방송권한을 이용한 고장감내 방송통신 프로토콜

O
한 인, 홍영식
동국대학교 컴퓨터공학과

A Fault Tolerant Broadcast Protocol Using Broadcast Token

In Han, Young-Sik Hong
Department of Computer Engineering, Dongguk Univ.

요 약

많은 분산 시스템에서 사용되는 방송 통신 프로토콜들은 노드의 고장에 대한 대비를 한다. 순차기 기반 방송 통신 프로토콜은 선출과정을 거치고, 논리적 링 기반 방송 통신 프로토콜은 고장난 노드를 제외한 노드들로 새로운 링을 구성하는 과정을 거치게 된다. 이러한 기존의 방법들은 정상 동작중인 노드들의 상태를 조사하기 위해 너무 많은 메시지의 전송이 이루어진다. 따라서 고장을 검출한 시점부터 새로운 방송 메시지를 전송할 수 있을 때까지 상당히 많은 시간이 요구된다. 본 연구에서는 방송권한을 이용하는 전체 순서화 방송 통신 프로토콜에서 노드의 고장이 발생할 경우 적은 수의 메시지 전송만으로 새로운 방송 메시지를 전송할 수 있는 프로토콜을 제안하였다. 본 연구의 프로토콜은 노드 수의 증가와 무관한 메시지 전송이 이루어져 대규모 분산 시스템에 적합하다.

1. 서 론

분산 시스템에서 방송통신 프로토콜의 필요성은 꾸준히 증대되어 왔고, 방송메시지의 순서를 일정하게 유지하고자 하는 연구들이 상당히 많이 이루어 졌다. 대표적인 방법으로 중앙노드를 사용하는 방법[2,5]과 논리적 링 기반 방송통신 프로토콜[1,3,4]이 있다. 그러나, 중앙노드를 사용하는 방송통신 프로토콜에서 중앙노드의 고장이나 논리적 링을 사용하는 프로토콜에서 링을 구성하는 노드의 고장이 발생하면, 방송메시지를 전송할 수 없거나 방송메시지의 정보를 논리적 링을 따라 회전시킬 수 없게 된다.

일반적으로 중앙노드를 사용하는 프로토콜은 새로운 중앙노드를 선출하는 과정[5]을 거치게 되고, 논리적 링 기반 프로토콜은 고장난 노드를 제외한 정상 동작 노드들로만 새로운 링을 재구성하는 단계[1,3,4]를 거치게 된다. 그러나, 선출과정과 새로운 링의 재구성 단계는 정상 동작 노드의 정보를 조사하기 위해 너무 많은 메시지 전송이 이루어진다. 또한, 전송되는 메시지 수는 방송도메인을 구성하는 노드 수가 증가함에 따라 비례적으로 증가하는 단점이 있다. 따라서, 고장을 검출한 시점부터 새로운 방송메시지 전송이 가능한 시점까지의 지연 시간이 크게 증가한다.

본 연구에서는 노드 수 증가와 무관하며, 적은 수의 메시지만으로 노드 고장을 처리할 수 있는 방법을 제안한다. 본 논문에서 적용할 전체 순서화 방송통신 프로토콜은 방송권한을 이용한 방법[6]이다.

본 논문의 구성은 2장에 논리적 링 기반 방송통신 프로토콜인 TOTEM의 고장 감내 특성을 살펴보고, 3장에서 본 연구에서 제안한 방법을 기술한다. 4장에서 두 모델의 비교 실험과 결과를 분석하고, 5장에서 결론을 맺는다.

2. 링 기반 방송통신 프로토콜의 고장감내

방송 도메인을 구성하는 노드들을 논리적 링 형태로 구성하고, 방송 메시지의 순서 번호를 링에 따라 회전시키는 링 기반 방송통신 프로토콜의 대표적인 시스템은 Totem[1,3,4]이다.

Totem 시스템은 다음 [그림 1]과 같은 4개의 상태를 갖는다.

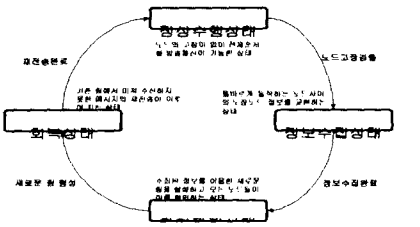

그림 1 Totem에서 노드고장 검출로 인한 각 노드의 상태전이도

어떠한 노드의 고장도 발생하지 않는다면, 모든 노드는 [그림 1]의 정상 수행 상태를 갖는다. 고장이 발생하면 정보 수집 상태와 확인 과정 상태, 그리고 회복 상태를 거치면서 새

로운 링을 재구성하게 된다.

Totem 프로토콜은 방송 도메인을 구성하는 노드들이 방송 메시지를 전송하지 않을 때, 지속적으로 토큰을 회전시키는 속성이 있다. 따라서, 노드 고장이 발생하면 더 이상 토큰 회전이 발생하지 않아 고장을 시간초과(Timeout) 알고리즘으로 쉽게 검출할 수 있다. 고장 발생을 검출하면 검출한 노드는 정보 수집 상태로 전환된다.

노드의 고장을 처음 검출한 노드가 정보 수집 단계로 전환된 즉시 모든 노드들에게 노드의 고장이 검출되었음을 알리는 방송 메시지를 전송한다. 정상 수행 상태의 모든 노드들은 고장 검출 통보 메시지를 수신하면 모두 정보 수집 상태로 전환되고, 자신이 유지하고 있는 고장 노드 정보 리스트를 방송메시지를 이용하여 교환한다. 다음 [그림2]는 정보수집단계의 동작을 나타낸다.

```
The Gather_State
Broadcast(F_node_list)
N_node_list ← T_node_list - F_node_list
while( !consensus() )
   Recv(Fail_info)
   if (F_node_list<>Fail_node)
      Union(F_node_list, Fail_node)
      N_node_list←T_node_list - F_node_list
      Broadcast(Fail_node_list)
   end if
end while
Goto Commit_State
```

그림 2 Totem의 정보 수집 단계 알고리즘

[그림 2]에서 각 노드는 자신이 유지하고 있는 고장 노드 정보와 다른 정보를 수신하게 되면 자신의 정보를 변경하고, 다른 모든 노드들에게 자신의 정보를 방송 메시지로 전송한다. 따라서 각 노드의 정보가 자주 변경되면 방송메시지의 수는 크게 증가한다.

모든 노드로부터 고장 노드 리스트를 수신하여 고장 노드의 정보를 일치시킨 후, 정상 노드들 중 제일 작은 노드 식별자를 갖는 노드는 정보수집으로 새롭게 생성된 링을 확인하는 메시지를 새로운 링을 따라 일대일 통신 메시지로 전송한다. 확인 과정 상태에서 확인 메시지는 새롭게 생성된 링을 2회전하게 된다.

확인 과정 단계의 두 번째 링 회전이 끝나면 모든 노드는 회복 단계로 전환된다. 회복 단계에서는 확인 과정의 두 번째 링 회전으로 수집된 재전송 요구를 수행하는 단계로 링의 모든 노드들이 노드 고장으로 인해 수신하지 못한 메시지의 재전송을 받게 된다. 모든 재전송이 완료되면 새롭게 생성된 링을 이용하여 새로운 방송 메시지를 전송할 수 있다.

식(1)은 Totem에서 노드의 고장을 검출하고 새로운 방송 메시지를 전송할 수 있는 상태까지 전송되는 메시지 수를 계산하는 식이다. 식(1)에서 사용되는 M_{Broad}는 방송 메시지 수이고, M_{Ptp}는 일대일 통신 메시지 수를 나타낸다. N은 정상 동작하는 노드 수이고 회복 단계에서 어떠한 재전송 메시지도 발생하지 않는다고 가정한다.

$$N_{MSG} = (N+\alpha)\cdot M_{Broad} + 2\cdot N\cdot M_{Ptp} \cdots\cdots \text{식(1)}$$

α는 임의의 노드에서 정보의 변경으로 발생하는 추가적인 방송 메시지 수를 나타낸다. 식(1)을 통해 Totem은 노드 수 증가에 따라 새로운 링을 구성하기 위해 더욱 많은 메시지 전송이 발생함을 알 수 있다.

3. 제안된 방송통신 프로토콜의 고장감내

본 논문에서 제안하는 방송통신 프로토콜은 노드 고장에 대비하여 각각의 노드들이 과거 방송 권한인 토큰을 소유했던 노드의 정보를 유지하게 된다. 따라서 메시지를 방송하고자 하는 노드는 가장 최근에 방송권한을 소유했던 노드인 방송 노드 정보의 첫 번째 노드에게 방송 권한을 요청하게 된다. 그러나 자신의 방송 노드 정보의 첫 번째 노드로부터 일정시간동안 어떠한 응답도 받지 못한다면, 그 노드의 고장을 검출할 수 있고, 자신의 방송 노드 정보의 다음 노드에게 방송 권한 요청 메시지를 전송하게 된다.

제안된 프로토콜에서 유지하는 방송 권한 정보는 방송도메인을 구성하는 노드 수만큼의 메모리 공간을 필요로 한다.

방송 노드가 방송 권한 요청 메시지를 수신하면 자신의 방송 메시지를 모두 전송한 후 요청한 노드에게 토큰을 전송한다. 다음 [그림 4]는 방송 노드가 방송 권한을 요청 받았을 때의 동작 과정을 간략히 표현한 것이다.

```
The_Token_Request_Processing
if ( recv(request_token) = TRUE )
   if ( Bnode = TRUE )
      while(Bcast_Msg = TRUE)
         broadcast( Msg )
         TOKEN.SeqNo ← TOKEN.SeqNo + 1
      end while
      send( TOKEN, ReqNode )
   else
      NewToken.VerNo ← OldToken.VerNo + 1
      NewToken.SeqNo ← 0
      send( NewToken, ReqNode )
   end if
end if
```

그림 3. 방송권한(Token) 요청을 받았을 경우의 동작 과정

[그림 4]에서 방송권한을 요청 받은 노드는 가장 먼저 자신이 방송권한을 갖고 있는지를 조사한다. 만약, 방송권한을 소유하고 있다면, 자신이 방송하려 하는 모든 메시지를 방송한 후, 방송권한을 요청한 노드로 전송한다. 그러나, 방송권한이 없는 상태에서 방송권한 요청메시지를 수신하면 방송권한을 소유하고 있는 노드의 고장이 검출된 경우이고, 자신이 가장 최근에 방송한 노드이므로 과거 토큰의 버전번호를 하나 증가시키고, 순차번호 정보를 0으로 설정한 후, 방송권한을 요청한 노드로 방송권한을 전송한다.

버전 번호가 증가된 토큰을 수신한 노드는 방송 메시지를 방송할 권한을 갖고 메시지를 방송한다. 버전번호가 증가된 방송 메시지를 수신한 모든 노드는 기존 버전의 메시지보다 늦은 메시지로 취급하고, 추후 메시지를 방송하기 위하여 방송권한을 새로운 버전의 방송권한을 갖고 있는 노드로 요청하게 된다. 따라서, 기존의 방송권한을 소유한 노드가 네트워크나 노드의 작업량에 의해 고장으로 검출되는 현상인 거짓 고장 검출 현상이 발생하더라도 모든 메시지의 순서를 일정하게 유지할 수 있다.

Btoken에서 고장을 검출하고 새로운 방송통신이 가능하기까지 소요되는 메시지의 수는 M_{Ptp}를 일대일 통신 메시지라

할 때, 식(2)와 같다.

$$N_{MSG} = 2 \cdot M_{PTP} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{식(2)}$$

식(2)를 통해 고장이 검출된 후 새로운 버전의 방송권한을 얻기까지 전송되는 메시지 수는 방송노드의 고장이 없을 경우 전송되는 메시지 수와 같음을 알 수 있다.

4. 실험 및 결과 분석

고장 감내 속성을 실험하기 위하여 논리적 링 기반 프로토콜의 모델은 Totem 시스템으로 하였다. 실험은 네트워크 시뮬레이터인 Ns-2.1.b4로 시뮬레이션하였다. 사용된 네트워크 모델은 10Mb의 이더넷 환경으로 하였으며, 전송되는 모든 패킷의 크기는 1000바이트로 하였다. 모든 시스템은 고장 검출을 시간초과(TimeOut) 알고리즘을 사용하였으며, 고장 발생 시간은 임의의 시간에 발생하도록 하였다. 본 실험에서 고장은 붕괴(Crash) 고장만을 고려하였고, 네트워크의 분할은 없다고 가정한다

이미 선행된 연구[6]에서 Totem의 경우 방송도메인을 구성하는 노드의 수가 증가하면 논리적 링의 크기가 커지기 때문에 방송메시지 수가 줄어들고 방송시간이 크게 증가함을 보였다. 본 연구의 실험에서 고장이 발생하더라도 정상적인 전체 순서화 방송메시지의 전송이 가능함을 확인하고 고장의 피해를 조사한다.

실험은 고장이 발생되지 않은 상황과 단일노드 고장이 발생하였을 경우에 대해 1000개의 순서화 방송 메시지를 방송하기 위해 요구되는 시간(초)을 측정하였다.

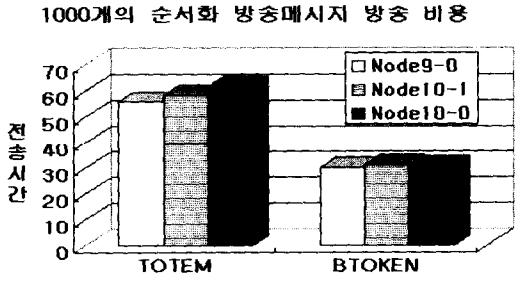

그림 4 단일노드 고장 상황에서 1000개의 순서화된 방송 메시지를 전송하기위해 소요된 시간(초)

[그림 4]에서 Node9-0과 Node10-0은 방송도메인의 크기가 각각 9와 10이고 고장이 발생하지 않을 경우의 전송시간을 나타내고, Node10-1은 방송도메인 크기 10에서 단일노드 고장이 발생할 경우를 나타낸다.

실험 결과를 통해 Totem의 경우, 고장이 발생하면 논리적 링의 크기가 작아지기 때문에 토큰의 링 회전 시간이 짧아져 방송 메시지의 수가 증가한다. 하지만 방송도메인을 구성하는 노드의 수에 무관한 BTOKEN에서는 단일노드의 고장으로 약간의 지연시간이 있음을 알 수 있다.

위 실험으로 단일 노드 고장으로 인한 피해를 명확히 확인할 수 없으므로 고장 검출시점부터 새로운 방송 메시지 전송이 가능한 시점까지 각 시스템에서 요구되는 메시지 수를 조사해 보면 [그림 5]와 같다.

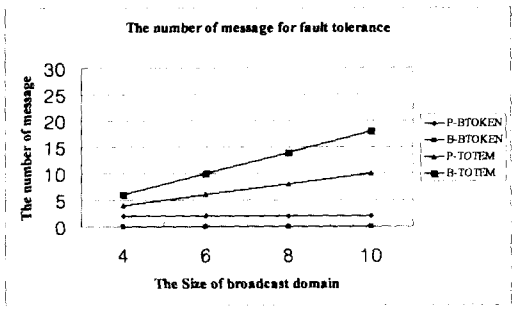

그림 5 고장감내를 위해 요구되는 메시지 수

[그림 5]를 통해 TOTEM에서 새로운 링을 구성하기 위해 상당히 많은 메시지를 필요로 하게 되고, 이 메시지 수는 방송 도메인을 구성하는 노드 수 증가에 따라 크게 증가함을 알 수 있다.

5. 결론 및 향후 연구과제

방송통신 프로토콜에서 고장감내 속성은 매우 중요한 속성이다. 고장감내 방송통신 프로토콜에서 사용하는 대표적인 방법은 선출 알고리즘이나, 논리적 링의 재구성 알고리즘을 사용한다. 그러나, 기존 방법은 고장 검출시점부터 새로운 방송 메시지 전송이 가능한 시점까지 많은 수의 메시지 전송이 요구되고, 그로 인한 소요시간이 증가하게 된다.

본 연구에서는 노드의 수 증가에 영향을 받지 않으면서, 적은 수의 메시지 전송만으로 고장감내 속성을 구현할 수 있는 방법을 제안하고 실험하였다. 실험 결과를 통해 링 기반 방송통신 프로토콜보다 뛰어난 성능향상이 있음을 확인했다.

본 연구는 일반적인 전체 순서화 방송통신이 진행되는 과정에서 노드의 그룹 참여와 고장으로 인한 탈퇴를 통한 성능변화의 측정이 진행되어야 할 것이다.

[참고문헌]

[1] Y. Amir, L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, and P. Ciarfella, "The Totem Single-Ring Ordering and Membership Protocol", ACM Transactions on Computer Systems 13, 4 (November 1995), 311-342.

[2] Jo-Mei Chang and N.F. Maxemchuk, "Reliable Broadcast Protocols", ACM Transaction on Computer Systems, Vol.2, No.3, pp.251~273, 1984.

[3] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, C. A. Lingley-Papadopoulos, and T. P. Archambault, "The Totem System", Proceedings of the 25th International Symposium on Fault Tolerant Computing, Pasedena, CA (June 1995), 61-66.

[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] M. Frans Kaashoek and Andrew S. Tannenbaum, "Group Communication in the AMOEBA Distributed Operating System", The 11th International Conf. on Distributed computing System, pp.222~230, 1991.

[6] 한인, 홍영식, "방송권한을 이용한 전체 순서화 방송통신 프로토콜", 한국정보과학회 봄 학술 발표논문집, vol.26, no.1, pp.152-154, 1999.