이동통신망을 위한 멀티캐스트 메시지 전달 알고리즘의 설계 및 평가

Design and Evaluation of Multicast Message Delivery Algorithm for Mobile Networks

장익현
동국대학교 정보통신공학부

Ik-Hyeon Jang(ihjang@dongguk.ac.kr)

요약

본 논문에서는 이동통신망을 위한 효율적인 멀티캐스트 인과순서 알고리즘과 채널전환 프로토콜을 제안하였다. 메시지 전달 순서를 유지하기 위한 제어정보의 크기는 이동통신망에서의 채널전환과 메시지 전송 성능에 큰 영향을 주므로 제어정보의 크기를 최소화할 필요가 있다. 이를 위해 모든 유효한 통신패턴을 분석하여 인과순서를 유지하는데 필수적이지 않은 중복정보를 가능한 이른 시기에 찾아내어 제거하고, 전송되는 제어정보를 최소화하는 채널전환 프로토콜을 사용하였다. 시뮬레이션을 통해 제안한 알고리즘이 기존의 알고리즘보다 더 좋은 성능을 보임을 보였다.

■ 중심어 : | 인과순서 | 채널전환 | 이동통신망 |

Abstract

In this paper, we proposed an effective multicast causal order algorithm with hand-off protocol for mobile networks. Since the size of control informations needed to enforce message transfer order has much influence on the performance of hand-off and message transfer in mobile networks, size of control information need to be minimized. We reduced the size of control information by analyzing all the valid communication patterns and pruning redundant information not required to enforce causal order as early as possible, and used hand-off protocol which requires minimal amount of control information to be transferred. By simulation, we found that the proposed algorithm showed better performance than other existing algorithms.

■ keyword : | Causal Order | Hand-off | Mobile Networks |

I. 서론

분산시스템은 메시지를 통해서만 통신하는 여러 개의 프로세스로 구성된다. 시스템을 구성하는 프로세스들은 메모리를 공유하지 않으며, 전역시간이나 완벽하게 동기화된 지역시간도 제공되지 않는다. 따라서 서로 다른 프로세스에서 발생하는 사건들의 순서를 맞추는 것은 분산시스템의 근본적인 문제 중의 하나인 비결정성(non-determinism)의 제어에 속하는 문제이다[1][9].

메시지의 전송시간은 유한하지만 전달시간의 차이 때문에 송신하는 메시지들의 순서와 전달되는 메시지들의 순서가 일치하지 않을 수 있다. 이러한 통신에서의 비결정성을 줄이기 위한 노력으로 Birman은 메시지의 인과순서전달을 제안하였다[2]. 동일한 목적지로 보

접수번호 : #090826-003
접수일자 : 2009년 08월 26일
심사완료일 : 2009년 09월 24일
교신저자 : 장익현, e-mail : ihjang@dongguk.ac.kr

내진 두 개의 메시지 m_1과 m_2를 가정하자. 만약 m1이 m_2보다 먼저 보내졌고 m_1이 m_2보다 목적지에 먼저 전달된다면 메시지 전달의 인과순서가 지켜졌다고 하고 그렇지 않을 경우 인과순서가 지켜지지 않았다고 한다.

중요한 점은 메시지를 인과순서에 맞게 전달하기 위해서는 많은 양의 관련 정보를 함께 전달해야 한다는 것이다. 이로 인한 많은 오버헤드는 알고리즘의 확장성을 제한한다.

메시지의 인과순서 전달은 데이터의 중복 관리, 자원의 할당, 분산 공유 DB, 전자 게시판, 분산시스템의 모니터링 등의 응용분야를 가지고 있으며, 특히 사람과의 대화를 포함하는 정보제공, 증권거래, 원격회의 등의 분산이동시스템에 적합하다[1][6][8][9].

고정망을 위해 설계되었던 인과순서 알고리즘을 이동통신망에 적용하기 위해서는 이동통신시스템의 특성을 고려하여야 한다. 이동통신시스템에서는 이동유닛의 메모리 크기와 프로세서의 성능, 배터리의 용량이 제한되어 있고, 이동유닛과 네트워크간의 연결이 단절될 수도 있으며, 이동유닛이 셀과 셀 사이를 이동하며, 무선통신채널의 대역폭은 유선통신채널의 대역폭보다 적다[1].

따라서 이동통신시스템을 위한 알고리즘은 채널전환을 효율적으로 처리하는 프로토콜을 포함해야 하며, 통신채널의 대역폭이 적다는 점을 고려하여 인과순서를 위해 메시지와 함께 전송되는 제어정보의 양을 최소화하고, 이동유닛에서의 작업이 많지 않아야 한다.

본 논문에서는 인과순서화를 기초로 하는 멀티캐스트 알고리즘과 이동통신망에서 효율적인 채널전환 프로토콜을 제안한다.

인과순서를 유지하기 위하여 필요한 제어정보를 이동유닛이 관리하는 경우에는 메시지를 보낼 때마다 제어정보를 기지국에 전달해야 하므로 이동유닛과 기지국 사이의 대역폭이 낭비되고 이동 유닛의 부하가 커진다. 따라서 본 논문에서는 효율적인 알고리즘을 구현하기 위하여 인과순서화에 대한 정보를 각각의 이동유닛이 속한 기지국이 관리하도록 하였다. 인과순서를 위한 제어정보의 크기 감소로 네트워크의 대역폭을 효율적으로 이용하고 시스템의 성능을 향상시키기 위하여 Jang[5]이 유선망을 위하여 제안한 인과순서 알고리즘을 이동통신망에 적용시킬 수 있도록 수정하였다.

Alagar가 사용한 Raynal의 알고리즘[1][8] 및 Prakash의 알고리즘[7]과 시뮬레이션을 통해 비교하여 제안한 알고리즘의 성능이 우수함을 보였다.

본 논문의 구성은 다음과 같다. 2장에서는 이동통신망의 모델과 인과순서에 관해 소개한다. 3장에서는 효율적인 인과순서메시지 전달을 위해 제어정보의 크기를 줄이는 방법에 대해 다루고, 4장에서는 이동통신망에서 멀티캐스트를 지원하는 인과순서 알고리즘과 채널전환 프로토콜을 제안한다. 5장에서는 제안한 알고리즘을 시뮬레이션을 통해 성능을 평가한다. 마지막으로 6장에서는 결론을 맺는다.

II. 시스템 모델 및 관련 연구

2.1 이동통신망

이동통신망은 분산시스템의 일종으로 [그림 1]와 같이 이동유닛(mobile host, MH)과 고정유닛인 기지국(mobile support station, MSS)으로 구성된다. 이동유닛과 고정유닛의 통로역할을 하는 기지국은 고정유닛 중의 하나일 수도 있고, 독립적으로 존재할 수도 있다. 기지국들은 유선통신채널로 서로 연결되어 있어 다른 고정유닛들과 유선네트워크를 통해 통신한다. 기지국들

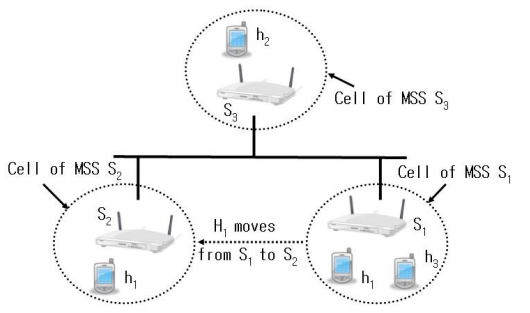

그림 1. 이동통신망 모델과 채널전환

은 무선통신채널을 통하여 이동유닛과 통신을 하는데 하나의 기지국이 이동유닛과 통신할 수 있는 지리적인 영역을 셀(cell)이라고 한다. 이동유닛이 기지국과 통신하기 위해서는 이동유닛이 그 기지국이 관리하는 셀 안에 위치해야 하며 하나의 이동유닛은 임의의 시간에 최대 한 개의 셀에만 속해 있어야 한다[1].

임의의 두 기지국 사이의 메시지 전달은 신뢰할 수 있으며 순차적이라고 가정한다. 또한 기지국과 이동유닛 사이의 메시지 전달은 선입선출의 특성을 갖는다고 가정한다. 즉, 기지국과 이동유닛 사이에서 이동되는 메시지는 특별한 순서화기법을 사용하지 않더라도 전송되는 순서와 전달되는 순서가 일치한다고 가정한다.

[그림 1]의 H_1과 같이 이동유닛이 셀 사이를 이동할 때 이동유닛이 포함되는 셀이 바뀜에 따라 기지국도 바뀐다. 이때 이동하기 전의 기지국이 가지고 있던 이동유닛에 대한 정보는 새로운 기지국에 전달되어야 하고 이동유닛은 셀이 변경되어도 작업을 계속 수행할 수 있어야 한다. 이와 같이 이동유닛의 셀이 변경됨에 따라서 수행되는 일련의 작업을 채널전환이라 한다[1].

2.2 인과순서 알고리즘

분산시스템은 여러 개의 프로세스들로 구성되어 있고, 이 프로세스들은 서로 메시지 교환을 통해 통신을 한다. 그러나 메시지들의 발생시간이 서로 다르고 전송되는 시간의 차이 때문에 순서에 맞지 않게 목적지에 도착할 수 있다. 따라서 메시지들의 순서를 유지하기 위한 방법으로 인과순서가 제안되었다[2].

send(m)을 메시지 m을 전송하는 이벤트, deliver(m)을 메시지 m을 전달하는 이벤트라고 한다면 인과순서 메시지 전달은 다음과 같이 정의할 수 있다[2].

정의 (인과순서 메시지 전달) : 메시지 m_1과 m_2가 같은 목적지를 가지며, send(m_1)이 send(m_2)보다 먼저 발생한다면 deliver(m_1)도 deliver(m_2)보다 먼저 발생하여야 한다.

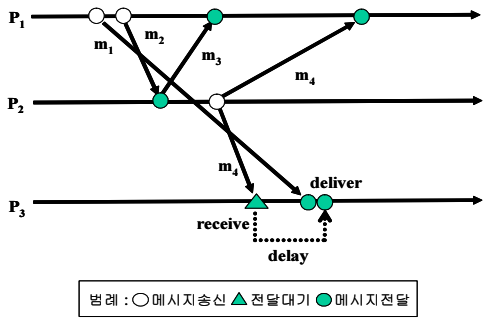

그림 2. 인과순서 메시지 전달

[그림 2]는 인과순서 메시지전달의 예를 보여주는 것으로 가로축은 시간을 나타낸다. m_1과 m_4는 모두 P_3를 목적지로 하는 메시지이고 m_4는 m_1보다 뒤에 발생한 이벤트이다. 따라서 m_1보다 m_4가 P_3에 먼저 수신되었지만 m_1이 전달되기를 기다려서 그 후에 전달되어야 한다. 그러나 동일한 메시지인 m_4가 P_1에 전달되는 경우에는 m_4보다 이전의 메시지인 m_3가 이미 P_1에 전달되었기 때문에 자신보다 앞선 메시지를 기다리지 않고 바로 전달될 수 있음을 보이고 있다.

인과순서 메시지 전달에 대해서는 Birman이 ISIS 시스템을 위하여 제안한 이후 많은 연구가 있었지만 고정망을 위한 연구[2][3][5][6][8]가 주를 이루었으며, Birman의 결과가 너무 많은 오버헤드를 가졌기 때문에, 연구의 주된 이슈는 오버헤드를 줄이는 것이었다. 대표적인 연구인 Raynal의 알고리즘은 각각의 메시지에 NxN 매트릭스의 제어정보를 포함시키는 것으로 최악 공간복잡도는 $O(n^2)$이다. 나머지 대부분의 연구도 유사한 결과를 보였다. Jang[5]과 Kshemkalyani[6]는 인과순서의 유지를 위한 조건을 연구하여 최악공간복잡도는 $O(n^2)$로 동일하지만 평균 공간복잡도의 크기를 크게 줄였다.

한편 Alagar[1]와 Jang[4], Prakash[7]는 통신망의 변화에 맞춰 이동통신망을 위한 인과순서화 알고리즘에 대한 연구를 진행하였다. 그러나 제안한 알고리즘과 다른 알고리즘과의 상대비교는 이루어지지 않고 있다. 따라서 본 논문에서는 이동통신망을 위한 인과순서 알고리즘을 제안하고, 제안한 알고리즘과 대표적인 다른 알고리즘의 성능을 비교한다.

III. 제어정보의 관리

3.1 인과관계의 정리

인과순서에 맞게 메시지를 전달하기 위해서는 먼저 메시지들 간의 인과관계를 파악할 수 있어야 한다. 인과관계를 파악하는 방법은 제어정보를 어떻게 정리하느냐에 따라 두 가지로 분류할 수 있다.

첫 번째 방법은, 인과관계를 파악할 수 있는 제어정보를 목적지에 따라 정리하는 것으로 지금까지 발표된 대부분의 알고리즘[1][2][3][7][8]이 여기에 속한다. 이 방법의 장점은 메시지의 전달조건 검사가 쉽다는 것이다. 그러나 인과관계들 간의 우선순위를 파악하기 위해서는 제어정보 전체를 확인해야 하므로 많은 시간이 필요하다.

또 다른 방법은 발신지를 기준으로 인과관계를 정리하는 것으로 전송되는 메시지와 그와 함께 전송되는 제어정보를 하나의 객체로 처리한다[5][6]. 이 방법은 목적지를 기준으로 인과관계를 검사해야 하는 전달조건의 검사는 상대적으로 어렵다. 전달조건을 검사하기 위해서는 전체 인과관계를 확인해야 하기 때문이다. 그러나 메시지들 간의 우선순위 관계를 파악하는 것이 쉽고 메시지와 함께 전달해야할 제어정보의 양을 줄일 수 있어 앞의 방법보다 유리한 점이 있다.

본 논문에서는 제어정보들 간의 우선순위를 찾는 것이 중요하므로 두 번째 방법을 채택하며, 전달조건의 신속한 확인을 위해 송신 시에 전달조건 확인을 위한 정보를 목적지를 기준으로 별도로 추출한다.

3.2 메시지 패턴의 분류

전송되는 메시지 m에 대한 정보를 가지는 상태를 송신프로세스와 목적프로세스를 기준으로 분류하면 모든 프로세스는 다음 4가지 중 하나의 상태에 속한다[5].

① 송신프로세스의 상태
② 목적프로세스의 상태
③ 송신과 목적프로세스 사이의 프로세스의 상태
④ 목적프로세스 이후에 있는 프로세스의 상태

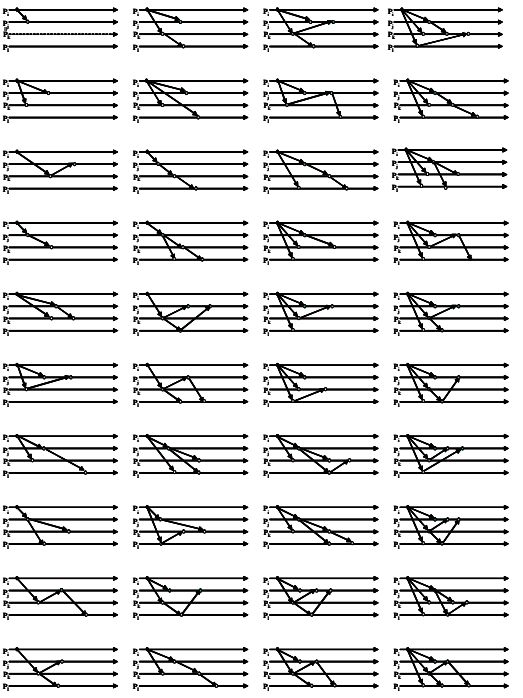

그림 3. 유효한 모든 통신 패턴

만약 두 개의 프로세스가 동일한 상태 정보를 가지고 있다면 그 두 프로세스는 메시지 m에 관한 정보에 대해서는 하나의 프로세스로 볼 수 있다. 따라서 분산시스템은 4개의 프로세스로 이루어진 시스템으로 변환이 가능하며, 어떤 메시지에 관한 정보의 흐름 관점에서

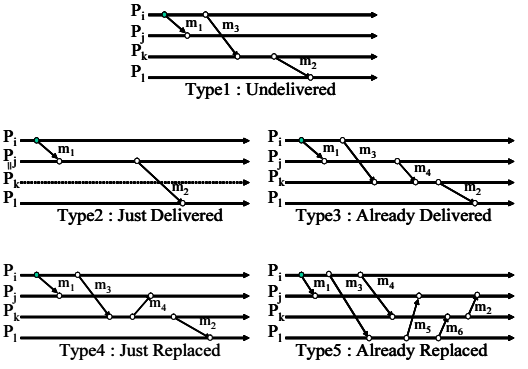

그림 4. 추상통신패턴 (P_k의 m_2 전송시 m_1에 대한 정보의 흐름을 기준으로 분류)

보면 다중전송(multicast)도 단일전송(unicast)으로 변환되므로 [그림 3]에 나타낸 인과순서를 제공하는 유효한 모든 통신패턴은 [그림 4]에 있는 추상통신패턴 중의 하나로 변환이 가능하다[5].

[그림 4]는 P_k가 m_2 전송시에 m_1에 대한 정보를 어떤 식으로 유지하고 있는지를 기준으로 정리한 추상통신패턴으로 다음과 같이 정의된다.

발신프로세스와 발신지와 목적지 사이에 있는 프로세스들은 메시지가 목적지에 전달되었는지 알 수가 없으므로 이런 프로세스들은 Type1이 된다. 목적지에서는 해당 메시지가 전달되었다는 것을 바로 알 수 있으며 Type2가 된다. 목적지 이후에 있는 프로세스들도 m_1이 목적지에 이미 전달되었다는 것을 알 수 있으며 Type3이 된다. 목적지와 발신지 사이에 있는 프로세스들은 m_1 이후에 인과순서상 뒤에 있는 다른 메시지가 동일 목적지로 전송될 경우, 그들 간의 인과순서를 유지하기 위하여 앞의 메시지는 뒤의 메시지보다 먼저 전달될 것이므로, 인과순서상 뒤에 있는 메시지에 대한 정보만 유지해도 된다[5,7]. Type4에서 m_1은 m_4에 의해 대치되었다고 한다. Type4를 일반화하면 Type5가 된다.

3.3 중복정보의 제거

만약 Type2 프로세스들이 어떤 이유로든 메시지 m_1에 대한 정보를 가지고 있다면 그 정보는 인과순서를 유지하는데 필수적이지 않은 중복정보이다. 즉 중복정보는 Type2부터 Type5 사이의 통신패턴을 가지는 프로세스가 인과순서를 유지하는데 필수적이지 않은 정보를 가지고 있을 경우로 이러한 정보의 파악과 제거는 제어정보의 크기를 줄이는 핵심이 된다.

동일한 프로세스에서 전송된 메시지들 간의 인과순서는 메시지 송신순서와 같으므로 쉽게 파악이 된다. 하나의 프로세스에는 동일한 프로세스에서 전송한 메시지에 대해서는 최종 송신메시지에 대한 정보만 유지하면 인과순서의 파악이 가능하다. 따라서 나머지 메시지에 대한 정보는 중복정보이므로 삭제할 수 있다.

중복정보는 전송된 메시지에 부속된 제어정보와 각 노드에 있는 제어정보를 비교하여 파악하며 인과순서상 가장 뒤의 메시지에 대한 정보만 유지하도록 하여 중복정보를 제거한다. 단 이 경우 모든 송신 노드에 대해 최소한 하나의 메시지에 대한 정보는 남겨서 이후에 인과순서 파악에 사용할 수 있도록 한다.

IV. 이동인과순서 알고리즘

4.1 자료구조 및 전달조건

프로세스 P_i의 제어정보는 (k, t)의 집합 형태로 발신지별로 분류하여 CI_i 벡터로 관리하며 메시지를 보낼 때마다 함께 보낸다. 여기서 t는 메시지를 보낸 순서를 나타내는 논리시간으로 메시지를 송신할 때마다 하나씩 증가되며, k는 메시지의 목적지를 나타낸다. 멀티캐스팅의 경우 k는 복수의 목적지가 될 수 있다. (k, t)가 $CI_i[j]$에 포함되어 있다면 프로세스 P_i가 프로세스 P_k에게 보내는 메시지는 프로세스 P_j의 t번째 메시지가 프로세스 P_k에 전달된 뒤에만 전달될 수 있다는 것을 나타낸다. 자료구조에 표기되는 아래첨자 i, j는 자료구조의 소속 프로세스를 나타내고 m은 메시지로 수신된 것임을 나타낸다.

P_i가 P_k에 메시지를 보내는 경우에 목적지에서의 전달조건을 신속하게 확인하기 위하여, CI에 P_j가 보낸 메시지의 목적지로 P_k를 포함하는 것이 있으면($k \neq j$), (P_j, t)를 $DCi[k]$로 별도로 분리하고, P_k를 CI에 있는 제어정보의 목적지에서 제거한 후, CI와 DC를 메시지와 함께 P_k로 보낸다.

한편 각 프로세스 P_i는 전달정보를 관리하는 정수형 배열 DLV_i를 가진다. DLV_i에는 각 프로세스로부터 P_i에 전달된 최종 메시지의 논리시간이 기록된다.

프로세스 P_i에 도착한 메시지는 메시지와 함께 전송된 제어정보에 포함된 인과관계 중에서 P_i를 목적지로 하는 메시지들이 모두 P_i에 전달되었다는 것이 확인되면 P_i에 전달될 수 있다. 이를 전달조건이라 하며 다음 식과 같이 표현한다.

$\forall\ (k, t) \in DC_m[i] : (t \leq DLV_i[k])$

전달조건이 만족되지 않으면 해당 메시지는 전달조건이 만족될 때까지 목적프로세스에 전달되지 않고 대기하여 인과순서를 유지하도록 한다.

4.2 알고리즘

이동통신망을 위한 알고리즘은 고정모듈과 채널전환 모듈로 구성된다. 고정모듈은 이동유닛이 특정 셀에 있을 때 실행되는 모듈로 송신모듈과 수신모듈이 있으며, 채널전환 모듈은 이동유닛이 한 셀에서 다른 셀로 이동할 때 실행된다. 이동유닛에 들어오고 나가는 모든 메시지는 해당 기지국을 통하므로, 이동유닛의 제한된 자원을 감안하여, 각 이동유닛에 대한 모든 제어정보는 기지국에서 관리하도록 한다.

4.2.1 고정모듈

S_k에 있는 이동유닛 h_i가 S_l에 있는 이동유닛 h_j에게 메시지를 보낼 때 두 이동유닛이 동일한 셀에 있을 경우에는 제어정보에 대한 처리가 동일 기지국내에서 일어나므로 제어정보는 기지국과 이동유닛 사이에서 전송되지 않는다. 그러나 두 이동유닛이 서로 다른 셀에 있으며 제어정보도 메시지와 함께 전송되어야 한다.

S_k의 셀에 있는 이동유닛 h_i가 S_l의 셀에 있는 이동유닛 h_j에게 메시지를 보내는 과정은 다음과 같다.

이동유닛 h_i는 먼저 메시지 m을 자신의 기지국 S_k에 보낸다. S_k는 이동유닛 h_j가 속해있는 기지국 S_l로 CI_m과 DC_m을 작성하여 메시지 m과 함께 보낸다.

메시지 m과 제어정보 CI_m, DC_m을 수신한 기지국 S_l은 전달조건을 검사하여 전달조건이 만족되지 않으면 이들을 MH_PEND_j에 대기시키고, 전달조건이 만족되면 메시지 m을 이동유닛 hj에게 보내고, DLV_j의 값을 갱신한 후 m과 C_l를 $PEND_ACK_j$에 대기시킨다. 이동유닛 h_j로부터 ACK를 받으면 기지국 S_l은 m과 C_l를 $PEND_ACK_j$에서 삭제하고 CI를 갱신한다. 기지국 S_l은 MH_PEND_j에 대기 중인 메시지 중에서 전달 가능한 메시지가 있으면 위와 동일한 절차를 진행시킨다.

기지국과 이동유닛간의 통신에는 기존에 나와 있는 어떤 프로토콜도 사용할 수 있다. 아래의 송수신 알고리즘에서 기지국과 이동유닛 사이에 전달되는 ACK 교환과 채널전환의 *register* 전송의 표시는 생략하였다.

가. 송신알고리즘

이동유닛 P_i가 P_j로 메시지 m을 송신할 때 이동유닛 P_i가 속해 있는 기지국에서 실행되는 모듈로 P_j는 P_i의 멀티캐스트 목적지 중의 하나가 될 수 있다.

```
begin
   t_i := t_i + 1                          // 송신 논리시간 증가
   DC_i := ∅;
   for j ∈ m.D and m^τ_k ∈ CI_i    // Type-4 중복정보 제거
      if (j ∈ m^τ_k.D) then
         DC_i[j] := DC_i[j] ∪ {(k, τ)};
         m^τ_k.D := m^τ_k.D - {j};
      endif
   endfor
   for m^τ_k, m^υ_k ∈ CI_i              // Type-5 중복정보 제거
      if (m^τ_k.D = ∅ and τ < υ) then  CI_i := CI_i - m^τ_k;
   endfor
   for j ∈ m.D
      SEND (p_i, τ_i, m.D, CI_i, DC_i[j], m) to p_j;
   endfor
   CI_i[i] := CI_i[i] ∪ {(τ_i, m.D)};
end
```

나. 수신알고리즘

이동유닛 Pi가 Pj로부터 메시지 m을 수신할 때 이동유닛 Pi가 속해 있는 기지국에서 실행되는 모듈로 이 모듈은 원자적으로(atomic) 실행되어야 한다.

```
begin
   wait (∀(k,τ) ∈ DC_m : (τ ≤ DLV_i[k]));
   Deliver m to p_i;
   DLV_i[j] :=τ_m;
   CI_m[j] :=CI_m[j]∪{(τ_m, m.D-i)}; // Type-2 중복정보 제거
   for m^τ_k ∈ CI_i, m^υ_k ∈ CI_m  // Type-3,5 중복정보 제거
      if (m^τ_k ∉ CI_m and τ < υ) then CI_i := CI_i - m^τ_k;
      if (m^υ_k ∉ CI_i and τ > υ) then CI_m := CI_m - m^υ_k;
   endfor
   for m^τ_k ∈ CI_i, m^υ_k ∈ CI_m and τ= υ  // for multicast
       m^τ_k.D := m^τ_k.D ∩ m^υ_k.D;
       CI_m := CI_m - m^υ_k;
   endfor
   CI_i := CI_i ∪ CI_m;
   for m^τ_k, m^υ_k ∈ CI_i      // Type-4 중복정보 제거
      if (τ < υ and l ∈ m^τ_k.D ∩ m^υ_k.D) then
         m^τ_k.D := m^τ_k.D - {l};
   endfor
   for  m^τ_k,  m^υ_k ∈ CI_i   // 최종메시지는 남김
      if (m^τ_k.D = ∅ and τ < υ) then CI_i := CI_i - m^τ_k ;
   endfor
end
```

4.2.2 채널전환프로토콜

이동유닛 h_i가 기지국 S_k에서 다른 기지국 S_l로 이동한다고 가정하자. 이 경우 채널전환절차는 기지국 S_k와

S_l에서 진행된다.

이동유닛 h_i가 기지국 S_l의 셀에 들어가면 이동유닛 h_i는 먼저 *register(h_i, S_k)*를 기지국 S_l에 보내서 자신이 기지국 S_k의 셀에서 이동해 왔음을 알리고 이전의 S_k의 셀에서 ACK를 받지 못한 메시지들을 S_l에게 다시 보낸다. 이 메시지를 받은 기지국 S_l은 *handoff_begin(h_i)* 메시지를 기지국 S_k에 보낸다.

handoff_begin(h_i) 메시지를 받은 기지국 S_k는 CI_i, DLV_i, MH_PEND_i, $PEND_ACK_i$를 기지국 S_l에 보내고 마지막으로 *handoff_over(h_i)*를 S_l에 보낸다. 기지국 S_k에서는 *handoff_over(h_i)*를 보낸 뒤에 이동유닛 h_i로부터 오는 모든 메시지는 폐기한다.

기지국 S_k가 보낸 메시지와 제어정보를 받은 기지국 S_l은 먼저 $PEND_ACK_i$에 있는 메시지 중 전달조건을 만족하는 모든 메시지를 순서대로 처리한다. 또한 이동유닛 h_i로부터 재전송된 메시지를 해당 목적지로 보내고 채널전환을 종료한다.

가. 기지국 SI에서 실행되는 모듈

① h_i로부터 *register(h_i, S_k)*를 받은 후 *handoff_begin(h_i)* 을 S_k로 전송

② CI_i, DLV_i, MH_PEND_i, $PEND_ACK_i$와 *handoff_over(h_i)*를 S_k로부터 수신

③ $PEND_ACK_i$의 메시지 중 hi로부터 ACK를 받았으나 제어정보의 정리가 필요한 메시지는 다음을 수행 후 PEND_ACKi의 메시지 삭제

$\forall$ k, l : $CI_i[k,l] = \max(CI_i[k,l], CI_m[k,l])$

④ $PEND_ACK_i$에 있는 메시지 중 아직 ACK를 못 받은 메시지는 차례로 h_i로 보내어 ACK를 받은 후 C_l를 갱신하고 m을 PEND_ACK에서 삭제

⑤ 이동유닛 h_i가 보낸 메시지 중 미전송 메시지가 있으면 해당 목적지로 메시지 전송

나. 기지국 S_k에서 실행되는 모듈

① 기지국 S_l로부터 *handoff_begin(h_i)*을 받은 후 CI_i, DLV_i, MH_PEND_i, $PEND_ACK_i$와 *handoff_over(h_i)*를 S_l로 차례대로 송신

V. 성능평가

5.1 알고리즘 복잡도 분석

기존의 인과순서 알고리즘에서 전송되는 제어정보의 크기에 대한 최악공간복잡도는 $O(n^2)$이다[3][6-8].

제안된 알고리즘의 공간복잡도를 계산하기 위해서는 CI, DC, DLV 세 개의 주요 자료구조를 고려하면 된다. 여기서 DC는 CI의 일부분이므로 고려에서 제외할 수 있고, DLV는 일차원 배열로 공간복잡도가 O(n)이므로 전체 공간복잡도는 CI에 의해 결정된다. CI 벡터의 개수는 n이며 각 항목에는 (n, t) 형태의 자료가 들어있고 그 수는 시스템 내의 프로세스 수 n으로 제한된다. 따라서 CI 벡터의 최악공간복잡도는 기존의 알고리즘과 같은 $O(n^2)$이 된다. 그러나 평균적으로는 다음의 모의실험 결과가 보여주듯이 중복정보의 조기 제거로 기존의 알고리즘에 비해 훨씬 우수함을 알 수 있다.

5.2 시뮬레이션을 통한 평가

시뮬레이션은 Sun Enterprise 450에서 실시되었으며, 시스템의 구성은 이동유닛과 기지국으로 구성되어 있다고 가정한다. 본 시뮬레이션에서는 일반적인 셀의 구성을 감안하여 기지국의 수를 7개로 한정하였다. 유선통신채널의 대역폭은 100Mbps 전파지연시간은 7ms이고, 무선통신채널의 대역폭은 1Mbps 전파지연시간은 500us로 가정하였다. 메시지전송이벤트 사이의 시간 간격(tm)과 채널전환 사이의 시간간격은 모두 지수분포를 따른다고 가정하였다.

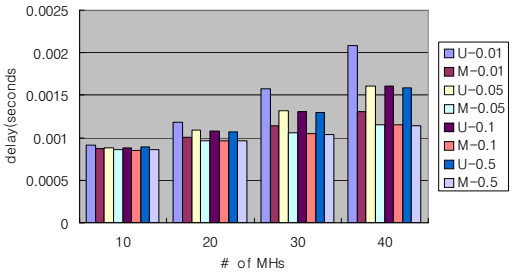

그림 5. 평균 메시지 전송 시간

본 논문에서 제안한 알고리즘은 멀티캐스팅을 기반

으로 하고 있으며 [그림 5]는 제안 알고리즘에서 t_m 값에 따른 유니캐스팅과 멀티캐스팅의 평균 메시지 전송시간을 보여주고 있다. 그림에서 범례는 U-0.1은 t_m 값이 0.1인 유니캐스팅을, M-0.5는 t_m 값이 0.5인 멀티캐스팅을 나타내는 식으로 해석한다. t_m값이 작아짐에 따라 메시지 발생확률이 커지므로 기지국의 메시지 처리에 대한 부하가 그만큼 증가하며 실제 메시지의 전송지연시간이 인과순서에 미치는 영향이 커지게 된다. 따라서 전달 조건을 만족시키지 못하고 MH_PEND에 들어가는 메시지가 많아지므로 통신 지연시간이 커지게 된다.

이동유닛의 수가 늘어남에 따라 전송지연시간도 늘어나고 있지만 모든 경우에 있어서 멀티캐스팅은 유니캐스팅에 비해 좋은 성능을 보이고 있음을 알 수 있다.

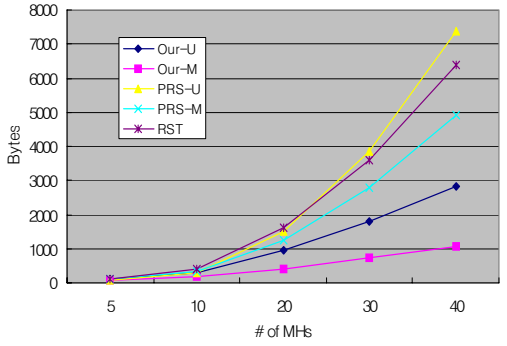

그림 6. 평균 오버헤드의 크기

[그림 6]은 t_m값이 0.01초인 환경에서 제안알고리즘과 PRS[7], RST[8] 알고리즘과의 오버헤드 크기에 대한 상대비교 결과를 보여주고 있으며 [그림 7]은 동일한 조건에서 전송되는 제어정보의 갯수에 대한 비교결과를 보여주고 있다. RST 알고리즘은 멀티캐스팅을 지원하지 않으므로 유니캐스팅에 대한 결과만 보인다.

모든 경우에 있어서 제안 알고리즘은 PRS나 RST에 비해 월등한 성능을 보이고 있음을 알 수 있다. 특히 이동유닛 수의 증가에 따른 제어정보의 크기가 거의 선형적으로 증가하는 것을 보이고 있어서 기존 알고리즘에 비해 확장성이 뛰어남을 알 수 있다.

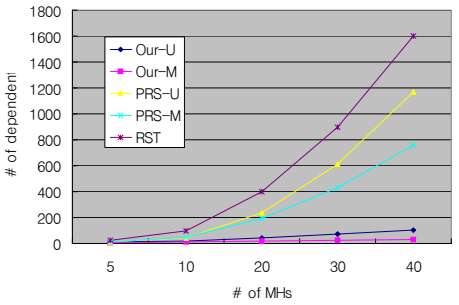

그림 7. 제어정보의 평균 갯수

제어정보의 개수와 오버헤드의 크기는 직접적인 연관관계를 가지고 있고, 일반적으로 멀테캐스팅의 경우 하나의 제어정보의 크기가 유니캐스트보다 크므로 제어정보의 개수에 비해 오버헤드의 크기에 있어서는 그 차이가 더욱 벌어짐을 알 수 있다.

VI. 결론

본 논문에서는 유효한 모든 전송패턴을 분석하여 인과순서의 유지에 필수적이지 않은 중복정보의 유형을 정의하고, 중복정보의 조기 제거를 통해 전송되는 제어정보의 양을 최소화하는 Jang[5]의 알고리즘을 이동통신망에서 동작할 수 있도록 멀티캐스트 메시지전달 알고리즘을 제안하였다. 이를 위해 채널전환 알고리즘도 제안하였다.

제안한 알고리즘의 최악공간복잡도는 기존의 알고리즘과 같은 $O(n^2)$이다. 그러나 평균적으로는 기존의 대표적인 알고리즘인 Raynal과 Prakash의 알고리즘에 비해 우수함을 시뮬레이션을 통해 보였다.

참 고 문 헌

[1] S. Alagar and S. Venktesan, "Causal Ordering in Distributed Mobile Systems," IEEE Trans. on Computers, Vol.46, No.3, pp.353-361, 1997.

[2] K. Birman, A. Schiper, and P. Stephenson,

"Lightweight Causal Atomic Group Multicast," ACM Trans. on Computer Systems, Vol.9, No.3, pp.272-314, Aug. 1991.

[3] W. Cai, B. Lee, and J. Zhou, "Causal Order Delivery in a Multicast Environment: An Improved Algorithm," Journal of Parallel and Distributed Computing, Vol.62, pp.111-131, 2002.

[4] I. Jang, "A Efficient Causal Order Algorithm for Mobile Networks," Proc. of ICCC2008, Hangzhou, China, pp.584-593, 2008(12).

[5] I. Jang, J. Cho, and H. Yoon, "An Efficient Causal Multicast Algorithm for Distributed System," IEICE Trans. on Information and Systems, Vol.E81-D, No.1, pp.27-36, 1998.

[6] A. Kshemkalyani and M. Singhal, "An Optimal Algorithm for Generalized Causal Message Ordering," in Proc. 15th Symposium on PODC, pp.87-94, 1996(5).

[7] R. Prakash, M. Raynal, and M. Singhal, "An Efficient Causal Ordering Algorithm for Mobile Computing Environments," in Proc. 16th ICDCS, pp.744-751, 1996(5).

[8] M. Raynal, A. Schiper, and S. Toueg, "The Causal Ordering Abstraction and a Simple Way to Implement It," Information Processing Letter, Vol.39, pp.343-350, 1991(9).

[9] A. Tanenbaum and M. Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed., Prentice Hall, 2007.

저 자 소 개

장 익 현(Ik-Hyeon Jang) 정회원

- 1984년 : 서울대학교 계산통계학과(이학사)
- 1986년 : KAIST 전산학과(공학석사)
- 1998년 : KAIST 전산학과(공학박사)
- 1986년 ~ 1999년 : (주)데이콤 책임연구원
- 2006년 ~ 2007년 : Indiana University 방문교수
- 1999년 ~ 현재 : 동국대학교 정보통신공학과 교수

<관심분야> : 분산시스템, 컴퓨터통신, 인터넷 프로토콜, 인터넷 응용