간 환경을 요구하는 형태로 변화하고 있다. 분산 실시간 멀티캐스트 프로토콜에서의 통신은 높은 신뢰성을 요구하고, 참가자들의 정보인 뷰(view)의 일관성을 유지하는 작업은 멀티캐스트 프로토콜 기능 중 가장 중요하다고 볼 수 있다. 이러한 뷰의 관리는 멀티캐스트 프로토콜의 전반적인 성능에 매우 큰 영향을 주는 부분이다. 따라서 본 논문에서는 논리적 링 구조로 뷰를 관리하는 실시간 멀티캐스트 프로토콜의 그룹 연산 성능을 분석하기 위하여, TMO 모델을 사용한 실시간 시뮬레이션과 프로토콜 구현을 통한 실험을 하였고, 그 결과를 비교하였다.

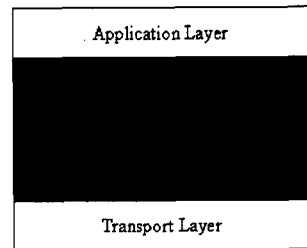
## 1. 서 론

인터넷 기반 네트워크 컴퓨팅이 확산되며 멀티캐스트 기술을 사용한 응용 서비스가 많이 사용되고 있고, 데이터의 형태 또한 실시간성을 요구하는 형태로 변화하고 있다. 그러므로 실시간 데이터를 신뢰성 있고 효율적으로 전송할 수 있는 분산 실시간 멀티캐스트 프로토콜에 대한 요구도 함께 증가하고 있으며, 이러한 실시간 멀티캐스트 프로토콜의 개발은 실시간 분산 처리 시스템의 개발에 필요한 요소라고 할 수 있다. 분산 실시간 멀티캐스트 프로토콜에서의 통신은 높은 신뢰성을 요구하고, 이를 위해 메시지 원자성(atomicity), 전체 순서화(total ordering), 고장 감내(fault tolerance) 등을 제공하는 신뢰성 있는 멀티캐스트 알고리즘들이 연구가 되어왔다. 멀티캐스트는 참가자들의 정보인 뷰(view)에 기반하여 이루어지기 때문에, 송신자와 수신자 모두 뷰의 일관성을 유지하는 작업이 필요하다.

본 논문에서는 논리적 링 구조로 뷰를 관리하는 실시간 멀티캐스트 프로토콜의 그룹 연산에 대해 검토한다. 그룹 관리기를 설계하기 위해서 TMO (Time-triggered Message-triggered Object)[3] 모델을 사용하여 시뮬레이션을 통해 분석하였고, RFRM(Release-time based Fault-tolerant Real-time Multicast Protocol)[4]을 구현하여 비교 분석하였다. 제 2장에서는 RFRM의 구성요소와 멀티캐스트 알고리즘에 대하여 설명하고, 3장에서는 뷰 관리 기법에 관하여 기술한다. 4장에서는 시뮬레이션과 구현한 결과를 검토하고 5장에서는 본 논문의 결론을 제시한다.

## 2. RFRM[3]

RFRM은 통신 채널, 멀티캐스트 관리자, 고장 관리자, 그룹 관리자의 네 가지의 객체로 구성된다.



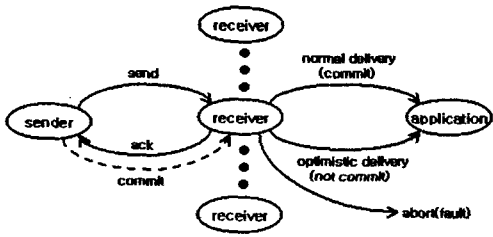
[그림 1] 프로토콜 스택

통신 채널은 UDP 방식으로 메시지 송/수신을 하며, 각 관리자에게 메시지를 전달하는 역할을 한다.

멀티캐스트 관리자는 응용프로그램으로부터 요청받은 멀티캐스트 메시지를 그룹 멤버들에게 전송한다. 수신된 멀티캐스트 메시지는 멤버들이 유지하고 있는 큐에 삽입되고 ORT(Official Release Time) 동안 유지된다. 멀티캐스트 송신자가 ORT 동안에 모든 그룹 멤버들로부터 ack 메시지를 수신 받는다면 Commit 메시지를 멤버들에게 전송하여 신뢰성 있는 메시지임을 알려주고, 멀티캐스트 메시지는 응용 프로그램에 전달된다. 만약 ORT가 지나도 모든 그룹 멤버들로부터 ack 메시지를 수신 받지 못 한다면 낙관적(optimistic) 멀티캐스트 메시지로 응용프로그램에 자동 전달된다. 그렇지 않고 ORT가 지나지 않은 동안 고장이 발생하는 등초포 멀티캐스트 메시지를 취소해야 한다면 멤버들에게 ORT가 종료되기 전에 큐에서 삭제하도록 abort 메시지를 보낸다.

[그림 2]는 RFRM의 메시지 전달 구조를 간단하게 나타내고 있다. 멀티캐스트 메시지의 송신자로부터 commit 메시지가 도착하는지에 따라 일반적인 멀티캐스트 메시지인지, 낙관적 멀티캐스트 메시지인지가 결정된다.

본 연구는 한국과학재단 특정기초 연구(R01-2000-000-00284-0)의 결과물임



[그림 2] 메시지 전달 구조

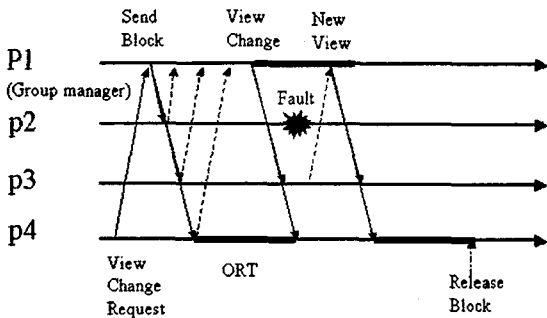
그룹 관리자는 멤버의 가입과 탈퇴와 같은 그룹 연산과 그룹 뷰의 변경을 담당한다. 또한 각 관리자에게 현재 그룹 정보를 제공하는 역할도 한다.

고장 관리자는 SNS(Supervisor-based Network Surveillance) 방법[5]과 유사하게 동작하며, 논리적 링 구조에 기반한 방법으로 이웃노드의 고장을 탐지한다. 중앙 집중식 감독자(supervisor)가 존재하고, 모든 노드들은 자신의 존재를 알리기 위해 이웃 노드에게 주기적으로 heartbeat signal을 보낸다. 또한 송신자나 수신자의 순간적인 에러(transient error)를 방지하기 위해 각각의 heartbeat 시그널은 두 번씩 중복 전송한다. 반대로 모든 노드들은 이웃 노드로부터 주기적으로 heartbeat 시그널을 수신하고 일정 기간동안 heartbeat 시그널이 도착되지 않는다면 이웃 노드의 고장을 감지할 수 있다.

### 3. 그룹 연산

그룹통신에서의 그룹관련 연산은 각 멤버간의 동일한 뷰를 유지하는 것이 필수이다. 이를 보장하기 위한 특성인 가상 동기화(virtual synchrony)를 위해서 그룹 뷰를 수정하기 전 각 멤버들의 blocking을 요구하고, 모든 멤버들이 blocking 상태가 된 것을 확인한 후 새로운 뷰를 적용한다.

[그림 3]은 그룹 뷰 변경 처리 과정을 보여 주고 있다.



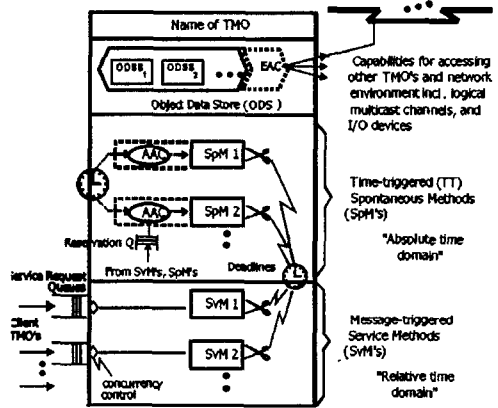
[그림 3] 뷰 변경 과정

일관성 유지를 위한 가상 동기화를 보장하기 위해서는 뷰 변경 중에 일어날 수 있는 간섭을 방지해야 하므로 blocking이 필요하고, 뷰 변경 요청을 받은 그룹관리자는 각 멤버들에게 blocking 요청을 멀티캐스트한다. Blocking 요청 메시지를 받

은 각 멤버들은 그룹 관리자에게 ack 메시지를 전송하고, 그룹 관리자는 그룹 멤버 모두에게서 ack 메시지를 수신하였다면 새로운 뷰(normal view)를 전송한다. ORT동안 모든 멤버로부터 ack 메시지를 수신하지 못한 경우 낙관적 뷰(optimistic view)로 구별하여 새로운 뷰를 멤버들에게 전송한다. 새로운 뷰를 받은 그룹 멤버들은 ORT가 지난 후 뷰를 변경 적용하고 blocking을 해제한다. 뷰를 전송받은 후 ORT 안에 고장이 발생한다면 그룹 관리자는 이전 뷰를 취소하고 새로운 뷰를 전송한다.

### 4. 실험 및 성능 분석

본 논문에서 제안한 모델을 시뮬레이션하기 위해서 TMO 모델을 기본 모델로 채택하였고, [그림 4]와 같이 TMO모델은 세 가지 객체로 구성되어 있다.



[그림 4] TMO 시뮬레이션 모델

SpM(spontaneous method)은 주어진 시간 조건 AAC(automatic activation condition)가 만족되면 자동으로 호출되며, SvM(service method)은 이벤트에 반응하는 메소드이다. 또한 ODSS(object data store)는 SpM과 SvM사이의 데이터 공유 및 동기화의 역할을 수행한다.

TMO 모델을 사용한 실험에서 사용한 파라미터는 다음과 같다. 제안된 그룹관리기법을 실험하기 위해 실험파라미터는 노드수를 4개에서 6개로 증가시키면서 실험하였고 ORT는 200ms로 실험하였다[8].

Message	Multicast	80%
	Join	10%
	Leave	10%
HB signal 주기		1 sec
노드 수		4,5,6개
ORT		200ms
실험시간		500sec

[표 2] 실험 파라미터

RFRM에서 그룹 연산 성능을 측정하기 위하여 동일한 실험

파라미터를 적용하였고 측정된 결과는 [표 3]과 같다.

노드수	Multicast		Join		Leave	
	발생 횟수	평균 처리 시간	발생 횟수	평균 처리 시간	발생 횟수	평균 처리 시간
4	151	202	24	406	24	406
5	150	203	27	405	27	405
6	152	204	29	406	30	406

[표 3] RFRM에서의 그룹 연산 측정

TMO를 이용한 실험에서 그룹 연산의 처리시간이 한 번의 멀티캐스트 처리 시간에 비하여 10ms정도의 시간이 증가된 것을 알 수 있었다[8]. 이는 blocking 요청 메시지만 ORT 동안 유지하고 release하였기 때문이다. 구현에서는 뷰의 일관성을 향상시키기 위하여 blocking 해제 메시지도 ORT 동안 유지하고 release 하였다. 그러므로 구현된 그룹 연산 기법에서는 그룹 뷰를 변경하기 위하여 두 번의 멀티캐스트를 행하였고, 이는 일관성 유지를 위해서는 감수할 만한 것으로 평가할 수 있다.

### 5. 결론 및 향후과제

그룹 관리 기능은 멀티캐스트 프로토콜에 있어 효율성과 신뢰성을 좌우하는 핵심적인 기능이다. 본 논문에서는 실시간 환경의 시간제약을 만족하고 신뢰성을 제공하기 위해, 본 논문에서는 뷰 변경과정에서 ORT와 고장처리를 사용하였다. 이는 실시간 멀티캐스트 응용 프로그램에서 제어 메시지의 양을 줄일 수 있다는 장점을 가지고 있다. 그리고 그룹 관리의 비용을 측정하기 위하여 TMO 모델을 이용하여 시뮬레이션 했고, 실시간 멀티캐스트 프로토콜 RFRM을 구현하고 그룹 연산 시간을 측정했다. 이는 blocking으로 인한 시간낭비를 최소화하기 위해서 메시지 blocking없이 ORT시간 안에 고장이 발견되지 않으면 각 멤버간에 새로운 뷰를 release하는 낙관적인 방식으로도 사용가능하다. 이 경우 ORT안에 고장이 발견되면 그동안의 메시지는 모두 롤백(rollback)해야 하는 단점이 있다.

향후 과제로는 그룹 연산 시 지연을 줄이기 위한 낙관적인 방법의 적용에 따른 그룹 연산 비용 및 롤백 비용 등을 측정할 계획이다.

### 6. 참고문헌

[1] Y. Chu, S. Rao, and H.Zhang. "A Case For EndSystem Multicast." In Proceedings of ACM Sigmetrics, Santaclar, CA, June 2000.

[2] G. Grnsteidl and H. Kopetz, "A Reliable Multicast Protocol for Distributed Real-Time Systems", Proceedings of the 8th IEEE Workshop on Real-Time Operating Systems and Software, Atlanta, U.S.A., 1991.

[3] Y.S. Hong, "Distributed object-oriented real-time

simulation of the multicast protocol RFRM", Proc. of 7th IEEE int. Workshop on Object-Oriented Real-Time Dependable Systems(WORDS 2002), pp.215-218, 2002.1.

[4] K. H. Kim, "Object Structures for Real-Time Systems and Simulators", IEEE Computer, August 1997, pp.62-70.

[5] K. H. Kim and C. Subbaraman, "Dynamic Configuration Management in Reliable Distributed Real-Time Information Systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, 1999.

[6] Jeremy Sussman, Idit Keidar, and Keith Marzullo, "Optimistic Virtual Synchrony.", In the 19th IEEE Symposium on Reliable Distrubted Systems, page 42-51, October 2000.

[7] Jeremy Sussman, Idit Keidar, and Keith Marzullo, "Optimistic Virtual Synchrony.", In the 19th IEEE Symposium on Reliable Distrubted Systems, page 42-51, October 2000.

[8] 나성국, 홍영식, "분산 실시간 시스템을 위한 가상 토폴로지에서의 그룹연산", 한국정보과학회, 제29권 2호, pp322-324, 한국, 2002년 가을.