QoS 멀티캐스트 라우팅을 위한 계획된 트리 재구성 방법

(Pre-Planned Tree Reconfiguration Mechanism for QoS Multicast Routing)

한 승 재 † 박 선 주 ††
(Seungjae Han) (Sunju Park)

요 약 전송-수신 쌍들을 연결하는 많은 수의 경로들로 이루어진 멀티캐스트 트리에서 네트워크 구성요소의 실패는 멀티캐스트 트리의 일부를 손상시킬 수 있다. 그러나 하나의 구성요소의 실패를 복구하기 위해 전체 멀티캐스트 트리를 다시 만드는 것은, 실패의 영향을 받지 않은 경로를 사용하는 그룹 멤버들까지도 서비스의 중단을 겪어야 하기 때문에 바람직하지 않다. 본 논문은 QoS 멀티캐스트 트리에서 재구성해야 할 영역을 줄이면서 재구성의 성공 가능성을 최대화하는 계획된 재구성(Pre-Planned Reconfiguration: PPR) 정책을 제안한다. PPR 방식은 멀티캐스트 트리의 전송-수신 쌍을 연결하는 각 경로에 재구성 경로를 미리 만들고, 이들 경로에 필요한 자원을 미리 예약해 둔다. 이를 위해 우리는 기존 멀티캐스트 트리의 변화를 최소화하며 손상되지 않은 부분들의 서비스를 최대한 유지하는 재구성 경로의 라우팅 방법을 고안하였으며, 효율적 자원 공유 방법을 사용하여 재구성 경로들을 위해 예약된(실패가 일어나지 않을 경우 사용되지 않는) 자원의 양을 줄인다. PPR 방식은 실패 복구를 위해 여러 멀티캐스트 세션들이 동시에 엄청난 경쟁을 하는 것을 막을 수 있다. 시뮬레이션을 통해 최단경로 라우팅을 사용하는 전송자 중심 멀티캐스트 트리와 공유 멀티캐스트 트리에서 각각 성능을 평가한 결과, PPR 방식은 적당한 오버헤드내에서 모든 그룹 멤버들에게 성공적인 재구성을 제공한다. 또한 PPR 방식은 그룹 멤버쉽이 동적으로 변화할 때에도 잘 적응한다.

키워드 : 멀티캐스트, fault tolerance, 자원공유, 라우팅

Abstract A multicast tree includes several, possibly a large number of, paths connecting source-receiver pairs, and network failure may disable part of the multicast tree. Reconstruction of the entire multicast tree to recover from a component failure is highly undesirable, because some group members have to suffer service disruptions even though the communication paths to/from them are not affected by the failure. To limit reconfiguration region and to maximize the likelihood of successful reconfiguration, we propose and evaluate a pre-planned reconfiguration policy for QoS multicast sessions. Specifically, we equip a reconfiguration path (RP) with each end-to-end path that connects a source-receiver pair in the multicast tree, and reserve resources in advance along the RPs. Efficient resource-sharing techniques are applied to reduce the amount of resources reserved for RPs but not used in the absence of failures. This way, we prevent uncontrolled competition among different multicast sessions which may simultaneously try to recover from failures. We evaluate the performance of the proposed scheme using simulation on randomly-generated networks. We use the shortest-path routing for QoS multicast sessions, and simulate both source-based and shared multicast trees. The evaluation results indicates that successful pre-planned reconfiguration can be achieved for all group members with reasonable overhead. Our scheme is also shown to adapt well to dynamic changes of group membership.

Key words : multicast, fault tolerance, resource sharing, routing

· 본 연구는 정보통신부 및 정보통신연구진흥원의 해외교수초빙지원사업의 연구결과로 수행되었음. 두번째 저자는 연세경영연구소의 부분적 지원을 받았음

† 종신회원 : 연세대학교 컴퓨터과학과 교수
sjhan@cs.yonsei.ac.kr
†† 종신회원 : 연세대학교 경영학과 교수
boxenju@yonsei.ac.kr

논문접수 : 2006년 3월 13일
심사완료 : 2007년 1월 16일

1. 서 론

하나의 메시지 전송으로 그룹에 속한 모든 멤버들에게 메시지 전송이 가능한 일대다(one-to-multipoint) 통신인 멀티캐스트는 여러 각도에서 활발하게 연구되어왔

다. 그래프 이론 연구자들은 멀티캐스트 라우팅의 핵심 연구주제로 스타이너(Steiner) 트리 문제에 초점을 맞추어[1], NP-complete인 스타이너 트리 문제를 푸는 다양한 polynomial-time 휴리스틱 알고리즘들을 개발하였다[2-4]. 한편 인터넷 연구자들은 멀티캐스트 라우팅을 위한 여러 방법들을 제안해왔다. 예를 들어, IP 멀티캐스트에서는 스타이너 트리 방식의 저조한 확장성(scalability)을 피하기 위해 최단 경로(shortest path) 트리를 사용하였다[5,6]. 멀티캐스트 그룹 멤버쉽이 동적으로 변화하는 경우에는 확장성이 특히 중요하며 최단경로 방식이 더 효과적이라고 알려져 있다[7]. 다대다(multi-point-to-multipoint) 통신을 요구하는 응용분야에서 확장성을 높이는 다른 방법으로는 공유 트리 방식이 제안되었다[8,9].

유니캐스트와는 달리, 멀티캐스트 세션에는 여러명의 수신자가 존재하므로 메시지 라우팅이나, 손실(loss) 없는 메시지 전송, 순서에 맞는 메시지 전송 등을 지원하는 것이 어렵다. 또한 서비스 질의 정도에 대한 요구도 응용분야에 따라 크게 좌우된다. 예를 들어, 연속적인 미디어의 전송에서는 메시지 전송의 신뢰도나 정해진 수신자 순서에 맞는 메시지 전달 등을 엄격하게 요구하지 않지만, 분산 시뮬레이션이나 분산 처리 시스템은 이를 엄격하게 요구한다. 대신, 연속 미디어 전송은 시기를 맞춘 제때의(timely) 메시지 전송을 요구한다. 미디어 전송 응용분야의 이러한 QoS 요구를 Mbone에서는 UDP와 RTP를 사용하여 부분적으로 만족시켜 준다[10]. 앞으로 QoS를 지원하는 라우터들이 배치되면 QoS-sensitive 통신의 완전한 지원이 가능할 것이다.

연속 미디어 데이터의 전송은 인터넷 멀티캐스트중 가장 중요한 응용분야 중의 하나이다. 늘어나는 QoS-senstive 멀티캐스트 서비스에 대한 요구를 만족하기 위해서 QoS 멀티캐스트 라우팅에 대한 연구가 있고[11-14], QoS-senstive 유니캐스트에서의 재구성 방법에 관한 연구는 활발하나[16-19], QoS멀티캐스트 트리의 링크나 라우터의 실패를 어떻게 다루어야 할 지에 대한 연구는 미약한 편이다[26-30]. 네트워크 구성요소의 실패를 효과적으로 다루는 것은 아주 중요한 문제이다. 예를 들어, Mbone과 같은 오버레이 구조에서는 IP 라우팅을 사용함으로써 터널 구성요소들의 실패는 허용되지만, 멀티캐스트 라우터 자체의 실패를 회복하기 위해서는 멀티캐스트 트리의 재구성을 필요로 한다. 하나의 라우터나 하나의 링크의 실패를 복구하기 위해 전체 멀티캐스트 트리를 재구성하는 것은, 실패의 영향을 받지 않은 통신 경로를 사용하는 그룹 멤버들까지도 서비스의 중단을 겪어야 하기 때문에 바람직하지 않다. 특히 인터넷 라우터의 수가 증가하여 라우터들의 실패 가능성이 높아지는 상황에선, 실패의 발생시 기존 멀티캐스트 트리를 효과적으로 재구성하는 것이 매우 중요하다.[1] 실패 복구를 위한 재구성은 기존 그룹 멤버들 사이에 새로운 경로를 찾는다는 점에서, 멀티캐스트 트리에 새로운 멤버를 연결하거나 탈퇴한 멤버를 제거하는 그룹 멤버쉽의 동적 변화 지원과는 다르다. 실패 복구를 위한 재구성은 손실 없는 메시지 전송을 다루는 전통적인 멀티캐스트 신뢰도 문제와도 다르다.

본 논문은 재구성에 소요되는 시간과 오버헤드를 줄이고 재구성의 성공 가능성을 최대화하기 위한 계획된 재구성(Pre-Planned Reconfiguration: PPR) 방법을 제안한다. PPR 방식에서는 각 QoS 멀티캐스트 세션을 위해 1차(primary) 트리와 백업(backup) 트리를 만들고, 1차 트리의 손상된 부분을 백업 트리를 이용하여 복구한다. 즉, 백업 트리는 1차 트리의 전송-수신 쌍들을 연결하는 재구성 경로(Reconfiguration Path: RP)들을 제공한다. 본 논문은 멀티캐스트의 1차 트리는 널리 알려진 알고리즘 중 하나를 이용하여 만들어 진다고 가정하고, (1) 백업 트리의 라우팅 정책(즉 재구성 경로를 구축하는 방법론)과 (2) 1차 트리와 백업 트리 간의(또한 백업 트리간의) 효과적인 자원 공유(resource sharing) 정책에 초점을 맞춘다.

이 논문은 다음과 같이 구성되어 있다. 2장은 이 논문이 겨냥하는 멀티캐스트 통신 모델을 소개한다. 3장에서는 본 논문에서 제안한 PPR 방식을 설명한다. 4장에서는 PPR 방식의 성능을 실험을 통해 분석한다. 5장에서 결론을 맺는다.

2. 멀티캐스트 모델

미래의 실패를 미리 계획된 재구성 경로들을 통해서 복구하는 것은 빈번한 경로의 변화나 멤버쉽의 변화가 있는 단기간의(short-lived) 멀티캐스트 세션의 연결에는 적합하지 않다. 그러나 응용분야의 자원에 대한 요구가 크거나 멀티캐스트 세션의 수가 많다면, 네트워크는 여러 멀티캐스트 세션들이 실패 복구를 위해 동시에 엄청난 경쟁을 하는 것을 막을 수 있는 계획된 재구성 경로 방식을 효과적으로 사용할 수 있다. 이 장에서는 본 논문이 제안하는 PPR 방식에 가장 적합한 대상 멀티캐스트 모델을 설명한다.

응용분야: PPR 방식은 자원에 대한 요구가 높고 비교적 긴기간동안 지속되는 응용분야에 적합하다. 예를 들어, 비디오나 오디오 회의, 원거리 학습, 주문형 비디오 서비스 등의 연속 미디어 전송이 주 대상 분야이다.

1) 인터넷 경로는 전화 서킷보다 훨씬 신뢰도가 낮다고 알려져 있다. 대부분의 인터넷 백본(backbone) 경로는 평균 25일정도의 실패를 경험한다[15].

이러한 서비스는 대부분의 텍스트 데이터나 정적 이미지 전송(예를 들어, 이메일이나 웹을 통한 정보 검색)보다 훨씬 많은 네트워크 대역폭을 요구한다. 연속 미디어 전송 세션은 비교적 긴 시간 계속되며 다른 응용분야의 세션들보다 변화가 적은 편이다. 또한 이러한 응용분야에서는 그룹 멤버쉽도 상대적으로 천천히 변화한다.

멀티캐스트 라우팅: 본 논문은 멀티캐스트 1차 트리를 구축할 때 어떤 특별한 라우팅 알고리즘을 요구하지 않는다. 다만 (4장에 있는) 성능평가를 위한 실험시에, 새 멤버들을 추가할 때마다 전체 멀티캐스트 트리를 다시 구축하는 스타이너 트리 휴리스틱대신 기존의 트리에 하나씩 멤버를 추가하는 점진적인 최단 경로 트리 방법을 사용하였다.[2] 재구성된 경로가 원 경로보다 길어지는 경향을 고려하면, 최단 경로 트리 방법은 백업 트리를 만들 때에도 적절하다고 본다. PPR 방식은 전송자 중심(source-based) 트리 라우팅과 공유(shared) 트리 라우팅 모두에 적용할 수 있다.

경로의 경직성: 본 본문은 멀티캐스트 트리가 고정된 브랜치(branch) 노드들과 두 브랜치 노드들을 연결하는 하위 경로(sub-path)를 나타내는 정적인 세그먼트(segment) 들로 구성되어 있다고 가정한다. 이러한 고정 경로 방법은 per-flow reservation-based QoS 통신 방식에서 주로 사용된다[3][22].

확장성(scalability) 및 이질성(heterogeneity): 본 논문은 연결성이 충분히 높은 네트워크를 대상으로 한다. 멀티캐스트 세션의 참가자는 지역적으로 흩어져 있으며, 각 멤버들은 네트워크에 각기 다른 용량의 링크로 연결되어 있을 수 있고, 각 멤버들의 대역폭 요구량도다를 수 있다. 수신자들의 대역폭 요구량의 이질성은 예약, 패킷 필터링, 또는 레이어드 코딩 등으로 해결된다고 가정한다[23,24]. 또한, 각 멤버들의 실패에 대한 내성이 각기 다를 수 있다고 가정한다. 어떤 멤버는 실패 복구에 대한 완벽한 보장을 요구할 수도 있고, 다른 멤버는 낮은 보장만을 요구할 수도 있다.

3. 계획 재구성(Pre-Planned Reconfiguration: PPR) 방식

제안된 PPR 방식은QoS 멀티캐스트 세션에 재구성 경로(Reconfiguration Paths: RPs)를 미리 갖추어 멀티캐스트 1차 트리에 발생하는 실패가 하나일 경우에는 언제나 복구할 수 있도록 하는 것을 목표로 한다.[4] 아래에서 간단하게 PPR 방식을 설명하고, 멀티캐스트 트리의 재구성 문제를 여러 개의 중요 이슈들로 나누어, 제안된 방식이 각 문제들을 어떻게 처리하는지 자세히 살펴본다.

3.1 개요

하나의 구성요소의 실패가 멀티캐스트 트리의 일부 경로들을 손상할 때, 네트워크는 멀티캐스트 트리의 손상된 부분을 지나지 않는 대체 경로를 찾아야 한다. 즉, "어디로"(where), "언제"(when) 대체 경로를 구축할 지를 결정해야 한다. PPR 방식에서 "어디로" 대체 경로를 구축할 지의 문제는 기존의 멀티캐스트 트리의 변화를 최소화하고 손상되지 않은 부분들의 서비스를 유지하는 방법으로 해결하고자 한다. 또한 실패에 영향을 받지않은 그룹 멤버들은 재구성에 참여하지 않도록 하여 확장성을 성취한다. "언제" 대체 경로를 구축할 지에 관한 문제는 실패가 일어나기 전에 미리 재구성 경로를 설정해두는 방법을 통하여 푼다.

따라서 네트워크는 PPR 방식을 통해 동시다발적인 재구성 시도에 의한 무절제한 경쟁을 막으면서 미리 선정된 경로를 사용하여 재구성을 시도한다. PPR 방식은 재구성 경로에 필요한 예상 자원을 예약함으로써 재구성 이후의 QoS 지원을 보장할 수 있다. 우리는 이러한 예약된 자원들을 스페어(spare) 자원이라고 부르며, QoS 지원을 위해 실제 사용하는 액티브(active) 자원과 구분한다. 계획된 재구성 경로의 선정과 자원 예약은 재구성의 성공도를 높이지만, 예약 자원들이 QoS 보장 연결을 위해 사용되지 못함으로써 실패가 없는 경우에는 용량면에서 오버헤드를 일으킨다. 따라서, 얼마나 많은 자원을 스페어 자원으로 예약할지가 중요한 이슈이다. 이 문제를 해결하기 위해 우리는 효율적인 자원 공유 정책을 제안한다. 각 링크에 필요한 스페어 자원의 양을 계산하는 분산 알고리즘은 액티브 자원과 스페어 자원 사이의 자원 공유와, 동일한 또는 서로 다른 멀티캐스트 트리의 스페어 자원 사이의 자원 공유 등을 효과적으로 지원한다.

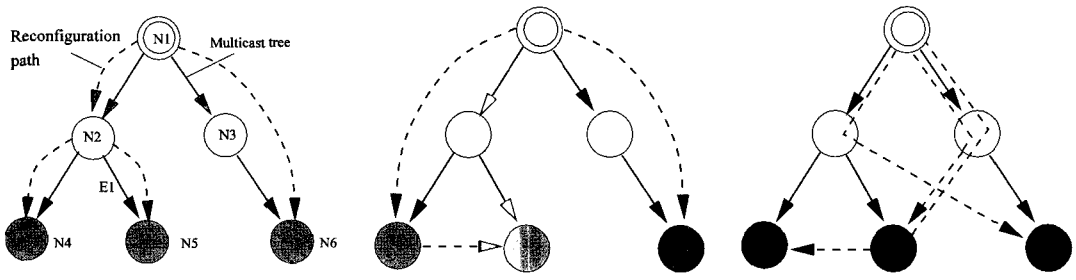
또 다른 이슈는 멀티캐스트 그룹의 동적인 변화를 어떻게 처리하는 가이다. 우리는 멀티캐스트 트리의 변화를 최소화하고자 한다. 그룹의 변화에 따른 재구성 경로의 유지에 관한 자세한 사항들은 이 장의 마지막에서 다루며, 이때 그룹 동적변화에 따른 멀티캐스트 트리의 수정이나 그룹 어드레싱과 관리 방식 등은 주어져 있다고 가정한다.

---

2) 많은 연구가 지연(delay)에 민감한 응용분야에서는 스타이너 트리 방법이 적절치 못하다고 지적하고 있다[21].

3) 한편 MBone은 고정된 브랜치 노드들과 동적 경로 세그먼트들을 사용한 터널의 형태를 활용하여, 동적 경로의 실패를 비연결(connection-less) 라우팅으로 해결한다. 따라서 본 논문이 제안하는 방식은 MBone에서의 브랜치 노드들의 실패를 다룰 때에도 사용될 수 있다.

4) 단 전송자 중심 멀티캐스트 트리에서 전송자 노드가 실패한 경우를 제외한 실패를 말한다.



(a) SD (b) TD (c) PD

그림 1 재구성 경로를 만드는 방법

3.2 재구성 경로 라우팅 방법

자원 효율성(즉, 정해진 자원 오버헤드 내에서 이룰 수 있는 fault-tolerance 레벨)과 제어 효율성(즉, 라우팅 테이블의 크기, 제어 트래픽 오버헤드 등)은 재구성 경로들을 어떻게 구축하는 지에 따라 크게 영향을 받는다. 아래에서 우리는 3가지 재구성 경로 제공 방법을 제안하고, 전송자 중심 트리의 경우와 공유 트리의 경우에서 3가지 방법을 비교해 살펴본다.

가장 단순한 방법은 그림 1(a)에서처럼 멀티캐스트1차 트리의 각 세그먼트마다 재구성 경로를 제공하는 것이다. 그림 1(a)에서 이중 원으로 표시된 노드는 전송자이고, 까만 노드들은 수신자[5]들이다. "세그먼트"는 두 브랜치(branch) 노드[6] 사이나 엔드 노드 사이의 경로(sub-path)를 말한다. 예를 들어, 그림 1(a)에는 (N1, N2), (N2, N4), (N2, N5), (N1, N6) 의 네 개의 세그먼트가 존재한다. 그림 1(a)의 재구성 경로들은 각 해당 세그먼트와 겹침 없이 디스조인트(disjoint)하게 구축되므로, 이러한 방법을 세그먼트-디스조인트(Segment Disjoint: SD)라고 부른다. SD 방식의 가장 큰 단점은 멀티캐스트 트리가 브랜치 노드의 실패로부터 보호되어 있지 않다는 데 있다.

이를 피할 수 있는 다른 방법으로는 그림 1(b)처럼 백업 트리를 각 엔드 노드들을 제외하고는 기존의 1차 트리의 링크나 노드들로부터 모두 디스조인트하게 제공하는 것이다. 이를 트리-디스조인트(Tree Disjoint: TD)라고 부른다. TD 방식은 single-point 실패로 인한 SD 방식의 취약성을 피할 수 있으나, 1차 트리와 완전히 디스조인트한 백업 트리가 존재하지 않을 수 있다는 단점이 있다. 예를 들어 만일 전송자에서 출발하는 모든 링크들이 1차 트리의 세그먼트라면, 백업 트리를 만들 수 있는 에지가 존재하지 않게 된다. 이런 상황을 피하기 위해서는 1 차 트리를 구성할 때 백업 트리 라우팅을 고려해야 하는 문제점이 있다. SD 방식은 [30]에서, TD 방식은 [28]에서, 그리고 앞으로 소개될 PD 방식은 [29] 등에서 채택되었다.

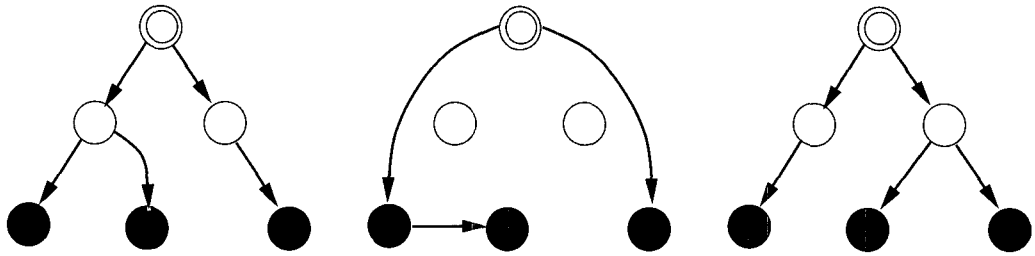
SD 방식과 TD 방식의 문제점을 해결하기 위해, PPR 방식은 두 방법의 장점을 결합한 경로-디스조인트(Path Disjoint: PD) 방식을 사용한다. PD 방식에서는 각 전송-수신 쌍에 재구성 경로를 제공한다. 즉, 재구성 경로는 기존 1차 트리의 전송-수신 쌍 사이의 '경로'와 디스조인트하게 정해진다. 이때 (TD 방식과는 달리) 재구성 경로들이 전체 1차 트리와 디스조인트할 필요가 없으므로 재구성 경로가 만들어지지 않는 경우는 거의 없다. PD 방법은 그림 1(c)에 나타나 있다. 그림 1(b)와 1(c)에서 재구성 경로 라우팅 방식과의 차이를 유의해서 보면, PD 방식에서는 재구성 경로 라우팅을 위해 1차 트리의 구성요소들을 재사용할 수 있다.

그림 1(a)에서 표시된 에지 E1의 실패가 발생했을 경우, 세가지 라우팅 방식에 의한 재구성의 결과가 그림 2에 비교되어 있다. SD와 PD 방식은 E1의 실패의 영향을 받은 수신자들을 연결하기 위해 기존 멀티캐스트 트리를 부분적으로 수정하였으나, TD 방식은 전체 멀티캐스트 트리를 바꾸었다.

각 라우팅 방식의 자원 오버헤드를 자세히 살펴보기 위해 그림 1에 있는 트리들을 최단 경로 라우팅을 이용하여 그림 3의 사각형 그리드 네트워크에 표현하였다. 모든 수신자들이 그들의 경로에 있는 각 링크에서 단위 자원만큼씩을 사용한다면, PD 방식은 5 스페어 자원을, SD 방식은 8 스페어 자원을, TD 방식은 10 스페어 자원을 필요로 한다. (그림 3에 자원 예약을 필요로 하는 링크들을 점선으로 표시해 두었다.) 재구성 경로가 1차 트리의 링크와 오버랩되는 경우에는 스페어 자원을 예약할 필요가 없음을 유의하자. 또한 다수의 재구성 경로가 동일한 링크를 지나는 경우에는 오직 하나의 스페어 자원만 예약하면 된다. 그림 3에서 보여진 것처럼, 재구성 경로들은 각 방식이 요구하는 디스조인트 제약조건이 만족하는 하에서 1차 트리의 링크를 최대한 이용하

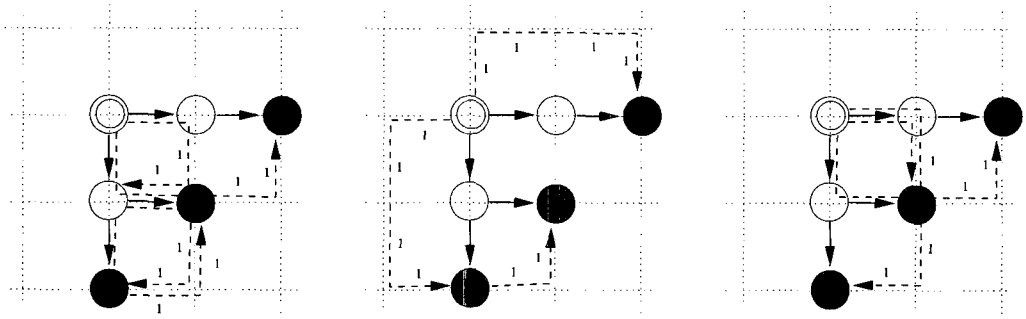
5) 수신자가 반드시 엔드노드일 필요는 없다.

6) 브랜치 노드는 트리상에서 경로가 나누어지는 지점에 위치한 노드를 지칭한다. 그림 1(a)에서 브랜치 노드는 N2이다.



(a) SD (b) TD (c) PD

그림 2 재구성의 결과



(a) SD (b) TD (c) PD

그림 3 사각형 그리드 네트워크에서의 라우팅 예

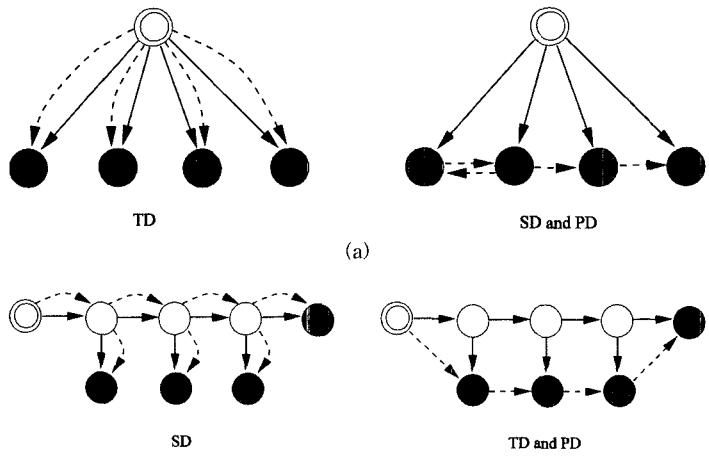


그림 4 다른 예제들

여 구축되었다.

그림 4에 보여진 것처럼, TD와 SD 방식의 상대적인 효율성은 경우에 따라 다르다. 그림 4(a)의 경우에는 SD와 PD 방식이 최소한의 스페어 자원 예약을 필요로 하며, 그림 4(b)의 경우에는 TD와 PD 방식이 최소한의 스페어 자원 예약을 요구한다. 그러나 PD 방식은 다른 두 방식보다 자원 사용면에서 항상 우수하다. 이를 간단하게 증명하면 아래와 같다.

정리 1. PD 방식은 TD나 SD 방식보다 항상 적거나 같은 양의 자원을 사용한다.

증명. PD 방식의 상대적인 우수성을 증명하기 위해서, SD와 TD 방식을 통한 재구성 경로 라우팅의 결과에 PD 방식을 적용하면 더 적은 자원을 쓰도록 수정될 수 있다는 것을 보인다. 첫째, TD 방식의 재구성 경로

구축 결과는 만일 더 적은 자원을 사용하는 경로가 존재한다면 현재 존재하는 전송-수신 쌍을 연결하는 재구성 경로대신 1차 트리의 다른 부분과 오버랩되는 경로를 선택하도록 수정함으로써 개선할 수 있다. 둘째, SD 방식의 재구성 결과는 수신자가 아닌 브랜치 노드들을 통한 재구성 경로를 허용함으로써 더 적은 자원을 사용하도록 개선할 수 있다. □

공유 트리에서의 재구성 경로를 설정하는 경우는 코어 노드의 실패를 다루어야 한다는 점을 제외하면 위에 설명된 전송자 중심 트리에서의 경우와 비슷하다. 그룹 형성 단계에서 멀티캐스트 그룹의 코어 노드가 무작위로 지정된다고 가정했을 때, 모든 그룹 참가자들은 양방향(two-way) 채널을 통해 코어 노드와 연결된다. 따라서 코어 노드의 역할은 전송자 중심 트리의 전송자와 다음의 두 가지 면을 제외하고는 동일하다. 첫째, 모든 트리 세그먼트들은 양방향이다. 둘째, 전송자 중심 트리에서 전송자 노드의 실패가 고려하지 않았던 것과는 달리, 공유 트리에서는 코어 노드의 실패도 고려하여야 한다. 코어 노드의 실패를 허용하기 위해서 그림 5처럼 백업 코어 노드를 정하고 백업 코어 노드에서부터 출발하는 백업 트리를 설정할 수도 있다. 그러나 이러한 백업 트리에는 PD 방식을 적용할 수 없고 오직 TD 방식만이 가능하며, 이는 전송자 중심 트리와는 달리 멀티캐스트의 1차 트리와 백업 트리의 루트가 다르기 때문에 확장성면에서 매우 나쁜 결과를 보이게 된다. 예를 들어 그림 5의 E2가 실패할 때 백업 코어 노드와 멤버 M1 사이의 재구성 경로만을 활성화한다면 M1이 다른 멤버들과 계속하여 격리되므로, 전체 멀티캐스크 트리를 백업 트리로 바꾸어야 한다. 따라서 우리는 별개의 백업 코어 노드를 사용하지 않기로 한다. 이는 백업 코어를 사용하지 않음으로써 코어 노드의 실패로 인한 single-node 실패를 감수하는 대신 효율적인 PD 방식을 공유 트리에 적용하고자 함이다.

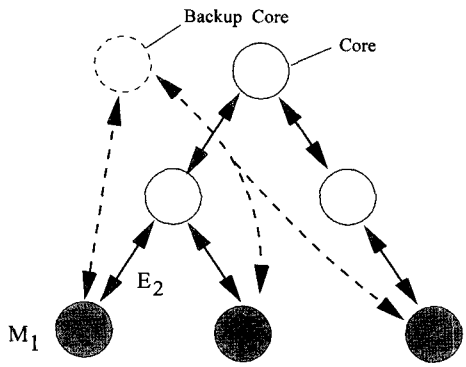


그림 5 공유 트리에서 백업 코어를 이용한 재구성

3.3 자원 공유(Resource Sharing) 정책

일반적으로, 멀티캐스트에서의 자원 공유는 메시지 전송의 레벨에서 고려된다. 예를 들어, [25]에서 제안된 자원 공유 방식에서는 한 순간에 대화할 수 있는 회의 참가자들의 수를 제한하고, 참가자들이 공유 자원을 사용하여 각기 다른 시간대에 메시지들을 전송하도록 한다. 이는 트래픽 전송자들 사이의 자원 사용에 관한 명시적인 약속을 활용한다는 점에서 트래픽 전송의 무작위적인 burstiness를 활용하는 통계적 멀티플렉싱과는 다르다. 모든 공유 트리 방법들은 위와 유사한 자원 공유 방식을 사용한다.

본 논문에서는 예약하는 스페어 자원의 양을 줄이는 2가지 자원 공유 방식을 제안한다. 첫번째 자원 공유 방식은 하나의 멀티캐스트 트리상에서 자원 예약사이의 관계를 최대한 활용한다. 재구성 경로가 1차 트리의 링크와 오버랩이 있는 경우에는 오버랩된 링크에서는 스페어 자원을 예약할 필요가 없다. 보다 정확히 말하자면, 재구성 경로의 대역폭 요구가 1차 트리를 위해 지정해 둔 대역폭보다 크지 않는 경우에는, 오버랩된 링크에 스페어 자원을 예약하지 않으며, 재구성 경로의 대역폭 요구가 1차 트리를 위해 지정해 둔 대역폭보다 큰 경우에는 둘의 차이를 메꿀 수 있는 정도만큼만의 스페어 자원을 예약한다. 비슷한 논리가 동일한 멀티캐스트상의 다수의 재구성 경로들이 같은 링크를 지날 때에도 적용된다. 따라서 디스조인트 제약조건이 만족하는 한, 1차 트리의 링크와 오버랩이 되도록 또는 다른 재구성 경로들과 오버랩이 되도록 재구성 경로를 라우트하는 것이 좋다. 다만 1차 트리나 다른 재구성 경로들과의 오버랩을 최대화하고자 함에 따라 재구성 경로의 길이가 길어질 수 있으므로, 경로 지연(delay) 요구조건을 만족시키는 것을 유의하여야 한다.

두번째 형태의 자원 공유는 서로 다른 멀티캐스트 트리들에서 사용되는 재구성 경로들 사이에서 일어난다. 기본 아이디어는 두 개의 멀티캐스트 트리가 동시에 실패하지 않는다면, 오버랩하는 재구성 경로들의 스페어 자원을 각각의 재구성 경로가 공유할 수 있다는 것이다. 예를 들어, 각 멀티캐스트 트리에서 하나의 구성요소 실패를 가정한다면, 두개의 멀티캐스트 1차 트리가 서로 오버랩이 없을 경우에는 두 멀티캐스트 트리의 재구성 경로들 사이에서는 자원을 공유할 수 있다. TD방식에서는 전체 1차 트리들을 비교하여 자원 공유를 허락할지를 결정하면 된다. PD 방식에서는 실패시 다른 멤버들은 원래 경로를 그대로 사용하고 연결이 끊어진 멤버들 사이의 재구성 경로들만이 활성화(activate)된다. 따라서 두 재구성 경로들 사이에서 자원 공유를 허락할지를 결정할 때, 전체 1차 트리들의 오버랩을 보는 대신 각 1차

트리의 해당 경로들이 오버랩하는지를 보는 것으로 충분하다. 비슷한 이유로 SD 방식에서는 세그먼트의 오버랩을 확인하도록 한다. 즉 오버랩이 있는지를 검사할 때 작은 단위를 사용하는 것은 더 많은 자원 공유를 가능하게 한다.

우리는 [16]에서 제안된 자원 공유 방식을 기본 틀로 삼고, 이를 멀티캐스트 환경에 적용하였다. 네트워크 구성요소의 실패 확률을 어떤 정해진 값으로 가정하고, 동일한 링크를 사용하는 두 개의 재구성 경로가 동시에 활성화될 확률을 구한다. 이 확률을 응용분야에서 요구하는 신뢰도를 반영하는 임계치(threshold)와 비교하여, 동일한 링크위에 멀티플렉싱할 재구성 경로들의 집합을 결정한다. 각 재구성 경로 링크의 스페어 자원 필요량은 재구성 경로들의 자원 요구들을 모두 합하여 그들중 가장 큰 값을 사용한다. 이때 위에 설명된 자원 공유 방법을 고려한다. 즉 1차 트리의 링크를 사용하는 재구성 경로들은 제외하며, 각 트리에서는 가장 큰 대역폭을 요구하는 재구성 경로만을 고려한다. 그림 6은 PD 방식에서 링크 l의 스페어 자원의 양을 계산하는 알고리즘을 보여준다. 자원 공유의 단위가 세그먼트와 트리라는 점을 제외하고는 SD와 TD 방식을 위한 알고리즘도 동일하다. 그림 6의 알고리즘은 수신자들간의 다양한 대역폭 요구도 처리할 수 있다. 한편 수신자들의 다양한 신뢰도 요구는 두 레벨에서 처리할 수 있다. 첫째, fault-tolerance를 요구하지 않는 수신자들에게는 재구성 경로를 제공하지 않는다. 둘째, 멀티플렉싱의 여부를 결정할 때 서로 다른 임계치 값을 사용함으로써, 낮은 fault-tolerance 보장을 요구하는 수신자들을 위한 재구성 경로들은 적은 양의 스페어 자원을 사용하도록 한다.

3.4 라우팅 최적성(optimality)와 그룹 동적변화

QoS 멀티캐스트 라우팅의 최적성은 여러 각도에서 정의할 수 있다. 첫째, 최적성은 멀티캐스트 트리를 만드는데 사용된 자원의 양으로 평가할 수 있다. 둘째, 각 전송-수신 쌍들 사이의 end-to-end 지연이 응용분야에서 요구되는 지연 정도보다 적은지를 알아보는 것도 하나의 방법이다. 세째, 같은 그룹내의 수신자들 사이의 end-to-end 지연의 분산(variance)이 기준이 될 수도 있다. 새로운 멤버들이 추가되고 기존의 멤버들이 사라지는 경우에는 라우팅의 최적성을 유지하는 것은 매우 어려운 문제이며, 대부분의 경우 전체 멀티캐스트 트리의 재구성을 필요로 한다. 본 논문에서는 멤버쉽의 변화에 따른 1차 트리의 재구축의 문제는 다루지 않고, 멤버쉽 변화에 따라 재구성 경로들을 어떻게 구축할지의 문제를 다룬다.

재구성 경로 라우팅의 최적성 또한 위와 비슷하게 (예를 들어, 스페어 자원의 양이나 응용분야에서 요구하는 지연 정도의 만족 등) 정의될 수 있다. 그러나 재구성 경로 라우팅은 그룹 동적변화뿐만 아니라 자원 공유

```
Max ← 0;
for each reconfiguration path RPi on link l
{
        Sum, X[ ] ← 0;
        /* Ignore RPs which do not need spare resources on l. */
        if the multicast tree of RPi runs over l
                continue;
        /* Consider highest bandwidth requirement for each tree under consideration. */
        for each RPj on l,which is not multiplexable with RPi
        {
                If X[tree of RPj] < bwj          /* bwj is the bandwidth requirement of RPj */
                        X[tree of RPj] ← bwj;
        }
        /* Calculate sum of bandwidth requirement by non-multiplexable RPs, and select
         maximum value among the sums for all RPs on l. */
        Sum ← Σ X[ ];
        if Sum > Max
                Max ← Sum;
}
Sl ← Max;                        /* Sl is the amount of spare resources to be reserved at l. */
```

그림 6 PD 방식을 위한 스페어 자원의 양을 계산하는 알고리즘

를 고려하여야 하기 때문에 더욱 어렵다. 멀티캐스트 멤버쉽이 변하지 않는 경우에도 다른 멀티캐스트 트리와의 상호작용때문에 하나의 멀티캐스트 트리에 대한 자원 필요량이 변할 수 있다. 즉 재구성 경로 라우팅의 최적성은 자신의 그룹뿐만 아니라 다른 그룹에 의해서 영향을 받을 수 있다.

우리는 새로운 재구성 경로를 만들때 최적성을 고려하지 않는 "greedy" 최단 경로 라우팅을 사용한다. Greedy 방식은 멀티캐스트 트리의 현존하는 링크들을 최대한 이용하여 새 멤버를 위한 재구성 경로를 만들어 준다. 이 방법은 멀티캐스트 트리상에 있지 않은 링크들 사이의 자원 공유를 고려하지 않기 때문에 스페어 자원의 최소 사용을 보장하지는 못한다. 멀티캐스트 그룹의 멤버가 탈퇴할 경우에는, 탈퇴 멤버를 위한 1차 경로와 재구성 경로를 둘 다 제거한다. 이때 (1차 트리와 오버랩하는 재구성 경로들은 스페어 자원을 예약해 두지 않기 때문에) 탈퇴하는 멤버들에 의한 1차 트리의 변화가 동일한 멀티캐스트 그룹의 재구성 경로에 필요한 스페어 자원의 부족을 야기할 수 있다. 따라서 이러한 링크를 위해서는 추가적인 스페어 자원을 할당받거나 새로운 재구성 경로를 만들어 주어야 하므로, 1차 트리와 링크를 공유하도록 라우팅했던 재구성 경로의 장점이 사라지며, 새로 만들어진 경로의 길이가 길어져 자원 사용에 비효율성이 생길 수 있다. 따라서 그룹 동적변화는 (특히 멤버의 탈퇴는) greedy 방법의 성능을 저하시킬 수 있다. 이러한 비효율성을 제거하기 위해 재구성 경로들을 다시 만들 수 있으며, 이러한 작업은 현재 제공되는 멀티캐스트 서비스에 영향을 미치지 않고 배후에서 할 수 있다.

4. 성능 평가

이 장에서는 체계적이고 광범위한 시뮬레이션을 통하여 PPR 방식의 single-node fault tolerance 성능을 평가한다. 시뮬레이션은 다양한 네트워크 조건하에서 세가지 방법의 자원 효율성을 비교하는 데 초점을 맞추고 있다.

4.1 시뮬레이션 환경

시뮬레이션 토폴로지는 [20]에서 사용된 방법을 이용하여 무작위로 생성하였다. 이러한 네트워크에서는 가까운 노드들이 서로 연결될 확률이 높으며, 두 이웃 노드들 사이의 연결은 (각 방향으로 하나씩) 두개의 같은 용량의 링크들로 연결된다. 에지의 길이와 노드의 디그리를 조절하기 위해, α와 β라는 두 파라미터를 사용한다. α가 크면 노드를 연결하는 에지들이 길고, β가 크면 평균 노드 디그리가 증가한다.

분리된(partitioned) 그래프를 피하고 재구성 경로가 생성될 수 있는 최소한의 조건을 만족시키기 위해, 기존의 방법을 수정하여 모든 노드들이 적어도 두개의 에지를 갖도록 하였다. 저밀도(sparse) 네트워크와 고밀도(dense) 네트워크를 각 경우마다 5개씩의 네트워크를 생성하여 실험하였다. 노드의 갯수는 49개로, 각 링크의 용량은 100 Mbps로 정하였다. 실험에 사용된 네트워크 토폴로지의 특성은 표 1에 정리되어 있다.

고밀도 네트워크는 3,000명의 참가자가, 저밀도 네트워크는 1,500명의 참가자가 존재한다. 그룹 크기의 영향을 조사하기 위해 다양한 그룹 크기를 실험하였다. 여기에서 그룹의 크기는 전송자 중심 트리에서는 수신자들의 수이며, 공유 트리에서는 그룹 멤버의 수이다. 예를 들어, 고밀도 네트워크에서 그룹 사이즈가 30이면, 멀티캐스트 트리의 갯수는 100개가 된다. 전송자 중심 트리에서는 각 에지가 1 Mbps 대역폭을 요구하며, 공유 트리에서는 각 에지가 0.5 Mbps 대역폭을 요구한다고 가정하였다.

전송자 중심 트리와 공유 트리 모두에서 최단 경로 라우팅을 통해 멀티캐스트 1차 트리를 만든다. 멀티캐스트 1차 트리는 하나씩 점진적으로 만들어지며, 각 멀티캐스트 트리의 멤버들도 하나씩 추가된다. 각 트리의 재구성 경로는 1차 트리가 만들어진 이후에 생성되며, 앞에서 설명한 재구성 경로와 1차 트리 사이의 자원 공유 방식을 적용하였다. 이때 재구성 경로의 길이가 대응하는 멀티캐스트 트리의 경로의 길이보다 2 홉(hop)을 초과하지 않는다면 경로 지연 조건이 만족되었다고 가정한다.

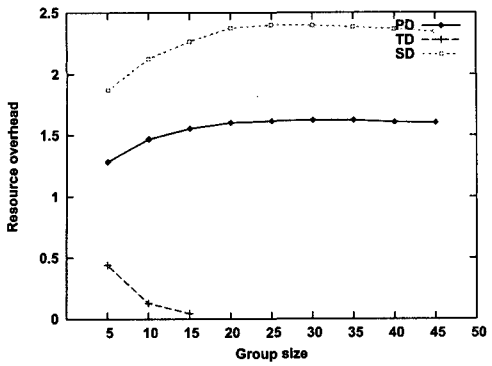
4.2 성능 비교

이 절에서는 다른 트리의 재구성 경로와의 자원 공유(멀티플렉싱)을 고려하지 않은 경우하에 SD, TD, PD 방식의 성능을 비교하였다. 멀티캐스트 트리 간의 멀티플렉싱의 영향은 다음 절에서 분석한다. 그룹의 크기를 5 노드 그룹에서 45 노드 그룹까지 (즉, 전체 네트워크 노드의 10%에서 90%까지) 변화시키면서 비교해 보았다. 이때 4가지의 성능 기준을 이용하였다.

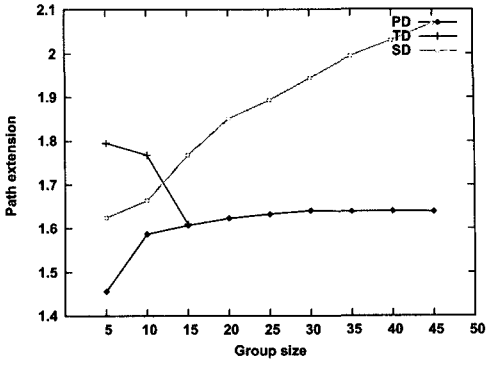
•

표 1 시뮬레이션 네트워크의 특성

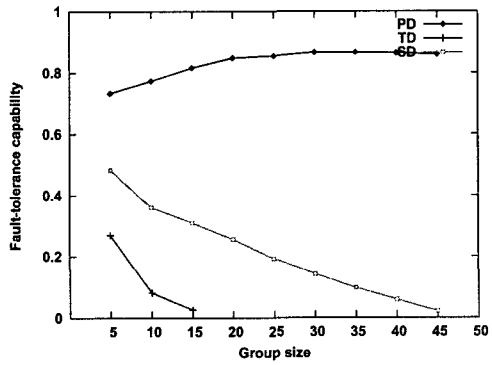
	(α, β)	Nodes	Mean degree	Mean diameter	Mean path length
Sparse	(0.15, 0.3)	49	2.77	8.4	4.13
Dense	(0.2, 0.45)	49	4.0	6.2	2.98



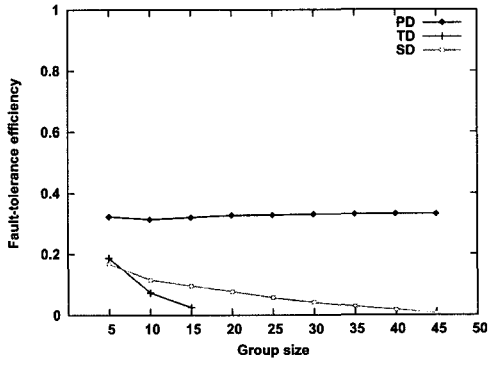
(a) 자원 오버헤드



(b) 경로 연장



(c) fault-tolerance 능력



(d) fault-tolerance 효율성

그림 7 저밀도 네트워크에서의 실험 결과

- 자원 오버헤드 $= \frac{\text{스페어 자원}}{\text{액티브 자원}}$
- 경로 연장 $(extension) = \frac{\text{재구성 경로의 경로 길이}}{\text{원 경로의 경로 길이}}$
- $Fault-tolerance$ 능력 $(capability) =$

$$\frac{\text{재구성 경로에 의해 복구된 멤버의 수}}{\text{하나의 실패에 의해 연결이 끊어진 멤버의 수}}$$

- $Fault-tolerance$ 효율성 $(efficiency) =$

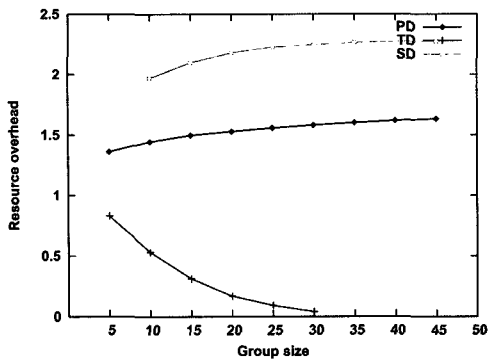
$$fault-tolerance \text{ 능력} \times \frac{\text{액티브 자원}}{\text{액티브 자원} + \text{스페어 자원}}$$

자원 오버헤드는 1보다 크거나 작을 수 있다. 자원의 부족, 디스조인트 제약조건, 경로 지연 제약조건 등에 의해 재구성 경로를 만들 수 없는 경우에는 fault-tolerance를 완벽하게 보장 할 수 없게 된다. SD 방식의 경우에는 브랜치 노드의 실패로부터 보호할 길이 없다는 점을 유의하자. Fault-tolerance 능력을 측정할 때에는 전송자나 수신자의 실패는 고려하지 않고 전송-수신 사이의 노드들의 fault-tolerance만 측정한다. 1이라는 효율성은 여유 자원을 하나도 사용하지 않고 완벽한 fault-tolerance를 이루는 것이므로, fault-tolerance 효율성은 항상 1보다 작다.

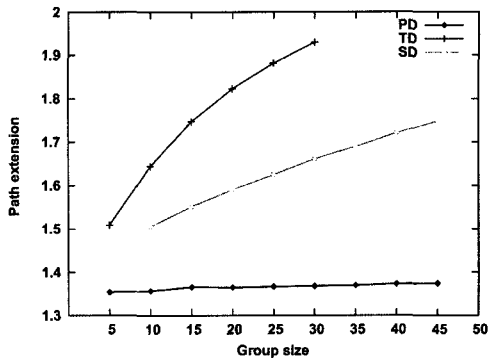
저밀도 네트워크에서의 실험 결과는 그림 7에, 고밀도 네트워크에서의 결과는 그림 8에 있다. 그래프상의 각 점은 50번의 시뮬레이션을 통한 결과이며, 그래프에 나타나지 않은 점들은 자원의 부족으로 인해 하나의 트리도 라우트하지 못한 것을 의미한다.[7] 실험에 의하면 그룹의 크기나 네트워크의 연결도와 상관없이, PD 방식이 항상 다른 방식들보다 우수한 결과를 보여준다. TD 방식은 작은 크기의 그룹에서만 적용이 가능하며, 작은 그룹에서도 경로 길이가 길어지는 단점을 보여준다. SD 방식은 자원 오버헤드가 높고 fault-tolerance가 낮아, 결과적으로 낮은 fault-tolerance 효율성을 보여준다.

실험을 통해 여러가지 관찰이 가능하다. 첫째, 고밀도 네트워크에서는 재구성 경로를 라우팅하는 짧은 경로들이 존재하므로, PD 방식의 성능은 저밀도 네트워크보다

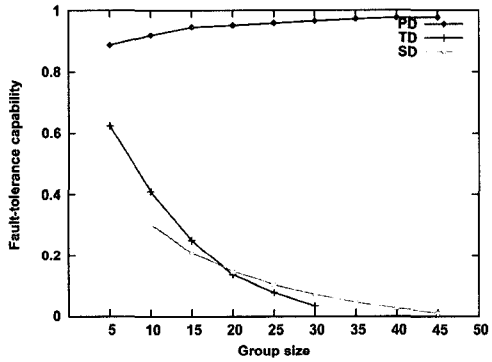
7) 하나의 트리를 라우트하는 데 실패했을 때는 네트워크이 꽉 찰 때까지 (saturate) 다른 트리들을 라우트하려고 시도한다.



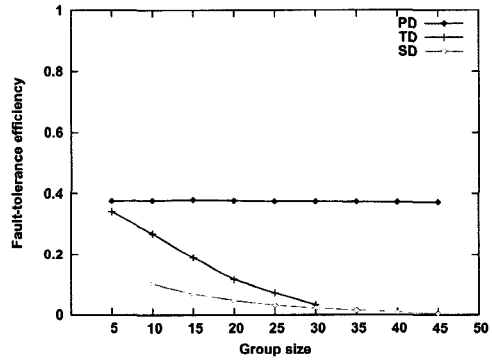
(a) 자원 오버헤드



(b) 경로 연장



(c) fault-tolerance 능력



(d) fault-tolerance 효율성

그림 8 고밀도 네트워크에서의 실험 결과

고밀도 네트워크에서 더 우수하다. 이는 경로 길이의 차이를 비교해 보면 알 수 있다. 둘째, 저밀도 네트워크에서 재구성 경로를 구성하지 못하는 가장 큰 이유는 경로 지연 제약조건 때문이다. 세째, 저밀도와 고밀도 네트워크 모두에서 fault-tolerance는 그룹 크기가 커지면 약간 개선된다. 네째, 큰 사이즈의 그룹에서는 더 많은 참가자들이 같은 자원을 공유하게 되므로, 재구성 경로를 라우트하는데 사용되는 자원이 더 많이 남아있게 되어 자원 부족으로 인한 문제가 발생하는 경우를 줄인다.

4.3 트리 간 자원공유의 영향

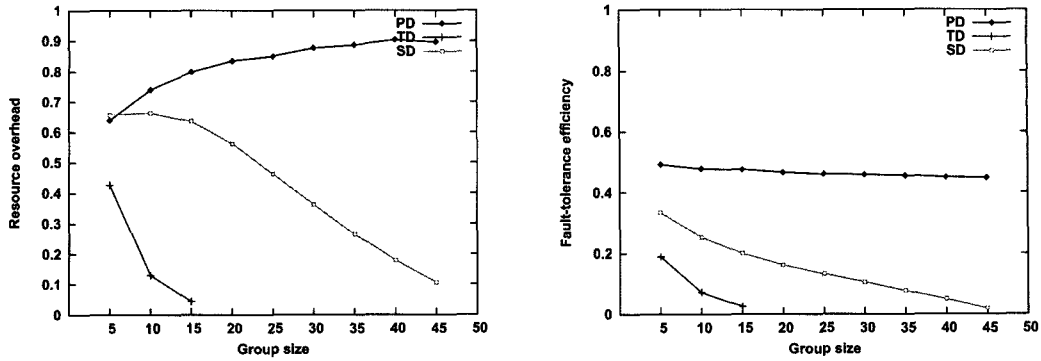
이 절에서는 다른 그룹의 재구성 경로와의 자원 공유(멀티플렉싱)의 영향을 살펴본다. 멀티플렉싱에 사용되는 임계치(threshold)의 값은 fault-tolerance 능력을 손상하지 않도록 신중하게 선정되었다. 그림 9와 그림 10은 각 재구성 경로 방식에 멀티플렉싱을 적용한 경우의 시뮬레이션 결과를 보여준다. 경로 길이와 fault-tolerance 능력에 관한 그래프는 앞 절의 결과가 거의 동일하므로 생략하였다. Fault-tolerance 능력은 멀티플렉싱을 적용한 경우, 재구성 경로를 라우트하는데 사용된 스페어 자원양의 감소로 작은 그룹 크기에서는 약간 향상되었다. (이는 멀티플렉싱 이전에는 작은 그룹의 경우 자원 부족이 일어났던 것과 비교된다.)

멀티플렉싱의 결과, TD 방식을 제외하고는자원 오버헤드가 눈에 띄게 줄어들었다. PD의 자원 오버헤드는 3배이상으로 줄었음을 알 수 있다. 멀티플렉싱의 단위가 전체 트리인 TD의 경우에는 멀티플렉싱이 전혀 효과적이지 못하다. SD 방식에서의 자원 오버헤드의 감소는 fault-tolerance 효율성을 개선하나 fault-tolerance 능력이 너무 낮아서, SD 방식은 좋은 대안이 되지 못한다.

PD 방식에서 그룹의 크기가 커지면 멀티플렉싱으로 인한 효과가 감소하는 것처럼 보인다. 이는 그룹이 커질수록 트리의 갯수가 줄어 다른 트리에 있는 재구성 경로와의 자원 공유 가능성이 줄기 때문이다. 즉, 멀티플렉싱의 효과가 감소한 이유는 그룹의 크기때문이 아니라 트리의 갯수때문이다. 트리의 갯수가 변하지 않는 경우에는 멀티플렉싱의 효과가 계속될 것이며, 트리가 많아질수록 멀티플렉싱의 효과는 더 높다.

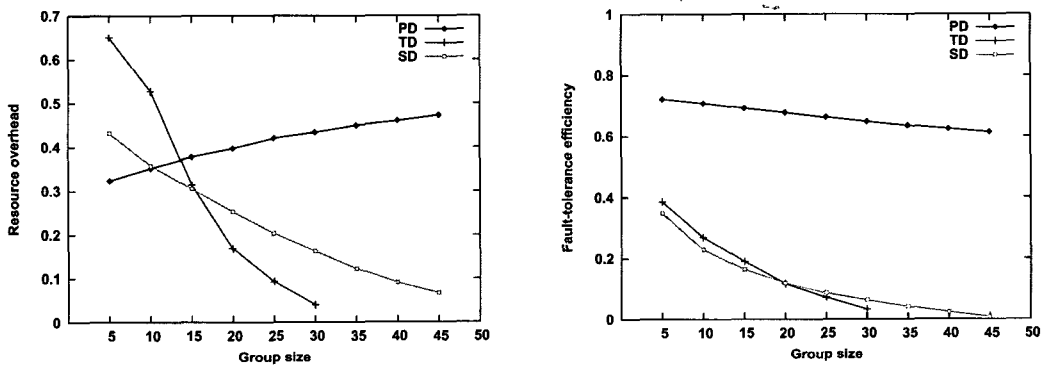
4.4 공유 트리의 경우

전송자 중심 트리의 전송자와 수신자를 라우팅하는 방법을 공유 트리의 코어 노드와 그룹 멤버를 연결하는



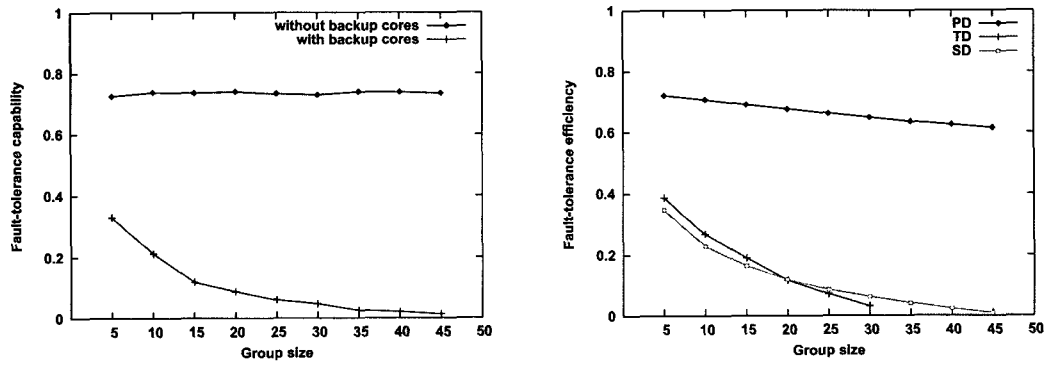
(a) 자원 오버헤드 (b) fault-tolerance 효율성

그림 9 저밀도 네트워크에서 멀티플렉싱후의 실험 결과



(a) 자원 오버헤드 (b) fault-tolerance 효율성

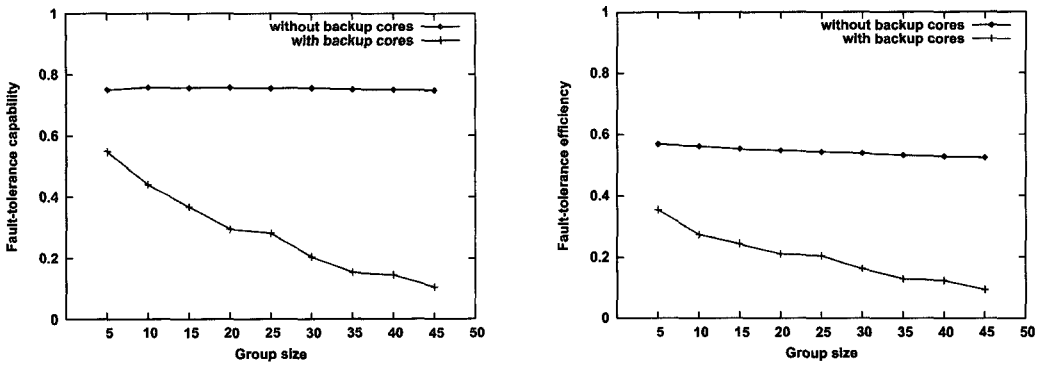
그림 10 고밀도 네트워크에서 멀티플렉싱후의 실험 결과



(a) 자원 오버헤드 (b) fault-tolerance 효율성

그림 11 저밀도 네트워크에서 공유 트리의 경우

데 사용하였기 때문에, 공유 트리에 대한 실험은 낮아진 fault-tolerance 능력을 제외하고는 앞의 실험과 매우 비슷한 결과를 보여준다. 전송자 중심 트리보다 fault-tolerance 능력이 낮아진 이유는 코어 노드의 실패가 fault-tolerance 능력을 저하시키기 때문이다. 백업 코어를 사용하면 이를 방지할 수는 있으나, 이는 자원 사용의 효율을 떨어트리며 재구성 경로의 라우팅을 막는 결과를 가져온다. 그림 11과 그림 12는 백업 코어가 있는 경우와 없는 경우를 비교한다. 백업 코어를 사용하면 fault-tolerance 효율성을 낮출뿐 아니라 fault-tolerance 능력 또한 나빠진다. fault-tolerance 능력은 그룹이 커질수록 낮아지는데, 이는 큰 트리의 백업 코어에서



(a) 자원 오버헤드 (b) fault-tolerance 효율성

그림 12 고밀도 네트워크에서 공유 트리의 경우

부터 출발하는 재구성 경로들을 만드는 것이 어렵다는 것을 의미한다.

4.5 그룹 동적변화의 영향

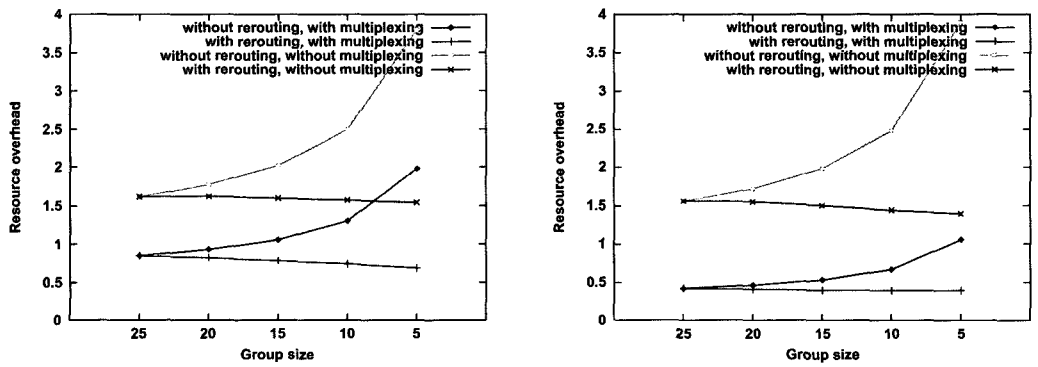
그룹 동적변화의 영향은 멤버의 가입과 탈퇴를 고려하여야 한다. 이제까지의 실험에서 각 트리에 멤버를 하나씩 추가하였으므로 멤버의 가입에 따른 성능은 이미 조사되었다. 따라서 이 절에서는 멤버의 탈퇴에 따른 성능을 조사하는데 초점을 두었다. 멤버가 그룹에 가입할 때마다 각 멤버의 재구성 경로가 하나씩 만들어지므로, 멤버의 탈퇴 순서를 가입 순서와 반대로 하면 탈퇴에 따른 성능의 변화를 측정할 수 없다. 우리는 멤버가 그룹에 가입한 순서대로 탈퇴하는 최악의 경우를 실험하였다. 그림 13은 그룹의 크기가 25인 경우에 자원 오버헤드의 증가를 보여준다. 멤버의 탈퇴시 재구성 경로들을 다시 라우트하면 자원 사용의 비효율성을 제거할 수 있음을 알 수 있다. 또한 멀티플렉싱을 사용하면 재구성 경로를 다시 라우트하는 경우와 다시 라우트하지 않는 경우의 차이를 줄일 수 있음을 알 수 있다.

5. 결 론

본 논문은 인터넷의 QoS 멀티캐스트 서비스의 신뢰도 보장을 위한 새로운 방법을 제안하였다. 계획된 재구성 정책을 통한 PPR 방식은 실패시 사용할 경로를 미리 예약해 둔다는 점에서 전화 네트워크의 높은 신뢰도 요구를 만족하기 위한 해결책과 약간의 유사성을 찾을 수 있다. 그러나 우리의 PPR 방식은 인터넷 QoS 멀티캐스트 특성에 맞도록 디자인되었으며, 용량의 오버헤드를 줄이기 위한 독창적인 자원 공유 방식을 사용하였다. 시뮬레이션을 통해 최단경로 라우팅을 사용하는 전송자 중심 멀티캐스트 트리와 공유 멀티캐스트 트리에서 성능을 평가한 결과, PPR 방식은 적당한 오버헤드내에서 모든 그룹 멤버들에게 성공적인 재구성을 제공한다. 또한 PPR 방식은 그룹 멤버쉽이 동적으로 변화할 때에도 잘 적응한다.

참 고 문 헌

[1] P. Winter, "Steiner problem in networks: A



(a) 저밀도 네트워크 (b) 고밀도 네트워크

그림 13 라우팅 재시도후의 자원 오버헤드의 감소

survey," *Networks*, vol. 17, no. 2, pp. 129-167, 1987.
[2] V. Rayward-Smith and A. Clare, "On finding steiner vertices," *Networks*, vol. 16, no. 3, pp. 283-294, 1986.
[3] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees in graphs," *Acta Informatica*, vol. 15, pp. 141-145, 1981.
[4] K. Bharath-Kumar and J. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Communications*, vol. 31, pp. 343-353, 1983.
[5] S. Deering, C. Patridge, and D. Waitzman, "Distance vector multicast routing protocol," Intenet RFC 1075, November 1988.
[6] J. Moy, "Multicast routing extensions for OSPF," *Communications of the ACM*, vol. 37, no. 8, pp. 61-66, August 1994.
[7] M. Doar and I. Leslie, "How bad is naive multicast routing?," in *Proc. IEEE INFOCOM*, pp. 82-89, 1993.
[8] A. Ballardie, P. Francis, and J. Crowcroft, "Core based tree: An architecture for scalable inter-domain multicast routing," in *Proc. ACM SIGCOMM*, pp. 85-95, 1993.
[9] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jaconson, C. Liu, F. Sharma, and L. Wei, "Protocol independent multicast-sparse mode (pim-sm): Protocol specification," Intenet-Draft: RFC 2117, 1997.
[10] H. Eriksson, "MBONE: The multicast backbone," *Communications of the ACM*, vol. 37, no. 8, pp. 54-60, August 1994.
[11] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: A survey of protocols, functions, and mechanisms," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 277-290, April 1997.
[12] K. Carlberg and J. Crowcroft, "Building shared trees using a one-to-many joining mechanism," *ACM Computer Communication Review*, vol. 27, no. 1, pp. 5-11, 1997.
[13] N. Maxemchuk, "Video distribution on multicast networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 357-372, April 1997.
[14] M. Faloutsos, A. Banerjea, and R. Pankaj, "QoSMIC: Quality of Service sensitive Multicast Internet protoCol," in *Proc. ACM SIGCOMM*, pp. 144-153, 1998.
[15] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of Internet stability and wide-area backbone failures," Technical Report CSE-TR-382-98, University of Michigan, 1998.
[16] *Reference omitted to preserve anonymity.*
[17] A. Banerjea, C. Parris, and D. Ferrari, "Recovering guaranteed performance service connections from single and multiple faults," in *Proc. IEEE Globecom*, pp. 194-205, 1994.
[18] A. Banerjea, "Simulation study of the capacity effects of dispersity routing for fault tolerant realtime channels," in *Proc. ACM SIGCOMM*, pp. 194-205, 1996.
[19] C. Dovrolis and P. Ramanathan, "Resource aggregation for fault tolerance in intergrated services networks," *ACM Computer Communication Review*, vol. 28, no. 2, pp. 39-53, 1998.
[20] B. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617-1622, December 1988.
[21] H. Salama, D. Reeves, and Y. Viniotis, "Evaluation of multicast routing algorithms for real-time communication on high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 332-345, April 1997.
[22] C. M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne, "Real-time communication in packet-switched networks," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 122-139, January 1994.
[23] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Network*, pp. 8-18, September 1993.
[24] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in Proc. ACM SIGCOMM, pp. 117-130, 1996.
[25] A. Gupta, W. Howe, M. Moran, and Q. Nguyen, "Resource sharing for multi-party real-time communication," in *Proc. IEEE INFOCOM*, pp. 1230-1237, 1995.
[26] J. Cui, M. Faloutsos, and M, Gerla, "An architecture for scalable, efficient and fast fault-tolerant multicast provisioning," IEEE Network, vol. 18, no. 6, pp. 26-34, 2004.
[27] M. Kodialalam and T. Lakshman, "Dynamic routing of bandwidth guaranteed multicasts with failure backup," IEEE ICNP, pp. 259-268, 2002.
[28] A. Fei, J. Cui, M. Gerla, and D. Cavendish, "A dual-tree scheme for fault-tolerant multicast," IEEE ICC, 2001.
[29] O. Banimelhem, A. Agarwal, and J. Atwood, "A Tree Division approach to support local failure recovery for multicasting in MPLS networks," IEEE ICW, 2005.
[30] C. Wu, S. Lee, and Y, Hou, "Backup VP Preplanning Stragtegies for survivable multicast ATM Neworks," IEEE ICC, 1997.

한 승 재

1989년 서울대학교 컴퓨터공학과(학사) 1991년 서울대학교 컴퓨터공학과(석사) 1998년 Univ. of Michigan, Ann Arbor, CSE (Ph.D.). 1999년~2005년 Bell Laboratories, Wireless Research Lab (연구원). 2005년~현재, 연세대학교 컴퓨터과학과(부교수). 관심분야는 차세대 이동통신망, 내장형 시스템, 멀티미디어 네트워킹, 센서네트워크

박 선 주

1989년 서울대학교 컴퓨터공학과 (학사) 1991년 서울대학교 컴퓨터공학과 (석사) 1999년 Univ. of Michigan, Ann Arbor, CSE (Ph.D.). 1999년~2005년 Rutgers University, MSIS Department(Assistant Professor). 2005년~현재 연세대학교 경영학과(조교수). 관심분야는 에이전트 시스템, 전자상거래, 옥션, 인공지능, SCM, 네트워크 가격정책