

# 무선 ad-hoc 망에서의 Overlay Multicast 지원 방안

\*김혜원

한양대학교 전자통신전파공학과  
e-mail : luckyhw@ihanyang.ac.kr

## Overlay Multicast in Wireless Ad Hoc Networks

\*hye-won Kim

Division of Electrical and Computer Engineering  
Hanyang University

### Abstract

Overlay multicast is proposed as an alternative approach for providing multicast services. A logical infrastructure is built to form an overlay network on top of the physical layer. In this paper, we propose an efficient overlay multicast in wireless ad hoc networks. The overlay multicast tree adapts to the changes in underlying networks. The multicast tree adjusted according to the local member information.

### I. 서론

현재 많은 일 대 다 통신의 이득에 매료되어 많은 연구가 진행되어 왔다. 그러나 기존의 개발되어 왔던 Network layer에서의 멀티캐스트는 많은 개발이 진척되고 있는데 에도 불구하고 실제 서비스에 적용하는데 있어서의 부진한 면을 보이고 있다. 그 이유 중에 하나로 기존에 Network Layer에서 제공하는 멀티캐스트는 라우터에서 패킷이 복사되고 전달하게 되기 때문에 기존의 망의 변화를 요구한다. 이에 반해 최근에 제시

되고 있는 overlay 멀티캐스트는 end-host에서 패킷이 복사되고 유니캐스트 방식으로 전달된다. overlay 멀티캐스트는 기존의 network layer 멀티캐스트보다 효율성이 떨어지는 면이 있지만 기존 망의 변화를 요구하지 않는다는 점에서 최근 핫이슈로 떠오르고 있다.

본 논문에서는 overlay 멀티캐스트를 ad hoc 망에 적용하고자 한다. 잦은 노드 이동으로 동적인 토폴로지 구성으로 인해 제한된 대역폭과 이동 단말의 제한된 처리 능력을 고려한 overlay 멀티캐스트 적용 방안을 제시하고자 한다. overlay 멀티캐스트는 end-host에 더 많은 처리량을 요구하고 있기 때문에 효율적인 데이터 전달을 위한 overlay tree를 구성하고 유지해야 한다.

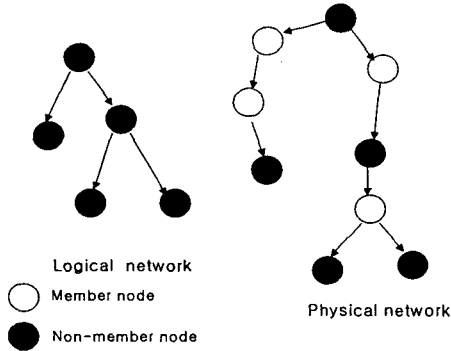
본 논문에서는 이러한 점을 고려한 overlay 멀티캐스트를 제시하고자 한다. section2에는 관련 연구를 살펴보고 나머지 section3에서는 새로 제안하는 overlay 멀티캐스트를 살펴보도록 하겠다.

### II. 관련 연구

#### 2.1 Overlay Multicast

overlay 멀티캐스트 기술은 기존 IP 멀티캐스트 모델을 완전히 탈피한 개념이다. 즉, 라우터가 아닌 종단 호스트 혹은 서버 레벨에서 멀티캐스트 데이터 포워딩을 수행하는 기술을 말한다. 즉, IP 멀티캐스트가 도입

되지 않은 라우터 혹은 네트워크에서 기존의 Unicast 전송 기술만을 이용하여 멀티캐스트 전송을 구현하는 것이다[3].



### III. 제안하는 overlay 멀티캐스트

#### 3.1 overview

root 노드와 멤버 노드로 구성되어 있다. sub\_root 노드는 모든 멤버들이 쉽게 접근 할 수 있고 충분한 처리 능력과 대역폭을 가지고 있는 노드를 선정한다. root 노드와 멤버 노드들은 멀티캐스트 트리를 구성하고 멀티캐스트 트리 연결은 유니캐스트로 한다. 멀티캐스트 데이터는 멀티캐스트 트리를 따라 전송하게 된다. loop를 없애기 위해서는 한 링크 내에 두 멤버는 부모(sub\_tree)와 자식으로 지정한다. sub\_tree는 멀티캐스트 트리, 멤버 등록에 관한 관리를 수행한다. 멤버 노드가 트리에 join 혹은 leave 할 때 네트워크 토폴로지 변화가 발생했을 때 sub\_tree노드에 보고 한다. sub\_tree 노드는 이러한 보고를 root 노드로 보고 한다. sub\_tree에 의해 주기적으로 전송된 보고를 통해 root노드는 업데이트 된 weight(처리 능력과 대역폭을 고려한 값)와 홉 수를 기반으로 steiner 트리를 구성한다.

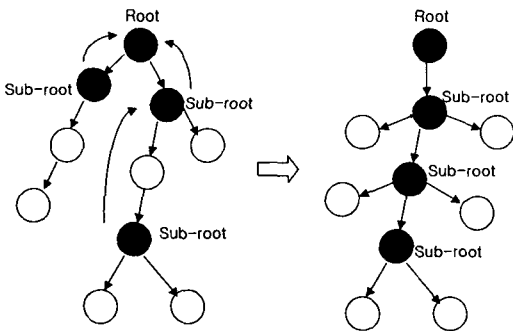


그림2. 새로 구성된 steiner 트리 모습

#### 3.2 overlay 멀티캐스트 라우팅 프로토콜

network layer 프로토콜들은 link와 라우터 자원에 직접 접근 할 수 있다. QoS를 만족하는 경로를 따라 자원을 예약하거나 네트워크 자원의 이용이 가능한지에 대한 여부도 파악할 수 있다. overlay layer 프로토콜의 경우에는 overlay 경로 상에 있는 이용 가능한 자원에 직접 접근 할 수 없고 network layer 자원을 컨트롤하거나 다룰 수 없다. 이러한 이유로 이용 가능한 대역폭 정보를 얻기 위해서는 measurement 방법에 의존해야 한다. 각 노드는 자신의 운영체제에게 수시로 자신의 처리 능력을 query를 통해 체크해야 한다. 각 노드는 이 두 가지 요소를 고려한 가중치 값을 가지고 있다. 가중치 계산 방법은 아래와 같다.

overlay link(i, j)의 가중치

$$\text{weight}(i, j) = 1 / B_{ij} * 1 / C_j$$

$B_{ij}$  : 이용 가능한 대역폭

$C_j$  : 이용 가능한 처리 능력

병목현상을 가지고 있는 링크의 경우 높은 가중치 값을 가지게 된다. 낮은 가중치를 가지고 있는 링크는 높은 자원 이용 가능성을 가지고 있는 경우이다. 가중치 값을 가지고 경로 비용이 결정되게 된다. 이러한 두 가지 요소를 사용해 트리 구성하고자 한다.

#### 3.2.1 멤버 발견 과정

멀티캐스트 세션으로 모든 멤버를 가상 mesh로 연결해 구성한다. 각 멤버 노드는 expanded ring search를 사용해서 멤버 발견 과정을 시작한다. Group\_REQ 메시지에 자신의 가중치 값을 포함해 TTL 값은 작은 값으로 제한한다. 어떤 노드 I가 가중치 값을 포함한 Group\_REP 메시지를 노드 J로부터 받았을 때 노드 J를 가상 mesh 안에 멤버이라고 member table에 기록한다. 기록할 때 노드 J와의 떨어진 거리와 가중치 값을 같이 기록한다. 가상 토폴로지의 최대 크기는 제한될 수 있다. 노드의 가상 멤버의 수가 최대 제한 수에 도달하면 노드는 멤버 발견 과정을 멈춘다. 만약 search TTL이 최대 값에 도달할 때 까지 아무런 멤버 노드를 발견하지 못한 경우에는 플러딩 기술을 사용한다. 이웃 발견 과정에서 Group\_REQ와 Group\_REP 메시지를 교환을 통해서 노드는 어떤 노드가 충분한 대역폭과 가중치를 가지고 있는 지 알 수 있다.

멀티캐스트 트리 구성 시 멤버 노드들 중에서 충분한 두 가지 요소를 가지고 있는 노드가 sub\_root 노드가 된다. sub\_root 노드가 된 노드는 멤버 노드들을 관리하게 되고 토폴로지 변화 시 root 노드에게 보고하게 된다. root 노드는 보고 받은 정보를 기반으로 새로

은 steiner tree 구성 알고리즘을 사용해 tree를 계산하는 역할도 한다.

### 3.2.2 트리 구성

트리 구성 방법으로는 steiner 트리 구성 알고리즘을 사용한다[1]. 소스노드는 총비용을 최소화한 멀티캐스트 virtual steiner 트리를 만든다. 소스 노드 주변의 가상 mesh는 이웃 발견 과정에서 발견된 member table을 정보를 참조하여 트리를 형성한다. 이러한 이유로 각 소스 노드 가까이 있는 멤버 노드들과의 정확한 토폴로지 정보를 유지하기 위해 주기적인 상태 정보를 교환한다. tree 생성 중 같은 비용을 가지는 두 개의 가상 링크가 존재하면 소스 노드와 가까이 있는 것을 선택한다.

소스 기반의 steiner 트리 알고리즘

$c(n1, n2)$ 는 가상 링크  $(n1, n2)$ 의 비용

적절한 비용(adapted cost)은 위의 두 가지를 곱한다.

$ac(n1, n2) = weight * c(n1, n2)$

adapted cost 사용하여 최소한의 총비용을 가지는 steiner tree 구성하기 위해 휴리스틱 알고리즘을 사용한다. 초기 트리 생성 시 소스 노드는 root로 정한다. 소스 기반의 steiner 트리 알고리즘 적용에 의해서 소스는 멀티캐스트 트리 내에 그들의 이웃을 child으로 만든다. 이 중 weight 값이 충분한 child 노드가 sub\_root가 되어 다른 서브 그룹들을 형성한다. sub\_root 노드의 가중치는 sub\_root 노드로써 수행할 수 있는 기준 weight\_threshold 값을 넘어서야한다. sub\_tree 노드가 자신이 관리하는 멤버 노드들의 상태를 살피고 주기적으로 root 노드에게 보고 함으로써 효율적인 트리 구성을 구성할 수 있다. 모든 두 노드 사이에는 parent 와 child 관계를 가지게 된다. parent 노드는 child 노드에게 패킷 전송 시 헤더 안에 각각의 서브 그룹 대한 주소를 넣고 헤더와 복사한 데이터를 붙인 후 패킷을 상용하는 child에게 유니캐스트로 전송한다. 각 parent 노드 역시 자신의 하위에 위치한 모든 노드들에게 데이터 패킷을 전달할 책임을 가지고 있다. 자신의 하위에 멤버 노드가 더 이상 없게 되면 처리는 멈추게 된다. root 노드가 sub\_tree 노드에서 보고된 정보를 이용해 전체적인 트리를 구성하기 때문에 확장성에는 한계를 가지고 있다. 본 논문에서 제시된 알고리즘은 스케일이 작은 네트워크에 적용할 수 있다.

ad hoc 네트워크는 수시는 노드가 동적으로 이동하는 특성을 가지고 있다. 만약 sub\_root 노드가 멀티캐스트

tree에서 탈퇴하거나 다른 곳으로 위치 이동 했을 때 혹은 루트 노드로서 역할을 수행하기에 자원이 부족한 상태 일 경우 더 이상 sub\_root로써의 역할을 할 수 없게 된다. 이때 sub\_root 노드는 자신 member 테이블에 있는 이웃들 중에서 가장 작은 가중치 값을 가지는 노드를 검색하게 된다. 이중 가중치가 작은 즉 충분한 대역폭과 처리 능력을 가진 노드를 새로운 sub\_root 노드로 선정해 선정된 노드에게 Group\_LV 메시지를 전송한다. Group\_LV 메시지에는 sub\_root 노드의 상위 노드의 주소를 포함한다. 새롭게 선정된 루트 노드는 Group\_LV 메시지 안에 있던 상위 노드에 자신이 새로운 sub\_root 노드임을 알리고 sub\_root 노드로서 역할을 수행하게 된다.

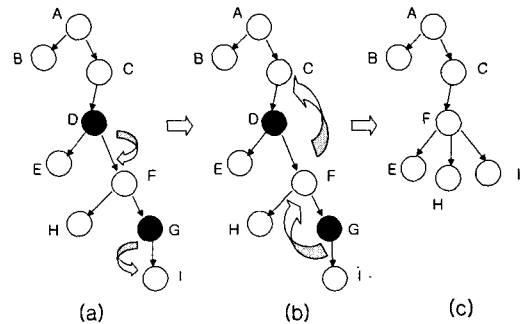


그림2. 트리 구성 과정

그림2에서 노드 D와 G가 그룹에서 탈퇴할 때 노드D는 sub\_root 노드로써 자신의 이웃 중 작은 가중치를 가지고 있는 노드 F에게 sub\_root 노드로써의 역할을 위임하고 자신의 탈퇴 여부를 알리게 된다. 노드 F는 자신이 sub\_root 노드가 되었음을 자신의 상위 노드 C에게 알리고 새로운 토폴로지 구성을 하게 된다. 노드 G역시 자신의 child 노드에게 이탈을 알리고 자신의 상위 노드에 대한 정보를 알려준다. 노드I 역시 메시지 안에 있는 상위 노드로 재 가입 하게 된다.

### 3.2.3 join and leave

기존의 join, leave과정과는 달리 overlay 멀티캐스트는 단순한 방식을 사용한 dynamic한 멤버쉽 관계를 유지한다.

a)join 과정

1) 노드가 멀티캐스트 그룹에 join 하고자 할 때 앞에서 설명했던 것과 같이 멤버 발견 과정을 시작한다. 다양한 그룹이 네트워크 내에 공존하기 때문에 Group\_REQ 패킷에 그룹 주소를 포함해 브로드캐스트

트 한다.

2) 의도된 그룹에서 멤버 노드가 Group\_REP 메시지로 응답함으로써 자신의 가상 이웃들에 대한 정보(전송 포트)를 얻고 자신과 링크를 형성할 수 있다. Group\_REP를 보낸 노드는 충분한 자원을 가지고 있는 노드여야만 한다. 링크 형성 시 Group\_REP를 보낸 노드는 parent 노드가 되고 새로 가입한 노드는 child 노드가 된다. 이러한 방식으로 loop를 생성을 막을 수 있다.

3) 해당 그룹 노드들은 그들의 이웃으로 새로운 멤버가 들어오게 되면 그들은 새로운 멤버와 링크 상태 정보를 교환하게 된다.

4) 새로운 멤버는 멀리 있는 소스노드가 자신을 인식하지 못한 상태라도 자신의 가상 이웃에 의해서 데이터 패킷을 받을 수 있다. 데이터 전송은 멀티캐스트 트리를 hop by hop 방식으로 전송한다. 두 인접한 멤버 사이에 데이터 전송의 reliable와 unreliable는 TCP 혹은 UDP 사용에 따라 달라지게 된다.

멀티캐스트 트리에서 각 parent 노드는 자신의 서브 그룹 안에 속한 모든 노드들에게 데이터를 전달할 때 모든 이웃 수신자들이 자신의 하위 그룹 안에 속해 있는지 여부를 판단하게 된다. 새로운 멤버가 여러 개의 가상 이웃을 가지고 있다면 같은 데이터 패킷에 대해 중복된 패킷을 받을 것이다. 중복된 패킷은 버리게 된다. 소스 노드가 새로운 멤버를 인식하고 후에 새로운 트리를 생성하면 더 이상 중복된 패킷을 받지 않는다.

#### b) leave 과정

1) 그룹을 떠날 경우에는 각 멤버 노드는 Group\_LV 메시지를 현재 가상 이웃들에게 유니캐스트로 전달한다.  
2) 해당 멤버와 이웃들 간에 이루어진 링크 state 교환 역시 멈추게 된다. 그것이 잠시 동안 그룹 내에 여전히 존재하더라도 데이터 패킷을 전달되지 않게 된다.

#### 3.3 control 패킷을 종류와 동작 방법

제안하는 overlay 멀티캐스트는 root 노드와 멤버 노드들의 통신을 위해 control 프로토콜에 의존한다. control 프로토콜은 멤버십 관리와 performance 관리에 관련된 일을 한다[2].

##### 3.3.1 메시지 종류

-Group\_JOIN : 멀티캐스트 그룹에 참여하려는 노드가 멀티캐스트는 멤버 노드에게 전달하는 메시지이다.  
-Group\_REP: 멀티캐스트에 참여하려는 멤버에게 주변의 동료 멤버 리스트 정보와 통신 포트 정보를 담아 응답해준다.

-performance monitoring: sub\_root 노드가 관리하는 토폴로지 변경 시 root 노드에게 weight 값과 흡수율

포함해서 주기적으로 전송하는 메시지이다. root 노드는 이 값을 주기적으로 업데이트한 후에 steiner 트리를 계산할 때 사용한다.

-Group\_LV: 멀티캐스트 그룹에서 나갈 때 노드가 그들의 부모, 자식 노드에게 이 사실을 알릴 때 사용한다. 만약 child 노드가 부모 노드에게서 이 메시지를 받는다면 그들은 그룹에 재가입해야 한다.

## IV. 결론 및 향후 연구 방향

본 논문에서는 최근 주목 받고 있는 overlay 멀티캐스트를 ad hoc 네트워크 망에 적용하는 방안을 제시했다. ad hoc 네트워크의 잦은 토폴로지 변화로 인한 적은 대역폭과 처리 능력을 고려한 멀티캐스트 트리 구성 방법을 제시하고 있다. 모든 노드들은 자신의 이웃 노드들의 weight값(대역폭과 처리 능력을 고려한 값)에 대한 정보를 가지고 있다. 이 값을 보고 가장 작은 weight 값을 가진 노드를 sub\_root 노드로 선정한다. root 노드는 sub\_root에 의해 보고 되는 정보를 통해 노드들을 정보를 파악하여 steiner 트리를 휴리스틱 알고리즘을 사용하여 구성한다. sub\_root 노드는 항상 이웃 노드를 주시하고 멀티캐스트 트리를 관리한다. data 전송을 이렇게 구성된 트리를 통해서 할 수 있다. 이러한 overlay 트리는 멤버 노드들 사이에서 logical하게 연결해 형성한다. 근본적인 멤버 노드들 사이에는 유니캐스트 방식의 통신하고 있다. 유니캐스트 방식 통신에는 현재 ad hoc 네트워크에서 활발하게 연구된 DSR, AODV, DSDV, TORA 등 많은 프로토콜이 있다. 이러한 기본 프로토콜에서 QoS를 보장한다면 overlay 트리를 통한 통신을 더 원활하게 할 수 있을 것이다.

### 참고문헌

- [1]Chao Gui and Prasant Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," Wireless Communication and Networking Conference(WCNC), 2003
- [2]Dimitrios Pendarakis and Sherlia Shi, "ALMI: An Application Level Multicast Infrastructure," Proceedings of the 3rd USNIX Symposium on Internet Technologies and Systems (USITS '01)
- [3]고석주, 강신각, "인터넷 멀티캐스트 신기술 동향" 전자통신동향분석 제16권 제2호 2001년 4월