링 기반 그룹 통신 프로토콜을 위한 효율적인 토큰 관리

○
이 창 우, 홍 영 식
동국대학교 컴퓨터공학과

An Efficient Token Management for Ring based Group Communication Systems

Chang Woo Lee, Young Sik Hong
Dept. of Computer Engineering, Dongguk Univ.

요 약

그룹 통신 분야에서는 효율적이고 일관된 그룹의 상태를 관리 할 수 있는 그룹 구성원간의 통신 방법에 대한 많은 연구가 진행되고 있다. 그 대표적인 방법으로 중앙 집중식 순차기를 사용하거나, Totem과 RMP와 같이 링 구조를 기반으로 하는 그룹 통신 시스템들이 있다. 그러나 링을 기반한 그룹 통신 시스템에서는 멤버의 수가 증가할 경우 링을 회전하는데 걸리는 시간이 길어지고 토큰 권한을 가지기 위해 주고받는 메시지 수의 증가로 인한 성능 저하가 발생한다는 단점이 존재한다. 본 논문에서는 이러한 단점을 보완할 수 있도록, 멀티캐스트를 자주 하는 멤버들만으로 구성된 미니그룹을 그룹 안에 따로 구성하여 노드 수가 증가되더라도 링 회전시간으로 인한 성능 저하가 매우 적은 링 기반 그룹 통신시스템 알고리즘인 미니그룹을 제안한다.

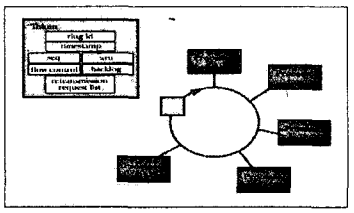
1. 서론

그룹 통신은 분산 시스템에서 하위 계층의 통신을 전담하는 부분이므로 그룹 통신 시스템의 성능이 전체 분산 시스템의 성능에 큰 영향을 끼친다. 이러한 그룹 통신을 구현하는데 있어서 효율적으로 일관된 그룹의 상태를 관리 할 수 있는 그룹 구성원간의 통신 방법에 대해서 많은 연구가 있었다.[1,2,3,4] 예를 들면 중앙 집중식으로 순차기를 사용하거나 링 구조를 이용한 방법에 대한 연구들이 있었는데 [7], 최근에는 Totem[3,4], RMP[1,2]와 같이 링 구조를 기반으로 하는 그룹 통신 시스템에 대한 여러 가지 연구가 활발히 진행 중이다. 링 기반 그룹 통신 시스템에서는 토큰(Token)을 사용하여 멀티캐스트 권한을 주고 받는다. 토큰은 멀티캐스트 되는 메시지들의 순서를 일정하게 유지시키는 역할을 한다. 토큰을 가진 노드는 두 가지 역할을 수행하게 되는데, 첫째, 토큰 권한을 가진 노드는 그룹 멤버의 추가나 삭제와 같은 그룹 멤버 관리를 책임진다. 두 번째 역할은 메시지를 멀티캐스트 하는 전송자의 역할이다. 즉, 노드가 토큰 권한을 소유하였을 때, 그룹의 다른 멤버들에게 멀티캐스트 메시지를 전송 할 수 있는 권한을 가지게 된다[6].

본 논문에서는 그룹 안에서 멀티캐스트를 자주 하는 멤버들만으로 구성된 미니그룹을 그룹 안에 따로 구성한 링 기반 그룹 통신시스템 알고리즘인 미니그룹을 제안한다. 2장에서는 단일 링 기반 그룹 통신 시스템에서 대표적인 시스템인 Totem과 RMP에 대하여 설명하고 3장에서는 본 논문의 제안 모델인 미니 그룹을 사용한 그룹 통신 알고리즘을 기술한다. 4장에서는 본 논문의 비교 모델인 RMP와 미니 그룹 알고리즘을 비교한 실험과 결과를 기술하고 분석한다. 5장에서는 결론을 맺는다.

2. 단일 링 기반 그룹 통신 알고리즘

Totem은 단일 링과 다중 링 그룹 통신 알고리즘을 지원하여 신뢰성 있는 전체 순서화된 방송 메시지를 전달하는 기능을 제공한다. 하나의 근거리 통신망, 즉 방송 도메인 안에서 단일 링 프로토콜은 하나의 그룹을 관리하고, 단일 그룹의 통신을 제어하며, 다수 링 프로토콜은 방송 도메인 사이의 통신을 제어하게 된다[4]. 본 논문에서는 하나의 방송 도메인 안에서의 Totem 단일 링 프로토콜만을 고려하고 있다.



[그림 2]. Totem의 논리적 단일 링과 토큰 구조 [3]

Totem 단일 링 프로토콜(SRP)은 논리적인 링 상에서 토큰 전달을 이용하여 그룹을 유지하고 방송 메시지를 전달한다. [그림1]과 같이 Totem에서 토큰은 일대일 통신 메시지로 전달되며, 메시지의 신뢰성 있는 전달과 전체 순서화에 대한 정보를 가지고 있다[4]. 그룹 안에서 멤버가 메시지를 방송하기 위해서는 토큰 메시지를 받아야 하며, 모든 노드는 링 안에서 순서대로 토큰 메시지를 수신 할 수 있다. [3].

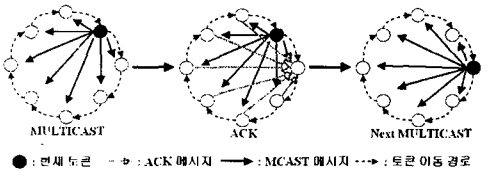
결국, Totem에서는 한번의 토큰 전송 메시지의 전달 후에 토큰 재전송 타임아웃 시간 안에 다음 순서의 방송 메시지가 발생하거나 자신에게 토큰 메시지가 전달되면 토큰이 올바로 전달된 것으로 가정한다. 만약, 그룹의 멤버 수가 N이고 링을 한바퀴 돌면서 방송된 메시지의 수가 M, 토큰 메시지 전송 시간은 T_{TM} 한번 방송하는데 걸리는 시간을 T_{BROD}이라 하면, 하나의 노드가, 토큰 메시지를 전달하고 다음 토큰 메시지를 받기까지 걸리는 시간은 [수식1]과 같이 표현된다.

$$Totem-R = N \times T_{TM} + M \times T_{BROD} \quad \text{.....[수식1]}$$

Totem은 토큰 메시지를 받은 노드만이 그룹에 방송 할 수 있도록 하고 있으며, 토큰 메시지를 통해 전체 그룹정보

를 관리하고 메시지 순서를 일정하게 유지하고 있다. 그러나 Totem과 같은 방식의 토큰 이동은 [수식1]에서 보이는 것 과 같이 $M \times T_{BROD}$외에 $N \times T_{TM}$ 시간의 토큰 메시지 교환의 부하가 생기며 메시지를 방송하기 위해 기다리는 시간이 최악의 경우 링을 한번 회전하는 시간이 걸린다는 단점을 가지고 있다.

RMP는 Totem 그룹 통신 시스템과 비교할 때 토큰 전송 메시지가 없이, 토큰 권한을 이동시킬 수 있다는 장점을 가지고 있다.[5]



[그림 3]. RMP 프로토콜의 메시지 전송

[그림2]는 RMP에서 토큰 권한을 가진 노드가 그룹 멤버들에게 멀티캐스트를 하고 멀티캐스트 메시지를 받은 노드들이 다음 토큰을 소유하게 될 노드에게 ACK 메시지를 보내는 상황을 보이고 있다. [그림2] 에서 보는 바와 같이 RMP는 토큰 이동에 따른 부가적인 메시지가 전혀 필요가 없다. 모든 노드는 그룹 안에서 토큰 이동 순서를 알고 있고 그에 따라 자신이 받은 멀티캐스트 메시지에 대한 ACK 메시지를 어떤 노드에게 전송해야 되는지 알고 있다. 다음번 토큰이 될 노드는 전송 받은 멀티캐스트 메시지로부터 다음 토큰 권한이 자신에게 있음을 알게 되고 ACK 메시지를 기다리게 된다. 이러한 순서로 이동하게 되므로 노드 고장이 없고 메시지가 손실되지 않는 다고 가정하면 일정시간 간격으로 부가적인 토큰 전송 메시지의 필요 없이 모든 멤버는 토큰을 소유할 수 있고 멀티캐스트를 할 수가 있게 된다.[5]

그룹의 멤버 수가 N이고 ACK가 전달되는데 걸리는 시간은 $T_{T\text{-}ACK}$, 한번 멀티캐스트 하는데 걸리는 시간을 T_{MUL}이라 하면, RMP에서 메시지를 한번 멀티캐스트하고 다음에 멀티캐스트를 할 수 있을 때까지 걸리는 시간은 [수식2]와 같이 표현된다.

$$RMP-R = N \times (T_{T-ACK} + T_{MUL}) \ldots\ldots\ldots [수식2]$$

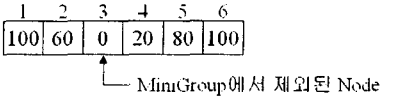
그러나, RMP역시 임의의 노드가 한번 멀티캐스트 메시지를 보낸 다음, 다음 번 멀티캐스트를 하기 위해서는 토큰이 링을 한번 돌 때까지 기다려야 한다는 단점이 존재한다. 특히 RMP는 Totem과는 달리 토큰권한을 가지더라도 단 한번의 멀티캐스트만을 할 수 가 있다. 따라서, 멤버의 수가 많거나 멤버중 임의의 노드가 집중적으로 멀티캐스트를 하게되면 멀티캐스트를 하지 않는 멤버들도 토큰 권한을 가지게 될 때마다 한번씩 널(Null) 메시지를 멀티캐스트 하게되고 이로 인해 전체적인 성능의 저하가 일어나게 된다.

3. 미니그룹 알고리즘

기존 연구인 Totem은 토큰 이동 메시지로 인해 토큰의 이동이 성능 저하의 원인이 되고, RMP는 멀티캐스트 하지 않는 모든 노드가 널(Null)멀티캐스트를 해야 링을 회전할 수 있는 단점이 존재한다. 멤버의 수가 많아지면 Totem과 RMP의 단점은 급격한 성능저하의 원인이 된다. 본 논문에서는 이러한 문제점을 해결하기 위해 그룹 안에 토큰 권한이 이동하는 작은 그룹을 유지해서 토큰 회전 시간을 최소화시킨 미니 그룹 알고리즘을 제안한다.

미니 그룹은 RMP와 유사한 방법으로 링을 구성하고 토큰 정보가 이동하게 된다. 그러나 미니 그룹은 축적된 정보를 바탕으로 멀티캐스트 메시지를 자주 전송하는 멤버들끼리만 논리적인 미니 링을 구성하여 토큰 권한을 가질 수 있게 한다. 따라서 메시지를 멀티캐스트 하려는 노드는 짧은 지연 시간을 갖게 되고, 토큰이 링을 회전하는 시간을 단축시킬 수 있다.

각 멤버는 전체 멤버가 얼마나 멀티캐스트 했는지를 우선순위 배열에 기억시키게 된다.



[그림 4]. 우선 순위 배열의 구조

[그림3]에서 배열의 인덱스는 링을 구성하는 노드들의 순서 정보이며, 모든 노드의 값은 100을 초기 값으로 갖는다. 이 배열의 최대 값은 100이며 각 멤버가 멀티캐스트를 하면 20씩 증가시켜 주고 널(Null) 멀티캐스트를 전송하면 20씩 감소해서 기록한다. 배열 안의 값이 0인 멤버는 토큰이 될 기회가 주어지지 않게 되고 미니 그룹 안에 포함되고 싶을 경우에는 토큰 요청 메시지를 미니그룹 안의 노드에게 보내야 한다.

아래의 [그림4] 와 [그림5]는 미니 그룹의 주요 알고리즘을 보이고 있다.

```
Switch (MsgType) {
MsgType== TokenRequest
        // 자신의 우선순위배열에 저장한다.
        IncPriorityArrayValue(ReqNode);
        // 토큰을 요청한 노드정보를 저장
        SaveTReqList(ReqNode);
MsgType== MCAST
        If ExistTReqList then
         // MCAST 메시지에 토큰요청 정보가
         // 있으면 우선순위 배열 안의 값을
         // 증가시켜준다.
         IncPriorityArrayValue(ReqNode);
        endif
        If IsNextToken(MyAddr) then
          //자신이 다음 토큰이면 ACK를 기다린다.
          WaitACK();
        else
          // 다음에 토큰의 될 노드를 우선순위
          // 배열의 값을 이용해 찾는다.
          TNode = FindNextMiniGroupMember();
          SendAck(TNode);
        endif
}
```

[그림 4]. 미니그룹의 **RecvMsg** Function

```
If ExistTReqList() then
    // 저장된 토큰 요청 정보가 있으면
    // MCAST와 같이 보낸다.
    SendMCAST with TReqList
else
    SendMCAST
```

[그림 5]. 미니그룹의 **SendMCAST** Function

이와 같이 우선 순위 배열을 이용한 미니 그룹 알고리즘

은 우선 순위 배열 안에서, 0 이 아닌 값을 가지는 멤버들만으로 이루어진 미니그룹 안에서만 토큰이 이동되게 된다. 그것을 그림으로 표시하면 [그림6] 같은 방식으로 알고리즘이 진행되게 된다. 아래의 그림에서 검은 노드들은 미니 그룹에 속해 있는 토큰을 소유할 수 있는 노드들을 의미한다. 만약 널(Null) 멀티캐스트를 하는 노드들이 증가되고 집중적으로 메시지를 보내는 노드들이 존재한다면 토큰은 [그림6]의 오른쪽 그림과 같이 토큰 권한이 이동 되게 되고 링 회전 시간은 대폭 줄어들게 된다.

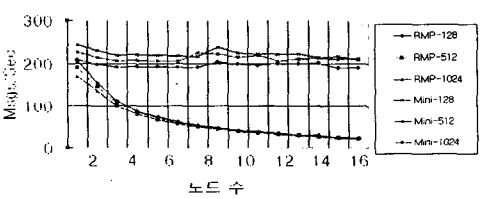
[그림 6]. 미니 그룹 알고리즘

만약, 우선순위 배열의 값이 모두 0이 되고 더 이상 미니그룹을 유지 할 수 없게 되면 토큰 이동은 멈추게 된다. 그러나 모든 멤버는 마지막으로 토큰 권한을 가지고 있던 노드를 알고 있고, 토큰 요청 메시지를 마지막으로 토큰 권한을 가지고 있던 노드에게 보내면, 토큰 권한은 다른 멤버로 이동된다.

4. 실험 결과 및 분석

미니그룹과 RMP를 시뮬레이션으로 성능을 비교하기 위하여 Linux RedHat 5.2 운영체제에서 ns-2.1b4를 사용하여 실험하였다. 그룹 안에서 노드의 고장에 따른 탈퇴나 멤버의 동적인 참여는 없고 메시지 손실은 존재한다고 가정한다. 메시지의 손실과 토큰 요청은 평균 0.5초의 지수분포를 나타내는 임의의 시간으로 발생되도록 실험하였다. 전체적인 통신시간을 초당 보낼 수 있는 메시지의 수로 환산하여 나타내었다.

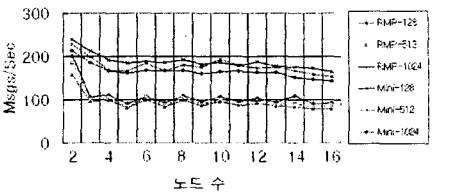
첫 번째 실험은 RMP논문에서 쓰인 가정[1]을 이용하여 N개의 멤버중 2개의 멤버가 계속해서 각각 100개의 멀티캐스트 메시지를 전송하는 시간을 비교하였다. 이 실험은 노드수의 증가에 따른 RMP와 미니그룹의 성능차를 보이는 실험이다.



[그림 7]. RMP & 미니그룹의 처리율 1

[그림7]에서 RMP는 링 회전 시간으로 인하여 2개의 메시지를 멀티캐스트 하기 위하여 N-2개의 '널(NULL) 멀티캐스트 메시지를 전송하게 되므로 노드수가 많아질수록 반비례로 성능이 저하된다. 미니그룹은 N개의 멤버가 있더라도 미니그룹에 포함된 멤버만이 멀티캐스트를 하게 됨으로 노드수가 증가하더라도 2개의 멤버만이 멀티캐스트 메시지를 전송하게 되고 처리율(Throughput)의 감소가 거의 없었다. 결국 그룹 멤버의 수가 많아질수록 미니그룹의 성능이 좋아짐을 알 수 있다.

두 번째 실험은 각각 연속적으로 50개의 메시지를 멀티캐스트 하는 노드들의 수가 평균적으로 전체 멤버 수의 절반을 유지하도록 하고 미니그룹과 RMP의 성능을 비교하였다.



[그림8] RMP & 미니그룹의 처리율 2

두 번째 실험에서 토큰이 링을 한번 회전하는데 소요되는 시간은 미니그룹 알고리즘이 RMP보다 평균 N/2개의 널 멀티캐스트 메시지 비용만큼 적다. 따라서 미니그룹 알고리즘의 성능이 좋음을 보이고 있다. [그림8]에서 미니그룹 알고리즘의 그래프는 노드 수의 증가에 따른 성능 저하가 [그림7]보다 더 크다. 이것은 미니그룹에 속한 노드수의 증가로 전체 링을 회전하는 시간이 길어짐에 따라 나타나는 현상이다. 만약, 미니그룹의 수가 전체 멤버 수와 같아진다면 미니그룹 알고리즘의 성능과 RMP의 성능은 비슷함을 보일 것이다.

5. 결론 및 향후 과제

본 논문에서는 단일 링 기반의 그룹 통신 시스템에서 토큰을 주고받는 메시지를 사용하지 않고 토큰을 전송하며, 자주 멀티캐스트 하는 멤버들로만 미니 그룹을 구성하여 토큰을 회전시켰다. 따라서, 그룹의 멤버 수 증가가 그룹의 전체 처리율에 거의 영향을 끼치지 않는 프로토콜을 제안하고, 실험을 통하여 본 논문의 제안 모델인 미니그룹이 노드수의 증가에 대해서, 시스템의 처리율이 거의 영향을 받지 않는 다는 사실을 보였다.

향후연구 과제는 미니그룹 알고리즘을 노드 고장이나 메시지 손실과 같은 고장을 감내할 수 있는 프로토콜로 발전시키는 것이다.

참고문헌

[1] Weijia Jia, Jiannon Cao, Edgar Nett and Jorg Kaiser, A High Performance Reliable Atomic Group Protocol, Proceeding of the 1996 International Conference on Parallel & Distributed System June 3-6, 1996 Tokyo, Japan

[2] Weijia Jia, Implementation of a Reliable Multicast Protocol, Software-Practice And Experience, Vol. 27(7), 813-850(July 1997)

[3] Y.Amir, L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, and P. Ciarfella, The Totem Single-Ring Ordering and Membership Protocol, ACM Transaction on Computer Systems, Vol 13, No. 4, November 1995, pp 311-342

[4] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, and C. A. Lingley-Papadopoulos, "Totem: A Fault-Tolerant Multicast Group Communication System", Communications of the ACM, April 1996.

[5] 권봉경, 정광수, "그룹의 중첩을 지원하는 순서화된 멀티캐스트 알고리즘", 한국정보과학회 봄 학술발표 논문집, Vol.23. No.1. pp.581-584. 1996

[6] 여인춘, 홍영식, "그룹 통신에서 계층적 중재자 모델을 위한 프로토콜 설계", 1997년 봄 동국대학교 석사 학위 논문.

[7] 한인, 홍영식, "방송권한을 이용한 전체 순서화 방송통신 프로토콜", 한국정보과학회 봄 학술발표 논문집, Vol.26. No.1. pp.152-154. 1999