

무선 ad-hoc 네트워크 환경에서 신뢰성 있는 멀티캐스팅을 위한 RAMRP 프로토콜

정희원 박 상 윤*, 엄 영 익*

RAMRP Protocol for Reliable Multicasting in Wireless Ad-hoc Network Environments

Sang Yun Park*, Young Ik Eom* *Regular Members*

요 약

동적인 무선 ad-hoc 네트워크 환경에서는 링크 실패로 인한 경로 재설정과 패킷 재전송 등이 유선 네트워크 환경 보다 빈번하게 발생할 수 있다. 따라서, 무선 ad-hoc 환경에서 멀티캐스팅이 수행될 경우에는 이러한 특수성이 반영된 멀티캐스팅 기법이 사용되어야 한다. 즉, 멀티캐스트 트리 상의 잦은 링크의 변화에 따라 발생하는 전송 지연 및 패킷 분실을 감소시킬 수 있는 멀티캐스팅 기술의 개발이 요구되는 것이다. 본 논문에서는 무선 ad-hoc 환경에서 효율적인 멀티캐스팅을 지원하는 RAMRP (Reliable Ad-hoc Multicast Routing Protocol) 프로토콜을 제안한다. RAMRP 프로토콜은 링크 건강도 기반 경로 설정 기법과 대리 수신 확인 기법 및 버퍼링 기술을 사용한다. 또한, RAMRP 프로토콜은 경로 재설정 및 패킷 재전송을 감소시킴으로서 무선 ad-hoc 네트워크에 적합한 신뢰성 있는 멀티캐스팅 방안을 제공한다.

ABSTRACT

In wireless ad-hoc network environments, packet retransmissions and route reconstructions caused by link failures can be more frequent than in the wired network environments. Therefore, when the multicasting is performed in wireless ad-hoc network environments, new multicasting schemes that are designed with the consideration of these problems should be used. In other words, for multicasting in wireless ad-hoc networks, development of the multicasting technologies reducing transmission delays and packet losses that occur due to the changes of links in a multicast tree are required. In this paper, we propose a RAMRP (Reliable Ad-hoc Multicast Routing Protocol) protocol that supports an efficient multicasting in wireless ad-hoc networks. RAMRP protocol uses a link soundness based route construction method, an agent based ACK strategy, and a buffering technology. It also provides a reliable multicasting scheme suitable for wireless ad-hoc networks by reducing route reconstructions and packet retransmissions.

1. 서 론

고정 라우터나 기지국 없이 이동 호스트들만으로 유동적인 무선 네트워크를 구성한 네트워크 환경을 무선 ad-hoc 네트워크라고 한다. 무선 ad-hoc 네트워크의 구성 요소인 이동 호스트는 라우터 역할을

수행하면서 동시에 이동성을 가지므로 무선 ad-hoc 네트워크 상의 토폴로지는 수시로 변화하며 노드간의 연결도 매우 약한 특성을 갖는다. 또한 제한된 대역폭과 약한 전력으로 잦은 링크 단절과 라우팅 실패가 발생할 수 있고 특정 경로에 트래픽이 집중되는 병목 현상이 발생하기도 한다. 이러한 무선 ad-hoc 네트워크 환경에서 메모리 복사를 통해 그

* 성균관대학교 정보통신공학부

논문번호 : 020044-0128, 접수일자 : 2002년 1월 28일

를 통신을 지원하는 멀티캐스팅 기술은 대역폭의 낭비를 줄일 뿐 아니라 일-대-다 통신이 빈번할 것으로 예상되는 무선 ad-hoc 네트워크에서 유용하게 사용될 수 있다.

무선 ad-hoc 네트워크에서 멀티캐스트 라우팅이 수행될 경우 안정성과 효율성의 관계가 상충하는 문제가 발생한다. 기존의 많은 멀티캐스트 라우팅 기법들은 라우터 내에 유지되는 멀티캐스트 그룹 멤버들의 상태에 의존하여 경로 설정 및 패킷 전달을 하기 때문에 저용량, 저전력의 이동 호스트들로 구성된 동적인 무선 ad-hoc 네트워크에서는 상당한 제어 트래픽이 발생할 수 있고 이로 인해 경로 설정에 많이 시간이 소요될 수 있다. 그러므로 경로의 안정성과 효율성을 모두 개선시킬 수 있는 새로운 기법이 요구되고 있다.

본 논문의 2장에서는 기존 무선 ad-hoc 멀티캐스트 라우팅 기술의 종류 및 특성에 대하여 소개하고, 3장에서는 링크 건강도 (link soundness) 기반 경로 설정과 대리 수신 확인 (agent-based ACK) 기능 등을 제공하는 RAMRP 프로토콜의 시스템 모델과 알고리즘들을 설명한다. 4장에서는 시뮬레이션을 통하여 본 프로토콜의 성능을 평가하며 5장에서는 요약 및 향후 연구 과제 등을 제시한다.

II. 무선 ad-hoc 멀티캐스트 라우팅 기술

고정 네트워크 상에서 멀티캐스팅 기술은 그룹 통신 응용을 위한 유용한 수단으로 사용되고 있다. 특히, 대역폭과 전원 등의 자원이 제한된 무선 네트워크 환경에서 멀티캐스팅은 더욱 효율적인 성능을 발휘할 수 있다^[1]. 그러나 고정 네트워크를 위한 멀티캐스팅 기술을 무선 네트워크에 그대로 적용할 경우 멀티캐스트 트리의 잦은 재구성으로 인한 과도한 제어 트래픽과 잦은 패킷 분실을 유발시킬 수 있다. 또한, 노드의 이동으로 인하여 멀티캐스트 트리에 루프가 발생하면 루프가 제거될 때까지 불필요한 트래픽에 의한 대역폭 낭비가 발생할 수 있다.

현재까지 무선 ad-hoc 네트워크를 위한 멀티캐스팅은 멀티캐스트 라우팅 프로토콜에 대한 연구가 주류를 이루고 있다. 대표적인 멀티캐스트 라우팅 프로토콜들로는 노드 식별자 기반의 트리 형성 기법을 사용하는 AMRIS (Ad-hoc Multicast Routing utilizing Increasing id-numberS), 트리 대신 그물 구조를 사용하는 ODMRP (On-Demand Multicast Routing Protocol), 그리고 트리 구조와 그물 구조

를 복합적으로 사용하는 AMRoute (Ad-hoc Multicast Routing Protocol) 등이 있다.

1. AMRIS

AMRIS^[2]는 네트워크 상태 변화에 대하여 전체 멀티캐스트 트리 재구성의 오버헤드를 줄이기 위해 각 노드에 할당되는 MSM-ID (Multicast Session Member-ID)를 이용하여 지역적으로 트리를 재구성하는 멀티캐스트 라우팅 프로토콜이다.

AMRIS에서 상위 노드는 하위 노드보다 항상 큰 ID 값을 갖는다. 세션 관리자인 송신자가 세션 개시 패킷을 방송하면 세션에 참여할 노드들은 이웃 노드들에게 참가 의사를 표시하고 MSM-ID 조정을 통하여 상하 관계를 형성한다. 링크 실패나 노드 이동 후에도 상위 노드를 찾아 협상을 통해 트리 구성에 다시 참여 할 수 있다. 이 프로토콜은 노드의 ID만으로 트리를 구성할 수 있고 트리 구성이 지역적으로 일어나는 장점을 갖지만 노드의 이동에 따라 트리의 모양이 경사 트리 (skewed tree)가 될 수도 있는 단점을 갖는다.

2. ODMRP

ODMRP^[3]는 주문식 멀티캐스트 라우팅 프로토콜로서 일반적인 트리 구조 대신 그물 구조의 멀티캐스트 그룹을 형성함으로써 트리 구조의 단점인 트래픽 집중화, 잦은 트리 재구성, 비 효율적인 경로 설정 등을 피하여 대역폭 부하를 줄이고 확장성을 증대시킨 프로토콜이다.

송신을 원하는 노드는 참가 요청 패킷을 방송하고, 이를 수신한 노드는 해당 정보를 캐쉬에 저장한 후 다시 방송한다. 목적지 노드는 참가 요청 패킷을 보낸 노드의 식별자를 포함하는 참가 테이블 패킷을 구성하여 송신자에게 전송한다. 참가 테이블 패킷을 수신한 중간 노드들은 자신을 포워딩 그룹으로 지정하고 여러 목적지의 참가 테이블 패킷들을 취합하여 송신자에게 전송한다. 데이터 전송은 상기 과정 이후에 발생한다.

ODMRP는 불필요한 트리 정보를 저장하지 않으므로 대역폭과 저장 장치의 공간을 상당히 절약할 수 있으며 효율적인 경로 설정이 가능하다. 트리 구성 문제가 없으므로 노드의 가입, 탈퇴 및 이동에 쉽게 대처할 수 있고 확장성도 높일 수 있다. 그러나 그물 구조의 특성에 따라 불필요한 트래픽을 유발시킬 수 있고 노드 수의 증가에 따라 참가 테이블 패킷의 크기가 점차 증가한다는 단점을 가지고

있다.

3. AMRoute

AMRoute^[4-5]는 잦은 트리 재구성으로 인한 과도한 시그널링과 패킷 분실을 감소시키고 동적인 환경에 안정적인 멀티캐스팅을 제공하기 위해 고안된 프로토콜이다. 멤버 노드 사이의 터널링을 통하여 이웃 노드들과 메쉬를 형성한 후 기존 IP 멀티캐스트의 CBT (Core Based Tree)^[6-7]에서와 같이 중심 (core) 노드를 선정해 메쉬 안에서 트리를 형성하게 된다^[8].

AMRoute는 CBT에서와 같이 강건한 (robust) 데이터 전송 효과를 가져올 수 있으나 메쉬 및 트리 구성 과정이 복잡하고 비 멤버 노드가 트리에 포함되지 않기 때문에 비 효율적인 경로가 설정될 수 있으며, 이에 따른 대역폭의 낭비와 패킷 지연을 유발시킬 수 있다. 또한 동적인 무선 네트워크 환경에 대하여 중심 노드를 유동적으로 변경시킬 수 있는 방안을 필요로 한다.

III. 제안 기법

무선 네트워크 상에서의 멀티캐스팅은 방송이라는 무선 매체의 기본 특성에 의해 그룹 통신에서 큰 효율성을 제공할 수 있다. 그러나 이동성을 지원해야 하는 무선 네트워크는 잦은 패킷 분실과 전송 오류를 유발하므로 무선 네트워크 환경에서의 신뢰성 있는 데이터 전송은 많은 패킷 재전송과 패킷 지연을 수반한다. 특히 이동 호스트들만에 의해 구성된 동적인 무선 ad-hoc 네트워크 상에서의 신뢰성 있는 데이터 전송은 더 많은 제어 패킷 교환과 패킷 재전송을 필요로 한다.

본 논문에서는 무선 ad-hoc 네트워크 상에서 신뢰성있게 멀티캐스트 데이터를 전송하고 패킷 재전송을 감소시켜 패킷 지연 시간을 단축시킬 수 있는 RAMRP (Reliable Ad-hoc Multicast Routing Protocol) 프로토콜을 제안한다.

1. 시스템 모델

그림 1에서 예시하는 바와 같이 무선 ad-hoc 네트워크 상의 각 노드들은 멀티캐스트에 참여하고 있는 멤버 노드와 참여하고 있지 않은 비-멤버 노드로 구분된다. 멤버 노드들은 송신자를 중심으로 멀티캐스트 트리를 형성하는데 무선 ad-hoc 네트워크의 특성상 멀티캐스트에 참여하지 않고 수동적인

패킷 전달 (passive forwarding)만을 담당하는 G, R, U와 같은 노드들이 발생할 수 있다.

다중 경로를 가진 노드들간의 링크의 선택은 링크의 안정성을 우선 기준으로 하고 회의용 응용보다는 방송용 응용의 경우를 모델로 한다. 멀티캐스트 트리 상에는 비-멤버 노드가 포함될 수 있으므로 모든 노드는 멀티캐스트 라우팅을 지원하는 것으로 가정한다.

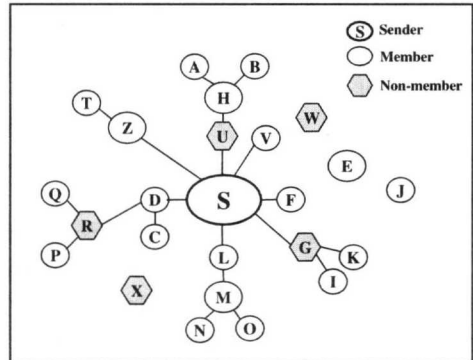


그림 1. 시스템 모델

2. 알고리즘

본 절에서는 라우팅 경로 설정을 위한 링크 건강도 기반 경로 선택 알고리즘과 무선 ad-hoc 네트워크의 동적 변화에 따른 링크 실패와 네트워크 분할을 고려한 수신 확인 알고리즘 및 경로 유지 (route maintenance) 알고리즘을 소개한다.

1) 링크 건강도 기반 경로 선택 알고리즘

기존 연구들에서는 라우팅 경로 선택의 기준으로 홉 카운트와 노드 아이디 등을 사용하였다. 이러한 방법들은 신속하고 용이한 경로 선택을 가져올 수는 있으나 노드의 이동이 잦고 링크간의 연결이 약해질 경우 경로 재설정으로 인한 많은 패킷 전송 지연을 유발할 수 있다. 이러한 문제점을 보완하기 위해 RAMRP 프로토콜에서는 링크 건강도 (link soundness) 기반 경로 선택 기법을 사용하여 링크 실패로 인한 경로 재설정 및 패킷 재전송을 감소시킨다.

링크 건강도 산출을 위한 기준은 여러가지가 될 수 있지만 대표적으로 신호 세기, 송신자로부터의 홉 카운트, 노드의 이동에 따른 호 실패 확률(CMR: Call to Mobility Ratio) 등을 들 수 있다. 노드간의 신호 세기가 임계값 이하로 떨어지면 경로가 유지되고 있어도 작아지는 대역폭으로 인하여 빈번한

패킷 전송 실패 및 지연을 유발할 수 있다. 송신자로부터의 홉 카운트가 커지면 두 노드간의 전송 경로의 실패 확률도 커지는 만큼 홉 카운트는 중요한 링크 건강도의 기준이 될 수 있다. 호 실패 확률은 앞서 언급한 기준들과 개념상 중복되는 부분도 있으나 대상 노드의 이동 속도 및 방향 등을 부가적으로 고려할 수 있다. 이러한 링크 건강도의 기준들은 독립적 또는 복합적으로 적용될 수 있으며 새로운 기준의 추가도 고려할 수 있다. 상기한 기준들간의 상관 관계에 의한 구체적인 링크 건강도의 산출은 본 논문에서 다루지 않는다.

그림 2는 RAMRP 프로토콜을 사용한 전송 경로 설정과 멀티캐스트 트리 구성 과정을 예시한다. 송신자 노드는 패킷 전송을 개시하기 위하여 멤버 노드들에게 경로 설정 참여를 요구하는 참가 요청 패킷 (JOIN_REQ)을 방송한다. 참가 요청 패킷을 수신한 노드는 수신한 패킷에 현재까지 지나온 홉 카운트 및 신호 세기 정보를 기록하여 다시 방송하고 일정 시간 동안 수신한 참가 요청 패킷들 중 링크 건강도가 가장 높은 경로를 선택하여 상위 노드에게 참가 패킷 (JOIN)을 전달하여 트리를 형성하며 다른 경로의 노드들에게 삭제 패킷 (PRUNE)을 전송하여 자신이 다른 경로를 선택했음을 알림으로서 더 이상의 패킷 전달을 막는다.

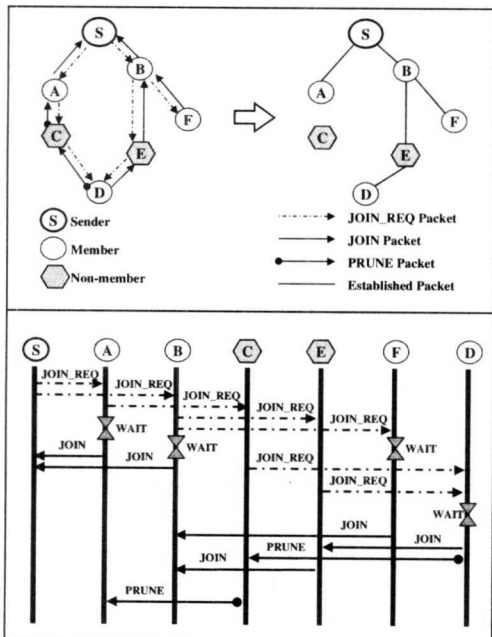


그림 2. 전송 경로 설정 및 트리 구성 과정

표 1. 참가 요청 처리 알고리즘

```

if (first_seen JOIN_REQ.SEQ)
    start JOIN_TIMER;
store JOIN_REQ.INFO;
if (HOP_CNT < TTL)
    flood JOIN_REQ;

while (Tjoin-req-timer) ::

PREF(JOIN_REQ().INFO);
if (NOT_SELECTED_LINK)
    send PRUNE(JOIN_NAK=true)
    to up-link nodes;
else
    send JOIN to up-link node;
    
```

표 1은 참가 요청을 수신한 노드의 참가 요청 처리 알고리즘을 예시한다. 최초 참가 요청 패킷이 수신되면 해당 노드는 타이머를 사용하여 일정 시간 동안 다른 경로를 통해 도착하는 동일한 참가 요청 패킷을 수신한다. 이 후, 해당 노드는 참가 요청 패킷에 포함된 링크 건강도 정보, 패킷 식별자 및 상위 노드 정보를 캐쉬에 저장하고 패킷의 TTL (Time To Live) 값이 아직 패킷의 홉 카운트 제한까지 도달하지 못하였을 경우 TTL 값을 증가시킨 후 참가 요청 패킷을 다시 방송한다. 일정 시간 후, 현재까지 저장된 링크 중에서 가장 건강도가 높은 링크를 선택하여 참가 패킷을 전송하며 다른 노드들에게는 삭제 패킷을 전송하여 더 이상의 패킷 전달을 막는다.

표 2. 참가 처리 알고리즘

```

if (this == FORWARDING_NODE) {
    send JOIN to up-link node
    along the existing path;
    set ROUTING_TBL with NEW_CHILD;
}
else {
    set FORWARDING_NODE to this node;
    set ROUTING_TBL with NEW_CHILD;
    send JOIN to PARENT node;
}
    
```

표 2는 하위 노드로부터 참가 패킷을 수신한 노드의 참가 처리 알고리즘을 예시한다. 해당 노드가 멀티캐스트 라우팅 기능을 수행하고 있다면 참가 패킷을 이미 설정된 경로를 통해 상위 노드에게 전달하고 참가 패킷을 수신한 하위 노드를 라우팅 테이블에 추가한다. 해당 노드가 아직 멀티캐스트 라우팅 기능을 수행하고 있지 않았다면 자신을 전달

노드로서 라우팅 테이블에 추가한 후 캐시에 저장된 기존 상위 노드로 참가 패킷을 전달한다. 삭제 패킷을 수신한 노드는 삭제 패킷을 송신한 노드에 게 더 이상의 패킷을 전달하지 않는다.

표 3. 데이터 송신 및 수신 확인 처리 알고리즘

```

while (packets to send remain) {
    if (event : T_snd) {
        if (data_cache is not full)
            flood next DATA packet;
    }
    if (event : ACK) {
        if (Already seen ACK: diff ACK_NUM) {
            discard ACK(DATA_NUM);
            continue;
        }
        if (#ACK(DATA.NUM)++
            == #MEMBER_TBL_ENTRY) {
            purge DATA(DATA.NUM)
            from data_cache;
            flood PURGE(DATA.NUM) packet;
        }
    }
}
    
```

표 3은 트리 구성 완료 후, 송신자 노드의 데이터 송신 및 수신 확인 처리 알고리즘을 예시한다. 멀티캐스트 트리 상에 패킷을 수신하지 못한 새로운 노드가 도착하면 트리 상의 각 노드는 새로운 노드에 대한 신속한 데이터 전달을 위하여 일정량의 데이터를 적재할 수 있는 버퍼를 관리한다. 동일 데이터가 중복 수신된 경우 나중 데이터는 무시되고 버퍼에 적재된 데이터는 모든 수신자로부터 수신 확인을 받은 후에 삭제될 수 있다. 모든 수신자로부터의 수신 확인이 끝나면 제거 패킷 (PURGE)을 발송하여 각 노드의 버퍼로부터 해당 데이터를 삭제하도록 한다.

2) 수신 확인 처리 알고리즘

무선 환경에서 임의의 두 노드 사이에 링크 실패가 발생할 확률은 항상 존재한다. 링크 실패는 직접 연결된 두 노드간 뿐 아니라 간접 연결된 두 노드 사이에서도 발생할 수 있다. 간접 연결된 두 노드 사이에 링크 실패가 발생하면 문제가 발생한다. 그림 3에서 D는 상위 노드 B에 수신 확인을 전송한 후 이동하여 E의 하위 노드가 되는데, 이 사이에 A, B간의 연결이 유실되면 B는 D의 수신 확인을 A에 전달할 수 없으므로 송신자 S는 타임아웃 후에 D에 대한 데이터 재전송을 실시한다. 이러한 상황은 무선 네트워크 상에서 노드의 이동성에 따라

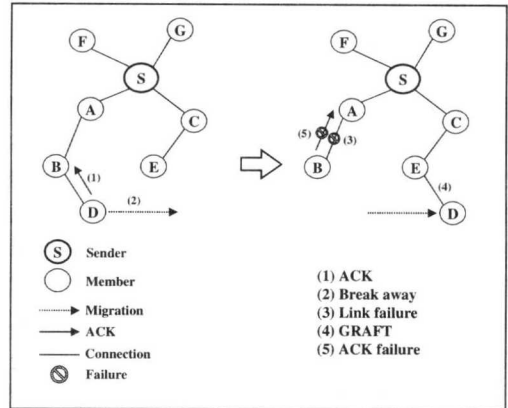


그림 3. 수신 확인 패킷의 분실

빈번하게 발생할 수 있다.

본 논문에서는 수신 확인 유실에 의한 재전송을 줄이기 위하여 대리 수신 확인 (agent-based ACK) 기법을 제시한다. 하위 노드로부터 수신 확인을 받은 상위 노드는 수신 확인 패킷을 송신자에게 전달할 책임을 맡는다. 상위 노드는 해당 데이터에 대한 대리 수신 확인을 하겠다는 의미에서 하위 노드에 수신 재확인 (RE_ACK) 패킷을 전달한다. 수신 재확인 패킷을 받지 못한 하위 노드는 타임아웃 후 또는 이주 시마다 수신 확인을 상위 노드에게 전송해야 한다. 표 4는 상위 노드의 수신 확인 처리 알고리즘을 예시한다.

표 4. 수신 확인 처리 알고리즘

```

if (same ACK is in ACK_TBL) {
    discard ACK packet;
}
else {
    add ACK.INFO to ACK_TBL;
    send RE_ACK packet
    to CHLD(ACK_SRC) node;
    send ACK_TBL packet to PARENT node;
}
    
```

상위 노드가 수신한 수신 확인 패킷이 수신 재확인 패킷 분실 등에 의해 하위 노드로부터 재전송된 것이라면 해당 패킷은 무시되고 그렇지 않으면 해당 패킷이 포함하고 있는 데이터 식별자 및 수신 확인 순서 번호 등의 정보를 수신 확인 패킷 테이블에 등록하며 수신 재확인 패킷을 하위 노드에게 전송하여 수신 확인 패킷이 상위 노드에 전달되었음을 알린다. 수신 확인 순서 번호는 하위 노드가

수신 재확인 패킷을 전달받지 못하여 재전송한 수신 확인 패킷의 개수를 의미한다.

그림 4에서 예시하는 바와 같이 D는 수신 확인 패킷을 B에 전송하고 B는 D에 수신 재확인을 전송한다. 이 후, D는 이주하고 A와 B간의 연결은 끊어진다. B는 새로운 상위 노드 C에 접합하고 D의 대리 수신 확인을 위하여 D의 수신 확인 패킷을 C에 전송한다. C는 D의 수신 확인 패킷을 송신자 S에게 전달함으로써 대리 수신 확인 절차를 완료한다.

3) 경로 유지 알고리즘

무선 ad-hoc 네트워크에서는 노드의 이동에 따라 네트워크가 여러 개로 분할될 수 있다. 경로 설정을 통한 멀티캐스트 트리 구성을 짧은 주기로 반복하게 되면 지속적인 트리의 유지가 가능하지만 트리 구성에 많은 시간을 소모해야하고 많은 제어 트래픽으로 인한 대역폭의 낭비를 초래할 수 있다. 본 논문에서는 이와 같은 오버헤드의 감소를 위하여 네트워크 분할이 발생했을 때 멀티캐스트 트리를 재접합 시키는 확장 링 탐색 (expand ring search) 기법을 사용한다.

그림 5에서 예시하는 바와 같이 멀티캐스트 트리로부터 분리된 노드 D는 트리에 다시 접합하기 위

하여 1홉 거리 (R=1)의 노드를 대상으로 접합 요청 (GRAFT_REQ) 패킷을 발송한다. 1홉 거리에는 A가 있지만 멀티캐스트 멤버가 아니므로 D는 임의의 시간 후에 다시 2홉 거리 (R=2)의 노드를 대상으로 접합 요청 패킷을 발송한다. 2홉 거리에 있는 B도 멤버가 아니므로 위의 과정을 반복하여 3홉 거리 (R=3)의 C를 만나게 된다. C는 접합 요청의 응답으로 접합 (GRAFT) 패킷을 B, A를 거쳐 D까지 전달하고 D는 접합 패킷 중 가장 먼저 도착한 패킷의 경로를 선택하여 접합 확인 (GRAFT_ACK) 패킷을 전송한다. 접합 확인 패킷을 수신한 A, B는 멀티캐스트 멤버는 아니지만 멀티캐스트 데이터의 수동적 전달을 지원해야 하므로 자신의 라우팅 테이블을 수정한다.

확장 링 탐색 과정이 반복되면 전송 경로의 길이가 계속 증가할 수 있는데 이는 링크 건강도 기반 경로 선택 기법을 사용하는 RAMRP 프로토콜의 취지에 위배되는 것이므로 주기적인 초기 경로 설정 과정의 반복으로 통해 안정적인 경로 설정을 제공하는 일이 필요하다.

표 5는 분리된 노드가 멀티캐스트 트리에 접합하기 위한 접합 요청 알고리즘을 예시한다. 분리 노드는 홉 거리 (GRAFT_RADIUS)가 1부터 멀티캐스

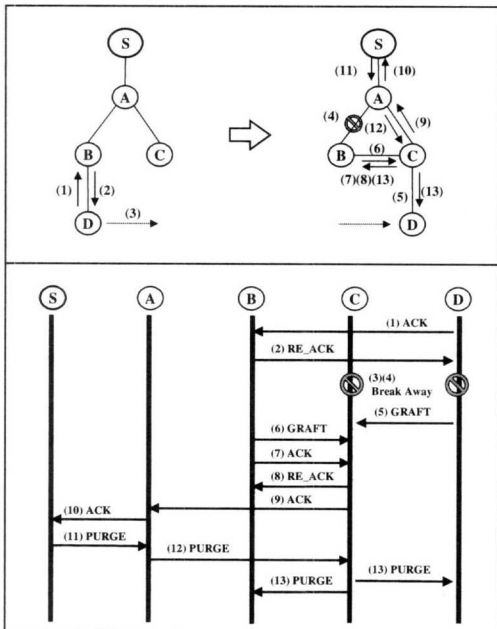
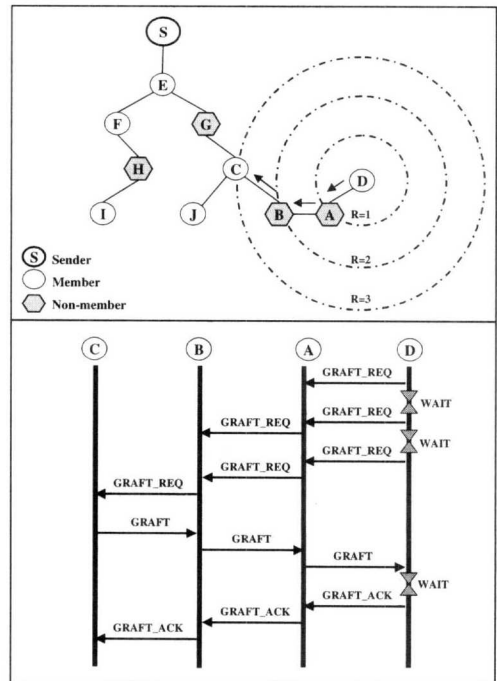


그림 4. 대리 수신 확인 과정



(그림 5) 확장 링 탐색 과정

트 멤버를 만날 때까지 접합 요청 (GRAFT_REQ) 패킷 전송을 반복하고 멤버 노드를 만나면 해당 노드 (SOURCE(GRAFT))까지 접합 확인 (GRAFT_ACK) 패킷을 전송한다.

표 5. 접합 요청 알고리즘

```
GRAFT_RADIUS = 1;
flood GRAFT_REQ(GRAFT_RADIUS) packet;
while (event : GRAFT packet) {
    while (T_graft) ::
        GRAFT_RADIUS++;
        send GRAFT_REQ(GRAFT_RADIUS) packet;
}
set SOURCE(GRAFT) node to PARENT node;
send GRAFT_ACK packet
to SOURCE(GRAFT) node;
```

표 6은 접합 확인을 수신한 노드가 접합 확인을 처리하는 과정을 예시한다. 접합 확인 패킷의 최종 목적지가 현재 노드가 아니면 현재 노드를 전달 노드로 설정하고 접합 확인 패킷을 최종 목적지까지 반복 전송하며 전달 목적으로 라우팅 테이블을 수정한다. 최종 목적 노드에 도착하면 접합 확인 정보 (GRAFT_ACK.INFO)를 라우팅 테이블에 추가한다.

표 6. 접합 확인 처리 알고리즘

```
if (Current_Node != DEST(GRAFT_ACK)) {
    set Current_Node to FOWARDING node;
    forward GRAFT_ACK
        to DEST(GRAFT_ACK) node;
    set ROUTING_TBL with NEW_CHILD;
}
else
    add GRAFT_ACK.INFO to ROUTING_TBL;
```

IV. 성능 평가

지금까지 무선 ad-hoc 네트워크 상에서 신뢰성 있는 멀티캐스트 라우팅을 위한 경로 설정, 수신 확인 및 경로 유지 등에 관한 알고리즘들을 소개하였다. 본 장에서는 시뮬레이션 모델을 정의하여 이러한 알고리즘들의 성능을 측정하고 그 결과를 비교 분석한다.

1. 시뮬레이션 모델

시뮬레이션 모델 정의 및 시뮬레이션에는 DEVS (Discrete EVent Simulation)^[9] 기반의 이산 사건

시뮬레이터인 DEVS Scheme과 DEVSIm++^[10] 등이 사용되었다. 모델에 대한 구성 및 가정은 다음과 같다.

- 1개 송신자 노드와 3개 멤버 노드가 존재한다.
- 송신자 노드는 멤버 노드들로부터 수신 확인 패킷을 모두 수신한 경우에만 해당 데이터에 대한 전송이 완료된 것으로 판단한다.
- 데이터 전송이 실패한 경우, 경로 재설정 및 패킷 재전송에 필요한 추가 시간에 따라 데이터 전송 시간은 정상적인 전송 시간의 2배로 한다.
- 패킷 전송 실패는 링크 건강도가 경로 실패 확률보다 작은 상황에서 발생한다.

2. 시뮬레이션 결과

시뮬레이션은 크게 링크 메트릭에 따른 비교와 수신 확인 및 데이터 전송 방법에 따른 비교로 구분되어 수행되었다.

1) 링크 메트릭에 따른 비교

송신자는 평균 10ms 주기로 데이터 전송을 시도하고, 송신자로부터 멤버 노드까지 데이터 전송 시간은 평균 15ms의 지수 분포 (exponential distribution)를 따른다고 가정하였다. 또한 링크 메트릭의 구성 요소로서 링크 건강도와 링크 지연 (link delay) 시간을 선정하였다. 링크 건강도를 기반으로 경로를 설정하는 기법과 참가 요청 패킷을 수신한 시간에 따라 경로를 선택하는 링크 지연 시간 기반 경로 설정 기법간의 비교를 위하여 링크 건강도의 변화에 따른 두 기법의 데이터 재전송 횟수와 데이터 전송 소요 시간을 측정하였다.

그림 6은 링크 메트릭에 따른 데이터 패킷 재전송 횟수의 변화를 예시한다. 데이터 전송 실패율 (data transfer failure ratio)은 전송 과정에서 발생하

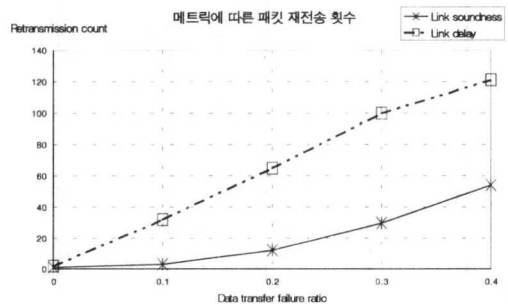


그림 6. 메트릭에 따른 패킷 재전송 횟수

는 링크 실패 확률을 의미한다. 두 기법 모두 데이터 전송 실패율이 증가함에 따라 패킷 재전송 횟수가 증가하고 있지만 링크 건강도 기반 경로 설정 기법의 경우 링크 지연 기반 경로 설정 기법 보다 재전송 횟수가 현저히 낮음을 볼 수 있다.

그림 7은 그림 6과 동일한 상황에서 100개의 데이터에 대한 수신 확인을 완료하는데 소요된 시간의 변화를 예시한다. 두 기법 모두 데이터 전송 실패 확률의 증가에 대해 패킷 전송 시간이 선형적으로 증가하고 있지만 링크 건강도 기반 경로 설정 기법이 더 완만한 경사를 보임에 따라 데이터 전송 실패 확률이 커질수록 링크 건강도 기반 경로 설정 기법이 더 좋은 효율을 가짐을 알 수 있다. 즉, 약 7% 이상의 데이터 전송 실패 확률이 존재하는 링크에서 안정적인 링크의 선택은 패킷 재전송 시간을 단축시켜서 링크 지연 기반 경로 설정 기법보다 빠른 데이터 전송을 이룰 수 있다.

2) 수신 확인 및 데이터 전송 기법에 따른 비교

각 노드에서 데이터에 대한 버퍼링을 하지 않는다면 각 노드는 데이터 전송의 신뢰성을 위해 이전 데이터에 대한 수신 확인을 받은 후 다음 데이터를 전송해야 한다. 그러나 각 노드는 새로 접합하는 멤버에 대한 데이터 전송 지연을 최소화하기 위해서 일정량의 데이터를 버퍼링하며 버퍼 크기 범위 내에서 수신 확인 이전에 다음 데이터를 보낼 수 있다. 버퍼와 대리 수신 확인을 사용하지 않는 경우, 각 노드는 타임아웃 후 데이터를 재전송하는데 타임아웃은 80ms로 가정하였고, 버퍼를 사용하는 경우, 송신자의 데이터의 발생 주기는 20ms로 가정하였으며 데이터 및 수신 확인 패킷의 전송 지연은 평균 20ms의 지수 분포를 따른다고 가정하였다. 특정 노드가 멀티캐스트 트리로부터 분리되어 재접합

할 때까지의 시간은 정상 데이터 전송 시간의 두 배 정도인 평균 40ms의 지수 분포를 따른다고 가정하였다. 시뮬레이션에 사용된 변수의 정의 (표 7) 및 수식 표현은 다음과 같다.

표 7. 시뮬레이션 변수 정의

확률 변수	P_{suc}	데이터 전송과 수신 확인이 링크 실패 없이 이루어질 확률
	P_{fail}	데이터 전송과 수신 확인 시 링크 실패가 발생할 확률
	P_{af}	데이터 전송 시 링크 실패로 인한 P_{fail}
	P_{df}	수신 확인 패킷 전송 시 링크 실패로 인한 P_{fail}
시간 변수	$T_{timeout}$	타임아웃 주기
	T_{data}	데이터 전송 소요 시간
	T_{ack}	수신 확인 패킷 전송 소요 시간
	T_{graft}	분리 후 다른 상위 노드에 접합하는 시간
기타	N_{hf-a}	ACK 전송시 실패한 홉의 개수
	N_{hf-d}	데이터 전송 시 실패한 홉의 개수

① 대리 수신 확인 및 데이터 버퍼링을 사용하지 않은 경우의 단일 데이터 패킷의 전송 소요 시간

• 최초 데이터 전송에 성공한 경우

$$T_{m-1} = (T_{data} + T_{ack}) \times P_{suc} \quad (1)$$

• 동일 데이터에 대하여 N번째까지 데이터 전송에 실패하고 N+1번째 전송에서 성공한 경우

$$T_{m-3} = \sum_{k=1}^N \{ (k \times T_{timeout} + T_{data} + T_{ack}) \times P_{fail}^k \times P_{suc} \} \quad (2)$$

$$T_{m-3} = \left[\left(\frac{P_{fail} \times (1 - P_{fail}^N)}{(1 - P_{fail})^2} - \frac{n \times P_{fail}^{n+1}}{1 - P_{fail}} \right) \times T_{timeout} + \frac{(T_{data} + T_{ack}) \times P_{fail} \times (1 - P_{fail}^N)}{1 - P_{fail}} \right] \times P_{suc} \quad (3)$$

② 대리 수신 확인 기법만을 사용하고 데이터 버퍼링을 사용하지 않은 경우의 단일 데이터 패킷의 전송 소요 시간

• 최초 데이터 및 수신 확인 전송에 성공한 경우

$$T_{m-1} = (T_{data} + T_{ack}) \times P_{suc} \quad (4)$$

• 최초 데이터 전송은 성공하고 수신 확인 전송은 실패한 경우

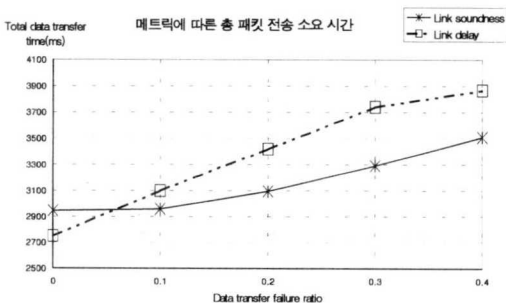


그림 7. 메트릭에 따른 총 패킷 전송 소요 시간

$$T_{m-2} = (T_{data} + N_{hf-a} \times T_{graft} + T_{ack}) \times P_{fail} \times P_{af} \quad (5)$$

- N번째까지 데이터 전송에 실패하고 N+1번째 전송에서 성공한 경우

$$T_{m-3} = \left[\left(\frac{(P_{fail} \times P_{df}) \times (1 - (P_{fail} \times P_{df})^n)}{(1 - P_{fail} \times P_{df})^2} \right) \right. \quad (6)$$

$$\left. - \frac{n \times (P_{fail} \times P_{df})^{n+1}}{1 - P_{fail} \times P_{df}} \right) \times T_{timeout}$$

$$+ \left(\frac{(P_{fail} \times P_{df}) \times (1 - (P_{fail} \times P_{df})^n)}{1 - P_{fail} \times P_{df}} \right)$$

$$\left. \times (T_{data} + T_{ack}) \right] \times P_{suc}$$

- N번째까지 데이터 전송에 실패하고 N+1번째 전송에서도 실패한 경우

$$T_{m-4} = \sum_{k=1}^n \{ (k \times T_{timeout} + T_{data} + N_{hf-a} \times T_{graft} + T_{ack}) \times (P_{fail} \times P_{df})^k \times P_{fail} \times P_{af} \} \quad (7)$$

$$T_{m-4} = \left[\left(\frac{(P_{fail} \times P_{df}) \times (1 - (P_{fail} \times P_{df})^n)}{(1 - P_{fail} \times P_{df})^2} \right) \right. \quad (8)$$

$$\left. - \frac{n \times (P_{fail} \times P_{df})^{n+1}}{1 - P_{fail} \times P_{df}} \right) \times T_{timeout}$$

$$+ \left(\frac{(P_{fail} \times P_{df}) \times (1 - (P_{fail} \times P_{df})^n)}{1 - P_{fail} \times P_{df}} \right)$$

$$\left. \times (T_{data} + N_{hf-a} \times T_{graft} + T_{ack}) \right] \times P_{fail} \times P_{af}$$

③ 대리 수신 확인 및 데이터 버퍼링을 모두 사용한 경우 단일 데이터 패킷의 전송 소요 시간

- 최초 데이터 및 수신 확인 전송에 성공한 경우

$$T_{m-1} = (T_{data} + T_{ack}) \times P_{suc} \quad (9)$$

- 어떠한 형태로든 전송이 실패한 경우

$$T_{m-2} = (T_{data} + N_{hf-d} \times T_{draft} + T_{ack} + N_{hf-a} \times T_{graft}) \times P_{fail} \quad (10)$$

그림 8은 재전송 기법에 따른 전송 소요 시간을 예시한다. 재전송 기법은 데이터 분실 후 타임아웃이 지난 후 재전송하는 경우, 대리 수신 확인을 통하여 타 노드가 중계해 주는 경우, 노드 분리 후 재

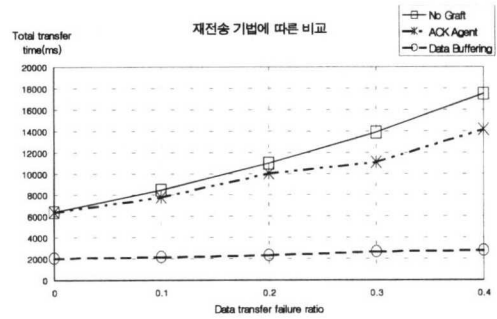


그림 8. 메트릭에 따른 총 패킷 전송 소요 시간

접합 시에 버퍼에 저장된 데이터를 전달하는 경우로 구분된다. 버퍼의 크기는 5개의 데이터 패킷을 저장할 수 있는 것으로 가정하였고 총 100개의 데이터 패킷을 발생시켜 모든 노드로부터 수신 확인을 받는 시점까지의 시간을 측정하였다.

버퍼링 기법은 데이터 전송 실패율에 거의 영향을 받지 않고 좋은 전송 효율을 제공함을 알 수 있고 타임아웃 기법이나 대리 수신 확인 기법은 모두 데이터 전송 실패율에 영향을 많이 받았으나 대리 수신 확인 기법이 데이터 전송 실패율이 커질수록 타임아웃 기법보다 좋은 효율을 제공함을 알 수 있다.

그림 9는 버퍼 크기에 따른 전송 소요 시간을 예시한다. 버퍼의 크기가 3일 때 전송 효율이 많이 떨어짐을 알 수 있고 버퍼 크기가 커짐에 따라 전송 효율이 개선됨을 알 수 있다. 그러나 버퍼 크기가 15일 때 전송 효율은 더 이상 개선되지 않고 일정하게 수렴함을 알 수 있다. 따라서 일정 크기 이상의 버퍼는 메모리 낭비를 초래할 수 있다.

그림 10은 버퍼 크기에 따른 평균 패킷 지연 시간을 예시한다. 버퍼의 크기가 커질수록 패킷이 버

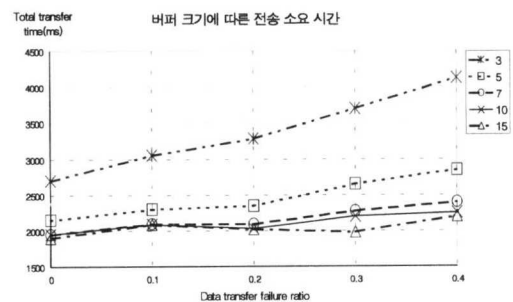


그림 9. 버퍼 크기에 따른 전송 소요 시간

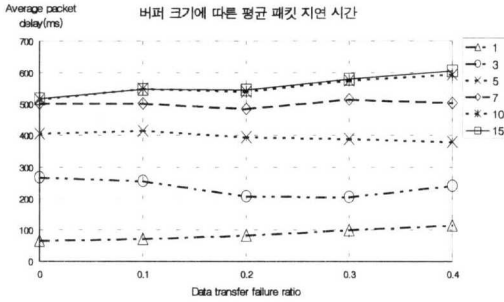


그림 10. 버퍼 크기에 따른 평균 패킷 지연 시간

퍼에 머무르는 시간이 길어지므로 패킷 지연 시간은 커짐을 알 수 있다.

V. 결론

무선 ad-hoc 네트워크 상에서의 멀티캐스팅에 대한 관심이 고조되고 있다. 본 논문에서는 무선 ad-hoc 네트워크 환경에서 효율적으로 멀티캐스팅을 할 수 있는 RAMRP 프로토콜을 제안하였다. RAMRP은 링크 건강도 기반 경로 설정 기법과 대리 수신 확인 기법을 사용하여 보다 안정적인 경로 선택을 지원하고 링크 실패로 인한 경로 재설정 횟수 및 패킷 재전송 횟수를 감소시킬 수 있다. 또한, 각 노드가 버퍼를 관리하는 경로 유지 기법을 통하여 링크 단절 시에 신속하게 후속 데이터를 전송할 수 있다. RAMRP 프로토콜은 노드들의 이동이 많아지거나 링크 상태의 변화 등으로 인해 패킷 전송 실패 확률이 높아지는 경우에 유용하게 사용될 수 있음을 시뮬레이션을 통해 확인 할 수 있었다.

무선 네트워크 환경에서는 패킷 분실을 최소화 할 수 있는 경로 설정 기법의 고안이 요구된다. 따라서 여러 상황에 대하여 링크의 건강도를 평가할 수 있는 메트릭들에 대한 좀 더 실질적인 연구 뿐 아니라 링크 상태의 변화를 유발할 수 있는 특수한 상황에 대한 고려도 필요하다.

참고 문헌

[1] K. Obraczka, G. Tsudik, "Multicast Routing Issues in Ad Hoc Networks," *IEEE 1998 International Conference on Universal Personal Communications (ICUPC '98)*, 1998.
 [2] C. W. Wu, Y. C. Tay and C. Toh, "Ad-hoc

Multicast Routing Protocol utilizing Increasing id-numberS (AMRIS) Functional Specification," draft-ietf-manet-amris-spec-00.txt, Internet-Draft, *IETF*, 1998.

[3] M. Gerla, G. Pei, S. J. Lee, and C. C. Chiang, "On-Demand Multicast Routing Protocol (ODMRP) for Ad-hoc Networks," draft-ietf-manet-odmrp-00.txt, Internet-Draft, *IETF*, 1998.
 [4] M. Liu, R. R. Talpade, and A. McAuley, "AmRoute : Ad-hoc Multicast Routing Protocol," Technical Research Report CSHCN T. R. 99-1, *Center for Satellite and Hybrid Communication Networks*, 1999.
 [5] M. Liu, R. R. Talpade, and A. McAuley, "AMRoute : Ad-hoc Multicast Routing Protocol," draft-manet-amroute-00.txt, Internet-Draft, *IETF*, 1998.
 [6] A. Ballardie, "Core Based Trees (CBT version 2) Multicast Routing Protocol Specification," RFC 2189, *IETF*, 1997.
 [7] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture," RFC 2201, *IETF*, 1997.
 [8] C. Semeria and T. Maufer, "Introduction to IP Multicast Routing," Technical Report, *3Com*, 1996.
 [9] B. P. Zeigler, *Object-Oriented Simulation with Hierarchical Modular Models*, Academic Press, 1990.
 [10] T. G. Kim, "DEVSIM++ : C++ Based Simulation with Hierarchical Modular DEVS Models," Technical Report, *KAIST*, 1998.

박 상 윤(Sang Yun Park)

준회원



1997년 2월 : 동국대학교 전자계산학과 학사

1999년 2월 : 성균관대학교 전기전자 및 컴퓨터공학부 석사

2001년 2월 : 성균관대학교 전기전자 및 컴퓨터공학부 박사과정 수료

<주관심 분야> 이동 컴퓨팅 시스템, 멀티미디어 통신, 이동 컴퓨팅 보안, 분산 시스템

엄 영 익(Young Ik Eom)

정회원



1983년 2월 : 서울대학교 계산
통계학과 학사

1985년 2월 : 서울대학교 대학
원 전산과학전공 석사

1991년 8월 : 서울대학교 대학
원 전산과학전공 박사

현재, 성균관대학교 정보통신공
학부 교수

<주관심 분야> 분산 시스템, 이동 컴퓨팅 시스템,
분산 객체 시스템