

Core 기반 멀티 캐스트의 효율적인 Core 선택 알고리즘에 대한 연구

정동재 , 하동문, 김용득
아주대학교 전자 공학부 컴퓨터 네트워크 연구실

The Study on effective Core selective Algorithm for Core based Multicast

Dongjea Jung , Dongmoon Ha , Yongdeak Kim
School of Electronics Engineering Ajou University
E-mail : metica2k@hanmail.net

Abstract

본 논문에서는 향후 인터넷 환경에서 필수적인 요건이 될 수 있는 인터넷 멀티 캐스팅의 여러 프로토콜 중 공유 트리 방식의 하나인 CBT 프로토콜에서 송신자와 수신자 사이의 경로 즉 비용을 줄이기 위해 기존의 코어 선택 알고리즘을 개선할 수 있는 알고리즘을 제안한다. CBT 프로토콜에서는 공유 트리의 구성이 코어 라우터를 중심으로 이루어지고 이 코어 라우터를 통해 그룹에 가입/ 데이터 전송/ 탈퇴 등이 이루어지게 되므로 코어 라우터의 선정은 전체적인 효율성 선정의 중요 요소가 된다. 여기서는 모든 노드의 전체 비용을 산출하여 가장 비용 요소가 적은 노드를 선정하는 알고리즘을 제안하며, 네트워크 시뮬레이터를 통해 사용율 및 평균 비용을 기존 방식과 비교하여 그 결과에 대해서 논한다.

I. 서론

네트워크의 발전에 따라 VOD, 인터넷 방송, 인터넷 공동 작업, 화상 회의 등의 인터넷 어플리케이션이 발전함에 따라 현재의 일반적인 인터넷 전송 방식인 일대일 전송 방식으로는 효과적인 정보를 제공할 수 없게 되었다. 즉 Unicast 전송 방법으로는 한명의 송신자가 다수의 수신자에게 데이터를 송신하려면 똑같은 데이터를 수신자의 수만큼 보내야 하기 때문에 네트워크

대역폭을 낭비하게 되고 과도한 네트워크 트래픽을 유발하게 된다. 이를 해결하기 위해 등장한 것이 멀티캐스트 기술이다.

멀티 캐스트 기술이란 한명 또는 복수의 송신자가 데이터 수신을 원하는 수신자들에게만 데이터를 전송하는 일대다 또는 다대다 전송 방식이다. 이 멀티 캐스트 방식은 수신하기 원하는 수신자에게만 보낼 수 있기 때문에 Broadcast 와 구별되며 Broadcast 에 비해 대역폭을 적게 사용한다. 또한 Unicast 와 다르게 한번의 데이터 전송만을 필요로 하기 때문에 네트워크 대역폭 낭비를 막을 수 있다.

본 논문에서는 기존의 멀티캐스트 프로토콜을 소개하고 공유 트리 방식 프로토콜인 CBT 프로토콜에서 대역폭의 효율적인 사용 및 혼잡을 감소시킬 수 있는 코어 선택 알고리즘을 제안한다. 그리고 네트워크 시뮬레이터를 통해 기존 알고리즘과 비교, 분석을 통해 제안된 알고리즘의 성능을 분석한다.

II. Multicast 라우팅 프로토콜

멀티 캐스트는 호스트의 특정 인터페이스 멀티 캐스트 그룹에 프로세스가 참여하는 개념이다. 따라서 특정 인터페이스에서 멀티캐스트의 그룹의 구성원 구성은 동적으로 이루어져야 하며, 프로세서는 자유롭게 그룹에 참여하고 떠날 수 있어야 한다.

이러한 멀티캐스트를 수행하기 위해 현재 개발된 프로토콜에는 Dense Mode Multicasting Protocol (DM 프로토콜) 과 Sparse Mode Multicasting Protocol (SM 프로토콜) 이 있다.

DM 프로토콜은 그룹 멤버들이 좁은 네트워크 환경에 밀집되어 있는 환경에 적합하도록 설계된 프로토콜이다. 이 DM 프로토콜은 송신자의 수만큼 트리를 가지고 있기 때문에 소스 중심 트리 프로토콜이라고 한다. 송신자의 수만큼 트리를 보유하기 때문에 라우터간의 정보 교환이 많고 시스템이 커지면서 유지해야 할 정보가 많아 확장성에 문제를 가지고 있는 프로토콜로 대역폭이 충분한 환경에서 주로 사용된다.

SM 프로토콜은 인터넷에서 멤버가 다양한 지역에 널리 분포되어 있는 상황에서 적합하게 만든 프로토콜이다. 그룹의 멤버들은 송신자와 관계없이 중심 노드와 연결된 하나의 트리를 공유하기 때문에 라우터간 교환해야 할 정보량이 적다. 따라서 이 SM 프로토콜은 공유 트리 방식이라고 불리며 네트워크의 확장성이 좋은 반면에 송신자와 수신자간의 최적의 경로를 제공해주지는 못한다. 이 SM 프로토콜을 사용하는 멀티 캐스트 알고리즘에는 다음에 논하게 될 CBT 와 PIM-SM 이 있다.

III. CBT (Core Based Tree) 알고리즘

본 절에서는 앞 절에서 제시한 공유 트리 방식의 CBT 에 대해서 확장성 측면과 그 알고리즘에 대해서 알아보고 또, 본 논문의 목적인 효과적인 코어 선택을 위한 알고리즘을 제안한다.

3.1 Core Based Tree 확장성 비교 및 알고리즘

소스 중심 트리 알고리즘의 경험을 통해 좀더 좋은 규모 확장성을 위해 CBT 는 공유 트리 방식을 사용한다. 다음은 소스 트리 중심 방식인 DVMRP 와 CBT 의 라우터 상태 정보량을 그룹 크기와 활성 소스 수를 다르게 설정하여 비교한 표이다.

그룹 수	10			100			1000		
구성 노드 수	20			40			60		
그룹당 소스 수	10%	50%	100%	10%	50%	100%	10%	50%	100%
DVMRP 라우터 테이블 항목수	20	100	200	400	2K	4K	6K	30K	60K

CBT 라우터 테이블 항목수	10	100	1000
-----------------	----	-----	------

[표 1] DVMRP 와 CBT 라우터 상태 정보량 비교

위 표에서 살펴 볼 수 있듯이 그룹 당 소스 수와 그룹의 크기에 따라서 DVMRP 는 라우터 테이블 항목수가 증가하지만 CBT 의 라우터 테이블 항목 수는 그룹 당 소스 수에만 영향을 받는다는 것을 알 수 있다.

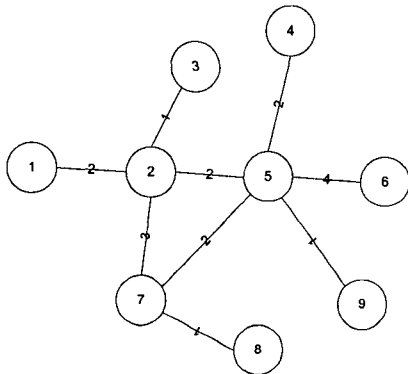
3.2 CBT Core 선택 알고리즘

소스 중심 트리 프로토콜은 RFP 등의 최단 경로 알고리즘을 사용하여 송신자와 수신자 간의 최단 거리를 제공한다. 그러나 CBT 프로토콜은 코어로 선택된 라우터를 중심으로 각 호스트간의 최단 경로를 이룸으로써 하나의 공유 트리를 구성하게 된다. 따라서 CBT 프로토콜에서는 전송하려는 데이터가 모두 코어 라우터로 전송되고, 수신을 원하는 수신자 역시 코어 라우터를 통해 모든 데이터를 전송 받게 된다. 또한 공유 트리의 유지 및 관리도 송신자와 코어 라우터,수신자와 코어 라우터를 통해 이루어진다. 따라서 CBT 알고리즘에서 코어 라우터의 선택은 멀티캐스트의 평균 딜레이, 맥시멈 딜레이, 총 비용 등에 영향을 주게 된다.

이런 코어 라우터를 선택하는 기존 방법에는 Random Method, Geographic Center Method, Maximum weight Method 등이 있다. Random Method 는 주어진 영역 내의 라우터 중 무작위로 코어 라우터를 선택하는 방법이다. Geographic Center Method 는 위치적으로 중앙에 가장 가까운 위치의 라우터를 선택하는 방법이다. 마지막으로 Maximum weight Method 는 각 노드간의 최단 경로를 계산한 뒤 각 쌍의 노드들 사이를 지나가는 횟수에 가중치를 두어 가장 가중치가 큰 라우터를 코어 라우터로 선정하는 방법이다. 위의 코어 선택 알고리즘중 Random Method 는 네트워크에 대한 지식이 필요 없고 네트워크 변화에 영향을 받지 않는다는 장점이 있고, Geographic Center Method 는 단순히 네트워크의 위치적인 중앙을 찾기 때문에 간단히 코어를 선택할 수 있으나 멤버들 간의 시간 지연이나 전체 비용 등에서 실용적이지 못하다. 또한 Maximum weight Method 는 코어 라우터를 선택하기 위해 네트워크 구성에 대해 정확히 알고 있어야 하고 코어 라우터를 선택하는 과정이 너무 복잡한 단점이 있다.

3.2 CBT Core 선택 알고리즘의 제안

CBT 프로토콜을 이용한 멀티캐스트 네트워크에서 전체 비용 및 전달 지연을 줄일 수 있는 코어 선택 알고리즘을 제시한다. 본 논문에서는 어떤 호스트가 CBT 멀티 캐스트 그룹에 참여하는지는 알지 못하지만 네트워크에서 CBT 그룹에 참여할 수 있는 호스트를 모두 알고 있다는 가정 하에서 CBT 알고리즘을 제안한다. 제안하는 알고리즘은 구성된 네트워크의 각 노드에서 자신을 제외한 모든 노드들간의 Dijkstra 알고리즘을 이용하여 비용을 계산하고 이렇게 계산된 값의 평균을 구한 뒤 가장 작은 평균 비용을 갖는 노드를 코어 라우터로 설정한다. 다음의 네트워크 모델 예제를 살펴보자.



[그림 1] 네트워크 각 노드 사이의 비용

위의 네트워크 구조에서 Dijkstra 알고리즘을 이용하여 초기 코스트 테이블을 작성하면 다음과 같다.

노드	1	2	3	4	5	6	7	8	9
1	0	2	x	x	x	x	x	x	x
2	2	0	1	x	2	x	3	x	x
3	x	1	0	x	x	x	x	x	x
4	x	x	x	0	2	x	x	x	x
5	x	2	x	2	0	4	2	x	1
6	x	x	x	x	4	0	x	x	x
7	x	3	x	x	2	x	0	1	x
8	x	x	x	x	x	x	1	0	x
9	x	x	x	x	1	x	x	x	0

[표 2] 그림 7의 네트워크 구조에서 구한 비용 테이블

그림 1에서 보면 노드 1에서 6으로 가는 경우는 2가지 경우가 있으며 첫번째 경우는

노드 1 → 노드 2 → 노드 5 → 노드 6

이며 비용은 2+2+4=8 이 된다.

두번째 경우는

노드 1 → 노드 2 → 노드 7 → 노드 5 → 노드 6

이며 비용은 2+3+2+4=9 가 된다 따라서 최단 비용이 드는 첫번째 경우가 노드 1에서 노드 6으로 가는 경로가 된다. 이런 식으로 모든 경로를 계산하면

노드(a)	1	2	3	4	5	6	7	8	9
1	0	2	3	6	4	8	5	6	5
2	2	0	1	4	2	6	3	4	3
3	3	1	0	5	3	7	4	5	4
4	6	4	5	0	2	6	4	3	3
5	4	2	3	2	0	4	2	1	1
6	8	6	7	6	4	0	6	7	5
7	5	3	4	4	2	6	0	1	3
8	6	4	5	5	3	7	1	0	4
9	5	3	4	3	1	5	3	4	0

[표 3] 각 노드별 비용

위의 표 3 과 같이 모든 노드들에 대한 경로 값을 구한 뒤 그 총합을 다음 식으로 구한다..

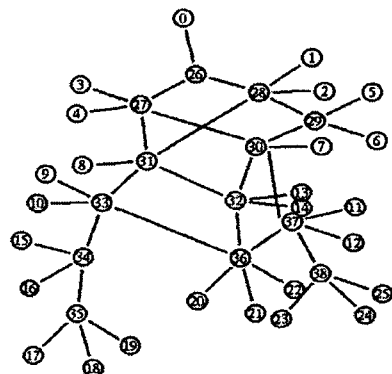
$Tc(a)=r(a, x)$, V:네트워크 내에 있는 노드 집합 즉, 한 노드를 기준으로 그 노드를 제외한 모든 노드로 가는 비용을 구한 뒤 그 비용을 비교하여 가장 적은 비용이 드는 노드를 코어로 선택한다.

노드(a)	1	2	3	4	5	6	7	8	9
Tc(a)	39	25	32	35	21	49	28	35	28

[표 4] 각 노드별 전체 비용

IV. 제안 알고리즘의 성능 평가

제안된 알고리즘의 성능을 분석하기 위해 IBNL 에 의해 개발된 NS 라는 네트워크 시뮬레이터와 C 언어를 이용한다. 먼저 가상의 네트워크 모델을 설정하고 Dijkstra 알고리즘의 성능 테스트를 위해 코어 라우터를 선정한다.



[그림 2] 시뮬레이션을 위한 네트워크 모델

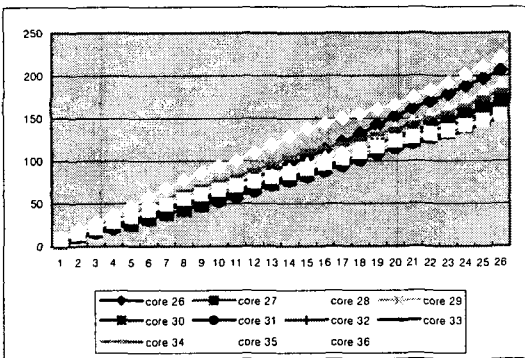
그림 2 의 네트워크 모델을 Dijkstra 알고리즘을 이용하여 최단 비용을 구하면 다음 표와 같다. 단 한 홉 사이의 거리는 1로 가정한다.

노드 번호	전체 비용	노드 번호	전체 비용	노드 번호	전체 비용	노드 번호	전체 비용
0	174	10	137	20	144	30	121
1	154	11	155	21	144	31	103
2	154	12	155	22	144	32	110
3	151	13	147	23	186	33	100
4	151	14	147	24	186	34	125
5	159	15	162	25	186	35	156
6	159	16	162	26	137	36	107
7	158	17	193	27	114	37	118
8	140	18	193	28	117	38	149
9	137	19	193	29	122		

[표 5] 각 노드별 전체 비용

표 5 는 각 노드간의 비용을 더한 결과를 나타내는 결과 표이다. 위 표에서 알 수 있듯이 33 번 노드가 최소 비용인 100 을 나타내는 것을 알 수 있다. 제안 알고리즘에 의해 노드 33 을 코어 노드로 선정한다.

이제 NS 시뮬레이터를 이용하여 다른 노드를 코어 노드로 사용했을 경우와 노드 33 을 코어 노드로 선정했을 경우 성능 비교를 수행한다.



[그림 4] 송신자 수에 따른 전체 비용의 증가량

그림 4 은 송신자의 수가 1 에서 26 으로 변할 때 비용의 증가를 나타낸다. 송신자의 수가 많은 경우에는 코어 라우터의 위치에 따른 비용의 증가 차이가 뚜렷하게 나타나는 것을 볼 수 있다. 그리고 위의 그림에서 알 수 있듯이 제안된 코어 라우터 알고리즘을 이용하여 코어를 선택했을 경우 가장 좋은 성능을 보인다는 것을 알 수 있다.

그 이유는 본 논문에서 제안한 알고리즘이 네트워크에 포함되어 있는 모든 라우터 사이의 비용을 계산하여, 평균적으로 가장 작은 비용을 가지는 라우터를 코어 라

우터로 선정하였기 때문에 송신자와 수신자의 위치나 수의 변화에 최적으로 적응을 할 수 있기 때문이다.

V. 결론

멀티 캐스트는 일반적인 네트워크 전송 방식인 유니 캐스트 전송방식으로 해결하기 힘든 어플리케이션을 지원하기 위한 새로운 전송 방식이다. 제안한 알고리즘은 네트워크 노드 쌍의 최단 비용을 Dijkstra 알고리즘을 이용하여 계산하고 자신을 제외한 모든 노드로의 비용의 합이 가장 적은 노드를 코어 노드로 선정한다. 이 방법은 비교적 간단한 계산에 의해 코어 라우터를 선정할 수 있고 또 본 알고리즘으로 선정된 코어는 다른 코어를 선정했을 때보다 송신자의 수가 증가함에 따라 회선 사용율이나 평균 비용이 3% ~ 70% 정도 낮은 것으로 나타나 기존의 알고리즘에 비해 향상된 성능을 나타낸다 것을 알 수 있다.

CBT 멀티 캐스트를 위한 향후 연구 과제는 코어 라우터로 집중되는 제어 및 데이터를 분산하기 위한 대책 및 멀티 캐스트 규모의 확장을 위해 하나 이상의 코어 라우터를 설정하여 효율적인 멀티 캐스트 전송을 이루기 위한 연구가 필요하다.

참고문헌

- [1] Tom Billhartz, J. Bibb Cain "Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols" IEEE Journal in Communications, April 1997, pp. 246-251
- [2] Clay Shields, J.J. Garcia-Luna-Aceves "The Ordered Core Based Tree Protocol" 1997 IEEE
- [3] Seok Joo Koh, Shin Gak Kang "Enhancement of the CBT Multicast Routing Protocol" IEEE 2001, pp. 209-213
- [4] A.D. Raghavendra and S.Rai "Multicast Routing in Internetworks Using Dynamic Core Based Trees" IEEE 1996, pp. 232-238
- [5] A. Ballardie "Core Based Trees (CBT version 2) Multicast Routing Architecture" RFC 2189, September 1997
- [6] S. Deering, C. Patridge, and D. Waitzman, "Distance Vector Multicast Routing Protocol" RFC 1075 Nov. 1988