CATMP : CAMP를 개선한 멀티캐스트 라우팅 프로토콜

임 수 정† · 이 미 정††

요 약

본 논문에서는 애드 혹 망을 위한 메쉬 기반 멀티캐스트 라우팅 프로토콜중 하나인 CAMP(Core-Assisted Mesh Protocol)를 보완한 CATMP (Core-Assisted Thick Mesh Protocol)를 제안한다. CAMP는 트리 구조를 사용하지 않으면서 데이터나 제어 패킷의 플러딩없이 라우팅 구조를 생성하는 유일한 프로토콜이다. 따라서 제어 패킷의 발생이 제한적이고 네트워크의 크기가 확장되어도 잘 적응할 수 있다. 그러나 CAMP는 메쉬의 중심부보다 가장자리가 연결성이 떨어지기 때문에 그룹 멤버의 수가 적은 경우나 멀리 떨어진 위치에 존재하는 그룹 멤버가 있는 경우 데이터 전달율이 저하된다. CATMP에서는 이를 보완하기 위해 메쉬의 연결 정도가 낮은 지역에서도 이웃의 참여를 유도하여 연결 정도를 높여줌으로써 결과적으로 전체 메쉬가 좀 더 균일하게 형성되도록 한다. 시뮬레이션을 통해 CAMP와 CATMP의 성능을 비교한 결과 CAMP의 성능이 저하되는 상황에서 CATMP가 성능을 향상시킬 수 있음을 볼 수 있었다.

CATMP : Multicast Routing Protocol to Enhance CAMP

Soo-Jeong Lim† · Mee-Jeong Lee††

ABSTRACT

In this paper we propose a multicast routing protocol referred to as CATMP to enhance CAMP. CAMP is the only multicast routing protocol not based on trees that avoids flooding of data or control packets to establish the routing structure for a group. In CAMP, the amount of control packet is limited and the scalability of the protocol is good. But CAMP has a disadvantage that the edge of a mesh is sparser than the center and a mesh similar to a tree is formed when the number of multicast group member is small. In this case, the probability of packet drop increases since there are few redundant paths. Where mesh is formed fragile, CATMP enhances mesh connectivity by inciting neighbors to join the mesh. So that the mesh becomes thicker and is formed more uniformly. The simulation results show that the proposed scheme increases the data delivery rate especially when CAMP shows performance degradation.

키워드 : 애드 혹(Ad-hoc), 멀티캐스팅(Multicasting), 멀티캐스트 라우팅 프로토콜(Multicast Routing Protocol), 메쉬구조(Mesh Structure), CAMP(Core-Assisted Mesh Protocol)

1. 서 론

애드 혹 네트워크는 고정된 네트워크 기반없이 이동하는 호스트들만으로 구성된 네트워크이다. 호스트들은 이웃과 정보를 교환할 수 있는 전파 범위에 한계가 있으므로 송신원에서 수신원까지의 경로는 대체로 다중 홉을 갖게 된다[1]. 또 애드 혹 네트워크에서 모든 호스트들은 단말일 뿐 아니라 트래픽을 포워딩해주는 역할을 담당해야하므로 각 호스트가 라우터의 기능까지 가진다. 그리고 호스트는 이동성과 휴대성이 있어야하므로 각 호스트들의 처리능력과 저장 공간등의 자원은 기존의 네트워크에 비해 상대적으로 제한이 많은 특징을 갖는다[2]. 따라서 애드 혹 네트워크에서는 이러한 특징에 부합하여 효율적으로 정보를 교환하고 데이터를 전송할 수 있는 기술이 요구된다.

애드 혹 네트워크는 기존의 네트워크 기반이 없는 곳에서 재난 복구, 탐색과 구조, 전투 훈련과 같은 상황에서 빠르게 망을 구성할 수 있는 장점을 갖는다. 이런 상황에서는 멤버들간의 통신을 위해서 멀티캐스팅의 필요성이 특히 매우 높다고 할 수 있다. 그러나 애드 혹 네트워크를 이루는 호스트들은 임의로 움직이기 때문에 네트워크 토폴로지의 변화가 매우 잦고 그에 대한 예측이 어렵다. 따라서 기존의 고정된 네트워크에서 사용되고 있는 멀티캐스트 라우팅 프로토콜들은 애드 혹 네트워크에서 좋은 성능을 내기 어려우므로 애드 혹 네트워크에 적합한 별개의 멀티캐스트 라우팅 프로토콜이 요구된다. 또한 애드 혹 네트워크에서 호

※ 본 연구는 한국학술진흥재단의 2001년 BK21 특화사업지원으로 수행되었음.
† 정 회 원 : 이화여자대학교 대학원 컴퓨터학과
†† 정 회 원 : 이화여자대학교 컴퓨터학과 교수
논문접수 : 2001년 2월 20일, 심사완료 : 2001년 5월 25일

스트들은 브로드캐스트 특성으로 인해 멀티캐스트 통신에 잘 적응할 수 있으므로 애드 혹 네트워크에서 기존에 연구되었던 유니캐스트 라우팅 프로토콜과도 별개로 연구되는 것이 더 효율적이라고 할 수 있다.

이에 애드 혹 네트워크에서의 멀티캐스트 통신을 위한 프로토콜들이 활발히 연구되고 있는데, 현재까지 제안된 프로토콜들은 라우팅을 수행하는 구조에 따라 크게 트리 기반의 프로토콜과 메쉬 기반의 프로토콜로 나누어 볼 수 있다[3-6]. 이 두 가지 방식은 각각의 장단점을 갖는데, 네트워크에 부과되는 오버헤드가 지나치게 크지않다면 멀티캐스트 그룹의 수신원들이 데이터를 성공적으로 받는 수신율을 가장 중요한 기준이라고 할 때, 단일 경로만을 제공하는 트리 기반의 프로토콜보다는 데이터의 전달 경로를 더 다양하게 갖는 메쉬기반의 프로토콜이 더 우수한 성능을 보인다[1].

그러나 메쉬 기반의 프로토콜들의 경우에도 메쉬를 생성하고 유지하는 다양한 방식에 따라 네트워크의 이동성이나 규모등에 얼마나 독립적으로 효율적인 데이터 전송을 수행할 수 있는지의 여부가 결정된다[5, 6]. 본 논문에서는 애드 혹 네트워크에서 제안된 멀티캐스트 라우팅 프로토콜인 CAMP의 메쉬 구성 방법을 보완할 수 있는 방안을 제안하였다. CAMP는 메쉬 기반의 프로토콜로 현재까지 제안된 멀티캐스트 라우팅 프로토콜 중에서 데이터 전송을 위한 메쉬를 생성하는데 있어서 제어 패킷이나 데이터 패킷을 플러딩하지 않고 데이터 전송 경로를 설정할 수 있는 유일한 프로토콜이다. 따라서 데이터를 전송하는 내부 구조인 메쉬를 생성하는데 있어서 네트워크에 무리하게 트래픽을 일으키는 오버헤드가 매우 적고, 플러딩 없이 유니캐스트로 제어패킷을 전송하므로써 네트워크의 크기에 크게 구애받지 않고 잘 적응할 수 있다는 큰 장점을 갖는다[6-9].

그러나 CAMP는 코어를 중심으로 메쉬를 형성하기 때문에 메쉬가 형성된 중심부에서 멀리 떨어진 송·수신원에 대해서는 프로토콜의 성능이 저하되는 것을 볼 수 있다. 수신원의 경우에는 멀리 떨어져 위치하는 해당 수신원의 수신율이 낮게 나타나고, 송신원이 메쉬에서 멀리 떨어져 위치할 경우에는 전체 그룹 멤버들의 수신율을 떨어뜨리는 결과를 낳는다. 또 멀티캐스트 그룹의 멤버가 적을 경우 또한 그룹의 멤버들에게 성공적으로 도착한 데이터의 수신율이 매우 낮아지는 경향을 보인다. 이는 멀티캐스트 그룹의 규모가 적은 경우 송·수신원이 메쉬에 연결되는 경로가 트리 구조처럼 단일한 경로가 될 확률이 커서 메쉬 구조의 장점인 다중 경로의 혜택을 제대로 받지 못하기 때문이다. 이에 본 논문에서는 CAMP에서 송·수신원이 단일 경로로 메쉬에 연결되는 경우 좀더 강하게 메쉬와 연결될 수 있는 방안을 추가함으로써 CAMP의 메쉬 구성상의 약점으로 보완하는 프로토콜을 제안하고 시뮬레이션을 통해 두 프로토콜의 성능을 비교하였다.

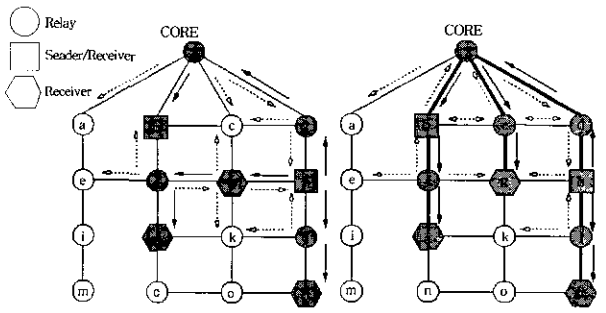
본 논문에서 제안한 CATMP는 그룹 멤버가 메쉬로 연결되는 경로를 설정할 때에 단일한 경로로 연결되는 경우인지를 파악하고, 그렇다면 주변에 있는 이웃 호스트들도 메쉬에 참여하도록 유도하여 메쉬와 연결되는 경로를 더 다양하게 확보하도록 한다. 따라서 CAMP에서는 단일한 데이터 전달 경로를 갖게되는 경우에도 CATMP에서는 다중경로를 갖게되고 결과적으로 좀더 균일하고 두터운 메쉬를 형성하게 된다.

본 논문의 구성은 다음과 같다. 제1장의 서론에 이어서 제2장에서는 본 논문에서 제안한 프로토콜의 기반이 되는 CAMP에 대해서 자세히 살펴본다. 제3장에서는 제안하는 CATMP를 설명하고 제4장에서는 CATMP와 CAMP의 성능을 비교하기 위한 시뮬레이션 및 결과를 살펴본다. 마지막으로 제5장에서는 본 논문의 결론에 대하여 기술한다.

2. CAMP

본 장에서는 CATMP가 기반으로 하는 애드 혹 네트워크를 위한 메쉬 기반 멀티캐스트 프로토콜인 CAMP에 대해서 자세히 살펴보도록 한다. 트리 기반 방식과 달리 메쉬 구조를 이용하는 멀티캐스트 기법에서는 전체 네트워크 토폴로지의 부분집합으로 송신원에서 각 수신원까지 최소한 하나 이상의 경로를 제공한다. 트리 구조의 프로토콜에서는 데이터를 정해진 트리 브랜치의 상위 호스트로부터만 받아들이지만 메쉬 구조의 프로토콜에서는 이웃에 위치해있는 어떤 호스트로부터도 데이터를 받아들여 전달할 수 있다. 따라서 멀티캐스트 메쉬의 멤버인 호스트는 다른 호스트로 다중 경로를 가지므로 토폴로지의 변화에의해 데이터의 흐름이 중단될 확률이 적고 패킷 포워딩을 지원하는 라우팅 구조를 재구성해야하는 요구가 적다.

(그림 1)은 메쉬 구조와 트리 구조의 차이를 보여주는 예로 어두운 색으로 표시된 노드들이 멀티캐스트 그룹의 멤버이다. 노드 g가 멀티캐스트 그룹에 마지막으로 참여한 노드라면, 노드 f와 h를 통해 메쉬에 참여하는 것이 가능하고 결론적으로 노드 c는 메쉬의 멤버가 될 필요가 없다. 또한 (그림 1)은 노드 h에서 그룹 멤버들에게 데이터가 포워드되는 것을 보여준다. 실선은 트래픽의 흐름을 점선은 브로드캐스트 특성으로 인한 오버헤드 트래픽을 보여준다. CAMP의 메쉬구조는 트리구조의 방식보다 데이터를 좀더 짧은 패스로 전송하는 모습을 보여주는데, 이는 패킷을 포워딩하는데 좀 더 적은 수의 노드로도 가능하다는 뜻으로 매우 중요하다고 할 수 있다. 또한 노드 h에 의해 보내진 패킷을 최소한 한번은 받는 노드의 수가 메쉬구조와 트리구조에서 동일하게 나타남을 보여줌으로써 메쉬구조 대신 트리구조를 사용하는 것이 브로드캐스트 링크를 가진 애드 혹 네트워크에서 트래픽 오버헤드를 줄어주는 것만은 아님을 보여준다.



(그림 1) 메쉬구조와 트리구조에서의 트래픽 흐름

CAMP에서는 각 멀티캐스트 그룹에 대해 하나이상의 코어를 두고 그 코어들에게 송·수신원들이 멀티캐스트 그룹 참여를 요청하는 방식을 취한다. 코어들은 송·수신원이 멀티캐스트 그룹에 참여하는데 있어서 제어 트래픽을 제한하는데 사용된다. 즉 송·수신원이 멀티캐스트 그룹의 메쉬에 연결되기 위해 JOIN Request를 발생할 때 코어를 향하여 유니캐스트하므로 메쉬를 생성하기 위한 제어 패킷이 제한된다. 이 JOIN Request는 유니캐스트의 목적지였던 코어 또는 코어로 향하는 중에 만난 기존의 메쉬 멤버인 호스트까지 진행되고 이 경로상의 모든 호스트는 메쉬의 멤버가 된다.

그리고 호스트가 메쉬에 참여할 때는 듀플렉스 모드와 심플렉스 모드 중 하나로 메쉬에 참여하게 된다. 이 두가지 모드의 차이는 듀플렉스 조인을 통해 메쉬에 참여하게된 호스트는 모든 방향에서 들어오는 데이터를 전달해주지만 심플렉스 조인을 통해 메쉬에 참여하게된 호스트는 자신에게 심플렉스 JOIN Request를 보냈던 송신원으로부터 메쉬쪽으로의 데이터만 전송하고 메쉬쪽으로부터의 데이터는 전달하지 않는다는데 있다. 멀티캐스트 메쉬에서 탈퇴하려는 호스트는 자신에게 연결되어야만 하는 이웃 호스트가 없다면 멤버쉽의 변화를 이웃들에게 알리는 것만으로 탈퇴할 수 있다. CAMP에서 메쉬에 참여하려는 호스트는 초기에는 기존의 메쉬 멤버인 이웃이나 코어로 향하는 JOIN Request를 통하여 메쉬와의 연결 경로를 확보하지만 데이터의 수신이 진행되면서 RSP(Reverse Shortest Path)를 통해 데이터가 전달될 것을 보장하기 위해 Heartbeat 메시지를 사용한다. 이를 위해 메쉬 멤버들은 송신원들이 보낸 패킷이 들어온 경로를 체크하며 RSP가 아닌 경로를 통해 들어오는 패킷수가 일정치 이상일 경우 유니캐스트 라우팅 테이블을 보고 송신원으로 향하는 다음 호스트에게 Heartbeat 메시지를 보낸다. 만약 메시지를 보낼 다음 호스트가 메쉬의 멤버가 아니라면 Push Join 리퀘스트를 생성하게 된다. Push Join 리퀘스트는 그 호스트로부터 송신원까지의 모든 호스트들을 메쉬에 참여하도록 해준다.

CAMP는 애드 혹에서의 멀티캐스트만을 담당하므로 이웃과의 연결이 추가되거나 손실되는 것을 찾아내는 것은 CAMP와 함께 작동하는 유니캐스트 라우팅 프로토콜의 역할이다. 따라서 CAMP가 올바르게 작동하기 위해서는 호스트의 실패나 네트워크 분리와 같은 상황에서 정확하게 작동하는 유니캐스트 라우팅 프로토콜이 필수적이다.

2.1 CAMP에서의 메쉬 생성

멀티캐스트 그룹에 참여하려는 호스트는 이웃 호스트 중 이미 그룹의 듀플렉스 멤버인 이웃 호스트가 있다면 멀티캐스트 라우팅 업데이트 메시지를 통해 직접 이웃들에게 자신이 심플렉스 멤버인지 듀플렉스 멤버인지를 알림으로써 멀티캐스트 그룹에 참여하게 된다. 만약 이웃 중 멤버인 호스트가 없다면 JOIN Request를 코어로 전송하기 위해 해당 멀티캐스트 그룹의 코어를 확인하고 자신에게 가장 가까이 있는 코어에게로 JOIN Request를 전송한다. 이때 코어로의 JOIN Request 전송은 유니캐스트로 전달되고 이를 위해 유니캐스트 라우팅 테이블을 참조하여 코어로 향하는 다음 호스트를 선택한다. JOIN Request는 유니캐스트를 통해 코어에 이르게 되고 메쉬에 참여하는 것을 허용하는 응답이 코어쪽으로부터 JOIN Request를 처음 발생시켰던 호스트까지 전달되면서 그 경로상의 모든 호스트들이 메쉬에 참여하게 된다. JOIN Request가 코어를 향해 유니캐스트되는 것은 제어 패킷 발생을 제한하기 위한 것으로 그룹에 참여하기 위해 반드시 코어까지 도달해야하는 것은 아니다. 만약 코어로 가는 경로중에 이미 메쉬의 멤버인 호스트가 JOIN Request를 받는다면 이 호스트는 JOIN Request를 보낸 호스트에게 JOIN Request에 대한 응답을 보내줄 수 있다.

이와 같은 요청과 응답의 과정은 멀티캐스트 그룹의 데이터를 받기 원하는 수신원뿐 아니라에서 데이터를 발생하는 송신원에서도 거쳐야하는 과정이다. 단, 데이터 패킷이 불필요하게 확산되는 것을 방지하기위해 송신원과 송신원의 요구에 의해서 메쉬에 참여하게 되는 호스트들은 심플렉스 모드로 메쉬에 참여하게 된다. 즉, 심플렉스 모드의 메쉬 멤버들은 송신원쪽으로부터 메쉬쪽으로는 데이터 트래픽을 전송하지만 반대의 경우에는 데이터 패킷을 받아들이지 않는다. 이와 달리 수신원의 JOIN Request로 인해 메쉬에 참여한 호스트들은 듀플렉스 모드로 메쉬에 참여하여 모든 방향으로부터의 데이터를 전송해 주게 된다. 만약 메쉬에 참여하려는 송신원이나 수신원이 코어로 이르는 경로를 전혀 알지 못한다면 ERS(Expanded Ring Search)를 이용하여 JOIN Request를 브로드캐스트하고 이 JOIN Request가 결국 메쉬 멤버중 하나를 만나면 메쉬에 연결된다.

2.2 CAMP에서의 메쉬 유지

CAMP에서 코어들은 서로 다른 코어들과 정기적인 메시지 교환을 통해 항상 연결성을 확보하도록 한다. 코어들끼리는 항상 완전 메쉬로서 연결성을 확보하고 있기 때문에

송신원이나 수신원이 어떤 코어에게로 연결되더라도 송신원에서 수신원으로의 데이터 전송이 가능하다. 메쉬에 참여한 모든 호스트들은 자신이 연결되어 있는 코어로 향하는 다음 호스트가 메쉬에 속해있는지를 정기적으로 확인하여 코어로의 경로를 확인한다. 만약 코어로 향하는 다음 호스트가 메쉬에 속해있지 않다면 다시 JOIN Request를 전송하므로써 코어로의 경로를 확보한다.

CAMP는 Heartbeat 메시지와 Push Join 메시지를 이용하여 데이터 전달 메쉬에 송신원과 수신원사이의 RSP가 포함되도록 한다. 이를 위해 수신원은 정기적으로 패킷 포워딩 캐쉬에 있는 데이터 패킷들을 전달해준 호스트가 RSP상의 호스트인지를 체크한다. 만약 성공적으로 받은 데이터 패킷 중에서 RSP를 통해 들어온 패킷의 수가 일정비율 이하라면 Heartbeat 또는 Push Join 메시지가 송신원쪽으로 전송된다. Heartbeat 메시지를 받은 호스트는 송신원으로의 RSP에 있는 다음 호스트가 이미 메쉬의 멤버라면 Heartbeat 메시지를 전송하고 다음 호스트가 메쉬의 멤버가 아닌 경우에는 Push Join 메시지를 보내고 응답을 기다린다. Push Join 메시지를 받은 호스트는 자신이 Push Join 메시지를 보낸 호스트가 지정한 다음 호스트이고 Push Join 메시지에 명시된 그룹의 멤버이며 Push Join 메시지가 가르키는 송신원까지의 경로를 알고있는 경우에 응답을 보낼수 있다. 응답을 받은 호스트는 응답을 보낸 호스트를 그 그룹의 앵커로 지정한다. Push Join 메시지를 보내고 응답을 받지 못한 호스트는 JOIN Request의 경우와 마찬가지로 재전송 메커니즘에 의해 다시 Push Join 메시지를 전송한다. 또한 만약 이미 Push Join 메시지를 보냈던 이웃과의 연결인 끊어졌을 경우에도 유니캐스트 라우팅 프로토콜에 의해 갱신된 정보에 의해 다른 이웃으로 Push Join 메시지를 전송한다.

2.3 CAMP에서의 메쉬 탈퇴

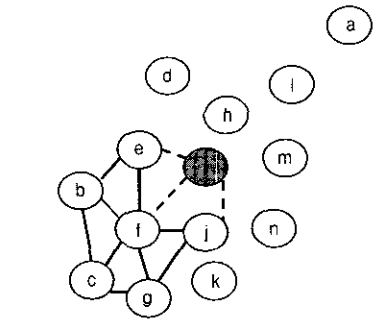
메쉬에서 탈퇴하려는 호스트는 자신의 멀티캐스트 라우팅 테이블을 변경하고 이웃들에게 탈퇴를 알리는 멀티캐스트 라우팅 업데이트 메시지를 보낸다. CAMP에서는 해당 호스트로부터 송신원으로 가는 RSP상의 다음 호스트를 앵커라는 정보로 가지고 있으며 이 정보를 이웃의 호스트들에게 알려준다. 탈퇴하려는 호스트는 자신을 앵커로 사용하고 있는 이웃이 없다면 탈퇴가 가능하다. 트리구조와 달리 메쉬구조에서는 패킷이 지나가는 경로가 정해져있지 않으므로 탈퇴 메시지에 대해서는 어떤 응답도 요구되지 않는다. CAMP에서는 코어 또한 그 코어를 앵커로 이용하는 호스트가 없는 한 메쉬에서 탈퇴할 수 있다.

3. CATMP

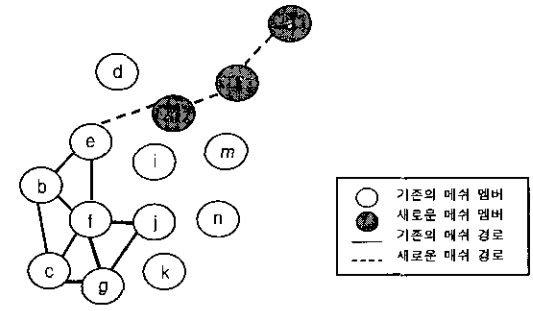
본 장에서는 2장에서 살펴본 CAMP를 보완한 CATMP를 제안한다. 먼저 CAMP의 성능이 저하되는 경우를 간략히 살펴보고, 그 문제점을 해결하기 위해 CATMP에서 제안한 방식에 대하여 살펴보도록 한다.

3.1 CAMP의 성능이 저하되는 경우

CAMP에서 메쉬에 참여하려는 호스트가 이미 메쉬의 멤버인 이웃들이 존재하지 않아 코어로 JOIN Request를 보내는 경우에는 호스트는 단일 경로를 통해 메쉬에 연결하게 된다. 이런 경우에 트리구조와 마찬가지로 단일 경로만을 제공받게 되고 호스트의 이동성에 의해 이 경로가 끊어질 경우에 데이터 트래픽을 전송받지 못하게 된다. (그림 2)는 메쉬에 참여하려는 호스트에게 메쉬 멤버인 이웃이 있을 때와 없을 때의 차이를 보여준다. 그림에서 호스트 i가 메쉬에 참여하려고 할 때 i는 기존 메쉬의 멤버인 이웃 호스트 e, f, j가 존재하므로 이 이웃들에게 메쉬 참여를 알리는 것 만으로 메쉬에 참여할 수 있고 3개의 연결 경로를 통해 메쉬에 연결된다. 그러나 호스트 a가 메쉬에 참여하려고 할 때는 기존 메쉬 멤버인 이웃이 존재하지 않으므로 JOIN Request를 통해 호스트 l과 h를 메쉬의 멤버로 만들면서 메쉬에 참여하게 되고 메쉬와의 연결 경로는 단일하게 된다. 그림에서 보는 것과 같이 메쉬의 중앙부에서 멀리 떨어진 송·수신원이 JOIN Request를 하는 경우 연결된 단일 경로가 여러 호스트를 지나는 멀티 홉을 갖게 되므로 그 호스트들 중 하나라도 전송 범위에서 벗어나면 메쉬와 호스트간의 연결이 끊어지게 되고 다시 경로가 설정될 때까지 데이터를 전송받지 못하게 된다. 또 그룹의 멤버가 적



<메쉬 멤버인 이웃이 존재할 때>



<메쉬 멤버인 이웃이 존재하지 않을 때>

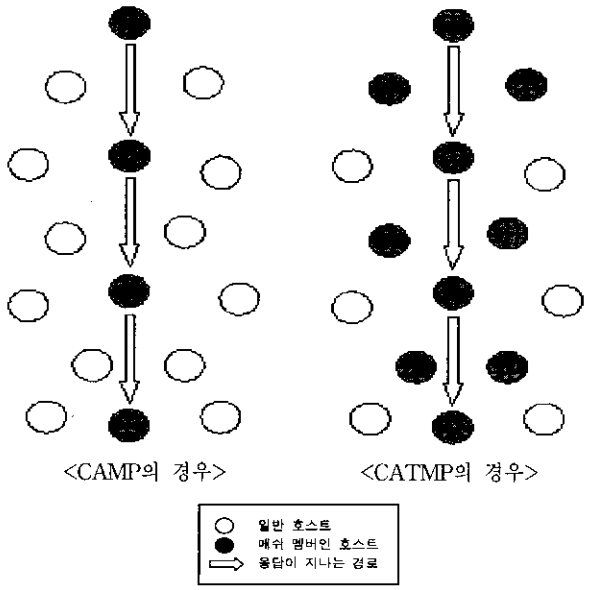
(그림 2) 메쉬 멤버인 이웃의 유·무에 따른 차이

은 경우에도 메쉬 멤버인 이웃을 통해 메쉬에 참여하는 것이 어려워지므로 경로가 단일해 지기 쉽고 이에 따라 데이터의 수신율이 저하된다. 즉 CAMP에서는 메쉬의 중앙부에 비해 가장자리에 위치하는 그룹 멤버들, 또는 적은 그룹 멤버로 인해 메쉬가 성기게 형성되어 단일경로를 갖기쉬운 그룹 멤버들의 경우와 같이 다른 메쉬 멤버들로부터 여러 경로로 데이터를 받아들이지 못하는 경우 호스트 이동성에 의해 전송 경로가 끊어지기 쉬워 데이터의 수신율이 낮아지게 된다.

3.2 CAMP를 보완한 CATMP

CAMP에서 데이터의 수신율이 낮은 경우는 모두 메쉬 구조가 메쉬의 특징인 다중경로를 제공하지 못하고 트리구조처럼 단일 경로만을 제공할 때 발생하였다. 이에 제안한 CATMP에서는 메쉬에 참여하는 호스트에게 이웃들만 존재한다면 그 이웃들을 통해 다중 경로를 제공할 수 있는 스킴을 CAMP에 추가해주었다.

CAMP에서 메쉬에 참여하려는 호스트는 이미 메쉬 멤버인 이웃 호스트가 존재하지 않을 경우 JOIN Request를 코어로 전송한다. 이를 받게 된 코어 또는 메쉬 멤버는 JOIN Request에 대한 응답을 발생한다. 이 응답은 JOIN Request가 경유해온 호스트들을 역으로 따라가며 전송되고 경유한 호스트들을 메쉬의 멤버로 승인해 준다. 이 경우에는 응답이 지나는 경로상에 있는 호스트들만이 메쉬의 멤버로 참여하게되고 호스트는 단일 경로를 통해 메쉬에 연결될 확률이 높다. 이에 CATMP에서는 응답이 전달되는 과정에서 응답이 지나는 경로상의 호스트만이 메쉬 멤버가 되는 것이 아니라 이웃들도 메쉬에 참여하도록 유도한다. 즉 JOIN Request에 대한 응답을 받은 호스트는 자신에게 메쉬의 멤버인 이웃이 몇개나 존재하는지를 살펴보고 MEMBER_NBR_COUNT(본 논문의 시뮬레이션에서는 3을 사용) 미만이라면 자신의 이웃에게 메시지를 보내 각 이웃이 갖고 있는 이웃 중 메쉬 멤버인 이웃이 몇 개나 되는지를 알아본다. 그리고 메쉬 멤버인 이웃을 많이 가지고 있는 이웃을 우선적으로 메쉬에 함께 참여하도록 유도한다. (그림 3)은 CAMP와 CATMP에서 JOIN Request에 대한 응답이 지나오는 경로를 따라 생성되는 메쉬 멤버의 변화를 보여준다. (그림 3)은 CAMP에서 JOIN Request에 대한 응답이 돌아오는 경로상의 호스트만이 메쉬 멤버가 되는 것에 비해 CATMP에서는 주변의 이웃들도 메쉬 멤버에 참여하게 됨을 보여준다.

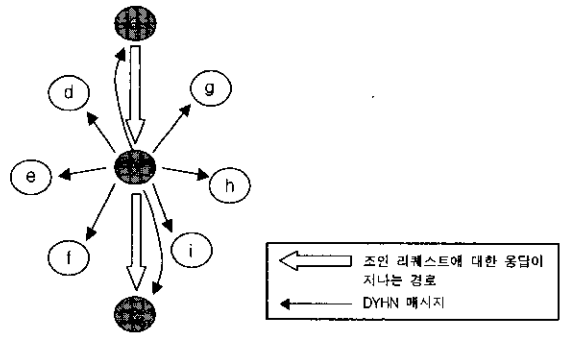


(그림 3) 응답을 통해 메쉬에 참여하는 멤버 수의 차이

만약 JOIN Request에 대한 응답을 전달받은 호스트가 이미 MEMBER_NBR_COUNT이상의 이웃을 메쉬 멤버로 갖는다면 CAMP와 마찬가지로 JOIN Request를 보냈던 호스트로 단순히 응답을 전달해 준다. 따라서 CATMP의 방식은 데이터의 신뢰성있는 전송을 위해 메쉬가 성글게 형성되는 곳에서만 일련의 과정을 통해 이웃들을 메쉬에 참여시킴으로써 메쉬를 강화해주고, 메쉬가 충분히 강하게 형성된 곳에서는 별도의 작용을 하지 않음으로서 불필요한 제어 패킷이나 데이터 패킷 복사본 수가 늘어나지 않도록 한다.

3.2.1 DYHN 메시지

메쉬 멤버로부터 JOIN Request를 보낸 호스트에게로 응답이 전송될 때, 이 응답이 지나가는 경로상에 있는 호스트들은 응답을 받으면 우선 자신에게 존재하는 메쉬 멤버인 이웃들의 수를 확인한다. 만약 메쉬 멤버인 이웃의 수가 MEMBER_NBR_COUNT미만이라면 자신의 모든 이웃들에게 DYHN 메시지를 전송한다. (그림 4)는 JOIN Request에 대한 응답을 받은 호스트가 가지고 있는 메쉬 멤버인 이웃의 수가 MEMBER_NBR_COUNT미만이라서 이웃에게 DYHN 메시지를 전송하는 것을 보여준다. DYHN 메시지는 모든 이웃들에게 브로드캐스트되며 이미 메쉬 멤버인 이웃들을 갖고 있는지의 여부와 갖는다면 몇 개의 이웃을 갖는지를 물어보는 메시지이다.



(그림 4) 응답을 받은 호스트가 발생하는 DYHN 메시지

DYHN 메시지의 포맷은 (그림 5)와 같다. MgAddr은 해당 멀티캐스트 그룹의 주소를 나타내고 ReqType은 DYHN

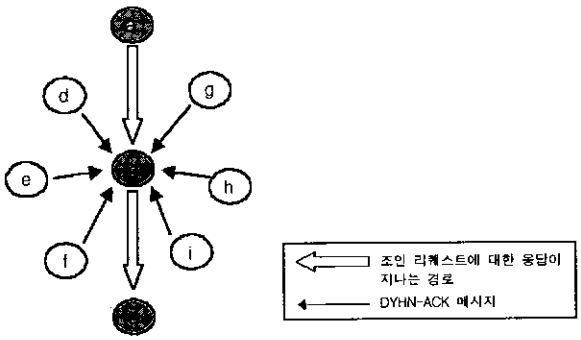
메시지를 전송하는 호스트의 멤버쉽 타입으로 심플렉스 모드인지 듀플렉스 모드인지를 알려준다. srcAddr은 메시지를 전송하는 호스트의 주소로 (그림 4)의 경우에 호스트 b의 주소를 명시해준다. exceptAddr1과 exceptAddr2에는 원래 JOIN Request에 대한 응답이 거치는 경로상에 있는 호스트들을 표시하여 이 이웃들은 DYHN메시지에 응답할 필요가 없음을 알려준다. 이는 여차피 CAMP에서 JOIN Request에 대한 응답이 경유하는 경로상의 호스트들은 메쉬의 멤버가 되므로 그 외의 이웃들을 메쉬에 참여시키기 위함이다. (그림 4)의 경우에 호스트 a와 호스트 c의 주소가 이에 해당한다.

MgAddr	ReqType	srcAddr	exceptAddr1	exceptAddr2

(그림 5) DYHN 메시지의 포맷

3.2.2 DYHN-ACK 메시지

DYHN-ACK 메시지는 DYHN 메시지를 받은 호스트들이 발생하는 메시지로 DYHN 메시지에 대한 응답이다. (그림 6)은 DYHN 메시지를 받은 이웃 호스트들이 DYHN-ACK 메시지를 응답하는 모습을 보여준다. 여기서 호스트 b에게 JOIN Request에 대한 응답을 전해준 호스트 a와 호스트 b가 JOIN Request에 대한 응답을 전달할 호스트 c는 DYHN-ACK 메시지를 발생시키지 않음을 볼 수 있다.



(그림 6) DYHN 메시지에 대한 DYHN-ACK발생

(그림 7)은 DYHN-ACK 메시지의 포맷을 보여준다. DYHN 메시지를 받은 호스트는 먼저 exceptAddr1 또는 exceptAddr2인지를 보고 자신이 둘 중 하나라면 DYHN-ACK 메시지를 생성하지 않는다. 그렇지않으면 ReqType을 보고 심플렉스 모드인 경우 자신이 갖고있는 심플렉스 모드의 이웃 수와 듀플렉스 모드의 이웃 수를 합하여 nbrCount에 넣어주고 ReqType이 듀플렉스 모드인 경우에는 자신이 갖고있는 듀플렉스 모드의 이웃 수를 nbrCount에 넣어준다. srcAddr에는 응답하는 호스트의 주소를 적어주고 destAddr에는 DYHN 메시지를 전송했던 호스트의 주소를 적어 DYHN 메시지를 보냈던 호스트에게 DYHN-ACK 메시지를 전송해 준다.

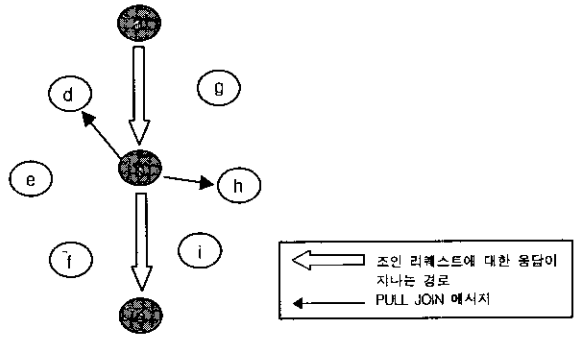
MgAddr	srcAddr	nbrCount	destAddr

(그림 7) DYHN-ACK 메시지의 포맷

3.2.3 PULL JOIN 메시지

DYHN 메시지를 전송했던 호스트는 자신의 이웃들로부터 DYHN-ACK 메시지를 받게되고 이를 바탕으로 메쉬에 참여할 것을 유도할 이웃을 선정한다.

메쉬의 형성을 더 강하게 하기 위해서 메쉬 멤버인 이웃을 많이 갖는 호스트들을 우선순위로 메쉬에 참여하도록 하는데, 이때 DYHN-ACK 메시지의 nbrCount 값을 근거로한다. (그림 8)은 DYHN-ACK 메시지에 보고 결정한 이웃으로 PULL JOIN 메시지를 보내는 모습을 보여준다. DYHN 메시지를 전송했던 호스트는 자신이 갖고 있는 메쉬 멤버인 이웃의 수와 MEMBER_NBR_COUNT의 차를 계산하여 그 차만큼의 이웃들에게 메쉬 멤버로 참여할 것을 요청한다.



(그림 8) 메쉬 참여를 유도하는 PULL JOIN 메시지

(그림 9)는 PULL JOIN 메시지의 포맷을 보여준다. ReqType에는 PULL JOIN 메시지를 받는 호스트가 메쉬에 참여하길 원하는 모드가 심플렉스 모드인지 듀플렉스 모드인지를 나타내고 destAddr에는 메쉬에 참여하길 원하는 이웃의 주소를 적어준다. srcAddr에는 PULL JOIN 메시지를 전송한 호스트의 주소를 명시하여 PULL JOIN 메시지를 받은 호스트가 메쉬 멤버인 이웃을 확인할 때 자신이 알고있는 이웃에서 제외하도록 한다. PULL JOIN 메시지를 받은 호스트는 먼저 자신이 destAddr에 명시되어 있는 호스트인지를 확인하고 명시되어 있을 경우에만 메시지 처리 과정을 수행하고 그렇지않으면 메시지를 삭제한다.

MgAddr	ReqType	srcAddr	destAddr

(그림 9) PULL JOIN 메시지의 포맷

PULL JOIN 메시지를 받은 호스트는 PULL JOIN 메시지를 전송한 이웃 외에 이미 메쉬 멤버인 이웃이 존재한다면 이웃들에게 자신이 메쉬 멤버가 되었음을 알림으로써 메쉬

에 참여한다. 그러나 PULL JOIN 메시지를 받은 호스트에게 메쉬 멤버인 이웃이 PULL JOIN 메시지를 전송한 호스트외에는 하나도 없다면 이 호스트는 코어로 JOIN Request를 보낸다. (그림 8)의 경우에 호스트 d와 호스트 h가 PULL JOIN 메시지를 통해 메쉬의 멤버가 된다.

4. 시뮬레이션 모델 및 구현

본 장에서는 CAMP와 CATMP의 성능을 평가하기 위한 시뮬레이션 모델을 설명하고 그 결과를 분석한다. 시뮬레이션은 멀티캐스트 그룹 멤버의 수와 송·수신원의 위치를 변화시켜 보면서 성능을 측정하였다. 시뮬레이션 환경은 NT 워크스테이션에서 C언어를 기반으로한 시뮬레이션 툴인 GloMoSim(Global Mobile Information System Simulator) 라이브러리를 사용하였다. GloMoSim은 무선 시스템을 위한 확장성이 있는 시뮬레이션 환경을 만들도록 도와준다[10]. 시뮬레이션에 필요한 파라미터 값들은 CAMP의 성능 평가를 수행하는 기존의 논문들을 참조하였다[1, 6, 8]. 4.1에서는 CAMP와 CATMP의 성능 평가를 위한 시뮬레이션 모델을 제시하고 4.2에서는 실험 결과들을 살펴보고 이를 토대로 두 방식의 성능을 비교한다.

4.1 시뮬레이션 환경

본 논문에서는 CAMP의 성능이 저하되는 상황을 특별히 연출하여 그때의 CAMP 성능과 CAMP의 개선안인 CATMP의 성능을 비교, 평가하기 위한 모델을 설정하였다[11].

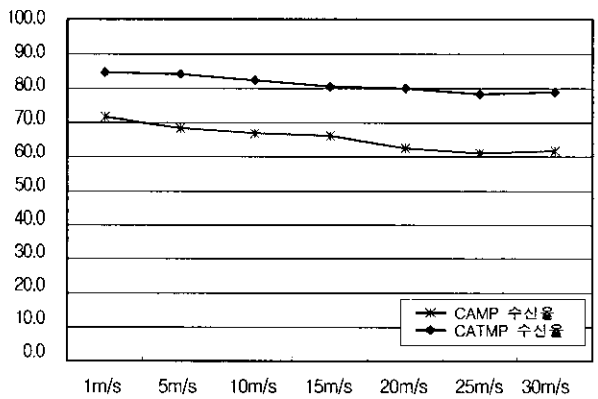
실험은 크게 두 가지로 나누어 진행되었는데, 첫 번째는 송·수신원의 위치에 따른 성능 평가이고 두 번째는 멀티캐스트 그룹 멤버의 수에 따른 성능 평가이다. 첫 번째 경우는 메쉬를 생성하는 그룹 멤버들을 서로 가까운 거리에 위치시켜 메쉬가 특정 지역에 조밀하게 생성되도록 한 후, 하나의 송신원이나 수신원만을 메쉬의 중심부에서 멀리 떨어져 위치하게 하므로써 수신원일 경우 그 호스트의 데이터 수신율을 비교하고 송신원일 경우에는 전체 그룹 멤버들에게 미치는 영향을 살펴보았다. 이 때, 그룹의 송신원은 하나이고 코어 역시 하나로 설정하였다. 두 번째 경우는 무작위로 선출된 멀티캐스트 그룹 멤버가 5개인 경우와 10개인 경우로 나누어 그룹 멤버의 수가 많을 때와 적을때의 성능을 비교하였다. 이 때, 그룹의 송신원은 하나이고 코어는 두 개로 설정하였다. 두가지 실험 모두 그룹의 멤버쉽은 시뮬레이션 시작부터 종료시점까지 지속되고 중간에 탈퇴하지 않는다고 가정하였다.

실험을 위한 네트워크의 크기는 1km×1km이고 설정하고 이 범위안에 호스트 50개를 임의로 배치하였다. 각 호스트들의 이동 속도는 1~30m/sec로 5m/sec의 간격으로 이동속도를 변화시키면서 시뮬레이션을 수행하였고 각 호스트들의 전파 전송 범위는 250m이며 채널의 용량은 2Mbits/sec이라고 가정하였다. 그리고 메쉬 멤버간에 전송되는 패킷의 크기는 512bytes라고 가정하였다. 또 Radio propagation모델은 Free space전송, MAC 프로토콜은 IEEE 802.11, 그리고 Radio 타입은 Radio-capture를 사용하였다. 데이터의 전송은 응용 계층에서 일정한 비트로 데이터를 발생하는 CBR을 사용하였고 1초당 2개씩 패킷을 생성하여 전송하였으며 각 실험에서의 시뮬레이션 시간은 600초로 하였다.

4.2 시뮬레이션 수행과 결과 분석

4.2.1 실험 1

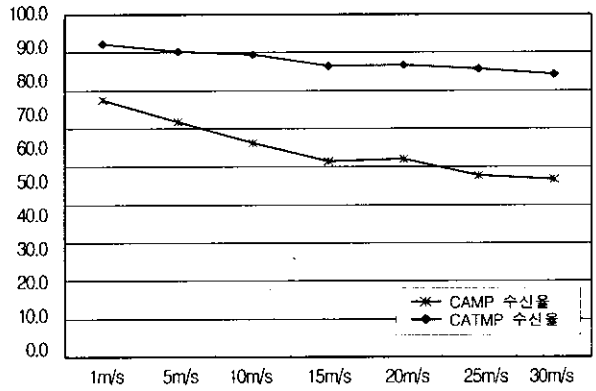
실험 1에서는 코어를 중심으로 메쉬가 강하게 형성되어 있는 지역에서 멀리 떨어져있는 송·수신원을 설정하여 그때의 데이터 수신율을 실험하였다. (그림 10)은 메쉬에서 멀리 떨어져 있는 멀티캐스트 그룹 멤버의 수신율을 이동 속도에 따라 보여준다. 그림에서 보듯이 멀티캐스트 그룹 멤버가 메쉬의 중심부에서 멀리 떨어져서 위치할 경우 CAMP에서는 해당 호스트에 데이터의 수신율이 상당히 낮고 또 이동속도에 따라 점차적으로 낮아지는 모습을 볼 수 있다. 그러나 CATMP에서는 이동 속도에 따라 수신율이 감소하기는 하나 그 차이가 CAMP보다 적고 거의 80%이상을 유지함을 볼 수 있다. 이런 결과는 CAMP의 경우 메쉬의 중심부에서 멀리 떨어져서 멀티캐스트 그룹 멤버가 위치하면 메쉬까지의 경로가 단일할 확률이 크기 때문에 이동성이 커짐에 따라 경로가 손실될 확률도 커지기 때문이다. 반면에 CATMP에서는 응답이 돌아오는 과정에서 이웃 호스트들을 메쉬에 참여시킴으로써 데이터를 전송해줄 이웃 호스트를 더 많이 확보하여 멀리 떨어져 있는 수신원에 이르는 경로가 손실될 우려가 적기 때문에 CAMP에 비해 높은 수신율을 유지한다고 볼 수 있다.



(그림 10) 메쉬의 중심부에서 멀리 떨어진 그룹 멤버의 수신율

(그림 11)은 데이터를 전송해 주는 송신원이 메쉬의 중심부에서 멀리 떨어져 위치할 경우의 멀티캐스트 그룹 멤버

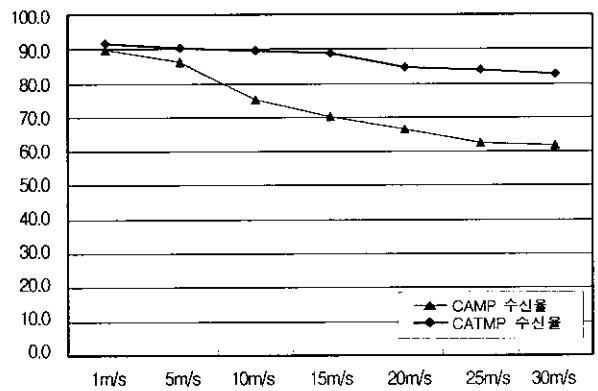
들의 수신율이다. (그림 11)을 보면 송신원이 메쉬와 단일 경로를 갖는 경우는 그룹의 멤버가 단일 경로를 받는 경우보다 전체 그룹에 미치는 영향이 훨씬 더 크다. 이는 데이터를 전송해주는 송신원이 메쉬에서 멀리 떨어진 경우 송신원과 메쉬의 연결 경로가 손실되면 대부분의 그룹 멤버가 영향을 받게 되기 때문이다. 이러한 CAMP의 경우와 달리 CATMP는 송신원과 메쉬의 연결 경로를 좀 더 강하게 확보해 주기 때문에 CAMP에 비해 더 높은 수신율을 보임을 알 수 있다.



(그림 11) 송신원이 메쉬에서 멀리 떨어진 경우 그룹 멤버의 평균 수신율

4.2.2 실험 2

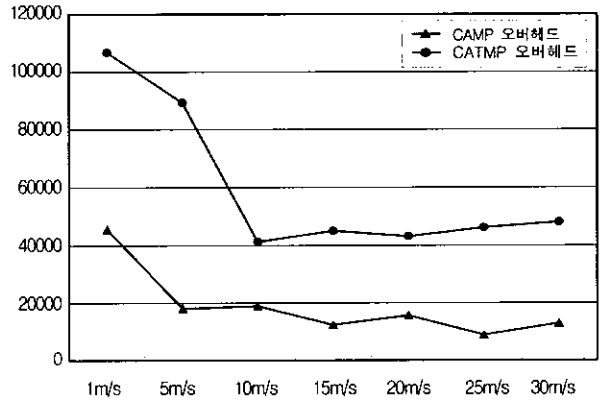
실험 2에서는 그룹의 멤버수가 5개일 때와 10개일 때의 프로토콜 성능을 측정하기 위해 수신원들이 성공적으로 데이터 패킷을 받은 수신율을 측정한다. 또 네트워크에 발생한 오버헤드를 측정하기 위해 메쉬 멤버인 각 호스트에서 중복으로 수신하는 데이터 패킷들의 합을 측정하였다. 제어 패킷도 오버헤드 트래픽이지만 두 프로토콜은 모두 제어 패킷이 플러딩되지 않아 제어 패킷의 양이 중복 데이터 패킷의 양에 비하여 매우 적기 때문에 중복 수신되는 데이터 패킷의 수만을 오버헤드로 측정하였다. 마지막으로 메쉬를 생성하기 위해 필요한 제어 패킷의 수를 측정하여 제어 패킷당 수신된 데이터 패킷의 양을 계산하여 효율성을 분석해보았다.



(그림 12) 멀티캐스트 그룹 멤버가 10개일 경우의 데이터 평균 수신율

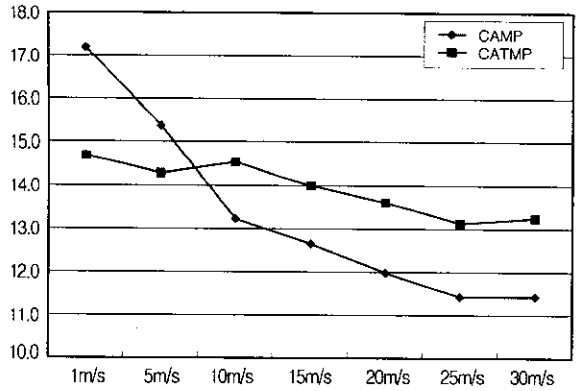
(그림 12)는 멀티캐스트 그룹 멤버가 10개일 경우의 수신율을 호스트의 이동 속도에 따라 보여준다. CAMP 수신율이 호스트의 이동성이 증가함에 따라 급격히 낮아지는데 비해 CATMP는 이동성이 커짐에 따라 약간 수신율이 감소하지만 80%이상의 수신율을 유지함을 볼 수 있다. 이는 CAMP가 호스트의 이동성이 커져 단일 경로가 끊어질 경우 데이터를 전송받지 못하는데 비해 CATMP는 호스트의 이동성이 커져도 데이터를 전송해주는 이웃이 상대적으로 더 많고 이 이웃들로의 연결이 모두 끊어질 확률이 적으므로 데이터를 성공적으로 받을 확률이 커짐을 보여준다.

(그림 13)은 (그림 12)의 수신율을 보일 때 네트워크에 발생한 오버헤드를 보여준다. CATMP의 경우 CAMP에 비해 데이터를 전송받는 중복 경로가 더 많기 때문에 (그림 13)에서 보이는 것과 같이 CAMP에 비해 약 2~3배 정도의 데이터 중복이 발생하게 된다.



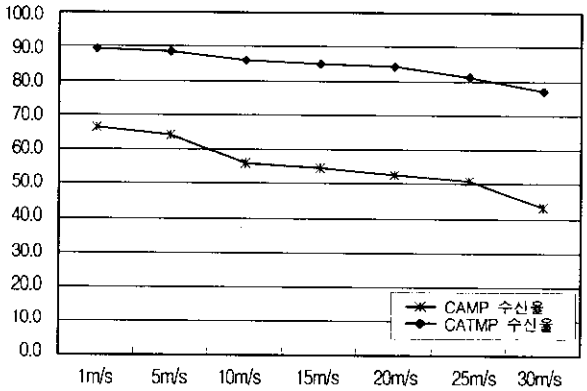
(그림 13) 네트워크에 발생한 오버헤드(그룹 멤버 수 : 10개)

(그림 14)는 네트워크에 발생한 총 제어 패킷에 대해 수신원에 성공적으로 도착한 패킷의 수를 보여준다. 제어패킷은 데이터 패킷을 제외한 메쉬를 생성하고 유지하기 위해 발생되는 모든 패킷을 뜻한다. 즉, CAMP에서는 JOIN Request, JOIN Request에 대한 응답, 멀티캐스트 라우팅 업데이트 메시지, Push Join 메시지, Heartbeat 메시지를 포함하고 CATMP에서는 CAMP에서의 제어 패킷에 DYHN 메시지, DYHN-ACK 메시지, PULL JOIN 메시지를 추가로 포함한다. CAMP의 경우 호스트들의 이동 속도가 증가함에 따라 제어 패킷당 수신원에 성공적으로 도착한 패킷의 수가 감소한다. 이는 속도가 증가함에 따라 메쉬의 연결성을 확보하기 위해 제어 패킷들은 더 많이 전송되고 수신원에서의 데이터 패킷 수신율은 떨어지기 때문이다. 이와 달리 CATMP에서는 제어 패킷이 어느 정도 증가하여도 데이터의 수신율이 일정하게 유지되므로 비율이 약간 감소하는 경향은 있으나 13~15정도로 비교적 고른 모습을 보여준다.



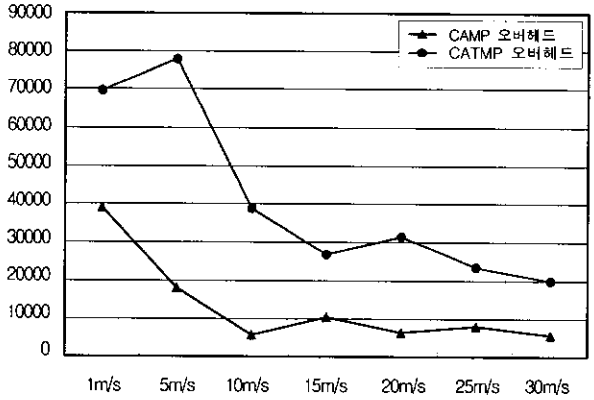
(그림 14) 제어 패킷 당 수신원에 성공적으로 도착한 패킷의 수 (그룹 멤버 수 : 10개)

(그림 15)는 멀티캐스트 그룹 멤버가 5개일 경우의 수신율을 호스트의 이동속도에 따라 보여준다. 이 경우는 앞에서 살펴본 멀티캐스트 그룹 멤버가 10개일 경우보다도 더 큰 성능의 향상을 보여준다. 멀티캐스트 그룹 멤버가 5개인 경우에는 10개일 때보다 더 메쉬가 성기게 형성될 확률이 높게되고 이에 따라 단일 경로들이 많아질 우려가 높다. 따라서 이 단일 경로가 손실될 경우 데이터의 전송이 이루어지지 못하게되어 수신율이 낮아지게 된다. 이와 달리 CATMP에서는 그룹 멤버의 수와 상관없이 메쉬에 참여하려는 호스트가 자신에게 데이터를 전송해 줄 이웃 호스트들을 일정 수 이상으로 확보하므로써 경로가 다양해지게 된다. 따라서 연결 경로가 끊어질 확률이 적으므로 데이터 전송에 있어 좀 더 높은 신뢰성을 가질 수 있다.

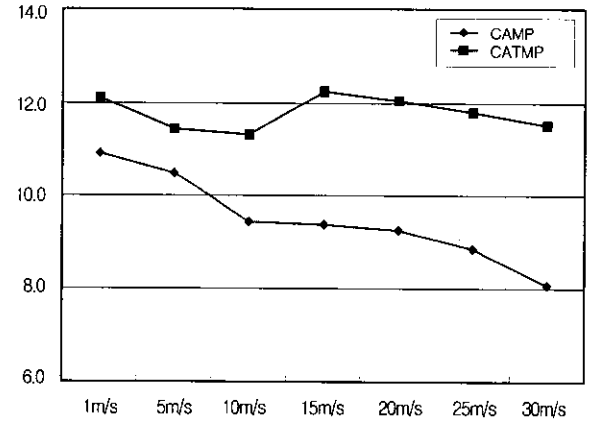


(그림 15) 멀티캐스트 그룹 멤버가 5개일 경우의 데이터 평균 수신율

(그림 16)과 (그림 17)은 (그림 15)의 수신율을 보일 때 네트워크에 발생한 오버헤드와 네트워크에 발생한 총 제어 패킷에 대해 수신원에 성공적으로 도착한 패킷의 수를 보여준다. 그룹 멤버의 수가 10개일 때 비교했던 것과 마찬가지로 (그림 16)은 CATMP가 CAMP에 비해 약 2~3배 정도의 데이터 중복이 발생함을 보여준다. 그리고 그림 14에서 보면 그룹 멤버의 수가 10인 경우에는 이동성이 1m/s~5m/s정도의 낮은 구간에서는 CAMP의 효율이 더 좋았다가 이동성이 커질수록 CATMP의 효율이 더 좋아지는데 반해 (그림 17)에서 보면 그룹 멤버의 수가 5개인 경우에는 모든 이동속도에 대해 CATMP의 효율이 더 좋음을 알 수 있다. 그리고 CAMP에 비하여 CATMP의 경우 이동속도에 대한 프로토콜의 효율이 더 일정하게 유지됨을 알 수 있다.



(그림 16) 네트워크에 발생한 오버헤드(그룹 멤버 수 : 5개)



(그림 17) 제어 패킷 당 수신원에 성공적으로 도착한 패킷의 수 (그룹 멤버 수 : 5개)

5. 결 론

본 논문에서는 애드 혹 네트워크에서의 멀티캐스트 라우팅 프로토콜인 CAMP를 보완한 프로토콜인 CATMP를 제안하였다. CAMP의 경우 메쉬를 형성하는 과정상의 특징으로 인해 메쉬의 중심부에 비해 메쉬 멤버의 수가 적은 가장자리에서는 다중 경로가 충분히 제공되지 못하는 약점을 갖는다. 이런 상황이 발생하는 곳에서 CATMP는 JOIN Request에 대한 응답이 돌아오는 경로 상의 호스트뿐만이 아니라 이웃 호스트들 또한 메쉬에 참여하도록 유도하는 스킴을 추가되어 결과적으로 좀더 연결성이 강하고 균일한 메쉬를 형성하도록 한다.

본 논문에서는 시뮬레이션을 통하여 CAMP에서 성능이

저하되는 두가지 상황을 연출하여 CAMP와 CATMP의 성능을 비교하여 보았다. 그 결과 본 논문에서 제안한 CATMP가 CAMP에 비해 높은 수신율을 보임을 확인할 수 있었는데, 이는 CATMP가 일정한 수준의 연결성이 강한 메쉬를 형성하여 데이터 트래픽을 전송하기 때문으로 CAMP에 비해 데이터 전송 경로가 손실될 확률이 훨씬 적기 때문이다. 다만 CATMP는 CAMP에 비해 메쉬 멤버의 수가 늘어나면서 중복된 데이터가 많이 발생하게 되어 2~3배의 오버헤드가 발생한다. CATMP는 메쉬와의 연결성이 약한 부분에서 이웃 호스트의 참여를 유도하여 모든 경로가 손실되는 확률을 줄임으로써 멀티캐스트 그룹 멤버들에게 성공적으로 도착하는 데이터의 수신율을 높일 수 있었지만, 메쉬의 멤버인 이웃이 증가함으로써 중복되어 수신되는 데이터 패킷의 양도 늘어났다. 특히 이동 속도가 빠르지 않은 경우에는 메쉬의 경로가 끊어질 확률이 적으므로 중복되는 데이터로 인한 프로토콜의 효율 감소가 있다. 따라서 어플리케이션의 이동 속도에 따라 CATMP에서 사용하는 MEMBER_NBR_COUNT 값을 조정함으로써 메쉬의 강화 정도를 결정해 주는 것이 효율성 면에서 좋으리라고 생각된다.

참 고 문 헌

[1] Sung-Ju Lee, William Su, Julian Hsu, Mario Gerla, Rajive Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," Proceeding of Inforcom'2000, 2000.

[2] Katia Obraczka, Gene Tsudik, "Multicast Routing Issues in Ad Hoc Networks," Proceeding of IEEE ICUPC'98, 1998.

[3] E. Bommaiah, M. Liu, A. McAuley, R. Talpade, "AMRoute : Ad-hoc Multicast Routing Protocol," Internet-Draft, draft-talpade-manet-amroute-00.txt, Aug. 1998.

[4] C. W. Wu, Y. C. Tay, C. K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification," Internet-draft, draft-ietf-manet-amris-spec-00.txt, Nov. 1998.

[5] S. J. Lee, M. Gerla, C. C. Chiang, "On-demand Multicast Routing Protocol," Proceedings of IEEE WCNC'99, Sep. 1999.

[6] J. J. Garcia-Luna-Aceves, E. L. Madruga, "The Core-Assisted Mesh Protocol," IEEE JSAC Vol.17, No.8, Aug. 1999.

[7] J. J. Garcia-Luna-Aceves, E. L. Madruga, "A Multicast routing Protocol for Ad-Hoc Networks," IEEE INFOCOM '99, Mar. 1999.

[8] E. L. Madruga, J. J. Garcia-Funa-Aceves, "Multicasting Along Meshes in Ad-Hoc Networks," IEEE ICC'99, Jun. 1999.

[9] E. L. Madruga, J. J. Garcia-Funa-Aceves, "Scalable Multicasting : The core Assisted Mesh Protocol," ACM, 1999.

[10] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim : A Library for Parallel Simulation of Large-Scale Wireless Networks," PADS'98, May 1998.

[11] 임수정, "CATMP의 simulation을 위한 source", http : // myhome.shinbiro.com/~sjlim98/catmp.c, 2001.

임 수 정

e-mail : tncud@hanmail.net

1998년 이화여자대학교 컴퓨터학과(학사)

2001년 이화여자대학교 컴퓨터학과(석사)

2001년~현재 LG전자 연구소

관심분야 : Ad-hoc 네트워크, 멀티캐스트 라우팅 프로토콜

이 미 정

e-mail : lmj@ewha.ac.kr

1987년 이화여자대학교 전자계산학과 졸업 (학사)

1989년 University of North Carolina at Chapel Hill 컴퓨터학과(석사)

1994년 North Carolina State University 컴퓨터공학과(공학박사)

1994년~현재 이화여자대학교 컴퓨터학과 부교수

관심분야 : 고속 통신 프로토콜 설계 및 성능 분석, 비디오 전송을 위한 트래픽 제어, 인터넷에서의 QoS 지원, Ad-hoc 네트워크