

# 유무선 통합 환경에서 QoS 향상을 위한 멀티캐스트 라우팅 알고리즘

## Multicast Routing Algorithm for QoS Improvement in the Wire/Wireless Integrated Environment

김 미 혜

Mi-Hye Kim

### 요 약

본 논문은 유무선 통합 환경에서 QoS 향상을 위한 멀티캐스트 라우팅 알고리즘을 제안하고, 통합환경에서 QoS의 요구사항 및 특징을 분석하여 유선망에서 노드 이동성을 추가하여 유무선 통합환경에서도 QoS 향상을 가능하게 해주는 새로운 알고리즘을 제안하였다. 이 새로운 알고리즘은 모바일 노드의 이동시 패킷 손실을 방지하고 대역폭을 절약하여 지연을 최소화하는 동적 멀티캐스트 트리를 구성한다. 다른 멀티캐스트 알고리즘과 비교한 시뮬레이션 결과 새로운 알고리즘은 유무선 통합망에서 가장 절약된 대역폭으로 지연이 비교적 적은 멀티캐스트 트리를 구성할 수 있음을 증명하였다.

### Abstract

In this paper, we proposed a multicast routing algorithm for QoS improvement in the wire/wireless integrated environment. We analyzed requests and characteristics of QoS, and then proposed a new algorithm that can improve QoS by adding node mobility to wire environment. This new algorithm constructs a dynamic multicast tree that can prevent a loss of packet and save the bandwidth. As a result of simulations comparing to another multicast algorithms, we showed that this new algorithm can simply and dynamically adjust the construction of multicast tree with little delay and the most reducible bandwidth resources.

Key words : QoS, Multicast Routing Algorithm, M\_Dijk Algorithm

### I. 서 론

현재 무선망과 인터넷과의 통합은 언제, 어디서든 자유로이 인터넷을 접속하여 이용할 수 있는 새로운 응용 영역이 될 것이며, 이러한 통합은 기존의 무선망과 인터넷의 자원을 비용측면에서 효율적으로 이용할 수 있어야 하며, 인터넷의 경우 기존의 전통적인 유선기반의 통신방식에서 다양한 방식의 유무선망과 통합되어 서비스를 제공할 것이며 응용 범위도 기업이나 대학 이외에 이동단말, 가전기기 등과 같

이 다양해져야 할 것이다. 향후 통신은 무선통신과 인터넷이 주요 통신 수단이 될 것이며, 이들을 중심으로 발전하고 통합되는 형태로 진화해 나갈 것으로 예측된다<sup>[1]</sup>.

무선환경에서는 노드 이동성이 있기 때문에, 송신노드와 수신노드가 최초로 협상하여 결정된 자원을 이용하여 패킷을 송수신하고 노드가 다른 지역으로 이동을 하게 되면 새로운 지역에서 자원에 대해 재협상을 해야 한다. 협상이 진행되는 동안에는 이동노드가 패킷을 수신하지 못하기 때문에 핸드오프 지

충북대학교 전기전자컴퓨터공학부(School of Electrical & Computer Engineering, Chungbuk National University)

· 논문 번호 : 20040420-049

· 수정완료일자 : 2004년 5월 14일

연이 발생되고, 이는 서비스 질의 저하를 초래하게 된다. 따라서 무선망이 제공하는 요소 기능인 핸드 오버나 로밍을 유연하게 제공하는 QoS 관리가 반드시 제공되어야 한다.

이를 위해 본 논문에서는 유선망에 노드 이동성을 추가하여 유선망뿐 아니라 유무선 통합환경에서도 QoS 향상을 가능하게 해주는 M\_Dijk 알고리즘을 제안하였다. 이 M\_Dijk 알고리즘에 따라서 각 수신자는 유무선 통합환경에서 망 자원을 효율적으로 활용할 수 있게 되어 가장 절약된 대역폭으로 지연이 비교적 작은 멀티캐스트 트리를 구성할 수 있게 된다.

## II. 유무선 통합 기술

유무선 통합망의 경우 다양한 이질망을 수용하고 있으므로 이질망 하부계층의 QoS 특성을 상위계층에서 하나하나 지정하는 것은 구조가 복잡해진다. 따라서, 상위 응용계층에서 단말의 해당 응용 서비스 QoS Profile을 확인하게 되면 곧바로 종단간 세션 설정의 파라미터로 적용된다. 즉 그림 1과 같은 구조를 가진다. 결국 지역 QoS의 협력을 통해 전체적인 글로벌 QoS를 생산적으로 가지는 구조가 되는 것이다<sup>[2],[3]</sup>.

그림 2는 Framework에서 네트워크 접속과 핸드오프 때 일어나는 시나리오를 나타낸다<sup>[4]</sup>.

- ① 이동호스트는 지역 에이전트에 MIP, SIP 등록을 수행한다.
- ② 이동호스트가 상대방 호스트와 응용계층의 연결을 시도할 경우, 사용자 QoS Profile에 기반하

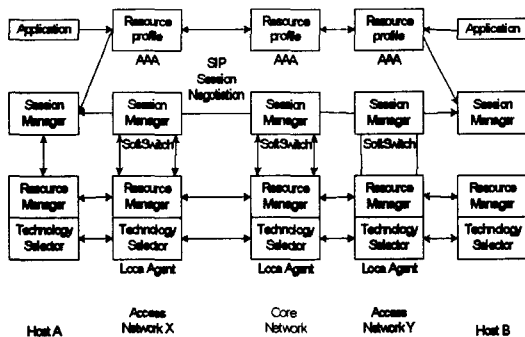


그림 1. 유무선 통합망 QoS 지원 프레임워크  
Fig. 1. QoS support framework of wire/wireless integrated network.

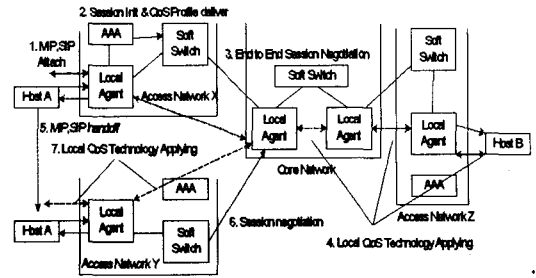


그림 2. 유무선 통합망의 QoS 지원 시나리오  
Fig. 2. QoS support scenario of wire/wireless network.

여 자원획득 권한을 AAA 서버에게 묻고 AAA 서버는 QoS Profile 정보를 SIP 세션 관리자에게 전달하게 된다.

- ③ SIP 세션 관리자는 SIP에 의존하여 종단간 세션 협상을 맺고, 이때 AAA 서버로부터 전달받은 QoS Profile 정보를 기반으로 QoS 협상 또한 맺게 된다.
- ④ 지역망 및 핵심 망 내의 QoS 자원관리를 맡게 되는 지역 에이전트들은 SIP 세션관리자의 종단간 세션 협상에 있어서 지역망의 특성에 알맞은 QoS 적용 기술들을 선택하여 지역망 내부 및 지역망 간 적용을 수행한다.
- ⑤ 호스트가 이질망으로 이동할 경우 지역 에이전트에 MIP, SIP 등록을 하여 핸드오프 절차를 수행한다.
- ⑥ 동질망이 아니고 이질망일 경우 QoS 적용은 또 다시 일어나야 한다. 따라서 이질망의 경계까지 세션 협상이 다시 일어나게 된다.
- ⑦ 세션 협상이 일어나는 구간까지 지역망의 특성에 알맞은 QoS 적용기술을 선택하여 적용하게 된다.

## III. QoS 향상을 위한 M\_Dijk 알고리즘

제안하는 M\_Dijk 멀티캐스트 라우팅 알고리즘에서는 QoS 요구가 최소 대역폭 요구가 주어진다고 가정하고, 링크 상태(link state) 라우팅에 기반하여 적정 경로를 계산한다. 따라서 라우터 서버가 파악해야 하는 QoS 메트릭은 각 링크의 가용 대역폭이다. 경로 계산 알고리즘으로는 변형된 Dijkstra 최단 경로 알고리즘을 사용한다. 변형된 Dijkstra 알고리

좁은 QoS 경로를 계산하는 시점에 링크의 가용 대역폭이 클라이언트가 요구한 대역폭보다 작게 남아 있는 링크들을 모두 제거한 후 남은 네트워크 토폴로지 상에 Dijkstra 알고리즘을 적용하여 widest-shortest 경로를 계산하는 것을 말한다.

새로운 알고리즘에서 데이터를 전송하기 위해서는 크게 경로 탐색과정, 확인 및 자원 예약 과정으로 나누어진다.

### 3-1 핸드오프 과정

모바일 노드를 MN(Mobile Node)이라고 부르고, 통신 상대방 노드는 CN(Corresponding Node)이라고 부른다. 모바일 노드는 홈 주소 HoA(Home Address)라고 불리는 고유한 IP 주소를 갖는데, 이 주소는 모바일 노드의 현재 위치와 무관하다. HoA를 포함하는 서브네트워크를 그 모바일 노드의 홈 네트워크라고 하며, 홈 네트워크 안에 있는 라우터인 홈 에이전트 HA(Home Agent)는 모바일 노드가 홈 네트워크를 떠나 있을 때 모바일 노드에게 보내진 패킷들을 전달하는 역할을 수행한다. 모바일 노드가 홈 네트워크에 위치할 때는 모바일 노드는 일반 노드와 전혀 차이가 없다. 모바일 노드가 홈 네트워크를 떠나 다른 곳에서 인터넷에 연결될 때, 접속 위치의 네트워크를 외지 네트워크, 외지 네트워크의 라우터를 외지 에이전트 FA(Foreign Agent), 외지 네트워크에서 임시로 부여받은 IP 주소를 위탁주소 CoA(Care of Address)라고 부른다.

Mobile IP의 동작은 다음과 같다.

- ① HA와 FA는 주기적으로 자신의 정보를 광고로 알린다. 이때 Agent 광고라는 Mobile IP 메시지를 사용한다.
- ② 이동노드는 이 Agent 광고메시지를 받아보고 자신이 어느 링크에 있는지를 판단한다. 자신이 Home 링크에 있으면 일반 IP 프로토콜로 동작하고, Foreign 링크에 있는 경우만 아래의 순서로 Mobile IP를 실행한다.
- ③ Foreign 링크에 있는 이동노드는 먼저 CoA를 얻어야 한다. FA CoA를 얻는 경우는 단순히 FA가 보낸 Agent 광고메시지 내에서 FA CoA를 읽어 사용하면 된다.
- ④ 이동노드는 위에서 얻은 CoA를 자신의 HA에 홈

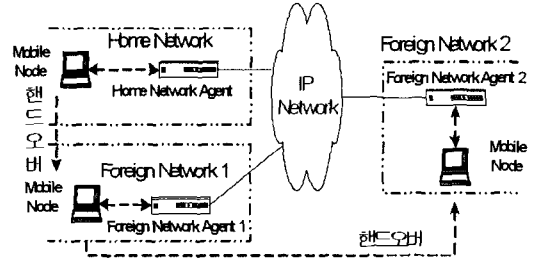


그림 3. 시스템 모델  
Fig. 3. System model.

등록하고 MN에 대해 검증한다.

- ⑤ 이동노드가 트리에 가입 신청할 때 HA 주소를 이용해서 신청한다.
- ⑥ 다음에 이 이동노드에 오는 모든 패킷들이 HA를 경로해서 FA에게 전달한다.

그림 3은 MN이 FA1에서 FA2로 이동한 뒤의 과정이다.

- ① MN은 FA1에서 FA2로 이동을 하고 FA1을 통해 MN에게 가던 1, 2번 패킷은 FA1의 Storage\_Buffer에 저장된다.
- ② HA에서 FA2를 통해 MN에게 보내지는 3, 4번 패킷은 FA2의 Storage\_Buffer에 저장된다.
- ③ FA1의 Storage\_Buffer에 저장되어 있던 패킷들을 FA2의 Provisional\_Buffer로 전송한다.
- ④ FA2는 FA1의 Storage\_Buffer에 있던 1, 2번 패킷이 FA2의 Provisional\_Buffer로 전송이 완료되면 Handoff\_End 메시지를 보내어 전송이 모두 완료되었음을 FA1에게 알려주고, FA1을 석방한다.
- ⑤ FA2는 먼저 FA2의 Provisional\_Buffer에 저장되어 있던 1, 2번 패킷을 MN에게 전송하고, Storage\_Buffer에 저장되어 있던 3, 4번 패킷을 MN에게 전송한다.

### 3-2 M\_Dijk 알고리즘

제안하는 M\_Dijk 알고리즘에서는 QoS 요구가 최소 대역폭 요구가 주어진다고 가정하고, 링크 상태(link state) 라우팅에 기반 하여 적정 경로를 계산한다. 따라서 라우터 서버가 파악해야 하는 QoS 메트릭은 각 링크의 가용 대역폭이다. 경로 계산 알고리즘으로는 변형된 Dijkstra 최단 경로 알고리즘을 사용한다. 변형된 Dijkstra 알고리즘이란 QoS 경로를

표 1. 알고리즘의 표기법  
Table 1. Marking methods of algorithm.

주요 파라미터 표기법	
$V$	네트워크상에 노드들의 집합
$E$	네트워크상에 링크들의 집합
$k$	노드의 개수
$s$	송신지 노드
$d$	수신지 노드
$M$	$s$ 로부터의 최소 비용 경로가 이미 발견된 노드들의 집합
$d(i, j)$	노드 $i$ 와 $j$ 사이의 비용 $d(i, j) = 0$ ; $d(i, j) = \infty$ 두 노드 사이에 연결된 링크가 없는 경우
$D(n)$	$s$ 로부터의 $n$ 노드까지의 최소 비용 경로의 비용
$b$	세션이 요구한 대역폭
$B(n)$	$s$ 로부터의 $n$ 노드까지의 최소 비용 경로의 대역폭

계산하는 시점에 링크의 가용 대역폭이 클라이언트가 요구한 대역폭보다 작게 남아 있는 링크들을 모두 제거한 후 남은 네트워크 토폴로지 상에 Dijkstra 알고리즘을 적용하여 widest-shortest 경로를 계산하는 것을 말한다.

본 알고리즘에서 사용되는 주요 파라미터는 표 1과 같다.

```

M_Dijk (s, d, b);
begin
    M := {s};      /* M의 초기값을 s로 둔다*/
    for each node n in (V-M) do D(n) = d(s, n);
        /* 송신지 노드와 그 외의 노드들 사이의
           비용을 구한다. */
    While d ⊆ M do
    begin
        Find a node w in (V-M) such that D(w) is
        minimum;
        /* 수신지 노드가 M에 속한다면, 최소 비용
           이 되는 노드 w를 찾는다 */
        if (V - M) = 0 then
            return FALSE;
        else
    
```

```

        add w to M;      /* V와 M 사이에 다른 노드
                           w가 있을 경우 M에 추가 */
        if w = d then
            return TRUE;
        else
            for each node n in (V-M) do D(n) := Min
                (D(n), D(w) + d(w, n));
            if (D(w)+d(w, n) = D(n)) then
                D(n) = Random ( );
            else return TRUE;
        end //While
    end //M_Dijk

    Random ( );
    begin
        select random path with probability based on its
        B(w);
        return path;
    end

```

제안한 M\_Dijk 알고리즘과 기존의 Dijkstra 알고리즘의 차이점은 다음과 같이 3가지로 정리할 수 있다.

첫째는 M\_Dijk 알고리즘을 통해서 멀티캐스트 트리를 생성할 때 망자원을 절약할 수 있다. 즉, M\_Dijk 알고리즘을 통해서 멀티캐스트 트리를 생성할 때 경로의 비용 같은 경우 대역폭을 제일 적합한 경로를 선택한다.

둘째는 M\_Dijk 알고리즘은 유선환경에 뿐만 아니고 무선환경까지 확장해서 사용할 수 있다. M\_Dijk 알고리즘은 FA 내에 존재하는 두 가지 버퍼(Storage\_Buffer 및 Provisional\_Buffer)를 이용하여 패킷의 손실에 대한 문제를 해결한다.

셋째는 M\_Dijk 알고리즘은 이동 노드가 다른 네트워크로 이동하였을 때 실행 중이던 애플리케이션의 중단 없이 사용할 수 있다.

### 3.3 동적 트리 생성 및 유지 관리

다음은 M\_Dijk 알고리즘을 이용한 트리의 생성 과정을 나타내었다.

- ① /\* Tree의 구성 \*/
- ① if Bandwidth(max) < 요구된 대역폭 값  
/\* Bandwidth(max): Link의 최대 대역폭 값 \*/

- ② Return Failed /\* 실행가능한 tree가 없다 \*/
- ③ Bandwidth  $\leftarrow$  0; Delay(s)  $\leftarrow$  0  
 $u \neq s$ 때 Bandwidth  $\leftarrow$   $\infty$ ; Delay(u)  $\leftarrow$   $\infty$
- ④  $T \leftarrow \emptyset$ ,  $Q \leftarrow V$  /\* V: 모든 링크들의 집합 \*/
- ⑤ while  $Q \neq \emptyset$  and  $\text{Min}(Q) \geq$ 요구된 대역폭 값 do
- ⑥  $u \leftarrow \text{Min}(Q)$  /\* next node를 선택한다 \*/
- ⑦  $T \leftarrow T \cup \{u\}$
- ⑧ for each top  $v \in \text{Adj}[u]$  /\* for each 인접된 node \*/
- ⑨ if  $\text{Bandwidth}(u, v) \geq$  요구된 대역폭 값 and  $v \notin T$
- ⑩ if  $\text{Delay}[v] > I_D(u) \text{ Delay}(u) + \text{Delay}(u, v)$   
 /\* node-v는 Tree에 속하지 않고 tree에 신청하고자한 node \*/
- ⑪  $\text{Delay}(v) \leftarrow I_D(u) \text{ Delay}(u) + \text{Delay}(u, v)$
- ⑫ 부모 node  $\leftarrow u$  /\* u는 부모 node가 된다 \*/

이렇게 생성된 Tree는 다음과 같이 관리되어진다.

트리에 가입할 때 기본 연산은 아래와 같다.

- JoinRequest(Group-id, NewMember-id, TTL):  
 자신의 id 정보(IP Address, Port No.)와 TTL 정보를 이용하여 그룹에 참여
- JoinAccept (Group-id, LGroupHeader-id, Receiver-id, TTL, (Parent-id, TTL<sub>p</sub>)):  
 지역 그룹 대표자(LGroupHeader) 정보와 TTL 정보를 포함하며, 대표자의 부모 노드의 정보 (Parent-id, TTL<sub>p</sub>) 등을 제공하여 지역 Tree에 참여

새로운 노드가 그룹에 참여하고자 할 때는 자신의 지역 그룹 대표자를 찾기 위하여 TTL 값을 1부터 증가시키며 참여 요청 메시지를 멀티캐스트 한다. 이 때 그룹 대표자는 좀 더 멀리 떨어져 있고 이미 그룹에 참여한 일반 수신자가 해당 메시지를 수신했다면 해당 영역의 지역 그룹 대표자 정보(ip address, TTL<sub>lg</sub>)를 송신한다. 새로운 노드는 송신 받은 지역 그룹 대표자 정보를 이용해 곧바로 유니캐스트 참여 요청을 한다. 만일 요청에 대해 여러 개의 지역 그룹 대표자에 대한 정보를 수신한다면 TTL 값이 가장 작은 대표자를 선택한다.

다음은 이 과정을 알고리즘으로 나타내었다.

- A member want to join a active group
- TTL= 1
- while TTL  $\leq$  255 or receive JoinAccept
- multicast JoinRequest
- if a LGH receives it
- if  $N(LG) < M_{\text{thresh}}$  // N(LG): 지역그룹 수신자수
- send JoinAccept
- if a Receiver receives it
- send LGH\_id, TTL<sub>lg</sub> // LGH: 지역그룹대표자

일반 수신자는 자신의 지역 그룹 대표자에 연결하면 자신의 대표자 이 외에 그의 부모 노드에 대한 정보(IP 주소, TTL<sub>p</sub>)를 저장한다. TTL<sub>p</sub>는 지역 대표자와 부모 대표자와의 TTL 값을 나타낸다.

이 정보를 이용하여 자신의 대표자가 오류로 인하여 응답 불가 상태가 되었을 경우 새로운 대표자를 신속하게 선정할 수 있다. 아래의 알고리즘은 이 과정을 나타낸 것이다. 일단 자신의 대표자가 예고 없이 그룹에서 이탈한 것을 감지하면 수신자는 새로운 대표자를 찾기 위해 TTL 값을 다시 1부터 ERS를 실행하는 것이 아니고 부모 노드에게 참여 요청을 하여 세션의 진행을 유지한다. 이 후 수신한 TTL<sub>p</sub> 값을 이용하여 최소 TTL 값을 갖는 새로운 대표자를 선정한다.

다음은 그룹 수신자가 LGH fail 감지 후 새로운 LGH 설정 과정을 보여준다.

- A receiver detect failure of LGH
- Report LGH\_fail into its local group
- send JoinRequest to parent\_of\_LGH
- while TTL  $<$  TTL<sub>p</sub>
- TTL = TTL<sub>lg</sub>+TTL<sub>p</sub> / 2
- // find if there is closer LGH than parent\_of\_LGH
- multicast JoinRequest with new TTL
- if receive any JoinAccept
- make new connection with the new LGH

그룹이 동적으로 변화하는 과정에서 하나의 지역 그룹 대표자에게 소수의 수신자만이 연결되는 형태가 반복되어 깊이가 깊어지는 트리가 형성될 위험이 있다. 이 경우 트리의 깊이에 따른 부하가 증가하는 특성 때문에 비효율적인 그룹 관리가 될 수 있다. 그러므로 본 논문에서는 지정된 임계값  $M_{\text{thresh}}$ 를 이

용하여 간단한 그룹 재조정을 동적으로 수행할 수 있도록 한다. 하나의 지역 그룹 대표자는 자신의  $M_{thresh}$  값을 자신의 부모 노드에게 주기적으로 전송하고 초기 트리 형성 과정을 벗어나면 부모노드는  $M_{thresh}/n$  값 이하인 지역그룹을 통합한다. 다음은 불균형 Tree를 만들 여지가 있는 지역 노드간의 병합 과정을 나타낸 알고리즘이다.

Local Group Tree Merge

```

if  $N(LGi) < M_{thresh}/n$  and
    $N(LGj) < M_{thresh}/n // i \neq j,$ 
   if tree level of  $i$  and  $j$  are the same
     LGHi becomes new LGH // LGHi:
       왼쪽 지역그룹대표자
     merge LGi and LGj
   if  $i$  is a parent of  $j$ 
     LGHi becomes new LGH
     merge LGi and LGj
    
```

이렇게 국소적으로 그룹을 통합하는 것을 궁극적으로는 전체 트리의 깊이가 깊어지는 것을 감소시킬 수 있게 되어 불균형 트리로 인해 발생하는 오버헤드를 줄일 수 있다.

IV. 실험 및 결과 분석

4.1 실험환경

본 논문에서 제안한 M\_Dijk 알고리즘의 성능을 평가하기 위해 이미 제안된 Dijk 알고리즘의 성능과 비교하였다. 성능비교는 네트워크 이용률, 경로 설정 성공률, 네트워크 비용 등 여러 방면에서 성능을 분석하였으며 Dijk 알고리즘<sup>[6]</sup>과 제안한 M\_Dijk 알고리즘을 비교하여 그래프로 나타내었다.

시뮬레이션에서 사용한 네트워크 크기는 노드 수를 5노드로 구성했으며 회선은 전이중 방식으로 155 Mbps로 구성하고, 각 회선의 큐잉 요소는 무시하였다. 또한, 회선의 지연은 대칭적이나 회선의 비용은 비대칭적으로 설정하였다. 즉, 회선의 길이가 같으면 지연도 같게 되지만, 회선의 비용은 다르다. 다음은 본 논문 시뮬레이션을 위해 사용한 가정들이다.

- ① 각 연결 요청 당 대역폭 요구사항은 64 Kbps에서 1.5 Mbps 범위에서 임의의 값을 사용.

- ② 연결 설정 후 데이터 전송 지속시간은 임의의 시간을 선택한다.
- ③ 연결 설정 요청은 5노드에서 무작위로 발생할 수 있으며 수신자는 송신자를 제외한 모든 노드가 될 수 있다.
- ④ 모든 회선의 지연시간은 일정하다. 따라서 송신자에서 수신자까지의 지연시간은 중간 경로 수로 결정된다.
- ⑤ 모든 알고리즘에서 사용되는 라우팅 테이블 갱신 주기는 모두 같다.

4.2 결과분석

그림 4에서 보는 바와 같이 본 논문에서 제시하는 알고리즘에서 네트워크 이용률은 네트워크 대역폭에 따른 승인된 대역폭 사용률로 정의한다. 네트워크 이용률 또한 정보의 불확실성에 의해 같은 연결요청에 의해 네트워크 이용률은 정보의 단계가 높을수록 낮아짐을 알 수 있다. 즉, 같은 수의 연결에 대해서 네트워크 이용률이 낮다는 것은 그만큼 성능이 우수하다는 것이다. 따라서 M\_Dijk 알고리즘의 성능이 가장 우수함을 알 수 있다.

그림 5는 같은 라우팅 테이블 갱신 주기를 사용하였을 때 총 연결요청 수에 대한 성공률을 보여주고 있다. 경로 설정 성공률은 총 연결 요청 수에 대해 연결이 성공한 요청 수로 정의한다. 경로 설정은 각 라우터에서 유지하는 라우팅 테이블 정보를 참조하여 목적지까지 경로를 설정하게 된다. 또한, 라우팅 테이블 정보의 정확도에 따라 경로 설정에 대한 성공률이 달라지게 된다. 즉, 라우팅 테이블 갱신주기에 따라 정보의 정확성이 달라진다. 그림 5는 각 라

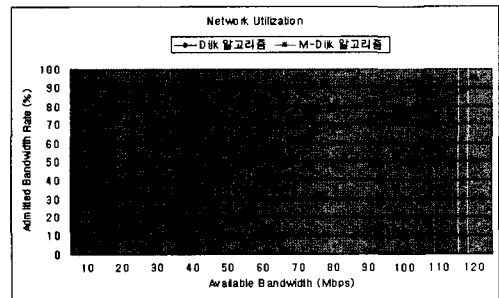


그림 4. 네트워크 이용률  
Fig. 4. Using rate of network.

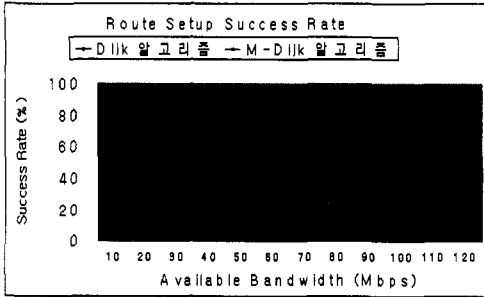


그림 5. 연결 당 경로 설정 성공률  
Fig. 5. Routing synchronization success rate per connections.

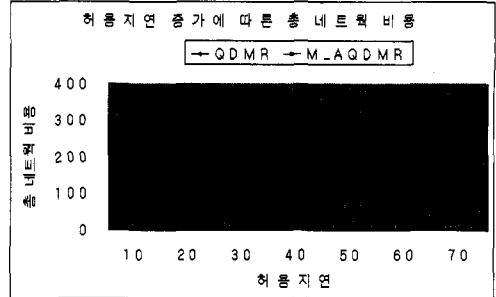


그림 7. 허용지연 증가에 따른 총 네트워크 비용  
Fig. 7. All network cost per increasing permission delay.

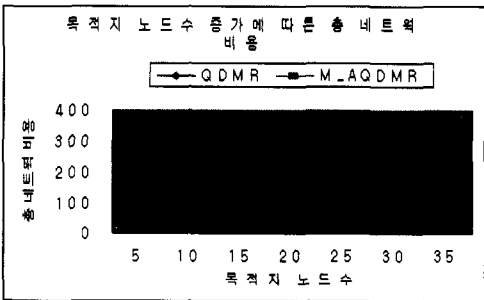


그림 6. 목적지 노드수 증가에 따른 총 네트워크 비용  
Fig. 6. All network cost per adding target node number.

우터에서 경로를 선택할 때 사용하는 라우팅 테이블 정보가 단계가 높을수록 정확성이 떨어지기 때문이다. 즉, 단계가 높을수록 네트워크 자원에 대한 상태 변화에 민감하지 못하기 때문이다. 또한, 네트워크의 대역폭이 작을수록 더 많은 연결 요청이 가능하며 이에 대한 성공률도 높아짐을 알 수 있다<sup>6)</sup>.

그림 6은 노드 개수가 50, 네트워크의 평균 연결도가 5이고, 허용지연  $\Delta=70$ 으로 동일하게 설정한 네트워크에 대해 목적지 노드 수를 증가시킬 때 기존의 멀티캐스트 알고리즘과 제안한 M\_Dijk 알고리즘의 총 네트워크 비용을 비교한 것이다. 목적지 노드수가 증가할수록 두 알고리즘 모두 총 네트워크 비용은 증가하지만, M\_Dijk 알고리즘이 효율적이라는 것을 알 수 있다.

그림 7은 허용지연의 증가에 따른 Dijk 알고리즘과 제안한 M\_Dijk 알고리즘의 성능을 비교한 것이다. 전체 노드의 개수가 70개, 네트워크의 평균 연결도가 7로 이루어진 네트워크에서 목적지 노드의 개수가

15로 주어진 경우에 대한 실험 결과이다. 그림에서 보면 허용지연 값이 작을 경우에는 Dijk 알고리즘은 제안한 M\_Dijk 알고리즘보다 더 작은 네트워크비용을 갖지만, 허용지연의 값이 증가하면 제안된 M\_Dijk 알고리즘이 더 좋은 성능을 나타낸다<sup>7)</sup>.

## V. 결론

본 논문에서는 유무선 통합환경에서 QoS 향상을 위한 멀티캐스트 라우팅 알고리즘 M\_Dijk를 제안한 Path 트리를 바탕으로 연구되었으며 통합환경에서 QoS를 향상하기 위한 효과적인 트리를 구축하여 멀티캐스트 라우팅을 실행할 수 있게 하였다.

M\_Dijk 알고리즘은 새로운 노드가 멀티캐스트 트리에 참여할 때 새로운 노드가 요구한 대역폭에 가장 가까운 대역폭을 선택하여 효과적인 트리를 구성하는 알고리즘이다. 네트워크 이용률, 연결당 경로 설정 성공률, 목적지 노드수 증가에 따른 총 네트워크 비용, 허용지연 증가에 따른 총 네트워크 비용 분석 등의 시뮬레이션 결과, M\_Dijk 알고리즘은 기존의 알고리즘에 비해 대역폭을 효율적으로 활용하면서 실행시간을 감소시킬 수 있으며 또한 가장 절약적인 네트워크 비용으로 지연이 비교적 적은 멀티캐스트 트리를 구성할 수도 있어 대규모 통합네트워크 상에서 QoS를 향상시킬 수 있는 효과적인 멀티캐스트 라우팅 알고리즘임이 증명되었다.

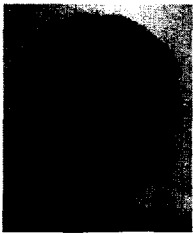
향후, 통신망은 통합 멀티미디어 서비스를 제공하는 방향으로 진화할 것이며, 현재까지 각 망의 QoS 지원정책이 수립되지 않는 상황이므로 이질성망의 통합에 따른 종단간 QoS 보장을 위한 자원 확보 및

유무선 통합환경에서의 QoS 정책이 수립되어야 할 것이다.

### 참 고 문 헌

- [1] 이승윤, 박정수, 김용진, "전세계 차세대 인터넷 망 구축 동향", 전자통신동향분석, 15(6), 2000년 12월.
- [2] 정일영, "차세대인터넷으로 진화방안에 관한 연구", 한국전산원 위탁과제 최종 보고서, 1999년.
- [3] 박상원, "차세대 유무선 통합망 구성", 텔레콤, 16 (1), pp. 66-76, 2000년.
- [4] 최태상, 정태수, "인터넷 QoS 기술 및 표준화 동향", 정보통신기술 15(2), pp. 5-8, 2001년 12월.
- [5] A. huja, R. K., T. L. Magnanti and J. B. Orlin, *Network Flows*, Prentice-Hall Inc., 1993. (Dijkstra Algorithm)
- [6] I. Cidon, R. Rom and Y. Shavitt, "Bandwidth reservation for bursty traffic in the presence of resource availability uncertainty", *Comput Commun*, vol. 22, no. 10, pp. 919-929, Jun. 1999.
- [7] A. Shaikh, K. Shin, "Destination-driven routing for low-cost multicast", *IEEE J. Select.Areas Commun.*, vol. 15, pp. 373-381, Apr. 1997.

김 미 혜



2001년 2월: 충북대학교 수학과 (이  
학박사)  
2001년 4월~현재: 충북대학교 전기  
전자컴퓨터공학부 초빙교수  
[주 관심분야] 퍼지이론, 정보이론