

멀티캐스트 그룹 관리를 위한 불리언 연산

이신형^o 유시환, 변상선, 유혁
고려대학교 컴퓨터학과
{shlee^o, shyoo, ssbyun, hxy}@os.korea.ac.kr

Boolean Operation for Multicast Group Management

ShinHyoung Lee^o SeeHwan Yoo, SangSun Byun, Hyuck Yoo,
Dept. of Computer Science, Korea University

요 약

기존의 멀티캐스트는 하나의 그룹 주소를 갖고 그 주소를 통해 여러 멤버 노드로의 데이터 송수신을 지원한다. 하지만 서로 연관된 많은 멀티캐스트 그룹이 존재 시 송신자의 각 그룹 관리가 힘들어지고, 노드의 잦은 멀티캐스트 그룹간 참가 및 탈퇴로 인해 지연시간이 생긴다. 또한 멀티캐스트 라우팅 테이블 유지에 많은 자원을 사용한다. 이에 서로 연관된 멀티캐스트 그룹간 불리언 연산을 통해 더 유연성 있는 운용을 가능하게 하는 기법을 제시하고 그에 따른 문제점과 해결방안을 살펴본다.

1. 서 론

멀티캐스트는 중복된 전송을 방지함으로써 효율적인 전송을 가능하게 하는 기법이다. 동일한 데이터를 여러 노드가 받게 될 때 송신자는 동일한 데이터를 각 노드에게 유니캐스트로 전송하지 않고 멀티캐스트로 전송함으로써 네트워크 대역폭의 효율성이 증대되고 송신자의 자원 관리에도 유리하다.

예를 들어, 온라인 게임 분야에서는 멀티캐스트가 유용하게 사용될 수 있다. 유니캐스트를 사용하여 동일한 게임 데이터를 전송하는 방식과 비교하여 멀티캐스트를 사용하면 게임 서버의 부하를 크게 줄일 수 있고 네트워크 대역폭도 효율적으로 사용할 수 있다.

현재까지의 멀티캐스트 그룹 관리 기법[1]은 하나의 그룹에 대한 참가(Join), 탈퇴(Leave) 등의 기법을 제시하고 있다. 특히, 온라인 게임과 같은 대규모 응용의 경우, 몇 개의 멀티캐스트 그룹을 사용하는 것이 일반적이다.

하지만 지금까지 멀티캐스트 관리 기법은 여러 그룹간 연산을 통한 데이터 송신을 지원하지 않는다. 기존 방식의 경우 여러 그룹간 연산을 통해 표현할 수 있는 그룹이더라도 새로운 멀티캐스트 그룹 주소로 지정하고 모든 노드가 새로운 그룹에 참가하여야 한다. 이로 인해 연관된 그룹간 잦은 참가 및 탈퇴로 인하여 지연시간이 길어진다. 또한, 멀티캐스트 그룹의 수가 증가함에 따라 라우팅 테이블의 크기가 증가한다.

본 논문에서는 이러한 단점을 보완하여 멀티캐스트 그룹간의 불리언 연산을 이용한 멀티캐스트 방법을 제시한다. 또한 그에 따라 예상되는 문제점을 살펴보고 해결방

안을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 멀티캐스트 주소와 그룹 관리에 대한 관련 연구를 살펴보고, 3장에서 멀티캐스트 그룹간의 불리언 연산을 기술한다. 이어서 4장에서 예상되는 문제점과 해결방안을 기술한다. 마지막으로 5장에서 결론과 향후 연구방안을 기술한다.

2. 관련 연구

IPv4에서 IP 멀티캐스트는 주소의 처음 4bit를 1110으로 표현한 클래스 D의 주소를 이용하여 그룹의 멤버로 가입한 노드에게 멀티캐스트 데이터를 전송한다[1]. IPv6는 처음 8bit를 1로 표현함으로써 멀티캐스트 주소를 표시한다[2].

멀티캐스트 그룹 관리를 위해 IPv4는 IGMP(Internet Group Management Protocol)를 사용한다. IGMP를 통해 노드는 원하는 그룹에 참가(Join)하여 멀티캐스트 데이터를 받을 수 있다. IGMP는 노드들 간에 정보를 교환하여, 멀티캐스트 데이터를 받아야 하는 라우터들 간의 트리를 형성한다. 멀티캐스트 트리는 주기적인 IGMP 쿼리를 이용하여 유지된다[1]. IPv6의 경우 MLD(Multicast Listener Discovery)를 이용하여 멀티캐스트 그룹을 관리 및 유지한다[3].

기존의 연구들은 하나의 멀티캐스트 그룹에 대한 연산(Join/Leave)을 수행할 때, 각 멀티캐스트 트리를 구성하고, 유지하는 방식을 제안하고 있다.

3. 멀티캐스트 그룹간의 연산

지금까지 멀티캐스트 관리 기법은 여러 그룹간 연산을 통한 데이터 송신을 지원하지 않는다. 본 논문에서는 두 개 이상의 멀티캐스트 그룹에 대한 불리언 연산을 정의

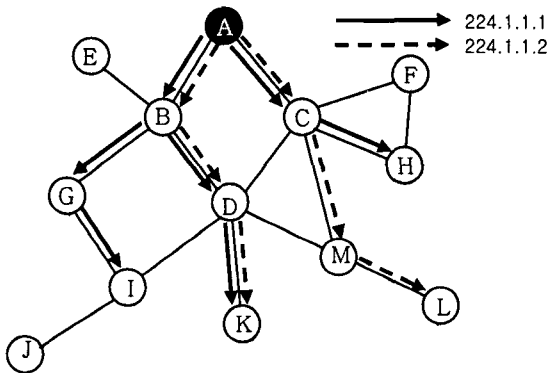
본 논문은 대학 IT 연구센터 육성 지원 사업의 연구결과로 수행되었음.

한다. \wedge 와 \vee 연산자를 사용한 그룹주소는 다음과 같다.

$0xEy.yy.yy.yy \wedge 0xEz.zz.zz.zz$
 $0xEy.yy.yy.yy \vee 0xEz.zz.zz.zz$

\neg 연산을 사용한 그룹 주소는 다음과 같다.

$\neg 0xEy.yy.yy.yy$



<그림 1> 224.1.1.1 과 224.1.1.2 멀티캐스트 트리 구성

두 개의 멀티캐스트 그룹 주소를 각각 MAddr_A, MAddr_B라 한다. NIC은 네트워크 인터페이스 카드, NIC_{Addr}는 NIC과 연관된 네트워크 주소라고 할 때, 해당 멀티캐스트 데이터를 포워딩하는 네트워크 인터페이스 카드인 McastNIC과 볼리언 연산은 다음과 같이 정의할 수 있다.

$McastNIC(A) = \{NIC \mid NIC_{Addr} \ni MAddr_A\}$
 $McastNIC(B) = \{NIC \mid NIC_{Addr} \ni MAddr_B\}$
 $McastNIC(A \wedge B)$
 $= \{NIC \mid NIC_{Addr} \ni MAddr_A \text{ 이고, } NIC_{Addr} \ni MAddr_B\}$
 $McastNIC(A \vee B)$
 $= \{NIC \mid NIC_{Addr} \ni MAddr_A \text{ 이거나, } NIC_{Addr} \ni MAddr_B\}$
 $McastNIC(\neg A)$
 $= \{NIC \mid NIC_{Addr} \ni MAddr_A\}^c$

\wedge 연산은 두 멀티캐스트 그룹 모두에 속한 노드와 연결된 네트워크 인터페이스 카드를 통해 데이터를 보낸다. 예를 들어 <그림 1>에서와 같이 224.1.1.1이라는 그룹과 224.1.1.2라는 멀티캐스트 그룹을 가정하면 224.1.1.1의 그룹과 224.1.1.2의 그룹에 동시에 가입된 노드에게만 멀티캐스트 데이터를 보내고자 할 때 \wedge 연산자를 이용한다. 라우터는 자신의 라우팅 테이블을 확인하고 해당하는 두 주소가 모두 연결되어있는 B, C, D, K 노드로 데이터를 포워딩한다. C 라우터와 같이 중간 라우터에서 조건에 맞는 네트워크 인터페이스가 없다면 자연스럽게 데이터는 드롭되어 더 이상의 네트워크 대역폭을 낭

비하지 않는다.

\vee 연산은 두 멀티캐스트 그룹 중 하나 이상 멤버인 노드에게 데이터를 보낸다. 보내려는 멀티캐스트 데이터의 주소가 \vee 연산을 한다면 해당하는 멀티캐스트 주소에 모두 포워딩함으로써 \vee 연산을 수행한다. 이 경우 하나의 인터페이스로 같은 데이터가 중복되어 보내지 않도록 하여 대역폭의 낭비를 줄여야 한다.

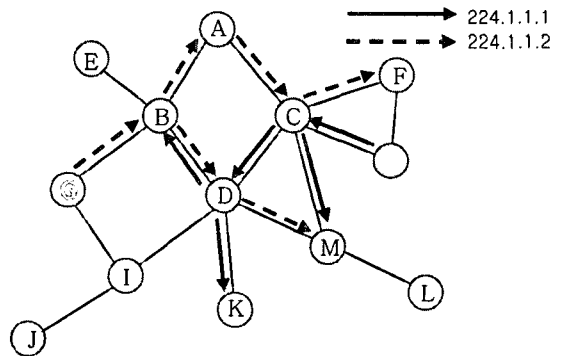
\neg 연산은 연산자에 이어 나오는 그룹의 멤버가 아닌 모든 노드를 의미한다. \wedge 또는 \vee 연산과 함께 사용하고 처음 멀티캐스트 주소는 \neg 연산을 사용하지 못한다. 하지만 원하는 결과를 얻지 못할 수 있으므로 \neg 연산을 사용할 때에는 원하는 결과를 얻을 수 있는지 매우 신중하게 생각하고 사용하여야 한다.

여러 개의 연산자가 사용될 경우 연산의 우선순위는 \neg 연산이 \wedge 와 \vee 연산보다 높다. 따라서 \neg 연산이 먼저 수행되고 \wedge 와 \vee 연산이 순차적으로 수행된다. 괄호 연산자 지원의 어려움으로 인해 연산으로 인한 결과의 예측이 힘들어지므로 송신자는 복잡한 연산자 사용을 매우 주의하여야 한다.

4. 예상되는 문제점과 해결방안

4.1 목적지 주소 표현

IP 헤더에 목적지 주소는 하나만 들어갈 수 있다. 하지만 멀티캐스트 그룹간의 연산을 위해서는 두 개 이상의 멀티캐스트 주소가 필요하다. 따라서 IP 헤더의 옵션필드를 이용하여 여러 개의 멀티캐스트 주소를 보낼 수 있도록 하여야 한다. 또한 어떤 연산을 할 것인지에 대해서도 옵션필드를 통해 표시하여야 한다.



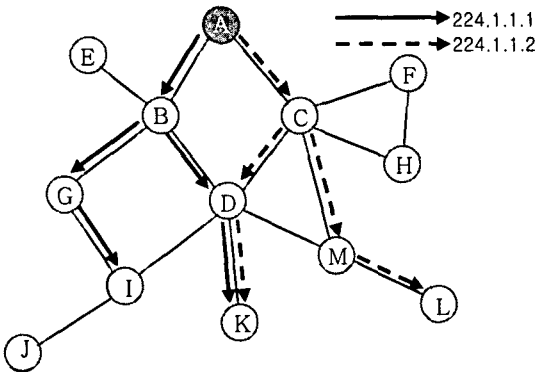
<그림 2> 소스가 분산되어 있을 때의 멀티캐스트 트리 구성

4.2 멀티캐스트 트리 관리

여러 개의 소스에서 시작되는 멀티캐스트를 가정하면 두 개의 멀티캐스트 그룹에 모두 가입되어 있음에도 불구하고 데이터를 못 받는 경우가 생길 수 있다. 예를 들어 <그림 2>와 같이 224.1.1.1과 224.1.1.2의 소스가 각

각 네트워크상에 분산되어 위치하고 DVMRP(Distance Vector Multicast Routing Protocol)[4]를 사용한다면 \wedge 연산에 의한 멀티캐스트 데이터는 전달되지 않는다.

이를 해결하기 위해 CBT(Core Based Tree)를 사용할 수 있다. CBT는 여러 개의 소스가 있더라도 하나의 코어 라우터를 루트로 멀티캐스트 트리가 만들어진다. 따라서 소스가 네트워크에 분산되어 위치하더라도 멀티캐스트 트리는 코어 라우터를 루트로 하여 생성되므로 문제를 해결할 수 있다[5].



<그림 3> D까지의 경로가 다른 두 멀티캐스트 트리

그룹 연산은 새로운 멀티캐스트 트리를 형성하지 않으므로, 정의한 연산에 따라 전달되지 않을 수도 있다. <그림3>에서 D 라우터는 224.1.1.1 그룹 멤버이면서 224.1.1.2 그룹 멤버이기도 하다. 하지만 224.1.1.1은 B 라우터를 통해 연결되어 있고, 224.1.1.2는 C 라우터를 통해 연결되어 있다. 따라서 A 라우터는 224.1.1.1 \wedge 224.1.1.2 주소로 보내야 하는 데이터를 받으면 자신의 아래에는 그와 같은 노드가 없는 것으로 판단하고 데이터를 보내지 않는다. 결국 D 라우터 이후의 K 노드는 데이터를 받지 못하게 된다.

이러한 문제는 연관된 멀티캐스트 트리의 경로가 같도록 기본이 되는 브로드캐스트 트리를 미리 구성함으로써 해결할 수 있다. 이를 위해 코어 라우터는 트리 구성 패킷을 브로드캐스트 한다. 구성된 브로드캐스트 트리는 멀티캐스트 트리의 구성의 기본이 된다. 모든 멀티캐스트 트리는 브로드캐스트 트리의 일부분이 되도록 구성한다. 결과적으로 연관된 멀티캐스트 트리의 경로를 같게 만들 수 있다. 브로드캐스트 트리는 네트워크 변화에 따라 같이 변화해야 한다. 따라서 코어 라우터는 주기적으로 브로드캐스트 트리를 위한 트리 구성 패킷을 브로드캐스트 한다.

5. 결론과 향후 연구 방향

기존의 멀티캐스트는 그룹 주소를 하나만 지원함으로써 그룹의 운용에 유연성이 부족했다. 그러나 다양한 응용프로그램들이 개발됨에 따라 멀티캐스트도 여러 그룹 간 연산을 통한 데이터 송신이 필요해지게 되었다. 따라

서 본 논문에서 불리언 연산을 통해 멀티캐스트 그룹을 더 유연하게 운용할 수 있는 기법을 소개하였다.

향후 실험을 통해 제시된 기법이 갖는 효용성을 정량적으로 분석하고자 한다. IP 헤더 옵션 필드의 변화가 필요하므로 IP헤더 옵션 필드 디자인이 요구된다. 각 그룹의 노드가 이동성을 갖는다면 동적으로 변하는 멀티캐스트 트리 관리가 매우 어려워진다. 따라서 이동성 지원에 대한 연구도 필요하다.

참고 문헌

- [1] Deering S., "Host Extensions for IP Multicasting," RFC 1112, Request for Comments, Aug. 1989.
- [2] R. Hinden and S. Deering, "IPv6 Multicast Address Assignments." IETF, RFC2375, July 1998. 14.
- [3] S. Deering, W. Fenner, and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6." RFC 2710, Oct. 1999.
- [4] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol (DVMRP)," Internet Engineering Task Force (IETF), RFC 1075, November 1988.
- [5] T. Ballardie, P. Francis, and J. Crowcroft, "Core based tree (CBT): an architecture for scalable inter-domain multicast routing," SIGCOMM, ACM press, 1993, pp. 85-95.