

대역폭 제약 그룹 멀티캐스트를 위한 다중 트리 구성 알고리즘

정회원 구 봉 규*, 박 태 근**, 김 치 하**

An Algorithm of Constructing Multiple Tree for Group Multicast with Bandwidth Constraint

Bong-Gyu Koo*, Taekeun Park**, Cheeha Kim** *Regular Members*

요 약

그룹 멀티캐스트는 그룹에 속한 모든 멤버들이 그룹 전체에 데이터를 송신할 수 있는 다자간 통신 방법이며, 그룹 멀티캐스트를 위한 라우팅 알고리즘들은 QoS (예: 대역폭) 제약 조건을 만족하면서 모든 그룹 멤버를 포함하는 멀티캐스트 트리의 구성을 목적으로 하고 있다. 현재까지 제안된 방법들로 소스 트리 구성 방법과 공유 트리 구성 방법이 있는데, 소스 트리 구성 방법은 그룹 크기에 따른 높은 트리 관리 오버헤드와 낮은 확장성 문제를 가지고 있고 공유 트리 구성 방법은 트리 구성 성공률이 낮다는 단점을 가지고 있다. 본 논문에서는 그룹 멀티캐스트를 위하여 구성되는 트리의 수가 망 부하에 따라 변화하는 휴리스틱 알고리즘을 제안한다. 시뮬레이션을 통하여, 제안하는 알고리즘이 공유 트리 구성 방법에 비하여 월등히 높은 수준의 트리 구성 성공률을 보장할 뿐만 아니라, 소스 트리 구성 방법에 비하여 현저히 낮은 트리 관리 오버헤드로 동일한 수준의 트리 구성 성공률을 보장함을 확인할 수 있다.

Key Words : Group Multicast, Source Tree, Shared Tree, Hybrid Algorithm

ABSTRACT

Group multicast refers to the kind of multicast in which every member of a group is allowed to transmit data to the group. The goal of routing algorithms for group multicast is to construct a set of low cost multicast trees including all the group members with QoS (e.g., bandwidth) constraint. There have been several algorithms proposed: source tree and shared tree approaches. However, the latter approach has a low success rate in constructing a shared multicast tree, and the former approach suffers from high control overhead and low scalability as group size increases. In this paper, we present a heuristic algorithm which varies the number of multicast trees according to the network load. The simulation results show not only that our algorithm outperforms the shared tree approach in terms of the success rate, but also that it has lower control overhead than the source tree approach while guaranteeing the same success rate.

1. 서 론

그룹 멀티캐스트는 그룹에 속한 모든 멤버들이 그룹 전체에 데이터를 송신할 수 있는 다자간 통신 방법으로, 원격 화상 회의, 분산 데이터베이스, 분산 시뮬레이션 등의 응용 프로그램들이 그룹 멀티캐스트 응용 프로그램 범주에 속한다^[1]. 이러한 그룹 멀티캐스트에서는 그룹 크기와 동일한 수의 송신자가 존재하기 때문에 하나의 송신자만 존재하는 멀티캐스트에 비하여 높은 대역폭을 필요로 하는데, 그룹 멀티캐스트 문제에 대하여

선 등의 응용 프로그램들이 그룹 멀티캐스트 응용 프로그램 범주에 속한다^[1]. 이러한 그룹 멀티캐스트에서는 그룹 크기와 동일한 수의 송신자가 존재하기 때문에 하나의 송신자만 존재하는 멀티캐스트에 비하여 높은 대역폭을 필요로 하는데, 그룹 멀티캐스트 문제에 대하여

* KT 기술연구소 네트워크 서비스 개발팀 (ihuman9@kt.co.kr),

** 포항공과대학교 컴퓨터공학과 네트워크 및 분산시스템 연구실 (tkpark99, chkim@postech.ac.kr)

논문번호 : 040036-0115, 접수일자 : 2004년 1월 26일

대역폭 요구를 고려한 최적의 해를 찾는 것은 NP-Complete^[3]임이 증명되어 있다^[3]. 따라서, 소스 트리 구성 방법 (Source Tree Approach)과 공유 트리 구성 방법 (Shared Tree Approach)으로 분류되는 많은 휴리스틱 알고리즘들이 제안되었다^[2,3,4,10].

소스 트리 구성 방법에 속하는 알고리즘으로 Jia & Wang 알고리즘^[2], GTM (Group Multicast Routing with Adapted TM Algorithm)^[3], FTM (Feasible TM)^[4] 등이 제안되었는데, Jia & Wnag 알고리즘은 Steiner 트리^[14] 구성을 위한 KMB (Kou, Markowsky & Berman) 휴리스틱 알고리즘^[7]을 기반으로 설계되었으며, GTM과 FTM은 TM (Takahashi & Matsuyama) 휴리스틱 알고리즘^[5]을 기반으로 제안되었다. 이들 알고리즘들은 모든 그룹 멤버를 송신자 (소스: Source)로 하는 다중 소스 트리를 구성하기 때문에, 그룹 멤버의 수가 m 인 경우 m 개의 소스 트리를 구성한다. 이와 같은 소스 트리 구성 방법은 공유 트리 구성 방법에 비하여 높은 트리 구성 성공률을 제공하는데 반하여, 그룹 멤버의 수와 동일한 수의 소스 트리 구성에 기인한 높은 트리 관리 오버헤드 및 낮은 확장성 문제를 내포하고 있다.

공유 트리 구성 방법에 속하는 알고리즘으로는 CBT (Core Based Trees)^[8] 및 STGM (Shared Tree for Group Multicast)^[10] 등이 제안되었으며, 이 알고리즘들은 모든 그룹 멤버들을 송신자 (소스)이자 수신자로 포함하는 한 개의 공유 트리를 구성한다. 단, CBT는 Steiner 트리에 관련된 휴리스틱 알고리즘이 아니기 때문에 본 논문에서는 더 이상 다루지 않는다. TM 알고리즘을 기반으로 설계된 STGM의 목적은 트리 유지를 위해 필요한 정보의 양을 줄임으로써 확장성을 높이는 것이기 때문에, 소스 트리 구성 방식에 비해 트리 관리 오버헤드가 낮고 확장성 측면에서 우수하다는 장점을 가지고 있지만 그룹 멤버의 수가 증가함에 따라 트리 구성 성공률이 급격히 낮아진다는 단점을 가지고 있다.

본 논문에서는 그룹 멀티캐스트를 위하여 구성되는 트리의 수가 망 부하에 따라 변화하는 휴리스틱 알고리즘인 MTGM (Multiple Trees for Group Multicast)을 제안한다. 기본적으로 MTGM은 각각의 그룹 멤버에 대한 Steiner 트리를 구성하는데 있어서 KMB 알고리즘보다 성능이 우수하다고 알려진 TM 알고리즘에 기반을 두고 설계되었다. MTGM은 그룹 멀티캐스트를 위한 하나의 공유 트리가 존재한다면, TM 알고리즘에 기반한 STGM과 동일하게 동작한다. 그러나, STGM은 그룹 멤버의 수가 증가하거나 망의 부하가 높을 때 트리

구성 성공률이 급격하게 낮아지는 단점을 가지고 있기 때문에, 그룹 멀티캐스트를 위한 하나의 공유 트리를 구성할 수 없는 경우에 MTGM은 이미 구성된 트리에 송신자로서 포함되지 못한 나머지 그룹 멤버들을 위한 소스 트리 구성을 시도한다. 즉, MTGM은 STGM과 같이 한 개의 공유 트리만 구성할 수도 있고, 몇 개의 부분 (Partial) 공유 트리와 몇 개의 소스 트리를 구성할 수도 있으며, 최악의 경우 소스 트리 구성 방법과 동일하게 그룹 멤버의 수와 동일한 수의 소스 트리를 구성할 수도 있다. 따라서, MTGM은 두 개의 양극단 접근 방식인 소스 트리 구성 방법과 공유 트리 구성 방법의 혼합 (Hybrid) 방식으로 분류될 수 있다. 시뮬레이션을 통하여, MTGM이 공유 트리 구성 방법에 비하여 월등히 높은 수준의 트리 구성 성공률을 보장할 뿐만 아니라, 소스 트리 구성 방법에 비하여 현저히 작은 수의 트리를 구성하기 때문에 낮은 트리 관리 오버헤드를 요구하면서도 소스 트리 구성 방법과 동일한 수준의 트리 구성 성공률을 보장함을 보인다.

본 논문의 구성은 다음과 같다. II장에서는 그룹 멀티캐스트를 위한 기존의 휴리스틱 알고리즘들을 고찰하고, III장에서는 그룹 멀티캐스트를 위하여 구성되는 트리의 수가 상황에 적절히 변화하는 휴리스틱 알고리즘인 MTGM을 제안한다. IV장에서는 시뮬레이션을 통하여 공유 트리 및 소스 트리 구성 방식에 대한 MTGM의 성능 향상 수준을 분석하며, 마지막으로 V장에서 결론을 맺는다.

II. 관련 연구

그룹 멀티캐스트 문제에 대하여 대역폭 요구를 고려한 최적의 해를 구하는 것이 NP-complete하다고 증명되었기 때문에^[3] 최적의 해에 가장 가까운 답을 얻기 위한 많은 휴리스틱 알고리즘이 제안되었다^[2,3,4,10]. 이런 알고리즘들은 우선 하나의 Steiner 트리를 구하기 위해 KMB^[7] 혹은 TM 알고리즘^[5]에 바탕을 두고 있다. 일반적으로 TM 알고리즘은 그래프에서 각 멤버 쌍 사이에 최소 비용 경로뿐만 아니라 Steiner 노드와 멤버 사이의 최소 비용 경로까지 계산하기 때문에 KMB 알고리즘보다 비용 측면에서 좋은 성능을 보인다^[12]. 그룹 멀티캐스트 문제를, 그래프 $G(V, E)$ 에 대하여 그룹 멤버 집합 M 을 위한 멀티캐스트 트리를 찾는 문제라고 할 때, 기존의 알고리즘들은 소스 트리 구성 방법 (Source Tree Approach)과 공유 트리 구성 방법 (Shared Tree Approach)으로 분류될 수 있다. 그래프 $G(V, E)$ 의 노드 수와 링크 수는 각각 $n(n = V)$ 과 $e(e = E)$ 로 표

현되며, $M \subseteq V$ 인 그룹 멤버 집합 M 의 멤버 수는 $m(m = |M|)$ 으로 표현된다.

소스 트리 구성 방법은 단방향 트리들의 집합 $\{T_1, T_2, \dots, T_m\}$ 을 구하는 것으로, 트리 T_i 는 멤버 노드 i 가 다른 그룹 멤버들에게 멀티캐스트 데이터를 전송하기 위하여 구성된 소스 트리다. 이 범주에 속하는 휴리스틱 알고리즘으로 Jia & Wang 알고리즘^[2], GTM (Group multicasting with adapted TM algorithm)^[3], FTM (Feasible TM)^[4]이 있다.

Jia & Wang 알고리즘은 Steiner 트리를 만들기 위해서 KMB 알고리즘을 수정한 것으로, Jia & Wang 알고리즘이 제안되기 이전의 알고리즘들이 소스 트리간의 연관성을 고려하지 않고 다중 소스 트리를 구성함으로써 최종 구성된 다중 트리들 중 각각의 트리를 놓고 보면 최소비용 트리이지만 전체 트리의 비용 합 측면에서 보면 최소가 아닌 결과를 초래한다는 현상에 초점을 맞추고 이를 개선하려고 시도하였다. 그래서 구성된 트리들의 총 비용 합을 최소화하기 위해서 각각의 트리를 만들 때 이들 작업간에 조율(coordination)을 수행하는데, 조율은 최소 "대안 오버헤드(alternative overhead)"를 지닌 트리가 대역폭이 포화된 링크를 놓고 경쟁하는 다른 트리를 위해서 트리 구성 과정 중에서 해당 링크를 포기하고 다른 대안 경로를 선택하는 방법이다. Jia & Wang 알고리즘의 시간 복잡도 (Time Complexity)는 최선의 경우에 $O(mn^2 + m^4 + m^2n)$, 최악의 경우에 $O(mn^2 + m^3n^2)$ 이 된다^[2].

GTM 알고리즘은 각 멤버의 소스 트리 구성을 위해 필요한 Steiner 트리를 구하기 위해서 KMB 대신에 TM 알고리즘에 기반하여 설계되었다는 것을 제외하고는 Jia & Wang 알고리즘과 동일하게 작동하며, $O(m^3n^2)$ 의 시간 복잡도를 가지고 있다^[3].

FTM 알고리즘도 TM 알고리즘에 기반하여 설계된 휴리스틱 알고리즘이다. Jia & Wang 알고리즘과 GTM 알고리즘의 목적은 구성 멀티캐스트 트리들의 총 비용 최소화이기 때문에 그룹 멀티캐스트를 지원하기 위한 트리 구성이 가능함에도 불구하고 대역폭 할당의 비효율성으로 인하여 종종 그룹 멀티캐스트 트리 구성에 실패할 수도 있는데 반하여, FTM 알고리즘은 그룹 멤버들에 도달하는 대역폭이 가장 큰 경로를 찾기 위해서 넓이 우선 검색(BFS)을 사용함으로써 멀티캐스트 트리 구성 성공률의 최대화에 그 목적을 두고 있다. 따라서, FTM 알고리즘은 Jia & Wang 알고리즘과 GTM 알고리즘보다 높은 트리 구성 성공률을 제공하기는 하지만, 구성된 트리의 비용이 이상의 두 알고리즘보다 높다는

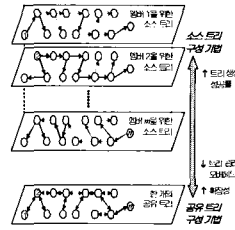


그림 1. 소스 트리 및 공유 트리 구성 방법

단점을 가지고 있다. FTM 알고리즘의 시간 복잡도는 $O(m(n + e)n^2)$ 이다^[4].

공유 트리 구성 방법에 속하는 휴리스틱 알고리즘으로는 STGM (Shared Tree for Group Multicast)^[10]이 제안되었으며, 이 알고리즘은 모든 그룹 멤버들을 포함하는 한 개의 양방향 공유 트리를 구성을 목적으로 한다. STGM은 TM 알고리즘에 기반하여 제안된 알고리즘으로, 그룹 멤버들을 부분 공유 트리로서 하나씩 연결함으로써 최종적으로 양방향 링크로 이루어진 하나의 공유 트리를 구성하는 방식을 취하고 있다. STGM은 모든 트리 상의 노드에서 이 노드를 통해 유입될 수 있는 대역폭의 양을 관리하는 "accounting"이라는 방법을 사용한다. 또한, STGM은 아직 공유 트리에 포함되지 않은 멤버 노드에서 부분 공유 트리까지의 연결 경로를 구하기 위해서 변형된 Dijkstra 알고리즘인 대역폭 제한 최소비용 양방향 경로 찾기 알고리즘을 사용한다. STGM은 그룹 멀티캐스트를 위한 공유 트리를 구성하기 위한 좋은 휴리스틱 알고리즘으로 알려져 있음에도 불구하고 가용 대역폭의 양이 작거나 멤버들의 대역폭 요구량이 크다면 트리 구성 성공률이 낮다는 단점이 있다. STGM의 시간 복잡도는 $O(m^2(n + e) \log n)$ 이다^[10].

그룹 멀티캐스트 문제를 해결하기 위하여 대역폭 제한을 고려한 소스 트리 구성 방법과 공유 트리 구성 방법의 장단점을 비교하면 그림 1과 같다. 소스 트리 구성 방법은 각 그룹 멤버를 소스로 하는 복수 개의 트리를 생성하는 방법으로 이 방법에 따라 생성된 트리의 요구 대역폭은 소스가 되는 멤버의 요구 대역폭에 불과하기 때문에, 생성된 트리의 요구 대역폭이 모든 그룹 멤버의 요구 대역폭 합과 동일한 공유 트리 구성 방법에 비하여 높은 트리 구성 성공률을 제공하는데 반하여, 그룹 멤버의 수와 동일한 수의 소스 트리 구성에 기인한 높은 트

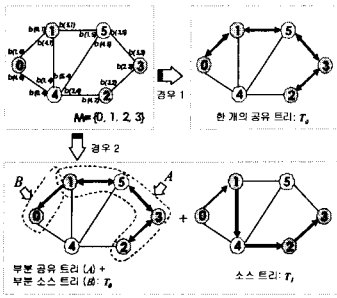


그림 2. MTGM이 구성 가능한 멀티캐스트 트리 예제

리 관리 오버헤드 및 낮은 확장성 문제를 내포하고 있다. 반면에, 공유 트리 구성 방식은 소스 트리 구성 방식에 비해 트리 관리 오버헤드가 낮고 확장성 측면에서 우수하다는 장점을 가지고 있지만 그룹 멤버의 수가 증가함에 따라 트리 구성 성공률이 급격히 낮아진다는 단점을 가지고 있다.

III. 제안하는 알고리즘

본 장에서는 제안하는 알고리즘인 MTGM의 기본 아이디어를 소개하고 구체적인 알고리즘을 서술한 후, MTGM의 시간 복잡도를 계산한다.

1. 기본 아이디어

제안하는 알고리즘인 MTGM은 그룹 멀티캐스트를 위한 트리를 구성하는데 있어서, 소스 트리 구성 방법보다 현저히 낮은 트리 관리 오버헤드로 소스 트리 구성 방법과 동일한 수준의 트리 구성 성공률을 보장하기 위하여 공유 트리 구성 방법과 소스 트리 구성 방법의 개념을 혼합 사용한다. 예를 들어, 그림 2와 같은 그래프와 방향성 링크의 가용 대역폭 $b(i, j)$ 및 그룹 멤버 집합 $M = \{0, 1, 2, 3\}$ 이 주어질 때, MTGM은 트리 관리 오버헤드를 절감하기 위하여 경우 1과 같이 한 개의 공유 트리 T_0 를 구성하려 시도하고, 이것이 실패하면 경우 2와 같이 두 개의 트리 (T_0 및 T_1)를 구성한다. 경우 2는 경우 1과 같이 그룹 멤버 0, 1, 2, 3을 위한 하나의 공유 트리 (경우 1의 T_0)를 구성하기 위한 작업 과정 중에 그룹 멤버 1, 2, 3을 위한 부분 공유 트리 A가 생성된 상태에서 마지막 그룹 멤버 0을 추가시키려다, 트리 A를 구성하는 링크들 중 하나 이상에 가용 대역폭이 부족하여

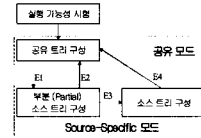


그림 3. MTGM의 기본적인 동작 단계

버 0의 추가가 불가능하기 때문에 그룹 멤버 1, 2, 3을 위한 트리 (경우 2의 T_0)와 그룹 멤버 0을 위한 트리 (경우 2의 T_1)를 생성한 예를 보여준다. MTGM은 최악의 경우에 단 한 개의 공유 트리도 없이 소스 트리 구성 방법과 동일하게 $|M|$ 개의 소스 트리를 구성할 수도 있다.

MTGM의 기본적인 동작 단계는 다음과 같다.

[1 단계] 그룹 멀티캐스트를 위한 트리 구성을 시작하기에 앞서, 트리의 구성이 가능한지 여부를 판단하기 위하여 실행 가능성 (feasibility) 시험을 수행한다.

[2 단계] "공유 모드"로 동작하며, 현재까지 구성된 트리들 (T_0 부터 T_1 까지)에 송신자 (소스)로서 추가되지 않은 그룹 멤버 중 하나를 선택한 후, 트리 T_i 로의 최소 비용 양방향 경로를 찾고 공유 트리 T_i 를 확장한다.

[3 단계] 공유 트리 구성이 더 이상 진행될 수 없으면 (그림 3의 E1), "Source-Specific 모드"로 전환하여, 트리 T_i 로부터 트리 T_i 에 속하지 않은 그룹 멤버로의 최소 비용 단방향 경로를 찾고 T_i 에 부분 소스 트리를 추가한다. 트리 T_i 에 모든 그룹 멤버가 포함될 때까지 반복한다. 부분 소스 트리 추가가 완료되면 2 단계로 돌아가서 새로운 트리 T_{i+1} 의 구성을 시도한다 (그림 3의 E2).

[4 단계] 2 단계와 3 단계에서 모든 그룹 멤버를 포함하는 트리를 구성하지 못하면 (그림 3의 E3), 2 단계와 3 단계에서 구성된 현재의 트리 T_i 를 제거한 후, 2 단계에서 최초로 선택된 멤버를 송신자 (소스)로 하는 소스 트리 T_i 를 구성한다. 소스 트리 T_i 의 구성이 완료되면 다시 2 단계로 돌아가서 새로운 공유 트리 T_{i+1} 의 구성을 시도한다 (그림 3의 E4)

그래프 $G(V, E)$ 와 그룹 멤버 집합 $M \subseteq V$ 에 대하여 그룹 멤버의 수를 $m (m = |M|)$, 노드 $v_i \in V$ 의 입력

Algorithm MTGM
Input: 그래프 $G(V, E)$, 비용 $c(i, j) \forall (i, j) \in E$ 가용 대역폭 $b(i, j)$
 $\forall (i, j) \in E$ 그룹 멤버 집합 M 및 대역폭 요구량 $B(i) \forall v \in M$
Output: 복수 트리 $T_1(E_1, V_1), \dots, T_k(E_k, V_k), 0 \leq k \leq |M|$
 1) if (가능성 시험 실패) return 실패;
 2) $M' = \emptyset$ // i 개의 트리에 송신자 s 로서 포함된 멤버 집합 M' 초기화
 3) for $j = 0; j < |M|; j++$ $T_j(E_j, V_j) \leftarrow \emptyset$ // 초기화
 4) $i = \alpha$ // 첫 번째 트리
 5) while ($M' \neq \emptyset$)
 6) $a = M'$ M' 에 속하면서, 현재 링크의 가용 대역폭 값들 중 최대값인
 가용 그룹 멤버 노드;
 7) 노드 a 를 트리 T_i 및 M' 에 추가;
 8) mode = "shared";
 9) while (mode = "shared")
 10) // T_i 로부터 T_j 에 속하지 않은 그룹 멤버까지의 경로들 중 대역폭
 // 요구를 만족하는 최소 비용 양방향 경로 P 와 멤버 v 를 찾는다.
 11) $P \leftarrow$ bi-directional-path-find-algorithm
 12) if (P 가 존재) {
 13) 트리 T_j 및 경로 P 에 속한 (i, j) 의 가용 대역폭 $b(i, j)$ 값 수정,
 14) 경로 P 의 멤버 노드 v 를 트리 T_j 에 추가하고 v 를 M' 에 추가;
 15) }
 16) else {
 17) mode = "source-specific";
 18) while (T_j 에 추가되지 않은 그룹 멤버 존재) {
 19) // T_i 로부터 T_j 에 속하지 않은 그룹 멤버까지의 경로들 중
 // 대역폭 요구를 만족하는 최소 비용 단방향 경로 P 와
 // 멤버 v 를 찾는다.
 20) $P \leftarrow$ uni-directional-path-find-algorithm
 21) if (P 가 존재) {
 22) if (T_j 에 한 개 멤버만 존재) return 실패;
 23) 모든 변수를 T_j 생성이전 값으로 복원한 후
 24) 라인(6)에서 결정된 노드 a 를 T_i 및 M' 에 추가;
 25) continue; // go to line 18)
 26) // end of if
 27) 경로 P 에 속한 (i, j) 에 대하여 가용 대역폭 $b(i, j)$ 값 수정,
 28) 경로 P 의 멤버 노드 v 를 트리 T_j 에 추가;
 29) // end of while
 30) } // end of else
 31) } // end of while
 32) } // end of while

그림 4. MTGM 알고리즘

링크 개수를 v' , 입력 링크 l 의 가용 대역폭을 B^l , 멤버 v_i 의 요구 대역폭을 $B(v_i)$ 라 할 때, 1 단계의 실행 가능성 시험은 다음과 같이 정의된다.

$$\sum_{l=1}^{v'} B^l \geq \sum_{k=1}^m (B(v_k); \text{except } v_i), \forall \text{ member } v_i$$

실행 가능성 시험의 성공은 모든 그룹 멤버들 각각의 수신 가능 대역폭이 자신을 제외한 나머지 그룹 멤버들의 요구 대역폭 합을 수용할 수 있음을 의미한다. 실행 가능성 시험 결과가 성공이면, 2, 3, 4 단계에서 MTGM은 다음의 두 개의 비용 함수를 이용하여 최소 비용의 양방향 혹은 단방향 경로를 찾은 다음 다중 멀티캐스트 트리를 구성하기 시작한다. 노드 i 에서 노드 j 로의 가용 대역폭을 $b(i, j)$ 로 표현한다.

Algorithm Bandwidth-Constrained Min-Cost Bidirectional Path Finding
Input: 그래프 $G(V, E)$, 비용 $c(i, j) \forall (i, j) \in E$ 가용 대역폭 $b(i, j) \forall (i, j) \in E$, s 와 t 를 포함한 트리 T_0 그룹 멤버 집합 M
 K 개의 트리에 s 로서 포함된 멤버 집합 M' 및 대역폭 요구량 B_1, B_2, \dots, B_K (대상 노드의 요구량, B_2 는 트리 T_0 의 요구량)
Output: T_1 로부터 $M-M'$ 에 속하는 그룹 멤버까지의 경로들 중 대역폭 요구를 만족하는 최소 비용 양방향 경로
 1) resultPathCost = ∞ ; resultPath = NULL;
 2) for ($M-M'$ 에 속한 멤버 v 에 대해) {
 3) for (그래프 G 의 모든 노드 w 에 대해) {
 4) vmark = false; wcost = ∞ ;
 5) } // end of for
 6) sc = 0; // 2)에서 선택된 s 의 초기 비용을 0으로 지정
 7) while (vmark(w) / false) 노드가 존재 {
 8) 이 둘 노드 중에서 w 가 t 와 최소인 노드 v 를 선택
 9) if (wcost == ∞) break;
 10) vmark = true;
 11) for (vmark = false) $h(w) \geq B_1$ 이면 $h(w) \geq B_2$ 인
 노드 w 의 비용을 $wcost$ 에 $c(s, w)$ 를 더함
 12) if (wcost < wcost) {
 13) wcost = wcost + $c(s, w)$;
 14) wpredecessor = v ;
 15) } // end of if
 16) } // end of for
 17) } // end of while
 18) 현재 구성된 트리 T_0 를 smark로 시작하여 MTGM이
 최소인 T_0 에 속한 노드 v 를 선택
 19) if (resultPathCost > wcost) {
 20) $P \leftarrow$ vpredecessor에 기반하여 추출된 경로
 21) if (v 를 제외한 P 의 모든 노드들이 T_0 에 속하지 않음) {
 22) resultPath = P ; resultPathCost = wcost;
 23) } // end of if
 24) } // end of if
 25) } // end of for
 26) return resultPath;

그림 5. 최소 비용 양방향 경로 찾기 알고리즘

$$\text{단방향 링크 비용 함수: } c_u(i, j) = \frac{1}{b(i, j)}$$

$$\text{양방향 링크 비용 함수: } c_b(i, j) = \frac{1}{b(i, j)} + \frac{1}{b(j, i)}$$

본 논문의 최소 비용 양방향 경로 찾기 알고리즘은 STGM (Shared Tree for Group Multicast)^[10]에서 제안한 변형된 Dijkstra 알고리즘에 양방향 비용 함수 $c_b(i, j)$ 를 적용한 것이고, 최소 비용 단방향 경로 찾기 알고리즘은 최소 비용 양방향 경로 찾기 알고리즘을 수정한 것에 단방향 비용 함수 $c_u(i, j)$ 를 적용한 것이다. 두 개의 알고리즘은 다음 절에서 상세히 서술된다.

2. 알고리즘

MTGM은 실행 가능성 시험에 성공한 그래프 $G(V, E)$ 와 멀티캐스트 그룹 멤버 집합 $M \subseteq V$ 에 대하여 "공유 모드" 및 "Source-Specific 모드"를 통하여 k 개 ($1 \leq k \leq |M|$)의 멀티캐스트 트리를 구성한다. MTGM의 Pseudo 코드는 그림 4와 같다.

```

Algorithm: Breadth-First-Search-Based M-In-Cost
Unidirectional Path Finding
Input: G=(V, E), E, S, T, C, B, M, M'.
Output: Path P from S to T.
1) Create Path Cost = C.
2) for (v in S)
3) if (v == S)
4)   for (w in G[v])
5)     if (w != S)
6)       if (w != T)
7)         if (w != S)
8)           if (w != T)
9)             if (w != S)
10)            if (w != T)
11)             if (w != S)
12)            if (w != T)
13)             if (w != S)
14)            if (w != T)
15)             if (w != S)
16)            if (w != T)
17)             if (w != S)
18)            if (w != T)
19)             if (w != S)
20)            if (w != T)

```

그림 6. 최소 비용 단방향 경로 찾기 알고리즘

이미 구성된 i 개의 트리에 송신자 (소스)로서 포함된 그룹 멤버들의 집합을 M 이라고 할 때, MTGM은 $M - M'$ 에 속한 그룹 멤버들 중에서 입력 링크의 가용 대역폭 값들 중 최대값이 가장 작은 그룹 멤버를 새로운 트리 구성을 위한 최초의 멤버로 선택하는데 (그림 4 라인 6~7), 이와 같이 선택하는 이유는 공유 트리 구성의 실패를 줄이기 위함이다. 멤버 $v(v \in M)$ 가 이미 구성된 i 개의 트리에 송신자 (소스)로서 포함되어 있다는 것은, 멤버 v 에 대한 양방향 경로를 포함하는 공유 트리가 구성되어 있거나 멤버 v 를 송신자 (소스)로 하는 소스 트리가 구성되어 있다는 것을 의미한다. 또한, 멤버 v 에 대하여 입력 링크의 가용 대역폭 값들 중 최대 값이란 $\max(b(u, v)), \forall (u, v) \in E$ 를 의미한다.

멀티캐스트 트리를 구성하는데 있어서, MTGM은 항상 "공유 모드"로 동작을 시작한다 (그림 4 라인 5~15). 현재 트리 T_i 의 최초 멤버가 선택되면, MTGM은 그림 5에 서술된 "최소 비용의 양방향 경로 찾기 알고리즘"을 이용하여 적절한 경로를 선정한다. 발견된 적절한 경로와 해당 멤버는 트리 T_i 에 추가되고, 트리 T_i 와 경로 P 에 속한 모든 링크 (i, j) 에 대하여 가용 대역폭 값 $b(i, j)$ 은 수정된다.

적절한 양방향 경로가 존재하지 않으면, MTGM은 "Source-Specific 모드"로 (그림 4 라인 16~29) 전환한다. "Source-Specific 모드"에서 MTGM은 현재까지 구성된 트리 T_i 로부터 T_i 에 속하지 않는 그룹 멤버까지의 경로들 중에서 대역폭 요구를 만족하는 최소 비용 단방

향 경로 P 를 그림 6에 서술된 "최소 비용의 단방향 경로 찾기 알고리즘"을 이용하여 찾아내고 트리 T_i 에 추가한다. 단, 적절한 단방향 경로도 존재하지 않는다면 모든 변수를 트리 T_i 구성 이전 시점으로 복원한 후, 트리 T_i 의 최초 선택 멤버를 송신자 (소스)로 하는 소스 트리 구성을 시작한다 (그림 4 라인 23~24). "Source Specific 모드"에서의 동작이 완료되면, MTGM은 "공유 모드"로 다음 트리 T_{i+1} 의 구성을 시도한다.

현재 구성중인 트리 T_k 로부터 M' 에 속하지 않는 그룹 멤버로의 "최소 비용 양방향 경로 찾기 알고리즘"은 그림 5와 같다. "최소 비용 양방향 경로 찾기 알고리즘"에서는, M' 에 속하지 않은 모든 멤버들로부터 트리 T_k 에 속한 노드들로의 양방향 경로들 중에서 대역폭 제약 조건을 만족하면서 최소 비용인 경로를 검색한다. M' 에 속하지 않은 그룹 멤버 s 와 트리 T_k 에 포함되는 노드 t 를 연결하는 양방향 경로 $(s \leftrightarrow t)$ 의 대역폭 제약 조건은, $b(s \rightarrow t) \geq B(s)$ 와 $b(t \rightarrow s) \geq B(T_k)$ 를 만족하는 것이다. 이러한 대역폭 제약 조건을 만족하는 양방향 경로는 트리 T_k 로부터 새로운 멤버 s 로 트래픽을 전달할 충분한 대역폭을 가지고 있을 뿐만 아니라 s 의 트래픽을 트리 T_k 에게 전달할 충분한 대역폭을 가지고 있다. 양방향 경로를 선택할 때 검사해야 할 또 다른 조건들로는, 양방향 경로 $(s \leftrightarrow t)$ 에서 t 를 제외한 어떠한 노드도 트리 T_k 에 포함되어 있지 않아야 한다는 것과, 트리 T_k 에 속한 모든 링크들이 멤버 s 의 추가에 의해 포화(saturate)되지 않아야 한다는 것이 있다.

양방향 경로를 찾는 알고리즘의 변형 알고리즘인 "최소 비용의 단방향 경로 찾기 알고리즘"은 그림 6과 같다. 수정된 부분으로는, 첫째, T_k 에 속하지 않는 그룹 멤버가 없을 때까지 알고리즘이 동작한다는 것 (그림 6 라인 2), 둘째, 단방향에 대해서만 대역폭 제약 조건을 검사하고 비용 함수도 단방향 비용 함수 $c_u()$ 를 사용한다는 것 (그림 6 라인 11~13), 셋째, 새로운 멤버의 추가에 의한 트리의 포화 여부를 검사하지 않는다는 것 (그림 6 라인 18)이 있다.

3. 시간 복잡도

그룹 멤버의 수를 m 이라 하고, 그래프를 구성하는 노드와 링크의 수를 각각 n 및 e 라 하자. 일반적으로 그룹 멤버의 수 m 는 그래프의 노드 수 n 에 비하여 상당히 작은 값이다. 이와 같은 환경에서, "최소 비용의 양방향 경로 찾기 알고리즘"의 시간 복잡도는 $O(m(n + e) \log n)$ 이기 때문에^[10], "최소 비용의 단방향 경로 찾기 알고리즘"의 시간 복잡도도 마찬가지로 $O(m(n + e) \log n)$ 이

다. 즉, 현재 생성중인 트리에 그룹 멤버 하나를 추가하기 위한 양방향 혹은 단방향 경로를 찾는 절차의 시간 복잡도는 $O(m(n+e)\log n)$ 가 된다. 따라서, 최상의 경우에 제안하는 알고리즘의 시간 복잡도는 오직 하나의 공유 트리를 구성하는 경우의 시간 복잡도인 $O(m^2(n+e)\log n)$ 가 된다. 그러나, 최악의 경우에 제안하는 알고리즘은 $(m-1)$ 개의 멤버를 위한 공유 트리를 만들었다가 마지막 1개 멤버를 공유 트리에 추가하는데 실패하여 하나의 소스 트리를 구성하고, 다음으로 $(m-2)$ 개의 멤버를 위한 공유 트리를 만들었다가 마지막에 실패하여 하나의 소스 트리를 구성하는 형태로 m 개의 소스 트리를 구성할 수 있기 때문에, 최악의 경우 시간 복잡도는 $O(m^3(n+e)\log n)$ 이 된다.

IV. 성능 분석

MTGM의 성능 분석에 사용되는 그래프들은^[6,11]에서 제안한 방법에 따라 임의의 (random)로 생성된 그래프 $G(V, E)$ 로서, V 에 속한 노드들은 직사각형 격자 (Rectangular Grid)에 임의로 배분되며 E 에 속한 링크 (u, v) 는 수식 (1)의 확률 함수^[11]에 따라 생성된다.

$$P_e(u, v) = \frac{k \cdot \bar{e}}{V} \alpha \exp\left(-\frac{d(u, v)}{\beta \cdot L}\right) \quad (1)$$

수식 (1)에서 $d(u, v)$ 는 노드 u 와 v 간의 거리이고, L 은 임의의 두 노드간의 최대 거리이며, α 와 β 는 $0 < \alpha, \beta \leq 1$ 과 같은 범위의 값을 가지는 상수이다. 상수 α 의 값이 커지면 노드간의 연결성이 커지는 반면에 상수 β 의 값이 커지면 노드간의 연결성은 떨어진다. 수식 (1)에서 k 는 임의의 두 노드간의 평균 거리이고, \bar{e} 는 노드의 평균 Degree이며, V 는 그래프 $G(V, E)$ 에 속한 노드의 수를 의미한다^[6,11]. 시뮬레이션에서는 α 와 β 의 값으로 각각 0.25 와 0.2로 지정하며, \bar{e} 는 5로 지정한다.

시뮬레이션에서 사용되는 그래프 $G(V, E)$ 의 노드 수와 격자(Grid)의 크기는 각각 100개와 30×30 이며, E 에 속한 모든 링크는 양방향 링크이다. 이상과 같이 생성된 그래프에 대하여, E 에 속한 링크들의 평균 대역폭 B_m 이 주어지면 링크 (u, v) 의 대역폭 $B(u, v)$ 은 $B(u, v) = B_m + (-1)^n \cdot r \bmod B_m$ 과 같이 계산된다. 변수 n 과 r 은 각 링크마다 임의로 선택되며, 그 결과 $B(u, v)$ 는 1부터 $2B_m - 1$ 사이의 정수 값이 된다. 또한, 그룹 멤버의 대역폭 요구량은 대역폭의 단위 값으로 1

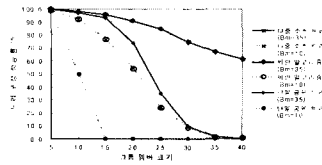


그림 7. 그룹 멤버 크기 대 트리 구성 성공률

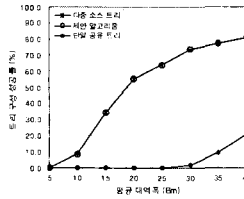


그림 8. 평균 대역폭 대 트리 구성 성공률 ($m = 30$)

이고, 시뮬레이션을 수행할 때마다 멀티캐스트 그룹 멤버는 V 에서 임의로 선택되며, 모든 결과 그래프는 1000번 시뮬레이션 결과의 평균값으로 작성된다. 성능 비교 대상이 되는 단일 공유 트리 알고리즘과 다중 소스 트리 알고리즘은 모든 링크의 비용을 1로 지정하거나^[10] 또는 링크 비용으로 노드간의 거리와 같은 고정된 값을 사용^[3]하는 대신에 각각 본 논문에서 정의된 양방향 비용 함수 $c_b()$ 와 단방향 비용 함수 $c_u()$ 를 사용하는 STGM과 GTM 알고리즘이다.

먼저, 멀티캐스트 그룹 멤버 크기에 따라 변화하는 멀티캐스트 트리 구성 성공률을 망의 부하가 낮은 경우와 높은 경우를 구분하여 분석하였다. 낮은 망 부하의 경우에는 링크들의 평균 대역폭 값 B_m 을 35로 지정하여 그래프 $G(V, E)$ 를 생성하였고, 높은 망 부하의 경우에는 B_m 을 10으로 지정하여 그래프를 생성하였다.

그림 7은 그룹 멤버의 크기 변화에 따른 트리 구성 성공률에 대한 분석 결과를 보여준다. 그림 7로부터 그룹

1) GTM 알고리즘에서의 트리간 조율 (Coordination)은 링크 비용이 노드간의 거리와 같이 변하지 않는 값일 때 동작하도록 정의되었기 때문에 본 시뮬레이션에서는 조율 방법을 사용하지 않았다.

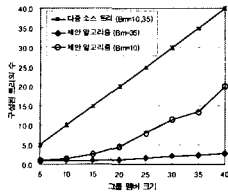


그림 9. 그룹 멤버 크기 대 구성된 트리 개수

멤버의 크기가 커질수록 트리 구성 성공률이 감소함을 확인할 수 있는데, 이러한 결과가 나타나는 이유는 그룹 멤버가 커질수록 구성된 트리의 링크에 대하여 요구되는 대역폭의 크기가 커지기 때문이다. 특히, STGM에 의해 구성된 공유 트리에 속한 링크는 모든 그룹 멤버의 요구 대역폭의 합을 수용할 수 있어야 하기 때문에 그룹 멤버 크기가 증가함에 따라 트리 구성 성공률이 급격히 감소하게 된다. 이에 반하여, 제안하는 알고리즘인 MTGM은 그룹 멤버의 수만큼 소스 트리를 구성하는 알고리즘과 동일한 트리 구성 성공률을 보일 뿐만 아니라, 공유 트리를 구성하는 알고리즘에 비하여 월등한 수준의 트리 구성 성공률을 보인다.

그림 8은 그룹 멤버 크기를 30으로 고정된 상황에서, 링크의 평균 대역폭 값 B_m 을 변화시키기에 따른 멀티캐스트 트리 구성 성공률 변화 분석 결과를 보여주는데, 그림 8에서도 그림 7과 동일한 의미의 결과가 도출되었다.

다음으로, 동일한 수준의 트리 구성 성공률을 보이는 소스 트리 구성 알고리즘과 MTGM을 구성되는 멀티캐스트 트리 개수 측면에서 비교 분석하였다. 라우터에서 상태 정보 관리에 필요한 멀티캐스트 트리 관리 오버헤드는 구성된 트리의 개수에 비례하여 증가하기 때문에^[9], 구성된 트리의 수는 중요한 성능 분석 항목이다. 트리 구성 성공률 분석에서와 동일하게 높은 부하와 낮은 부하 생성을 위하여 평균 대역폭 값으로 각각 35와 10을 사용하였다. 그림 9는 멀티캐스트 그룹 멤버를 위하여 구성되는 멀티캐스트 트리의 개수 분석 결과를 보여주는데, 이로부터 MTGM은 낮은 부하 ($B_m=35$)에서 그룹 멤버 수에 비하여 현저히 작은 수의 멀티캐스트 트리만 구성함을 확인할 수 있다. 단, 높은 부하 ($B_m=10$)에서 그룹 멤버의 수가 큰 경우에 구성되는 트리의 수가 상당 수준 증가함을 확인할 수 있는데, 그 경우에도 소

스 트리 구성 알고리즘에 비하여 여전히 낮은 수준을 유지하고 있다.

이상의 성능 분석 결과로부터, MTGM은 공유 트리 구성 알고리즘에 비하여 월등히 높은 수준의 멀티캐스트 트리 구성 성공률을 보장할 뿐만 아니라, 소스 트리 구성 알고리즘에 비하여 현저히 작은 수의 멀티캐스트 트리를 구성하면서도 동일한 수준의 트리 구성 성공률을 보장함을 확인할 수 있다.

V. 결론

본 논문에서는 기존에 제안된 알고리즘들과는 달리 대역폭 제약 그룹 멀티캐스트 지원을 위해 가변 수의 트리들을 구성하는 새로운 휴리스틱 알고리즘인 MTGM을 제안하였다. MTGM은 그룹 멀티캐스트를 위한 트리를 구성하는데 있어서, 소스 트리 구성 방법보다 낮은 트리 관리 오버헤드로 소스 트리 구성 방법과 동일한 수준의 트리 구성 성공률을 보장하기 위하여, 공유 트리 구성 방법에 따라 공유 트리 구성을 시도하다가 공유 트리 구성이 불가능한 경우, 이미 구성된 트리에 송신자로서 포함되지 않은 멤버들에 대하여 소스 트리 구성 방법에 따라 소스 트리를 구성한다. 그 결과, MTGM은 공유 트리 구성 방법에 비하여 월등히 높은 수준의 트리 구성 성공률을 보장할 뿐만 아니라, 소스 트리 구성 방법에 비하여 현저히 낮은 트리 관리 오버헤드로 동일한 수준의 트리 구성 성공률을 보장함을 시뮬레이션을 통하여 확인하였다. 본 논문에서 제안한 MTGM은 그룹 멤버 수가 정적인 경우를 가정하고 설계되었는데, 향후 연구 항목으로 동적으로 그룹 멤버가 바뀌는 응용 프로그램에 대하여 동작할 수 있도록 확장을 고려하고 있다.

참고 문헌

- [1] J.Hou and B. Wang, "Multicast routing and its QoS extensions: problems, algorithms and protocols," *IEEE Network*, vol.14, no.1, pp. 22-36, Jan./Feb. 2000.
- [2] X. Jia and L. Wang, "A group multicast routing algorithm by using multiple minimum Steiner trees." *Computer Communications*, vol.20, no.9, pp. 750-758, September 1997.
- [3] C. P. Low and N. Wang, "An efficient algorithm for group multicast routing with bandwidth reservation." *Proceedings of IEEE ICON*, pp. 43-49, Sept. 28 - Oct. 1, 1999.

[4] N.Wang and C.P.Low, "On finding feasible solutions to the group multicast routing problem", *Proceedings of Networking 2000*, May 2000.

[5] H.Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs", *Math. Japonica*, 24:253-577, 1980.

[6] B. M. Waxman, "Routing of multipoint connections", *IEEE Journal on selected areas in Communications*, vol.6, no.9, pp. 1617-1622, December 1988.

[7] L. Kou, G. Markowsky and L. Berman, "A fast algorithm for Steiner trees", *Acta Informatica* 15, pp. 141-145, 1981.

[8] A. Balladie, "Core based trees (CBT version 2) multicast routing: protocol specification", *RFC2189*, September 1997.

[9] Frank, A. J., Wittie, L.D., and Bernstein, A.J., "Multicast Communication on Network Computers", *IEEE Software*, Vol. 2, No. 3, pp. 55-89, 1992.

[10] A. Fei, Zhihong Duan, M. Gerla, "Constructing shared-tree for group multicast with QoS constraints", *IEEE GLOBECOM 2001*, vol.4, pp. 2389-2394, November 2001.

[11] M. Doar, I. Leslie, "How bad is naïve multicast routing?", *IEEE INFOCOM'93*. April, 1993.

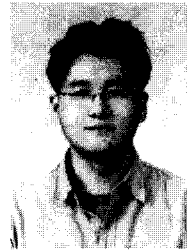
[12] Q. Sun and H. Langendoerfer, "An efficient Delay-Constrained Multicast Routing Algorithm", *Journal of High-Speed Networks*, vol.7, no.1, pp. 43-55, 1998.

[13] 김 철연, "그래프론과 알고리즘", *POSTECH PRESS*, 1997.

[14] F.K. Hwang and D.S. Richards, "Steiner tree problems", *Networks*, vol.22, no.1, pp.55-89, January, 1992.

구 봉 규(Bong-Gyu Koo)

정회원



2001년 2월 : 고려대학교 컴퓨터학과 졸업

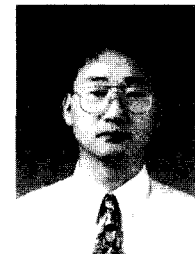
2003년 2월 : 포항공과대학교 컴퓨터공학과 석사

2003년 1월 ~ 현재 : KT 기술연구소 네트워크 서비스 개발팀 <관심분야> 교환 및 라우팅,

QoS, 초고속 정보통신, 이동 통신

박 태 근(Taekeun Park)

정회원



1991년 2월 : 포항공과대학교 컴퓨터공학과 졸업

1993년 2월 : 포항공과대학교 컴퓨터공학과 석사

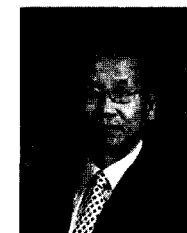
2004년 2월 : 포항공과대학교 컴퓨터공학과 박사

2004년 2월 ~ 현재 : 포항공과대학교 컴퓨터공학과 박사후 과정 연구원

<관심분야> 이동 통신 프로토콜, 유/무선 망 QoS, IP 기반 통합망, 멀티미디어 통신망

김 치 하(Checha Kim)

정회원



1974년 2월 : 서울대학교 전자공학과 졸업

1984년 8월 : University of Maryland Computer Science 석사

1986년 8월 : University of Maryland Computer Science 박사

1986년 9월 ~ 1989년 12월: State University of New York 교수

1989년 12월 ~ 현재 : 포항공과대학교 컴퓨터공학과 교수

<관심분야> 모바일 컴퓨팅 및 네트워킹, 컴퓨터 통신, 분산 시스템, 성능 평가