

## 2계층 구조의 신뢰적 멀티캐스트에서 제어 트리의 효율성을 고려한 그룹 대표 결정 기법\*

이동연<sup>o</sup>, 이승익, 고양우, 이동만

한국정보통신대학교 공학부

{dulee, silee, newcat, dlee}@icu.ac.kr

Group core election for efficiency of control tree in 2-layer reliable multicast

Dongun Yi<sup>o</sup>, Seungik Lee, Yangwoo Ko, and Dongman Lee

School of Engineering, Information and Communications University

### 요 약

인터넷의 발전에 따라 다자간 그룹 통신 환경이 주목을 받게 되었고 이를 위해 다대다 멀티캐스트를 이용한 통신 기법들이 제안되었다. 이들 중 GAM (Group Aided Multicast)은 다대다 멀티캐스트의 신뢰적 전달을 보장하기 위해 트리 기반의 손실 복구 기법을 제안하고, 제어 트리 관리 비용과 손실 복구 효율성간의 절점을 찾기 위해 그룹 개념을 도입하고 그룹의 대표로서 코어 노드를 정의한다. 코어 노드는 네트워크 상의 특정 지점에 미리 설치된 전용 노드로서 그룹 내의 손실 복구와 그룹 간의 손실 제어를 담당한다. 그러나 이러한 전용 코어 노드의 도입은 프로토콜의 가용성을 제한하고 코어 노드의 설치 부담을 가지게 된다. 따라서 본 논문에서는 별도의 코어 노드의 설치 없이 프로토콜이 동작할 수 있도록 세션 참가자 중에서 코어 노드를 동적으로 선택하고, 로컬 그룹 노드 중에서 제어 트리의 효율성을 최대한 보장하는 노드의 위치를 결정하는 기법을 제안한다.

### 1. 서론\*

인터넷이 발전하면서 많은 사람들이 멀리 떨어진 지역에 있는 다른 사람이나 정보에 쉽게 접근할 수 있게 되고 이를 통해 인터넷상에서 공간적 제약 없이 협동 작업을 쉽게 할 수 있게 되었다. 하지만 참가자의 수가 많은 협동 작업에서 복수의 유니캐스트 통신을 사용할 경우 많은 네트워크 자원의 낭비와 비효율성 문제를 일으키게 된다. 이를 위해 멀티캐스트 통신 기법이 제안되었으며 특히 협동 작업을 위한 응용 통신 방식이 기존의 일대다 (one-to-many) 통신 방식에서 다대다 (many-to-many) 통신 방식으로 확장되면서 응용의 신뢰성을 보장하는 다대다 멀티캐스트 (Many-to-many Multicast) 통신 기법들 [1,2]이 제안되었다.

GAM [1]은 트리 기반의 다대다 신뢰적 멀티캐스트 프로토콜로서 확장성과 효율성면에서 가장 성능이 좋은 기법으로 알려져 있다 [7]. GAM은 트리 기반의 다대다 멀티캐스트 환경에서 제어 트리의 효율성을 확보하면서도 트리 관리를 위한 비용을 적게 유지하기 위해 2-계층 (layer) 트리 관리 기법을 제안하였다. 하위계층은 지역적으로 근접한 멤버들로 로컬그룹을 구성하고 그 그룹은 코어 (Core)라고 불리는 노드를 루트로 하는 공유 트리를 구성한다. 상위계층은 각 그룹의 코어 노드들로 구성된 송신자별 트리를 구성한다. 각 그룹의 코어는 로컬 그룹을 대표하여 상위계층과 하위계층을 연결하는 기능을 가지고 있다. GAM은 2-계층 트리를 구성하여 상위계층은 각 송신자별 트리를, 하위계층은 공유 트리를 사용함으로써 트리를 관리하는 비용과 손실 복구의 효율성 사이의 절점을 찾고 있다.

하지만 GAM에서는 2-계층 트리를 연결하는 코어 노드가 네

트위크 상의 진입점(ingress point)에 미리 설치되어 있음을 가정하고 있다. 이는 코어 노드를 루트로 하는 로컬 그룹 내의 공유 트리가 멀티캐스트 라우팅 경로와 유사하게 만들어지게 해서 제어 트리의 효율성을 최대한 보장하기 위함이다. 하지만 네트워크 상의 어느 위치에 얼마만큼의 코어를 배치해야 하는지에 관련된 문제와 프로토콜 사용을 위해 별도의 인프라스트럭처를 구축해야 하는 문제는 프로토콜의 실제 적용을 제한하게 되고 또한 코어 노드가 정상 동작을 중지하면 전체 세션이 모두 중지되는 문제가 있다.

본 논문에서는 위에 언급된 GAM에서 코어 노드의 설치 및 복구 문제를 개선하기 위해 응용 세션 참가 노드들 중에서 제어 트리의 효율성을 최대한 보장할 수 있는 노드를 코어 노드를 선택하는 방법을 제안한다. 이를 통해 코어를 미리 특정 네트워크 위치에 설치하는 부담을 없앨 수 있고 코어의 실패 시 다른 노드를 코어 노드로 대체함으로써 프로토콜의 가용성을 증가시킬 수 있다.

본 논문의 나머지 장은 다음의 순서로 기술된다. 2장에서는 GAM에 대해 간략하게 소개하고 3장에서는 코어의 위치에 따른 프로토콜의 성능을 분석하였다. 4장에서는 설계고려 사항과 제안된 기법에 대해 소개하고 5장에서 시뮬레이션을 통한 비교 평가를 보이며 6장에서 결론 및 향후 연구 내용을 제시한다.

### 2. GAM (Group Aided Multicast)

트리 기반의 신뢰적인 멀티캐스트 프로토콜을 다대다 환경에 적용할 때 제어 트리의 효율성을 유지하기 위해서는 트리 관리의 비용이 상당히 증가하게 된다. 만약 다대다 환경에서 각 송신자에 대해 별도의 제어 트리를 관리하게 되면 그 비용은 송신자의 수에 비례하여 증가하게 된다. 이러한 비용을 줄이기 위해서 하나의 공유 제어 트리를 구성하는 연구가 소개되었지만 이는 실제 멀티캐스트 라우팅 트리에 근접하지 않으므로 손실 복구 효율성이 크게 떨어지게 된다.

GAM에서는 송신자와 근접한 노드들의 경우 멀티캐스트 라우

\* 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업과 정보통신부 및 정보통신연구진흥원의 디지털미디어연구소 지원사업의 지원에 의한 것임

링 경로를 많은 부분 공유한다는 관찰을 기반으로 지역적으로 근접한 노드들을 그룹으로 구성하여 각 그룹 내에서 공유 트리를 구성한다. 이렇게 함으로써 트리 관리에 따른 비용을 줄이고 제어 트리의 효율성은 크게 손상시키지 않을 수 있다. 손실 제어를 위한 트리를 그림 1과 같이 그룹 내 트리와 그룹 간 트리의 두 계층으로 나누어 관리함으로써 트리 관리에 따른 비용과 제어 트리의 효율성 사이의 절점을 찾은 것이 GAM이 제안한 2-계층 트리의 핵심적인 내용이다.

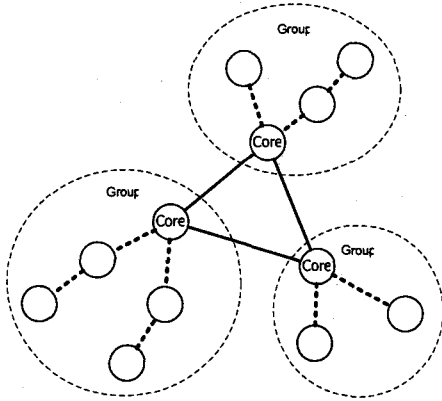


그림 1. 2-계층 트리

코어 노드는 로컬 그룹을 대표하여 1) 로컬 그룹 외부의 송신자가 보낸 데이터의 손실에 대해 해당 그룹의 코어 노드에게 손실 복구를 요청하고 2) 로컬 그룹 내부의 송신자가 보낸 데이터의 손실에 대해 외부 그룹 코어의 손실 복구 요청을 받아 손실 복구를 수행하여 그룹 간에 신뢰적인 멀티캐스트 데이터 전송이 가능하도록 한다. GAM에서는 이런 기능을 하는 코어가 분산 데이터베이스 서버나 웹 프록시(proxy)와 같이 미리 설치되어 있는 특정 노드로 가정하고 있고 제어 트리를 통한 지역적 손실 복구를 최대화하기 위해 네트워크상의 코어 위치를 해당 그룹이 속한 네트워크의 진입점으로 가정하고 있다.

하지만 코어를 미리 네트워크 상의 진입점에 설치하기 위해서는 네트워크에 대한 정보를 알고 있는 ISP (Internet Service Provider)의 도움을 받아 ISP내부의 진입점에 설치되어야 하기 때문에 프로토콜의 실제 적용하는데 제한 사항이 된다. 그리고 코어의 가용성을 높이기 위해서는 여러 세션에 동시에 참여하여야 하는데 이는 코어에 대한 부담을 증가시키고 코어 노드의 실패 시 관련된 세션 모두가 실패하는 상황이 발생한다. 또한 세션에 참여하는 노드들은 자신이 참가해야 할 코어를 알고 있다는 가정이 필요하거나 많은 코어를 중에서 근접한 코어를 선택해야 하는 부담을 가지게 되는 문제점들을 가지고 있다.

3. 코어의 위치와 프로토콜 성능의 상관 관계 분석

미리 설치된 노드가 아니라 세션에 참여한 노드들 중에서 코어를 선택할 경우 선택된 노드의 네트워크 위치가 프로토콜의 성능에 미치는 영향을 분석할 필요가 있다. 코어의 위치에 따른 프로토콜의 성능 변화를 분석하기 위해 각 노드가 가지고 있는 자원 (메모리, CPU, 네트워크 대역폭 등)은 동일하다고 가정한다.

신뢰적 멀티캐스트 프로토콜의 성능은 크게 제어 메시지의 비용과 손실 복구 지연 시간의 두 가지 기준으로 평가될 수 있다. 제어 메시지 비용은 전체 데이터 메시지의 개수에 대비한 제어 메시지의 개수의 비율을 따지는 것이고 손실 복구 지연 시간은 복구요

청을 시작한 후 복구를 받기까지 걸리는 지연 시간을 말한다. 제어 메시지 비용의 경우 세션에 참여하는 노드의 수나 링크에서 발생하는 손실에 영향을 받기 때문에 코어의 위치와는 큰 상관관계를 가지지 않는다.

하지만 손실 복구 지연 시간의 경우 코어의 위치에 따라 손실 복구 지연 시간에 영향을 줄 수 있다. 만약 코어가 멀티캐스트 라우팅 트리상의 종단(leaf)에 존재한다고 가정할 경우 코어 노드는 송신자로부터의 홑(hop) 증가로 인해 로컬 그룹내의 다른 노드들에 비해 손실 확률이 높아지게 된다. 이 경우 로컬 그룹내에서의 손실 복구 확률이 낮아지고 그 비율만큼 그룹간 손실 복구 횟수가 증가하게 된다. 예를 들어 MBone에서 측정된 멀티캐스트 손실 패턴을 보면 백본망이 아니라 네트워크의 종단이나 종단의 가까운 지역에서 더 많은 손실이 발생한다 [4]. 따라서 지역 손실 복구의 중요성은 커지게 된다. 일반적으로 로컬 그룹내의 노드들간의 네트워크 지연이 그룹간 노드들간의 네트워크 지연에 비해 훨씬 작기 때문에 손실이 발생한 메시지에 대해 로컬 그룹상에서 복구될 수 있는 지역적 복구 빈도를 최대화하면 손실 복구 지연 시간을 최소화할 수 있다.

3.1 손실 복구 지연 시간을 고려한 코어 노드의 선택 기준

코어 노드의 위치와 손실 복구 지연 시간의 상관 관계를 확인하기 위해 GT-ITM [6]을 사용하여 그림 2와 같은 네트워크 토폴로지(topology)를 생성하고 네트워크 시뮬레이터인 NS-2 [5]를 사용하여 코어의 위치에 따른 평균 손실 복구 지연 시간을 측정하였다. 실험에 사용된 토폴로지는 GT-ITM의 트랜짓-스텝 모델(Transit-Stub Model)을 사용하여 100개의 노드를 만들고 스텝(Stub) 간의 지연 시간, 트랜짓(Transit)과 스텝 간의 지연 시간 그리고 트랜짓 간의 지연 시간을 각각 평균 3, 20, 30ms로 설정하였다.

이 실험에서는 실험의 간편화를 위해 몇 가지 가정을 하였다. 첫 번째는 송신자는 하나가 존재한다는 것이다. 로컬 그룹 내부의 송신자에 대한 손실 복구 지연 시간은 로컬 그룹내의 지연 시간에 한정 지어질 수 있기 때문에 이 실험에서 고려대상이 되는 것은 로컬 그룹 외부의 데이터이다. 그러므로 송신자가 실험하는 그룹의 외부에 있다는 개수는 중요하지 않다. 두 번째는 로컬 그룹 내에서 만들어지는 공유 트리의 레벨(level)은 10이 된다. 실제 GAM에서는 로컬 그룹의 공유 트리는 여러 레벨로 이루어진다. 하지만 이 실험에서 멀티 레벨 트리로 인해 영향을 최소화하기 위해 트리의 레벨은 1로 실험하였다.

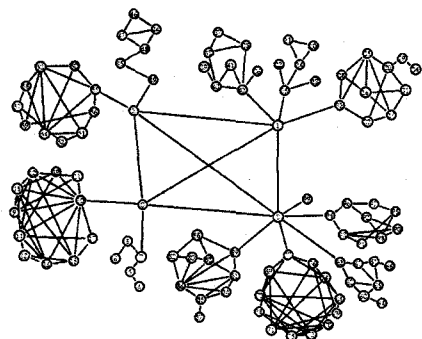


그림 2. 실험에 사용된 네트워크 토폴로지

그림 4는 그림 3과 같은 로컬 그룹 내에서 코어 노드의 위치에 따른 로컬 그룹 노드들의 평균 손실 복구 지연 시간을 보여주는 그래프이다. 그래프에서 그룹간 손실 복구는 코어 노드가 가지고

있지 못한 데이터에 대해 외부 그룹의 코어를 통해 손실 복구 수가 이루어졌을 때의 지연 시간을 말한다. 이 값의 경우 진입점과 노드 간의 홑수,  $k$ 와 상관 관계를 가짐을 알 수 있다. 그림 3의 화살표는 멀티캐스트 라우팅 경로를 보여주고 있다. 그림 3을 통해 알 수 있듯이 노드 28, 26, 31이 멀티캐스트 라우팅 경로를 공유하고 노드 33, 27, 30, 29, 34가 멀티캐스트 라우팅 경로를 공유하고 있다. 그림 3에서 평균 손실 지연 시간을 멀티캐스트 라우팅 경로를 공유하는 노드들 사이에서 비교하면  $k$ 가 증가함에 따라 평균 손실 복구 지연 시간도 증가함을 알 수 있다. 이를 통해 우리는  $k$ 가 증가할수록 손실 복구 지연 시간이 증가하는 경향을 볼 수 있다.

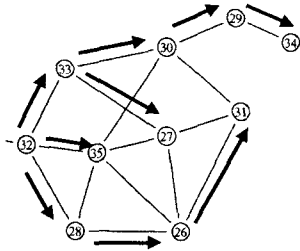


그림 3. 로컬 그룹의 토폴로지와 멀티캐스트 라우팅 경로

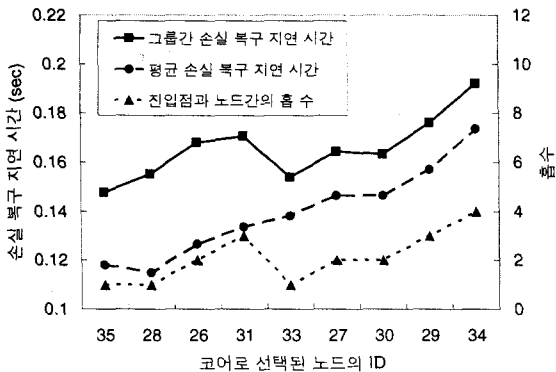


그림 4. 코어 노드의 위치에 따른 평균 손실 복구 지연 시간

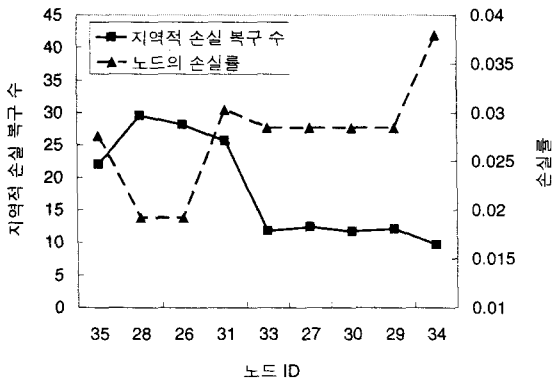


그림 5. 지역적 손실 복구와 노드의 데이터 손실률

하지만 라우팅 경로를 고려하지 않고 평균 손실 복구 지연 시간과  $k$  값을 비교할 경우 홑수의 증가에 따라 평균 손실 복구 지연이 증가하지 않음을 알 수 있다. 이는 각 노드의 손실률 차이에 따른 지역적 손실 복구의 정도에 의해 발생하는 것이다. 그림 5는 이 점을 보여주고 있다.  $k$ 가 1인 노드 33의 경우 지역적 손실 복구 수가 똑같이  $k$ 가 1인 35, 28 노드에 비해 평균 지역적 손실 복구 수가 크게 떨어질 것을 알 수 있다. 지역적 손실 복구 수가 작다는 것은 그만큼 지연 시간이 길어지는 그룹간 손실 복구를 해야 함을 의미하고 이것은 평균 손실 복구 지연 시간을 증가시키는 요인이 된다.

그림 5에서 지역적 손실 복구와 노드의 데이터 손실률, 두 값은 많은 부분 상관 관계를 보이지만 정확히 일치하지는 않는다. 이는 멀티캐스트 라우팅 경로의 차이에 의해 각 노드의 손실 패턴이 다르게 나타나기 때문이다. 예를 들어 노드 31의 경우 손실률이 높음에도 지역적 손실 복구 수는 크게 감소하지 않고 있다.

지금까지의 실험결과를 정리하면 1) 동일한 멀티캐스트 라우팅 경로 상에 있는 노드들의 경우 평균 손실 복구 지연 시간은 각 노드의  $k$  값이 증가할수록 증가하고 2) 동일한  $k$  값을 가지는 노드들의 경우 평균 손실 복구 지연 시간은 각 노드의 데이터 손실률이 작을수록 감소한다는 것을 알 수 있다. 결국 로컬 그룹의 평균 손실 복구 지연 시간을 최소화할 수 있는 노드는 진입점에 근접한 노드 중에서 가장 최소의 손실률을 보이는 노드이다.

위에서 설명한 코어 노드로서 적합한 노드를 찾기 위해서 4장에서 설명할 로컬 그룹 형성을 위한 기법에서 사용되는 랜드마크를 이용할 수 있다. 노드  $i$ 와 랜드마크 간의 측정된 RTT 값을  $R_i$ 라고 할 때, 진입점에 근접한 노드들은  $R_i$ 를 통해 추정할 수 있다. 그림 6은 그 점을 보여주고 있다. 4장에서 설명하듯이 각 노드는 로컬 그룹 구성을 위해  $R_i$ 를 측정한 후에 세션에 참여하기 때문에 이 값은 추가적인 비용 없이 로컬 그룹에서 적합한 코어 노드를 선택하기 위해 재사용될 수 있다.

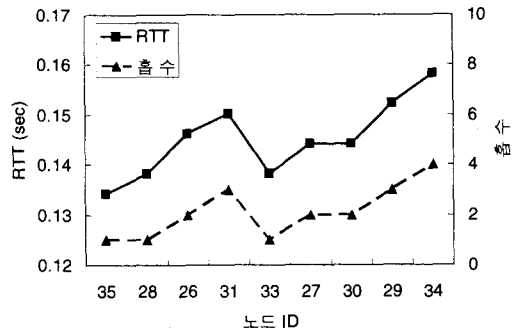


그림 6. 랜드마크와 노드의 RTT vs 진입점과 노드 간의 홑수

손실률은 각 노드가 세션에 참여하여 송신자로부터 데이터를 수신함으로써 얻어질 수 있는 값이다. 그렇기 때문에 안정적인 손실률을 얻기 위해서는 일정 크기 이상의 데이터를 수신해야 한다.  $R_i$  값이 초기 코어 노드 선택을 위한 기준으로 사용된다면 손실률은 코어 노드의 재 선택을 위한 기준으로 사용될 수 있을 것이다.

4. 제안 기법

4.1. 설계 고려 사항

기존 GAM의 경우 코어 노드가 네트워크 상의 특정 지점에 설치된 인프라스트럭처라고 가정하고 있다. 하지만 본 논문에서 제안하듯이 코어 노드를 세션에 참여하는 노드들 중에서 선택할 경우 다음과 같은 고려사항들이 있다.

4.1.1 로컬 그룹의 형성

GAM의 경우 로컬 그룹은 미리 정해진 코어를 기준으로 자동적으로 형성될 수 있었지만, 코어가 동적으로 선택될 경우에는 별도의 그룹 형성을 위한 알고리즘이 필요하게 된다. 로컬 그룹을 형성하는 알고리즘은 다음과 같은 사항이 고려되어야 한다.

첫 번째로 로컬 그룹은 지역적으로 근접한 노드들의 집합이 되어야 한다. 지역적으로 근접한 노드들의 집합은 멀티캐스트 라우팅 트리의 많은 부분을 공유하기 때문에 하나의 제어 공유 트리를 사용함에 따른 제어 트리의 효율성 저하를 최소화할 수 있다. 그리고 지역적으로 근접할 경우 노드들 간의 메시지 교환에 따른 지연 시간을 최소화할 수 있어 손실 복구 시 발생하는 지연 시간을 줄일 수 있다. 두 번째로 로컬 그룹은 그룹내 노드가 중복되지 않는 집합이 되어야 한다. 그룹내 노드가 중복될 경우 코어 선정이나 로컬 그룹 내에서 공유 트리를 형성하는데 문제를 야기한다. 마지막으로 로컬 그룹을 형성하는데 발생하는 비용은 노드의 증가에 큰 영향을 받지 않도록 확장성(Scalability)을 고려해야 하고 그룹 형성을 위한 알고리즘은 각 노드에서 분산된 방식으로 동작하되 프로토콜의 성능에 영향을 미치는 부분이 최소화되어야 한다.

4.1.2 코어 노드의 검색과 변경

세션에 참여하는 노드들은 임의의 시점에 세션에 참여하거나 탈퇴할 수 있다. 이 때 고려되어야 할 것은 다음 세 가지이다. 첫째로 세션에 새롭게 참여하는 노드가 어떻게 코어 노드를 찾을 것인가가 고려되어야 한다. 두 번째로 로컬 그룹 내에서 코어로서 더 적합한 노드가 새롭게 세션에 참여할 경우 코어 노드를 변경해야 한다. 마지막으로 코어 노드가 세션을 종료할 경우 그룹 내에서 다른 코어를 선택하기 위한 기법이 필요하다.

4.2. 제안 기법

앞에서 언급한 코어 선택 기준 분석과 설계 고려 사항을 바탕으로 로컬 그룹의 형성, 코어 노드 선택과 변경에 대한 기법을 다음과 같이 제안한다.

4.2.1 로컬 그룹의 형성

본 논문에서는 로컬 그룹의 형성을 위해 랜드마크(Landmark)를 이용한 Binning [3] 기법을 사용한다. 이 기법을 통해 로컬 그룹을 형성할 경우 지역적으로 근접한 노드들을 하나의 그룹으로 만들 수 있고 각 그룹은 중복되지 않는 집합으로 구성하는 것이 가능하다. 로컬 그룹을 형성하기 위한 비용은 극히 적으며 (랜드마크까지의 RTT 측정만이 요구됨) 각 노드가 분산된 방식으로 동작하게 된다. 이 기법은 인터넷상에 미리 잘 알려진 랜드마크를 이용하여 지역적으로 근접한 노드들을 하나의 그룹으로 묶는 기법이다. 세션에 참여하는 노드는 미리 정해진 몇 개의 랜드마크까지의 RTT(Round Trip Time) 값을 측정하여 각 랜드마크를 작은 RTT 순서에 따라 나열하여 얻어진 랜드마크 간 거리를 통해 그룹을 식별할 수 있다. 예를 들어 랜드마크 노드 L1, L2, L3의 RTT 값이 각각 80ms, 160ms, 126ms로 측정되었을 경우 그룹 식별자는 L1, L3, L2가 된다. Binning 기법을 이용할 경우 세션에 참여하는 노드는 세션에 참여하기 전에 랜드마크와의 RTT 측정을 통해 그룹 식별자를 얻음과 동시에 자연스럽게 로컬 그룹을 형성할 수 있다. 추가

적으로 각 노드가 얻은 랜드마크와의 RTT 측정 값은 본 제안 기법에서 로컬 그룹내의 코어 노드를 선택하는 기준으로 사용된다.

4.2.2 코어 노드 검색과 변경

3장에서 찾은 코어 노드 선택을 위한 기준을 통해 이 절에서는 세션에 새로 참여한 노드가 어떻게 코어 노드를 찾고 코어 노드의 변경은 어떻게 이루어지는지에 대한 과정을 설명한다.

코어 노드의 검색 세션에 참여하는 노드  $i$ 는  $R_i$  값을 가지고 있고 세션의 송신자들은 데이터 전송시 모든 데이터 패킷마다 자신이 속한 그룹의 ID와 코어 노드의 ID를 포함해서 전송한다.  $R_i$ 는 노드  $i$ 와 랜드마크들 간에 측정된 RTT의 평균값을 뜻한다. 노드  $i$ 는 세션에 참여한 후에  $W_s$  시간 동안 로컬 그룹내의 송신자가 보내는 데이터를 통해 코어 노드를 찾기 위해 기다리게 된다. 만약  $W_s$  시간내에 로컬 그룹 내의 송신자로부터 데이터를 수신할 경우 노드  $i$ 는 데이터 패킷내에 있는 코어 노드의 ID를 통해 로컬 그룹의 코어 노드를 알게 되고 코어 노드에 조인을 하여 로컬 그룹의 제어 공유 트리에 참여하게 된다. 만약  $W_s$  시간 동안 로컬 그룹내의 송신자로부터 데이터를 수신하지 못할 경우 노드  $i$ 는 일정 개수의 *CORE\_DISCOVERY* 메시지를 멀티캐스트로 전송하여 코어 노드를 직접 찾게 된다. 해당 로컬 그룹의 코어가 *CORE\_DISCOVERY* 메시지를 수신하면 *CORE\_DISCOVERY\_RESPONSE* 메시지에 자신의 주소를 포함해서 해당 노드에 유니캐스트로 전송한다. *CORE\_DISCOVERY\_RESPONSE* 메시지를 받은 노드는 코어 노드의 존재를 알게 되고 코어 노드에 조인하여 제어 공유 트리에 참여한다. 몇 번의 *CORE\_DISCOVERY* 메시지 전송에도 아무런 응답이 없을 경우 노드  $i$ 로컬 그룹에 코어가 존재하지 않다고 가정하고 스스로 로컬 그룹의 코어 노드가 된다.

코어 노드의 변경 코어 노드는 로컬 그룹 노드들이 조인하면서 얻은  $R$  값을 통해 어느 노드가 코어 노드로 가장 적합인지 추정할 수 있다. 세션에 참여한 후 일정량을 데이터를 수신한 노드는 자신이 수신한 데이터의 손실률을 계산하여 코어 노드에게 전달하여 차후에 코어 노드 변경을 위한 자료로 사용할 수 있도록 한다. 코어 노드의 변경은 각 노드의  $R$  값을 통해 진입점의 근접한 노드를 추정하고 그 노드들 중에서 손실률을 비교하여 가장 적은 손실률을 가지는 노드가 새로운 코어로 선택된다. 코어 노드 변경이 결정되면 기존 코어 노드는 새 후보 코어 노드에게 *CORE\_CHANGE* 메시지를 전달하고 이를 받은 새 후보 코어 노드는 기존 코어 노드에게 *CORE\_CHANGE\_CONFIRM* 메시지를 전달한다. 다음으로 기존 코어 노드는 로컬 그룹의 노드 리스트를 전달 받고 전달 받은 리스트의 노드들에게 코어 노드의 변경 사실을 전달한다. 이 과정은 코어 노드의 세션 탈퇴 시에도 동일하게 진행된다.

5. 성능 평가

성능 평가는 제안된 코어 선택 기준으로 수작업을 통해 코어를 선택하여 수행하였다. 측정을 위해 5개의 그룹 마다 9개의 노드를 배치하여 로컬 그룹의 평균 손실 복구 지연 시간을 측정하였다. 제안된 기법의 성능 차이를 보이기 위해 GAM에서 제시된 기법의 평균 손실 지연 시간 결과물 1로 두고 제안된 기법과 미리 측정된 최소 평균 손실 지연 시간을 비교 하였다. 최소 평균 손실 지연 시간은 미리 실험을 통해 각 로컬 그룹의 코어를 변경하면서 얻은 최소 값이다. 그림 7이 보여주듯이 우리가 제시한 코어 선택 기준으로 선택된 코어 노드가 로컬 그룹의 코어일 경우 평균 손실 지연 시간이 미리 측정된 최소 손실 지연 시간과 큰 차이를 보이지 않거나 동일한 값을 보여주고 있다.

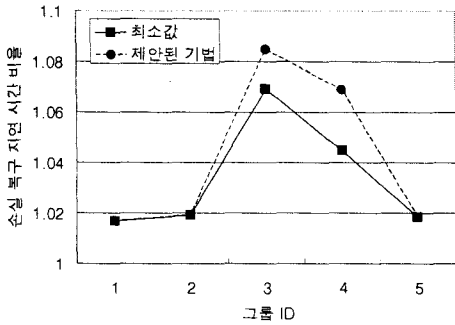


그림 7. 제안된 기법의 손실 복구 지연 시간 비교

### 6. 결론 및 향후 연구

다대다 그룹 통신 방법인 GAM은 네트워크의 특정 시점에 코어 노드가 미리 운용되고 있음을 가정하였다. 이런 추가적인 설치 비용은 GAM이 널리 사용되는데 장애가 된다. 이를 해결하기 위해서 본 논문에서는 세션에 참여하는 노드들 중에서 코어 노드를 선택하는 기법을 제안하였다.

본 논문에서는 실험을 통해 코어 노드를 선택하는 기준을 분석하여 진입점에 근접한 노드들 중에서 손실율이 가장 적어서 지역적 손실 복구를 최대한 보장할 수 있는 노드가 코어로서 적합하다는 것을 보였다. 이런 실험 결과를 바탕으로 Binning 기법을 통해 형성된 로컬 그룹 내에서 코어를 선택하고 변경하기 위한 방법을 제시하였다.

향후, 제시된 기법을 실제 구현하여 성능 평가를 수행하고 제안된 기법이 기존의 다른 그룹 기반의 신뢰적 멀티캐스트 기법들에 어떻게 적용될 수 있는지에 대한 연구를 진행할 것이다.

### 참고문헌

- [1] W. Yoon, D. Lee, and H.Y. Yoon, "On the Scalability of Many-to-many Reliable Multicast Sessions," *Concurrency and Computation: Practice and Experience*, November 2004
- [2] "Enhanced Communications Transport Protocol: Part 5," ITU-T SG17 Question 1: draft Recommendation X.608 (Working in Progress) | ISO/IEC JTC 1/SC 6/WG 7: WD 14476-5
- [3] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002.
- [4] M. Yajnik, J. Kurose, and D. Towsley, "Packet Loss Correlation in the Mbone Multicast Network," *IEEE GLOBECOM '96*, pp. 94-99, November 1996
- [5] NS-2: Network Simulator-2, <http://www.isi.edu/nsnam/ns/>
- [6] GT-ITM: Georgia Tech Internetwork Topology Models, <http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html>
- [7] W. Yoon, D. Lee, H.Y. Yoon, "Comparison of tree-based reliable multicast protocols for many-to-many sessions," *IEE Proceedings-Communications Vol. 152*, No. 6, December 2005