

## 네트워크 시스템에서 트래픽 부하에 따른 멀티캐스트 라우팅 방식

설남오

서남대학교 전기전자공학과

### Determination of Multicast Routing Scheme for Traffic Overload in network system

Seul, Nam-O

Seonam University Electrical & Electric Dept.

**Abstract** - The deployment of multicast communication services in the internet is expected to lead a stable packet transfer even in heavy traffic as in network system environment. The core based tree scheme among many multicast protocols is the most popular and suggested recently.

However, CBT exhibit two major deficiencies such as traffic concentration or poor core placement problem. so, measuring the bottleneck link bandwidth along a path is important for understanding th performance of multicast.

We propose not only a definition of CBT's core link state that Steady-State, Normal-State and Bottleneck State according to the estimation link speed rate, but also the changeover of multicast routing scheme for traffic overload. In addition, we introduce anycast routing tree, a efficient architecture for construst shard multicast trees.

#### 1. 서 론

인터넷의 고속화와 일반화는 사회 전반에 새로운 e-business를 창출하고 있으며, 유비쿼터스의 기반을 탄탄하게 받쳐주면서 유비쿼터스 시스템에 대한 수요와 공급을 폭발적으로 확대시키고 있다. 이러한 환경의 발전에 따라 모든 제어시스템들이 네트워크 기반으로 변환되거나 인간의 삶을 바꿔 놓을 것이다.

따라서 네트워크를 이용한 제어시스템의 원격제어는 계속 증가할 것이며 이에 따라 인터넷의 통신용량에 대하여 트래픽의 증가로 이어져 이를 해결할 이론이 필요하다.

이러한 문제를 해결하기 위한 방법으로, 인터넷을 경유하는 트래픽 부하의 경감을 실현하려는 다양한 시도가운데 멀티캐스트(Multicast) 통신 방식의 연구가 활발하게 진행되고 있다. 멀티캐스트 통신방식은 일대일 통신, 즉 Point-to-Point 방식의 유니캐스트(Unicast)와, 일대 전체의 일 방향성으로 Flooding 방식을 취하는 브로드캐스트(Broadcast)의 중간 형태를 취하는 통신방식으로 일대 다수(1:N), 또는 다수대 다수(M:N)의 통신이 가능하다. 멀티캐스트 통신 방식은 또한 이와 같은 접속의 유연성뿐만 아니라, 서버와 서버간이나 사용자들이 직접 연결되어 있는 서버까지의 단 한번의 패킷 전송만으로 수많은 사용자 그룹이 데이터 패킷을 공유할 수 있어 트래픽 부하를 현격하게 감소시킬 수 있다. 이에 따라 네트워크 자원의 이용률을 향상시킬 수 있는 대안으로 대두되면서 활용영역이 넓어지고 있다.[1] 멀티캐스트 통신 방식은 라우팅 트리의 구성 범위에 따라 Intra-Domain 방식과 Inter-Domain 방식으로 나누어지며, 트리의 구성 방법에 따라 소스 기반 멀티캐스트 라우팅 방식인 SBT(Source Based Tree)와 공유 트리 방식의

ShT(Shared Tree)방식으로 크게 구분할 수 있다.[2] 최근에는 유니캐스트(Unicast) 방식을 Multicast로 확장한 개념의 Anycast가 제안되고 있다.[9][10]

최근 활발히 연구되고 있는 CBT는 네트워크의 확장성이 유연하여, 폭발적으로 증가하고 있는 네트워크 트래픽을 기존의 다양한 멀티캐스트 라우팅 방식보다 비교적 우위에서 소화해 낼수 있다. 그러나 CBT 멀티캐스트 라우팅 방식이 갖는 고유 특성으로 인하여, 네트워크 제어 시스템에서 요구되는 실시간 처리와 멀티미디어 데이터를 수용해야 하는 트래픽 부하가 코어와 연결된 링크를 중심으로 링크속도(Link speed)를 증가시켜 병목현상(bottleneck)을 일으키고 링크 과도상태(Link Failure)로 이어질 수 있는 취약점도 또한 갖고 있다.[5][6]

본 논문에서는 네트워크 제어시스템의 효율적인 트래픽 제어와 안정적인 링크 유지를 위하여, CBT 멀티 캐스트 라우팅 방식에서 트래픽의 증가에 따른 링크의 트래픽 부하 밀도를 예측하고 링크의 트래픽 상태를 규정하였다. 또한 과도상태를 방지할 수 있도록 링크의 상태에 따라 멀티 캐스트 라우팅 방식을 전환하여 CBT 방식에서 Anycast 방식으로의 라우팅 전환 위한 알고리즘의 근거를 제시하였다.

#### 2. CBT와 링크 속도 예측

##### 2.1 CBT프로토콜

CBT 프로토콜은 최근에 제안되고 활발히 논의 되고 있는 멀티캐스트 라우팅 프로토콜로 코어를 중심으로 양방향성 공유트리를 형성하며, 네트워크 노드 수와 네트워크 트래픽 부하가 같은 조건에서 기존의 다른 멀티캐스트 프로토콜보다 적은 수의 라우터를 이용하여 트래픽의 처리가 가능하다. CBT 프로토콜은 다음과 같은 기본적인 메시지들을 처리하여 동작한다.

-JOIN\_REQUEST : 현재 사용중인 프로토콜이 CBT라 할때 멀티캐스트 트리 상에는 임의의 Core 라우터 c 가 존재하고 새로운 멤버 x 가 멀티캐스트 멤버로 가입을 요구할 경우 이 메시지를 전송한다.

-QUIT\_ACK : QUIT\_REQUEST에 대한 응답 메시지로 parent node나 상위 parent에 이상이 있을 경우에는 제한시간 내에 부정 응답 메시지로 처리된다.

-JOIN\_ACK : JOIN\_REQUEST에 대한 긍정 응답으로 새로운 멤버 x 가 CBT의 트리에 포함되어 있으면 부가적인 절차 없이 곧바로 멀티캐스트 서비스가 가능하다. 그러나 멤버 x 가 CBT의 트리에 포함되어있지 않으면 core 라우터 c로부터 멤버 x 에 이르는 최단경로가 형성 된다.

-JOIN\_NACK : JOIN\_REQUEST에 대한 부정 응답으로 새로운 멤버 x 가 멀티캐스트 트리에 JOIN 하기 위한 프로세서가 실패했을 때 처리되는 메시지다.

-QUIT\_REQUEST : 트리 상의 임의의 child node가 멀티캐스트 멤버로서의 종료될 위해 parent node에게

child node 조사와 종료를 요구하는 메시지이다.

CBT프로토콜의 가장 큰 장점은 멀티캐스트그룹의 모든 멤버에 의해 공유되는 단일 트리를 구성하므로 멤버나 소스의 수 변화에 트래픽이 안정적이다. 소스의 수를 S, 그룹 멤버의 수를 G 라 할 때 DVMRP(Distance Vector Multicast Routing Protocol)나 MOSPF(Multicast Open Shortest Path First)와 같은 SBT 의 트리 크기는 (S\*G)인 반면 CBT트리의 수는 G 와 같게 된다. 그리고 IGMP(Internet Group Management Protocol) 등의 프로토콜들처럼 그룹 관리를 위한 주기적인 정보를 요구하지 않아 할당된 대역폭을 순수한 멀티캐스트 트래픽을 위해 사용하므로 링크의 대역폭을 절약할 수 있어 네트워크의 자원을 효율적으로 사용하는 장점을 가지고 있다.[2]

그러나 전체 트리의 라우팅 경로가 코어로 중심으로 형성되고 코어를 경유해야 하므로 링크비용의 상승을 가져오며 트래픽의 코어집중으로 코어 링크에서 Bottleneck 을 일으킬 수 있다.

코어 링크의 대역폭을 초과하여 요구되는 링크 속도는 Bottleneck 현상을 일으켜 코어 링크의 교착상태로 이어지고 멀티캐스트 서비스의 중단을 가져올 수 있다. 코어 링크로의 트래픽 집중을 해소하기 위해 CBT의 코어를 두 개 이상(Multi Core)으로 두어 코어의 트래픽을 분산시킬 수 있으나 이 경우에도 하나의 멀티캐스트 코어를 이용하는 라우터 방식과 마찬가지로 어떻게 최적의 코어 라우터를 선택 하느냐 하는 문제에 봉착할 수 있다.

### 2.2 링크상태 예측 알고리즘

CBT 프로토콜의 장점에도 불구하고 최대 약점은 코어 라우터의 트래픽 집중에 기인하므로 코어 라우터의 트래픽 처리를 예측하여 해법을 찾을 수 있다. Bottleneck을 예측하는 알고리즘은 전력 스펙트럼 분석, Rate Based Congestion Control Scheme 등의 방법들이 있지만, 멀티캐스트 트리에서 Bottleneck 링크를 예측하여 멀티캐스트 트리를 재형성 하는 방법으로 사용되고 있다.[7][8] 이러한 방법들은 모두 별도의 BCS(Bottleneck Calculation Server)를 따로 두거나 멀티캐스트 트리 전체의 트래픽을 예측해야 하므로 전체의 멀티캐스트 관점에서 보면 Redundancy로 작용한다. 본 논문에서는 CBT 코어의 링크 트래픽 부하 밀도는 코어 로의 패킷 입력률(Link Speed Input Ratio)Bottleneck상태를 예측한다. 이를 기반으로 코어 라우터의 트래픽 처리상태에 따라 멀티캐스트 트리를 결정하는 알고리즘을 제안 하였다.

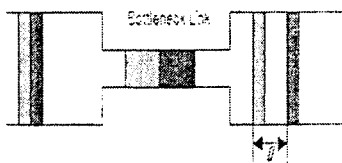


그림1. Bottleneck 링크

그림1 은 코어 라우터의 Bottleneck Link를 통과하는 패킷을 보여준다. 여기에서 패킷의 크기를 P (byte), Bottleneck 의 대역폭을 B (byte/sec)라 하면, 패킷 P 가 Bottleneck 링크를 통과하는 시간은 Qb는

$$Q_b = P / B$$

가 된다. 멀티캐스트 트래픽 전체에 대하여 고려하면 코어의 Bottleneck상태는 코어 라우터의 입력 링크 스피드 Bin 과 출력 링크 스피드 Bout 으로부터 산출할 수 있

으며 단일 Bottleneck Link Rate  $\Psi$  는

$$\Psi_{single} = Bin / Bout$$

가 된다. CBT코어 라우터의 연결이 단일 링크가 아닌 멀티 링크인 경우, 입력 링크 수를 m 출력 링크 수를 n 이라 하면

$$\Psi_{multi} = \sum(m/i=1)Bin(i) / \sum(n/j=1)Bout(j)$$

이 된다. 코어 라우터의 Bottleneck Link Rate  $\Psi$  가  $\Psi > 1.0$ 이 되면 패킷들은 코어 라우터를 통과하지 못하고 대기 하거나 폐기 되고 만다. 따라서  $\Psi$  값에 준하여 코어 라우터의 상태를 표.1과 같이 규정할 수 있다.

Core Link State	Definition	Link Speed I/O Ratio
SS(Steady state)	$0 < \Psi < 1.0$	Input < Output
NS(Normal state)	$\Psi = 1.0$	Input = Output
BS(Bottleneck state)	$\Psi > 1.0$	Input > Output

표 1. E값에 따른 Core 링크의 상태

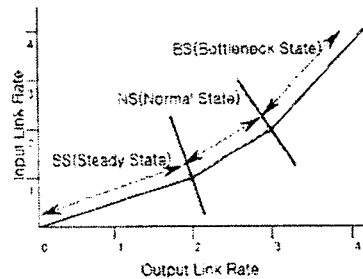


그림 2. E값에 따른 Core 링크의 상태

현재 네트워크의 트래픽 상태를 관찰하면 특정 시간대 에서 급속히 증가하는 경향을 알 수 있는데 이러한 특성은 곧 모든 멀티캐스트 트래픽이 집중되는 CBT의 코어를 과부하로 유도할 수 있다. 따라서 CBT멀티캐스트 방식으로 네트워크를 안정적으로 지원하기 위해서는 코어 라우터를 중심으로 트래픽의 상태를 모니터링 할 필요가 있으며 현재 트래픽의 상태가 SS(Steady State) 상태일 경우는 무관하나 NS(Normal State)상태로 진입할 가능성이 있다고 판단하고 멀티캐스트 네트워크에 트래픽 부하를 줄일수 있는 새로운 라우팅 방식으로 전환하는 멀티캐스트 전략이 마련되어야 한다.

### 2.3 예상효과

멀티캐스트 네트워크의 트래픽을 감소시키기 위해서는 게임 사용자의 데이터를 감소시키기는 매우 어려우며 네트워크 상에서 패킷 지연 요소를 줄이거나 중복되는 데이터를 제거하거나 그룹관리 정보를 최대한 간략한 해야 한다. 본 논문에서는 멀티캐스트 트리 전체의 트래픽 상태를 모니터링 하는 방식 대신 코어만의 트래픽 상태, 즉 코어의 링크 속도를 측정하는 BCS를 코어에 설치하고 코어로의 패킷 입력률과, 코어로부터의 출력률을 측정하여 링크의 속도 데이터를 기초로 CBT 코어의 트래픽 상태가 NS 상태에서 BS상태로 전이할 가능성이 있을 경우, 이를 사전에 회피하기 위한 새로운 라우팅 전략으로 Anycast 방식을 제안하였다.

Anycast 방식은 일대 다수(1:N) 통신을 지원하지만 멀티 캐스트 그룹내의 모든 멤버에게 패킷을 전송하는 방

식이 아니라 정보의 제공자 또는 게임 서버와 같은 Sender로부터 멀티캐스트 그룹 내의 가장 가까운 라우터나 최상의 조건을 가진 고유 서버에게만 데이터 패킷을 전송하므로 유니캐스트의 속성을 가지고 있다.

그러나 이러한 특성은 똑같은 데이터가 중복적으로 전송되는 네트워크의 대역폭을 충분히 감소시킬 수 있다는 측면에서 멀티캐스트의 속성과 일치하고, 또한 코어가

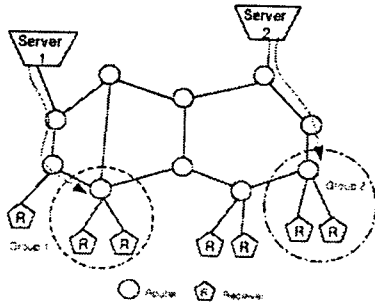


그림 3. E값에 따른 Core 링크의 상태

필요 없는 Non-Core 방식으로 CBT 멀티캐스트 라우팅 방식이 갖는 구조적인 문제점, 즉 코어 라우터로의 멀티캐스트 트래픽 집중과 코어 선정의 어려움 및 Poor Core의 구조적인 문제를 해결할 수 있다. 따라서 코어에서의 전송 지연, 패킷손실을 근원적으로 막을 수 있고 네트워크에 산재한 라우터들을 효율적으로 사용할 수 있다는 장점이 있다.

### 3. 결 론

본 논문에서는, 인터넷의 발전에 따라 새로이 등장하거나 유비쿼터스 방식의 서비스를 제공하는 제어시스템에서 활용가능한 멀티캐스트 라우팅 방식 중 코어를 중심으로 공유트리를 형성하는 CBT 멀티캐스트 라우팅 방식을 살펴보고 코어 라우터를 중심으로 한 입출력 링크 스피드를 측정하여 코어 라우터의 링크상태를 규정하는 알고리즘을 제안하였다. 제안한 알고리즘은 코어를 중심으로 한 네트워크 데이터 패킷의 트래픽을 모니터링하여 코어 라우터의 상태를 예측하고 SS상태(Steady State), NS상태(Normal State), BS상태 (Bottleneck State)로 구분하였다. 또한 NS상태에서 멀티캐스트의 성능을 저하시키며 코어 라우터의 교착 상태로 이어지는 BS상태로의 진입을 방지하기 위해 온라인 게임을 지원하는 멀티캐스트 라우팅 방식을 CBT에서 Anycast 라우팅 방식으로 전환하는 방안을 제안하였다. 이러한 제안은 CBT의 전형적인 문제들이 큰 대역폭을 요구하는 멀티캐스트 그룹에서 발생한다는 통계적 관찰에서 비롯된 것이다.

향후, 두 가지 멀티캐스트 라우팅 방식의 트래픽 처리량을 정량화하고 비교하기 위한 네트워크 모델의 설정과 프로토콜 시뮬레이션을 통한 성능 평가가 필요하다

### [참 고 문 헌]

[1] K. Almeroth and M. Ammar, "multicast Group Behavior in the Internet's Multicast Backbone", IEEE Communication, Jun, 1997.  
 [2] D. Meyer, "some Issue for an Inter-domain Multicast Routing Protocol", Internet-Draft, Nov, 1997  
 (3) A. Ballardie, P. Francis, and J. Crowcroft, "Core Base Tree(CBT) an Architecture for Scalable Multicast Routing," In Proc. of sigcomm/93, 1993.  
 (4) K. Calvert, E Zehura, and M. Donahoo, "core Selection

Methods for Multicast Routing," In Proc. fo ICCCN'95, 1995.

(5) L. Vicisano, L Rizzo J. Crowcroft, RCP-like congestion control for layered multicast data transfer", proceedings of INFOCOM'98, 1998

(6) S. Bhattacharyya, D. Towsley, and J. Kurose, "the Loss Path multiplicity problem in multicast congestion control", Proc. IEEE INFORCOM'99, pp. 856-863, New York, USA, March 1999.

(7) K. Yamamoto, M. Yamamoto, H. Ikeda, " Congestion control for reliable multicast achieving TCP fairness", Proc. SPECTS 2001.

(8) sylvia Ratnasamy and Steven McCanne, "Inference of multicast routing trees and bottleneck Bandwidth using End-to-End Measurements", Proc. of IEEE Infocom 1999.

(9) D. Katabi and J. Wroclawski, " A Framework for Global IP-anycast", Inernet-Draft, jun, 1999.

(10) D. Xuan, W. Jia, W. Zhao, H. Zhu, "a Routing Protocols for anycast messages", IEEE Transactions on Parallel and Distributed system, pp. 571-588, jun 2000.