

미래 인터넷을 위한 근접성 기반의 간접 멀티캐스트

박소영, 이호진, 진복균, 최훈규, 권태경, 최양희.
서울대학교 컴퓨터공학부

{sympark, lumiere, jeonbg, hgchoi, tk, yhchoi}@mmlab.snu.ac.kr

Proximity-Aware Indirection Multicast for the Future Internet

Soyoung Park, Hojin Lee, Bokgyun Jeon, Hoongyu Choi, Taekyoung Kwon, Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
{sympark, lumiere, jeonbg, hgchoi, tk, yhchoi}@mmlab.snu.ac.kr

요 약

미래 인터넷에서는 사용자의 개인 방송 등 대용량의 다양한 스트리밍 서비스가 발생할 것으로 예상된다. 따라서 이를 효율적으로 지원하기 위한 멀티캐스트가 필수적이다. 미래 인터넷의 기반으로 제안된 Internet Indirection Infrastructure(i3)는 단 하나의 오버레이 네트워크를 통해서 송신자에게 투명하게 유니캐스트, 멀티캐스트, 애니캐스트, 단말 이동성을 제공하는 방식으로, 기본적으로 멀티캐스트를 지원한다. i3의 멀티캐스트를 규모성있게 구체화시킨 End-host Controlled Multicast(ECM)이 제안되었다. 하지만, ECM은 물리적 근접성을 고려하지 않기 때문에 비효율적인 멀티캐스트 트리가 형성된다. 본 논문은 이를 해결하기 위해 물리적 근접성을 반영한 근접성 기반의 간접 멀티캐스트(Proximity-Aware Indirection multicast, PAIM)를 제안한다. PAIM은 i3의 식별자 앞부분에 물리적 위치를 반영하는 부분(근접성 정보)을 추가하고, 이미 멀티캐스트 트리에 참여한 노드 중 근접성 정보가 동일한 노드가 있으면 그 노드에 붙여 멀티캐스트에 참여한다. 그럼으로써 PAIM에서는 근접성 단위가 같은 노드들이 합쳐진 효율적인 멀티캐스트 트리가 형성되어 ECM보다 네트워크 자원을 절약하고 중단간 지연시간을 감소시킨다.

I. 서론

인터넷은 단순한 데이터 전송에서 대용량의 다양한 스트리밍 서비스를 제공하는 미디어로 변모하고 있다 [4] [5] [6]. 여러 사람에게 동일한 서비스를 동시에 제공하는 실시간 방송인 경우, 네트워크 자원을 절약하기 위해서 멀티캐스트 이용이 장려된다. 특히, 미래 인터넷에서는 User-Creative-Contents(UCC)를 넘어서 다양한 사용자의 적극적인 개인 방송 참여가 예상되므로 네트워크에서 이를 효율적으로 지원하기 위해 멀티캐스트는 필수적이다.

인터넷은 원래 고정된 위치 사이의 점대점 유니캐스트 통신을 제공하기 위해 설계되었다. 멀티캐스트를 위한 인터넷 프로토콜(L3) 기법으로 IP 멀티캐스트가 제안되었지만, 기술적인 문제와 배치 문제로 널리 사용되지 않고 있다[7]. 이를 해결하기 위해 랑데부 기반 통신을 통한 간접적 방식으로, 일반적 통신 추상을 지원하는 i3가 제안되었다[2]. 일반적 통신 추상은 송신자에게 유니캐스트, 애니캐스트, 멀티캐스트, 단말 이동을 단 하나의 오버레이 네트워크를 통해서 투명하게 제공하는 것을 말한다. 따라서 i3는 미래의 응용 프로그램에 많은 이점을 줄 수 있을 것으로 예상되어, 미래 인터넷의 기반으로 주목 받고 있다.

i3의 멀티캐스트를 규모성있게 구체화하기 위해 ECM이 제안되었다[3]. ECM은 인프라 기반의 효율성(예, Proxied Overlay Multicast[8])과 중단 단말 기반의 유연성(예, End System Multicast[7])을 모두 제공한다. 하지만 물리적 근접성을 고려하지 않기 때문에 집합적인

전송(agggregation)을 효율적으로 하지 못하고 멀티캐스트 트리가 최적으로 형성되지 않는다.

이 문제를 개선하기 위해 본 논문은 조인(join)시 물리적 근접성을 반영하여 멀티캐스트 트리를 형성하는 근접성 기반의 간접 멀티캐스트(Proximity-aware Indirection Multicast, PAIM)를 제안한다. PAIM은 ECM에 비해 멀티캐스트 트리의 효율성을 높여, 링크 스트레스와 중단간 지연을 감소시킨다. 또한 PAIM은 i3와 ECM 기반으로 동작하기 때문에, 이들이 가지고 있는 장점을 유지하여 일반적 통신 추상과 멀티캐스트 트리 형성 시 사용자 탄력성을 그대로 제공한다.

본 논문은 다음과 같이 구성된다. 우선 제안하는 기법을 설명하기에 앞서 2장에서 우리가 제안한 기법의 기본이 되는 i3, ECM에 대해 간략히 알아본다. 3장에서는 ECM의 문제점을 살펴본 후, 이 문제점을 해결하기 위하여 본 논문에서 제시하는 기법, PAIM에 대해 설명한다. 4장에서 PAIM과 ECM을 비교 분석한 실험 결과를 제시한 뒤에, 5장에서 본 논문을 정리하며 마친다.

II. 배경

2.1. Chord

Chord[1]는 Distributed Hash Table(DHT) 프로토콜 중의 하나로, 가상의 고리를 형성하여 이 고리를 통해 오버레이 네트워크를 구축한다. 각 노드는 M 비트 식별자 공간에서

무작위하게 식별자를 할당받으며, 시계 방향으로 식별자가 커지도록 배치되어 식별자 고리를 형성한다. 키도 노드와 같은 식별자 공간에서 식별자를 할당받으며, 할당된 식별자 이상인 노드들 중 최소 식별자를 지닌 노드가 해당 키를 담당한다.

각 노드는 M 개의 노드에 대한 레코드를 가지고 있는 **finger table** 을 유지한다. i 번째 레코드는 자신의 식별자 + 2^{i-1} 이상인 노드들 중 최소 식별자를 지닌 노드를 가리키며, 이 노드는 [자신의 식별자 + 2^{i-1} , 자신의 식별자 + 2^i) 영역의 키를 담당한다.

임의의 키 값에 대한 질의를 전달 받은 노드는 레코드의 담당 범위가 자신과 **Chord** 거리(시계방향으로의 거리)가 가장 먼 레코드부터 가까운 레코드 순으로 (M 번째 레코드에서 1 번째 레코드 순으로) 자신의 **finger table** 을 살펴보면, 키의 식별자가 해당 레코드의 담당 범위에 속하면 그 레코드 노드를 다음 홉으로 선택한다. 이렇게 **finger table** 내의 레코드 노드 중 목적 노드까지의 **Chord** 거리가 가장 가까운 노드로 탐욕적으로 전달하는 과정을 반복함으로써 $\log_2 N$ 홉 수 이내에 최종적으로 원하는 키를 찾아가게 된다.

2.2. Internet Indirection Infrastructure

Internet Indirection Infrastructure(i3)[2]는 **Chord** 프로토콜[1]을 이용한 인프라 기반으로 일반적 통신 추상을 제공하는 오버레이 네트워크이다. 여기서 일반적 통신 추상은 단 하나의 오버레이 네트워크를 통해서 송신자에게 유니캐스트, 애니캐스트, 멀티캐스트, 단말 이동을 투명하게 지원하는 것을 뜻한다. 따라서 **i3** 는 오버레이 네트워크에서 동작함으로써 IP 계층 해결책 고유의 배치 문제를 해결할 뿐 아니라, 단 하나의 오버레이 네트워크를 사용함으로써 기존 오버레이 접근 방식의 중복성 문제를 제거한다.

i3 는 송신과 수신을 분리한다. 수신자는 임의로 식별자를 선택하고 <식별자, 수신자의 IP 주소> (이후 이를 트리거로 칭한다.)를 오버레이 네트워크에 삽입한다. 오버레이 네트워크는 **Chord** 프로토콜[1]을 이용하여 **Distributed Hash Table(DHT)**를 구성하고 있으며, 식별자를 책임지는 **i3** 서버는 <식별자, 수신자의 IP 주소> 관계를 저장한다. 송신자는 수신자에게 <식별자, 데이터> 형태로 패킷을 전송하며, 이 패킷은 **Chord** 프로토콜에 의해서 해당 식별자를 책임지는 **i3** 서버에게 전송된다. 최종적으로 패킷을 전달받은 **i3** 서버는 식별자와 관계를 이루는 실제 IP 주소로 패킷을 전송한다. **i3** 는 유니캐스트 이외에 그 중 멀티캐스트, 애니캐스트, 단말 이동을 지원하는데 그 중 멀티캐스트를 지원 하는 방법은 대략적으로 다음과 같다.

- 멀티캐스트: 같은 멀티캐스트 그룹의 모든 소속원들은 같은 식별자를 지닌 트리거를 삽입한다. 그 식별자에 대응하는 패킷은 해당 식별자를 책임지는 **i3** 서버에게 전송되고, 그 **i3** 서버는 동일한 식별자를 지닌 트리거들의 수신자들(해당 그룹의 모든 소속원들)에게 각각 전송한다.

2.3. End-host Controlled Multicast

End-host Controlled Multicast(ECM)[3]는 **i3** 에서 제공하는 멀티캐스트를 구체화한 기법이다. **i3** 에서 언급한 기본적인 멀티캐스트는 한 **i3** 서버가 모든 멀티캐스트 그룹 소속원들에게 패킷을 전달하기 때문에 규모성이 부족하다. 따라서 큰 규모의 멀티캐스트를 지원하기 위해서 **ECM** 은 동일한 식별자를 지닌 트리거 수 (이를 **out-degree** 로 칭하겠으며, 멀티캐스트 트리의 가지 수로

생각하면 된다.)를 제한하고 트리거 연쇄를 이용하여 **i3** 서버들간에 트리를 구성한다.

단말은 두 개의 트리거를 삽입함으로써 멀티캐스트 트리에 참여한다. 한 트리거는 멀티캐스트 트리상의 **i3** 서버 중 **out-degree** 가 기준치 미만인 **i3** 서버 (**joinable server**)에 삽입된다. 이 때 사용되는 식별자(**id_{on-tree}**)는 그 **i3** 서버가 참여하고자 하는 멀티캐스트 그룹을 위해 이미 사용하고 있는 식별자이다. 또 다른 트리거는 호스트가 임의로 선택한 식별자(**id_{end}**)를 가지며, 단말 근처에 있는 **i3** 서버에 삽입된다. 전자의 트리거는 <**id_{on-tree}, id_{end}**> 형태로 이미 멀티캐스트 그룹에 참여한 **i3** 서버와 단말 근처에 있는 **i3** 서버를 연결해주는 역할을 하며, 후자의 트리거는 <**id_{end}, 호스트의 IP 주소**> 형태로 단말 근처의 **i3** 서버가 단말에게 해당 멀티캐스트 그룹의 데이터를 전달할 수 있도록 해준다. 이렇게 트리를 구성하면 **out-degree** 를 제한하면서도 자연스럽게 트리의 높이가 증가하게 되어 큰 규모의 멀티캐스트를 지원할 수 있다

III. 근접성 기반의 간접 멀티캐스트

3.1. 개요

ECM 은 발견적인 방식을 사용하여 최적의 노드에 조인(**join**)한다. 하지만, 이 때 근접성을 고려하지 않기 때문에 멀티캐스트 트리의 전체 성능이 상당히 떨어질 수 있다. 근접성을 고려하여 인근에 이미 트리에 속한 노드에 조인하면, 새로 조인하는 호스트는 최적의 노드에 조인하는 경우보다 성능이 다소 떨어지지만, 전체적으로 트리의 성능을 크게 높일 수 있다.

조인시 근접성을 이용하면, 조인을 개별 노드 단위 대신에 근접성 단위로 하게 된다. 따라서, 개별 노드에게 각각 유니캐스트로 전송하는 것을 근접성 단위로 합쳐서 전송하기 때문에 효율성을 높일 수 있다. 더욱이 근접성 단위로 조인하면, 상위 노드(멀티캐스트 트리의 뿌리 노드(**root**)에 가까운 노드)의 **out-degree** 가 절약되어, 상위 노드를 나중에 조인할 호스트들의 최적 노드로 사용할 수 있기 때문에 더 효율적인 멀티캐스트 트리를 형성할 수 있다.

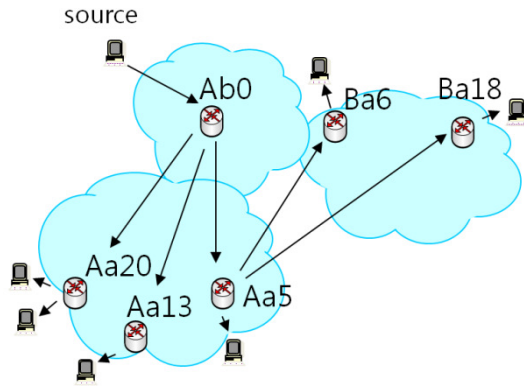
따라서 우리는 멀티캐스트 트리 형성 시, 근접성을 고려하여 멀티캐스트 트리에 참여함으로써 멀티캐스트 트리의 효율성을 향상시키는 근접성 기반의 간접 멀티캐스트, **Proximity-Aware Indirection Multicast(PAIM)**를 제안한다.

3.2 ECM 의 문제점

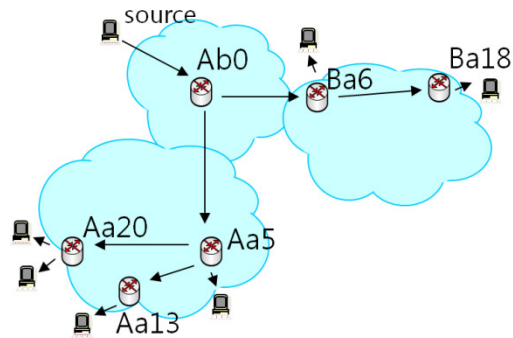
ECM 은 조인하려는 노드가 탐욕적으로 자신에게 최대한 유리한 위치에 조인한다. 이는 조인하는 순간에는 최적의 위치에 조인하지만 전체 멀티캐스트 트리 효율성은 떨어지는 문제점을 발생시킨다. 반면에 **PAIM** 은 근접성에 우선 순위를 두어 조인함으로써 전체 멀티캐스트 트리의 효율성을 향상시킨다.

[그림 1], [그림 2]는 각각 **ECM**, **PAIM** 방식으로 **Aa5**, **Aa13**, **Aa20**, **Ba6**, **Ba18** 순서로 조인했을 때 형성되는 멀티캐스트 트리를 보여준다. 식별자 앞의 두 알파벳은 근접성을 나타내는 부분으로 이 부분이 같으면 같은 근접성 범위 내에 있다고 판단한다.

ECM([그림 1])에서 **Aa13**, **Aa20** 은 탐욕적으로 자신과 뿌리와의 거리가 최소가 되도록 하는 **Ab0** 에 조인한다. 이는 근접성 **Aa** 범위에 세 개의 패킷을 각각 전달하도록 하여 네트워크 자원을 낭비한다. 또한 **Ba6** 이 조인할 때,



[그림 1] ECM 방식으로 형성된 멀티캐스트 트리 예시



[그림 2] PAIM 방식으로 형성된 멀티캐스트 트리 예시

Ab0의 out-degree가 모두 소모되었기 때문에 Aa5에 조인하게 되어 뿌리로부터의 거리가 멀어져 지연이 길어지는 문제점이 발생한다. PAIM([그림 2])은 Aa13, Aa20 근접성을 우선으로 하여, 이미 멀티캐스트 트리에 참여하였으며 자신과 근접성이 같은 Aa5에 조인한다. 따라서 근접성 Aa 범위내 세 번 보내던 패킷을 한 번만 보냄으로써 네트워크 자원을 절약한다. Ba6이 참여할 때 Ab0의 out-degree가 남아있기 때문에 Ab0에 조인하여 ECM보다 뿌리와 Ba6과의 거리가 크게 줄어든다. 이런 이점이 Aa13, Aa20이 근접성을 선택함으로써 뿌리와 멀어지는 비용을 보상하고 멀티캐스트 트리의 전체적인 지연을 감소시킨다.

3.3. 노드 식별자 할당

ECM에서는 노드의 식별자를 무작위하게 할당한다. 우리는 노드 간의 근접성을 알 수 있게 하기 위해서 ECM처럼 무작위하게 생성된 식별자(무작위 부분)의 앞부분에 물리적 위치를 반영하는 부분(물리적 부분)을 추가한다. 물리적 부분이 같은 노드들은 같은 근접성에 위치한다. 물리적 부분은 AS 번호, IP 주소의 네트워크 접두어(prefix), 네트워크 좌표(network coordinate)[8] 등을 이용해서 선택할 수 있다. 최적의 물리적 부분을 선택하는 방법은 우리 논문의 영역 밖이다.

3.4. 참여 과정

PAIM은 ECM의 방식 전에 근접성을 이용하여 참여 위치를 정하는 과정을 추가함으로써 더 효율적인 멀티캐스트 트리를 형성한다.

ECM은 멀티캐스트 뿌리부터 내려오면서 발견적인 방식으로, 참여하려는 노드와 멀티캐스트 트리의 뿌리와 거리를 최소화하는 참여 위치를 찾는다. 멀티캐스트 트리의 뿌리와 노드까지의 거리는 뿌리부터 참여

위치까지의 트리 상의 거리와 참여 위치부터 참여를 원하는 노드까지의 IP 소요 시간으로 구성된다. 트리 상의 거리는 트리 상의 경로에 대응되는 물리적 링크들의 IP 소요 시간의 합을 의미한다.

PAIM은 거리보다 근접성을 먼저 확인한다. 근접성이 같은 노드(식별자 중 물리적 부분이 같은 노드)가 멀티캐스트 트리에 있으면 그 노드 혹은 그 노드의 서브트리에 조인하여 멀티캐스트 트리에 참여하고, 그렇지 않으면 ECM과 동일하게 동작한다. PAIM의 참여 과정은 [그림 3]에 보여지는 순서를 따른다. 여기서 참여 가능(joinable) 노드는 out-degree가 기준치 이하인 노드, 참여 불가능(full) 노드는 out-degree가 기준치 초과인 노드를 말한다.

근접성을 반영하기 위해 현재 노드의 손자 노드들의 식별자 중 물리적 부분을 살펴보는 부분을 추가하였다. 현재 노드의 손자의 식별자를 보고 손자와 참여하려는 노드의 근접성이 같을 경우, 손자가 참여 가능하면 그 손자에 조인하고, 참여 불가능하면 손자 위치부터 다시 참여 위치를 검색한다. 이 때, 멀티캐스트 트리의 특정 서브트리만 깊이가 길어지는 것을 방지 하기 위하여 참여 불가능한 노드보다 참여 가능한 노드를 먼저 검색하며, 후보 노드 집합에서 한 노드를 선택할 때 무작위하게 선택한다. 참여 불가능한 노드가 참여 가능한 노드보다 서브트리의 깊이가 더 클 가능성이 크기 때문에, 참여 가능한 노드에게 우선 순위를 줌으로써 깊이가 균형있게 증가되도록 한다. 또한

```

join(S, nnew)
// 뿌리가 송신자 노드 S인 멀티캐스트 트리에 노드 nnew가 참여
min_distance = ∞
ncurr = S
end = 0 // 근접성 검사를 통해 참여 위치를 찾으면 1,
// 그렇지 않으면 0
do
candidate_set = same_proximity(ncurr's joinable grandchildren,
nnew)
If(candidate_set != ∅)
ncurr = random_select(candidate_set)
end=1
break
candidate_set = same_proximity(ncurr's full grandchildren, nnew)
If(candidate_set != ∅)
ncurr = random_select(candidate_set)
break
n = select( ncurr's joinable children, S, nnew)
if(min_distance > distance(S, n, nnew))
min_distance = distance(S, n, nnew)
njoin = n
If(ncurr's full children == ∅)
break
ncurr = select( ncurr's full children, S, nnew)
while(distance(S, ncurr, nnew) < min_distance || end == 0)
join_at(nnew, njoin)

same_proximity(a,b)
// a에 해당되는 노드들 중 b와 근접성이 같은 노드를 반환
random_select(a)
// a 중 무작위하게 하나를 선택하여 반환
select(a, b, c)
// a에 해당되는 노드들 중 b와 c의 거리를 최소화하는 노드를 반환
distance(a, b, c)
// a와 b의 트리 상 거리와 b와 c의 IP 소요 시간의 합을 반환
join_at(a, b)
// a는 b에 조인하여 멀티캐스트 트리에 참여

```

[그림 3] PAIM 참여 과정 알고리즘

후보에 속하는 노드들 중 한 노드를 고를 때 모든 노드의 선택 가능 확률을 동일하게 함으로써 특정 노드에게만 계속 조인하는 현상을 방지한다.

손자들 중 근접성이 같은 노드가 없으면 기존 ECM 방식을 사용한다. 트리를 타고 내려가면서 최소 거리를 줄일 수 있을 때까지 참여 위치를 검색한다. 참여 가능한 자식 노드 중 뿌리와의 거리를 최소화하는 노드를 고르고, 그 거리가 최소 거리보다 작으면 갱신한다. 그리고 참여 불가능한 자식 노드 중 뿌리와의 거리를 최소화하는 노드를 고른 후, 그 거리가 최소 거리보다 더 작으면 그 자식 노드가 있는 가지를 타고 내려가서 처음부터 (손자들의 근접성을 확인하는 과정부터) 같은 과정을 반복한다.

PAIM 은 발견적인 방법으로 근접성이 같은 노드를 검색하기 때문에, 멀티캐스트 트리에서 자신과 같은 근접성에 있는 노드를 완벽하게 찾아내지 못한다. 하지만 멀티캐스트 트리의 모든 노드들을 검색하는 것은 추가 비용이 크기 때문에 발견적인 방식을 선택하였으며, 뒤의 실험 결과를 보면 우리가 제안한 알고리즘이 접근성이 동일한 노드들을 대부분 합쳐준다는 것을 알 수 있다.

IV. 실험

4.1. 실험환경

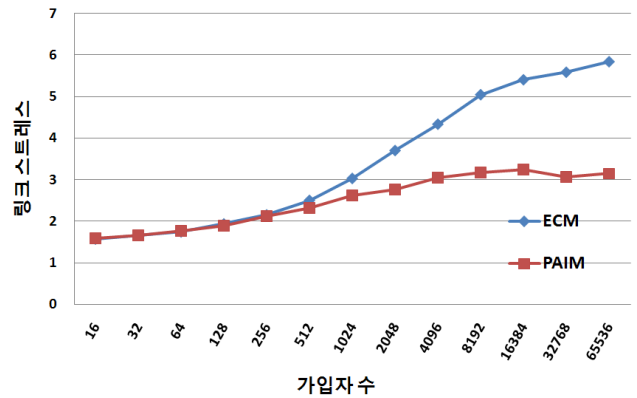
C 를 이용하여 [2]에 언급된 i3 프로토콜 위에서 ECM, PAIM 이 동작하도록 구현하였다. 토폴로지 생성은 GT-ITM 토폴로지 생성기[10]를 이용하였다. [11]의 결과를 바탕으로 GT-ITM 매개 변수를 설정하여 총 5000 개의 노드가 있는 DFN(영국 연구망, German Research Network) 같은 토폴로지를 10 개 생성하였다. 모든 노드는 i3 서버라고 가정하였다. 근접성 단위는 stub 으로 사용하였으며, 토폴로지 내에 총 600 개의 stub 이 존재한다. 멀티캐스트 가입자는 무작위하게 i3 식별자를 할당받고, Chord 에서 그 i3 식별자를 담당하는 i3 서버에게 조인한다. 멀티캐스트 그룹 크기와 관련된 규모성을 보여주기 위하여 멀티캐스트 그룹의 가입자 수를 변화시키며 실험하였다. 16 명의 가입자부터 시작하여 2 의 지수로 증가시켜가며 65,536 명의 가입자까지 실험하였으며, 각 가입자 수 별로 10 개의 토폴로지에서 실험을 하여 평균값을 냈다. 링크 사이의 지연은 transit 도메인 간의 링크는 [20ms, 50ms], transit 도메인과 stub 도메인 간의 링크는 [5ms, 10ms], transit 도메인 내 링크는 [1ms, 3ms], 그리고 stub 도메인 내의 링크는 [1ms, 2ms] 범위에서 균등하게 무작위 추출하였다.

4.2. 링크 스트레스

PAIM 은 근접성을 반영하여 멀티캐스트 트리에서 참여하기 때문에 ECM 보다 효율적인 멀티캐스트 트리를 형성하여 네트워크 자원을 절약한다. [그림 2]에서 본 바와 같이 PAIM 은 각각 유니캐스트로 전송하던 것을 근접성 단위로 합쳐 한 번만 전송하기 때문에 네트워크 자원 사용량을 감소시킨다. 이를 알아보기 위해 패킷을 전송할 때 발생하는 평균 링크 스트레스를 측정하였다.

링크 스트레스는 패킷을 전송할 때 같은 패킷이 한 링크를 지나가는 횟수이다. 멀티캐스트 트리를 따라 패킷 전송 시 링크 스트레스가 1 이면, 전송한 패킷이 물리적 링크들을 딱 한번씩 지나간 것으로 네트워크 자원을 최대로 절약하는 최적의 멀티캐스트 트리가 형성된 경우이다. 링크 스트레스가 1 보다 클수록 한 링크에

동일한 패킷이 여러 번 지나다녀 멀티캐스트 트리의 효율성은 줄어든다.



[그림 4] 가입자 수 변화에 따른 링크 스트레스

[그림 4]에서 보면 가입자 수가 512 명이 넘어가면 PAIM 이 ECM 보다 링크 스트레스가 작다. 가입자 수가 증가함에 따라 ECM 은 링크 스트레스 증가가 빠르게 증가하는데 반해, PAIM 은 서서히 증가한다. PAIM 은 근접성 단위로 합쳐지는 단말들에게는 한 번만 전송한 후, 합쳐진 단말들 내에서 패킷을 전달받기 때문에 ECM 에 비해 링크 스트레스가 작다. 또한 근접성 단위로 참여를 함으로써 절약되는 상위 노드의 out-degree 가 나중에 멀티캐스트 트리에서 참여할 노드의 최적 노드로 사용될 수 있기 때문에, 가입자 수가 증가하여도 링크 스트레스가 서서히 증가하게 된다. PAIM 이 ECM 보다 링크 스트레스가 낮다는 점은 PAIM 이 ECM 보다 네트워크 자원을 더 효율적으로 사용한다는 사실을 말해준다.

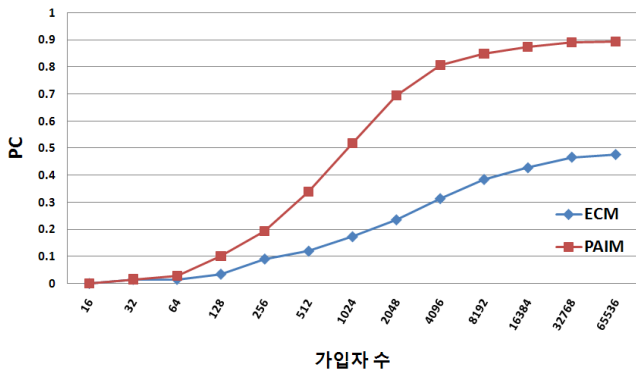
4.3. 물리적 결합력(physical coherence)

PAIM 은 근접성 단위로 합침으로써 효율적인 멀티캐스트 트리를 형성한다. 이 때 멀티캐스트 트리에서 접근성이 동일한 노드를 얼마나 잘 찾아가는가에 따라 멀티캐스트 트리의 효율성은 달라진다. 우리는 제안한 발견적인 방식이 얼마나 접근성이 동일한 노드들끼리 합쳐주는지 보기 위해 물리적 결합력(physical coherence, PC)라는 메트릭(metric)을 도입하였다. PC 계산 식은 다음과 같다.

$$PC = \frac{\text{근접성 단위의 도메인 내의 가지 수}}{\text{총 가지 수} - \text{근접성 단위의 도메인 수} + 1}$$

[식 1] 물리적 결합력

PC 는 총 가지 수에 비해 근접성 단위의 도메인 내의 가지가 얼마나 있는지에 따라 값이 달라진다. PC 가 1 인 경우가 최적 값으로, 접근성이 동일한 노드들끼리 모두 합쳐져서 멀티캐스트 트리를 형성한 경우이다. 이 경우 근접성 단위의 도메인 간의 가지는 한 개씩만 존재하고 다른 도메인과 연결된 노드를 제외한 도메인 내의 모든 노드들은 해당 도메인 내의 노드들과 연결된다. 따라서 한번씩 도메인을 잇는 가지들을 제외한 모든 멀티캐스트 트리의 가지들이 근접성 단위의 도메인 내에 존재하기 때문에 PC 가 1 이 된다. 접근성이 동일한 노드들끼리 합쳐지지 않아서 도메인 간의 가지 수가 많아질수록 PC 는 커진다. ECM, PAIM 의 PC 를 각각 측정한 결과는 [그림 5]와 같다.



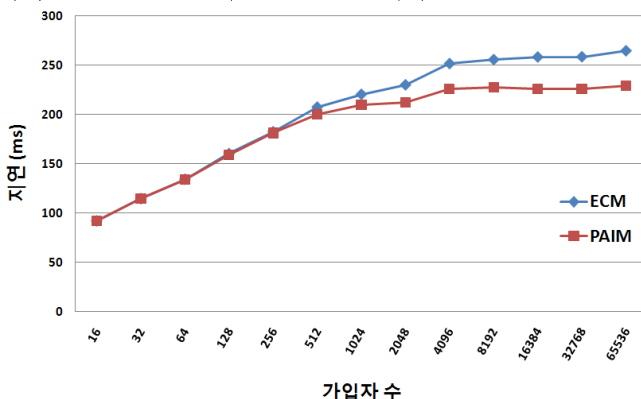
[그림 5] 가입자 수 변화에 따른 PC

근접성 단위를 stub 으로 설정하여 PC 를 측정하였다. [그림 5]에서 보면 전체적으로 PAIM 의 PC 가 ECM 보다 상당히 높다. ECM 은 근접성을 반영하지 않기 때문에 비효율적인 멀티캐스트 트리가 형성되어 PC 가 낮다. PC 가 최대인 가입자 수가 65536 명일 때도 0.5 를 넘지 못한다. 이는 멀티캐스트 트리의 가지 중 절반 이상이 다른 도메인 사이를 걸친다는 것을 의미하며, 이런 가지들 때문에 동일한 패킷이 도메인을 반복하여 드나들으로써 네트워크 자원을 낭비하고 지연을 증가시킨다. 반면, PAIM 의 PC 는 가입자가 1024 명을 넘어서면 0.5 를 넘어선다. 특히 가입자 수가 4096 명 이상이 되면 0.8 보다 큰데, 이는 우리가 제시한 알고리즘이 접근성이 동일한 노드들을 대부분 합쳐 효율적인 멀티캐스트 트리를 형성함을 말해준다.

4.4. 지연

근접성 단위로 잘 합쳐진(PC 값이 1 에 가까운) 멀티캐스트 트리는 링크 스트레스를 감소시킬 뿐만 아니라 각 가입자가 겪는 지연도 감소시킨다. 근접성이 동일한 노드가 존재하면 그에 조인함으로써 그 노드는 멀티캐스트 트리의 뿌리와 거리가 멀어질 수 있지만 상위 노드의 out-degree 가 절약되어 전체적인 지연은 감소시킬 수 있다. 멀티캐스트 트리의 뿌리부터 시작하여 각 가입자에게 도착할 때까지 거치는 링크들의 지연을 다 합친 후 평균 낸 결과는 [그림 6]과 같다.

링크 스트레스와 마찬가지로, 가입자 수가 512 명을 넘어서면 PAIM 의 지연이 ECM 보다 적다. 가입자 수가 적을 때에는 두 멀티캐스트 트리의 효율성이 크게 차이 나지 않지만 가입자 수가 많아질수록 PAIM 의 이점(링크 스트레스 감소, 지연 감소)이 증가한다. 따라서 PAIM 이 더 많은 수의 가입자가 존재하는 멀티캐스트를 지원할 수 있으며, 이는 규모가 큰 멀티캐스트가 빈번하게 발생하는 미래 인터넷 환경에서 필요한 요소이다.



[그림 6] 가입자 수 변화에 따른 지연

V. 결론

ECM 은 물리적 정보를 반영하지 않은 오버레이 네트워크에서 동작하기 때문에, 실제 물리적 네트워크에서 보면 비효율적인 멀티캐스트 트리가 형성된다. 이는 네트워크 자원을 낭비하고 지연을 길게 한다. 이 문제를 해결하기 위해 본 논문은 i3 식별자에 근접성 정보를 반영하고 이를 이용하여 멀티캐스트 트리를 형성하는 PAIM 을 제안하였다. PAIM 은 참여하려는 노드와 근접성이 같은 노드가 이미 멀티캐스트 트리에 존재하는 경우 그 노드 혹은 그 노드의 서브트리에 조인한다. 근접성이 같은 노드들을 합쳐서 하나의 패킷을 보냄으로써, ECM 보다 네트워크 자원을 절약하고 지연을 감소시킨다.

감사의 글

본 연구는 지식경제부 및 정보통신연구진흥원의 IT 성장동력핵심기술개발사업의 일환으로 수행하였음. [2007-F-038-02, 미래 인터넷 핵심기술 연구]

본 논문은 한국정보사회진흥원의 광대역통합연구개발사업을 통해 개발된 결과임을 밝힙니다.

이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다.

참고문헌

- [1] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications," *IEEE/ACM Transaction on Networking*, vol. 11, no. 1, 2003.
- [2] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, Sonesh Surana, "Internet Indirection Infrastructure," *IEEE/ACM Transactions on networking*, vol. 12 no. 2, 2004
- [3] Karthik Lakshminarayanan, Ananth Rao, Ion Stoica, Scott Shenker, "End-host Controlled Multicast Routing," *Elsevier Computer Networks*, vol. 50, no. 6, 2006.
- [4] YouTube, <http://www.youtube.com>
- [5] Afreeca, <http://www.afreeca.com>
- [6] Koen Casier, Bart Lannoo, Jan Van Ooteghem, Sofie Verbrugge, Didier Colle, Mario Pickavet, and Piet Demeester, "Adoption and Pricing: The Underestimated Elements of a Realistic IPTV Business Game," *IEEE Communications Magazine*, vol. 46, no. 8, 2008.
- [7] Y. hua Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, 2002.
- [8] Frank Dabek, Russ Cox, Frans Kaashoek, Robert Morris, "Vivaldi: A Decentralized Network Coordinate System," *in Proc. ACM SIGCOMM '04*, 2004.
- [9] J. Buford, "Hybrid overlay multicast framework," February 2008, internet draft, draft-irtf-sam-hybrid-overlay-framework-02.txt, work in progress.
- [10] GT-ITM, <http://www.cc.gatech.edu/projects/gtitm/>
- [11] O. Heckmann, M. Piringer, J. Schmitt, and R. Steinmetz, "On realistic network topologies for simulation," *MoMeTools '03: Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, pp.28-32, 2003.