

무선 애드-홉 네트워크를 위한 다중-홉 클러스터 라우팅 프로토콜

(Multi-Hop Cluster Routing Protocol in Wireless Ad-hoc Network)

전 형 국^{*} 김 문 정^{**} 엄 영 익^{***}

(Hyung-Kook Jun) (Moon-Jeong Kim) (Young-Ik Eom)

요 약 무선 ad-hoc 네트워크는 중앙의 특별한 관리 체계 없이 기존의 유선 네트워크 또는 기지국을 사용하지 않는 이동 호스트들만으로 구성된 네트워크를 말한다. 이러한 무선 ad-hoc 네트워크는 잦은 망 구성의 변화, 라우터의 수, 제한된 사용 자원 등 기존 유선 네트워크와는 다른 특성들을 가지게 된다. 따라서 기존의 유선 네트워크에서 사용하던 라우팅 프로토콜들을 무선 ad-hoc 네트워크에 그대로 적용시킨다면 많은 문제점들이 발생하게 된다. 본 논문에서 제안하는 라우팅 프로토콜은 무선 ad-hoc 네트워크의 특성을 고려하여, 네트워크 내의 이동 호스트를 멀티 홉을 갖는 클러스터로 묶고, 클러스터 헤드로 하여금 자신의 멤버 호스트들과 이웃 클러스터들의 헤드 정보를 유지하게 하여, 경로 설정에 대한 요구가 있을 때에 적은 지연 시간과 적은 패킷으로 목적지까지의 최단 경로를 설정할 수 있도록 한다. 또한 이동 호스트로 구성된 클러스터를 무선 ad-hoc 네트워크의 변화에 따라서 적절한 크기로 변화시켜 네트워크에 발생하는 경로 검색 오버헤드를 줄일 수 있게 하였고 경로 복구와 경로 유지에 드는 비용을 최소화하기 위한 알고리즘을 제시한다.

Abstract wireless ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any centralized administration or reliable support services such as wired network and base stations. Since wireless ad-hoc network is different from the conventional wired network by frequent changes in network formation, number of routers and resources, there are a number of problems in applying conventional routing protocol to wireless ad-hoc network. In this thesis, we suggested routing protocol which has lower routing delay and communication overhead considering ad hoc network environment. The proposed protocol divides the nodes of the ad hoc network into multi-hop clusters, and a cluster head maintains information of its members and neighboring cluster heads and discover the shortest route to the destination with lower routing delay, and network overhead upon request of route discovery. we reduce route discovery overhead by changing the size of the clusters formed with the nodes in terms of ad-hoc network environment variation and present algorithm that minimize route salvage and route maintenance cost.

1. 서 론

무선 ad-hoc 네트워크는 중앙의 특별한 관리 체계 없이

기존의 유선 네트워크 또는 기지국을 사용하지 않는 이동 호스트들만으로 구성된 네트워크를 말한다. 따라서 무선 ad-hoc 네트워크는 기존의 유선 네트워크의 도움 없이 필요시에 빠르게 구성될 수 있는 네트워크이다. 유선 네트워크를 구성하기 어려운 군사 통신이나 인명 구조시의 긴급 통신, 단기간 사용되는 사업장, 교육 통신 등을 위해 무선 ad-hoc 네트워크가 사용될 수 있다.

그림 1은 이동 호스트들만으로 구성된 무선 ad-hoc 네트워크 환경과 특성들을 보여준다.

그림 1에서와 같이 무선 ad-hoc 네트워크는 유선 네트

* 이 논문은 2000년도 두뇌한국21 사업에 의하여 지원 되었음.

† 정 회 원 : 한국전자통신연구원 인터넷정보기전연구부 연구원
hkjun@etri.re.kr

** 학생회원 : 성균관대학교 전기전자및컴퓨터공학부
tops@ecc.skku.ac.kr

*** 종신회원 : 성균관대학교 전기전자및컴퓨터공학부 교수
yicom@simsan.skku.ac.kr

논문접수 : 2000년 5월 22일

심사완료 : 2001년 1월 29일

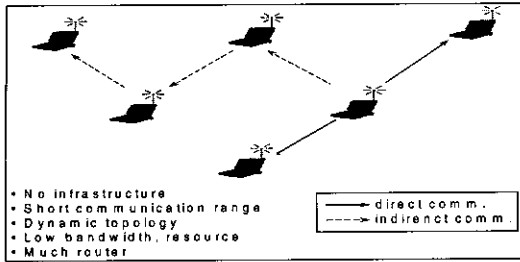


그림 1 무선 ad-hoc 네트워크 환경의 예

워크에 비해 상대적으로 라우터의 개수가 많고 이동 호스트들의 이동으로 인하여 네트워크 토폴로지가 동적으로 변하는 특성을 갖는다. 또한 무선 통신의 특성인 저 대역폭과 통신의 잦은 끊김이 발생하고 각 호스트들의 메모리, 배터리 등 시스템에 많은 제약이 존재한다. 이러한 특성들은 무선 ad-hoc 네트워크 환경을 위한 라우팅 프로토콜 설계 시에 반드시 고려되어야 할 중요한 요소들이다. 따라서 기존의 유선 네트워크 환경에서 사용되는 라우팅 프로토콜들을 무선 ad-hoc 네트워크에 그대로 적용시킬 경우에는 많은 문제점들이 발생하게 된다.

본 논문에서는 무선 ad-hoc 네트워크 환경 특성들을 고려하여, 동적으로 변하는 네트워크 환경에 적용 적이고 지연 시간과 오버헤드를 줄이면서 무선 ad-hoc 네트워크 내의 경로 정보를 획득할 수 있도록 하는 MHCR(Multi-Hop Cluster Routing) 프로토콜을 제안한다. MHCR 프로토콜은 무선 ad-hoc 네트워크 내의 이동 호스트들을 헤드와 멤버들로 구성된 멀티 홉 클러스터로 묶고, 클러스터 헤드로 하여금 자신의 클러스터 내에 존재하는 멤버들의 정보는 물론 이웃 클러스터 헤드의 정보를 유지하게 하여 경로 설정 과정에서 발생하는 네트워크 오버헤드를 최소화 하는 프로토콜이다. 또한 클러스터 크기를 일정 크기로 고정시키지 않고 이동 호스트들의 이동성을 바탕으로 클러스터의 크기를 재 설정할 수 있는 방법을 제안한다. 본 논문에서는 무선 ad-hoc 네트워크 내의 모든 이동 호스트들이 라우터의 역할을 수행하며 MHCR 프로토콜을 지원한다고 가정한다. 또한 각 이동 호스트간의 링크는 양방향 링크만을 고려하였으며, 이웃 호스트와의 링크 상태 감지는 MAC/link-layer protocol에 의해서 수행된다고 가정한다.

본 논문의 2절에서는 관련 연구를 소개하고 3절에서는 MHCR 프로토콜의 개요와 패킷 형식 및 자료 구조, 제안하는 경로 설정 방법 및 알고리즘에 대하여 설명한다. 4장에서는 제안하는 경로 설정 방법과 CBRP(Cluster Based Routing Protocol)[6]를 비교하였으며, 5장에서 결론 및 향후 연구과제를 설명한다.

2. 관련 연구

2.1 무선 Ad-hoc 네트워크 라우팅 프로토콜

현재 무선 ad-hoc 네트워크 환경을 위해 제안되고 있는 라우팅 프로토콜들은 크게 순향적(proactive) 설정 방식과 반응적(reactive) 설정 방식, 두 설정 방식을 혼합한 방식으로 분류될 수 있다.

순향적 설정 방식은 table driven 방식으로서, 각 이동 호스트가 자신의 라우팅 정보를 주기적으로 방송하여 각 이동 호스트로 하여금 네트워크 내의 경로 정보를 항상 유지하도록 하여 목적지에 대한 경로 정보가 필요할 때 즉시 경로 정보를 사용할 수 있는 방법이다. 이러한 방식의 프로토콜들은 경로에 대한 최신 정보를 항상 유지하기 위해 많은 네트워크 트래픽을 발생시키며 라우팅 정보의 방송 주기가 길어지면 대부분의 경로 정보가 쓸모 없게 되는 문제점을 갖는다. 제안되고 있는 순향적 설정 방식 라우팅 프로토콜로는 DSDV(Dynamic Source Routing Protocol)[7], OLSRP(Optimized Link State Routing Protocol)[8] 등이 있다.

반응적 설정 방식은 source-initiated on demand 방식으로서, 특정 목적지에 대한 경로 설정이 요구되는 경우 즉, 데이터 전송이 시작되기 바로 전에 경로 설정 절차를 수행하는 방식이다. 이러한 방식에서는 각 이동 호스트가 무선 ad-hoc 네트워크 내의 모든 경로 정보를 항상 유지하도록 요구하지 않기 때문에 네트워크 트래픽을 감소시키며 상대적으로 적은 메모리 공간을 필요로 한다. 그러나 경로 설정 과정에서 경로 설정 지연 시간이 발생하며, 데이터 전송 중 발생하는 경로 실패로 인해 경로 재 설정 작업이 요구될 수 있다. 제안되고 있는 반응적 설정 방식 라우팅 프로토콜로는 DSR(Dynamic Source Routing Protocol)[9], AODV(Ad Hoc On Demand Distance Vector)[10], RDMAR(Relative Distance Micro-discovery Ad Hoc Routing)[11], STAR(Source Tree Adaptive Routing)[12] 등이 있다.

혼합 방식은 순향적, 반응적 설정 방식의 혼합형태로 각 호스트가 사전에 임의의 홉 거리만큼 이웃 호스트의 라우팅 정보를 유지하고 홉을 넘어서는 호스트에 대해서만 경로 설정 절차는 요구하는 설정 방식을 말한다. 초기 목적지에 대한 경로 정보를 획득하기 위해서 각 호스트는 순향적 설정 방식을 통해서 유지된 라우팅 정보를 이용하고, 홉 안에 목적지 라우트 정보가 없을 경우 경로 설정 절차를 수행한다. 혼합 방식은 앞의 두 가지 설정 방식의 장점들을 사용함으로써 지역적인 네트워크 토폴로지 변화에 따른 전체 네트워크 토폴로지의 위상 변화를 막을 수 있

고, 경로 설정 과정에서의 지연 시간과 네트워크 오버헤드를 줄일 수 있다. 제안되고 있는 혼합 방식 라우팅 프로토콜로는 CBRP[6], ZRP[13] 등이 있다.

2.2 CBRP(Cluster Based Routing Protocol)

CBRP는 순방향 설정 방식과 반응적 설정 방식을 혼합한 형태로, 클러스터링을 통하여 클러스터 헤드와 멤버를 갖는 클러스터를 구성한다. 클러스터 구성 방법은 지역적으로 인접한 호스트들이 자신의 ID와 이웃 호스트의 ID를 비교하여 가장 낮은 ID를 가지고 있는 호스트를 클러스터 헤드로 선출하게 되고 클러스터 헤드가 아닌 호스트들은 클러스터의 멤버가 된다. 클러스터가 형성되면 클러스터 헤드는 멤버 호스트와 이웃 클러스터 헤드로의 라우팅 정보를 유지하게 되는데 이 정보는 후에 멤버들의 라우팅 정보 획득을 위하여 사용된다. 목적지에 대하여 라우팅 정보가 필요한 호스트는 자신의 클러스터 헤드에게 목적지에 대한 라우팅 정보 획득을 요청한다. 요청을 받은 클러스터 헤드는 자신의 멤버 호스트 중에 목적지 호스트가 있는지 조사하고, 없는 경우 자신의 이웃 클러스터 헤드에게 경로 요청 패킷을 전송하게 된다. 요청을 받은 이웃 클러스터 헤드는 앞의 클러스터 헤드와 같이 자신의 멤버 호스트 중에 목적지 호스트가 있는지 조사하고 만약 목적지 호스트가 존재하면, 소스 라우트 정보를 생성하여 경로 설정을 요구했던 송신 호스트로 소스 라우트 정보를 제공한다.

3. 제안기법

3.1 개요

본 논문에서 제안하는 MHCR은 기존 프로토콜들의 장단점을 절충하기 위하여 혼합 방식을 사용한다. 혼합 방식으로써 CBRP와 같은 클러스터에 기반을 두고 있는데, 클러스터는 무선 ad-hoc 네트워크 내에 계층적인 구조를 형성시켜 효율적인 네트워크 관리와 경로 정보 관리를 가능하도록 한다. 또한 이동 호스트들이 유지하는 경로 정보를 줄일 수 있고, 네트워크 변화에 따른 데이터 전송 지연과 오버헤드를 줄이며, 지역 토폴로지 변화에 따른 전체 토폴

로지 변화를 막을 수 있다.

무선 ad-hoc 네트워크 환경을 위해 제안된 기존 라우팅 프로토콜들은 각기 고유의 특성을 가지게 되는데, 이러한 특성들은 대부분 네트워크 경로 정보를 획득하기 위한 지연 시간과 네트워크 오버헤드간의 trade-off에서 발생한다.

본 논문에서 제안하는 MHCR 프로토콜이 적용된 네트워크 환경의 예를 그림 2에서 보인다. 그림 2에서는 클러스터의 크기를 2홉으로 가정하고 있고 앞으로 설명될 모든 예제에 적용되고 있으나, MHCR 프로토콜은 클러스터의 크기를 동적으로 변경 가능하도록 한다.

그림 2에서 MHCR 프로토콜은 무선 ad-hoc 네트워크 내의 각 이동 호스트들을 지역(2홉 거리)에 기반하여 헤드와 멤버를 갖는 클러스터로 구성한다. 클러스터가 형성되면 각 클러스터 헤드는 멤버들에 대한 정보와 이웃 클러스터들에 대한 정보를 유지하게 된다.

기존의 CBRP(Cluster Based Routing Protocol)[6]에서는 클러스터를 형성하기 위하여 고정된 1홉의 거리를 사용하고 있다. 이러한 고정된 크기의 클러스터는 무선 ad-hoc 네트워크가 확장되는 경우, 특히 동적으로 변화하는 무선 ad-hoc 네트워크에 잘 적용하기가 어렵다. 또한 1홉의 작은 클러스터는 상대적으로 많은 클러스터 헤드들을 생성시켜, 클러스터 헤드간의 경쟁 상황으로 인하여 추가적인 오버헤드를 발생시키게 된다. 이 외에도 클러스터의 구성이 최소 ID 값을 통해서 이루어지고 있는 것, 모든 경로 요구가 클러스터 헤드로 집중되는 것, 소스 라우팅을 위한 패킷 크기 확장 등 고려되어야 할 많은 문제점들이 있다.

따라서 본 논문에서 제안하는 MHCR 프로토콜은 고정된 크기의 클러스터를 구성하는 대신 크기가 동적으로 변경될 수 있는 멀티 홉 클러스터 기반의 라우팅 프로토콜로, 네트워크의 변화에 적응성 있게 대처할 수 있는 프로토콜이다. 또한 각 클러스터간의 연결을 위해 브릿지 호스트를 선출하여 클러스터 헤드간의 경로 정보와 최소 경로를 제공해 줌으로써 네트워크 내에 불필요한 패킷의 발생을 줄이고, 클러스터 헤드간의 최소 거리를 항상 제공해 줄 수 있도록 하였다. 또한 각 호스트를 기준으로 클러스터 크기만큼의 지역 경로 정보를 획득할 수 있게 하여 지역에 인접한 호스트에 대해서는 클러스터 헤드의 도움 없이 데이터를 전송할 수 있게 하였다. 대부분 기존 프로토콜에서 제안하고 있지 않은 시간의 경과에 따른 경로 변화와 실패에 대해서는 지역적으로 경로를 복구하는 방법과 RRHT(Route Request Head Table)를 이용하는 방법을 사용하여 불필요한 경로 재발견 과정을 줄였고, 클러스터의 크기가 커짐으로써 발생할 수 있는 경로 최적화의 문제

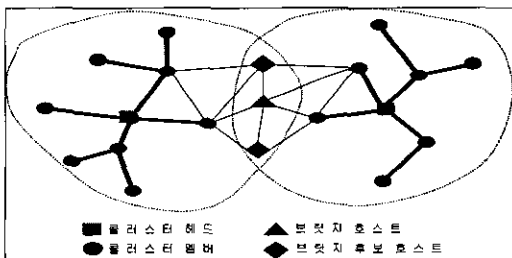


그림 2 무선 ad-hoc 네트워크 환경의 예

는 각 호스트가 유지하는 RT(Region Table)와 CFP(Cluster Formation Packet)를 통해서 해결하였다.

3.2 패킷 형식 및 자료 구조

각 이동 호스트는 클러스터를 형성하기 위하여 그림 3의 CFP를 사용한다. 이 CFP의 역할은 첫째, 무선 ad-hoc 네트워크 내에 클러스터를 형성하고 둘째, 각 이동 호스트가 클러스터 크기만큼의 지역(2홉 거리) 경로 정보를 얻을 수 있게 해준다. 클러스터가 구성되면 클러스터와 클러스터를 연결해주는 브릿지(bridge) 호스트를 선출해야 하는데 CFP의 확장 필드인 BE(Bridge Extension)가 이 역할을 수행한다. 브릿지 호스트의 선출은 클러스터 헤드간의 데이터 전송 통로 역할을 수행하여 네트워크에서 발생하는 오버헤드를 줄이고 이웃 클러스터로 향하는 최소 경로를 유지하는데 있어서 중요하다. 경로 설정 도중에 수행되는 경로 최적화를 위해서는 OE(Optimization Extension)가 사용된다. 그림 3에서 이들 패킷 형식을 보여주고 있으며 CFP의 Extension 부분에 경우에 따라 BE와 OE가 확장된다.

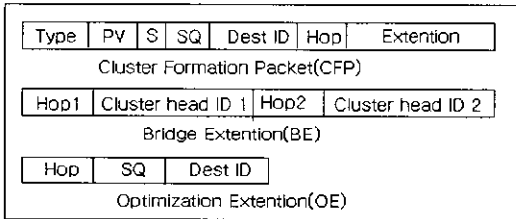


그림 3 CFP와 CFP의 확장 BE, OE의 형식

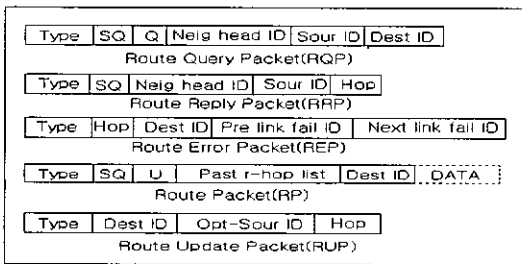


그림 4 경로 설정 패킷과 데이터 패킷의 형식

클러스터의 구성이 완료되면 목적지 경로를 알고자하는 이동 호스트는 경로 설정 작업을 수행해야 한다. 그림 4에서 경로 설정 패킷과 실제 데이터가 전송되는 데이터 패킷의 형식을 보여 준다. RQP(Route Query Packet)와 RRP(Route Reply Packet)는 경로 설정을 위해서 사용되는 패킷으로, RQP는 송신 호스트가 경로를 요청하기 위해서 사용하고, RRP는 중간 호스트나 목적지 호스트에 의해서 복

적지 호스트에 대한 경로 정보를 알려주기 위해 사용된다. REP(Route Error Packet)는 경로가 실패했을 경우, 경로 실패를 발견한 호스트에 의해서 경로 실패를 송신 호스트에게 알리기 위해서 사용되며, RP(Route Packet)는 실제 데이터가 전송되는 패킷의 형식으로 지역적인 경로 변화를 감지하고 복구한다. RUP(Route Update Packet)는 경로 최적화 과정이나 경로 복구가 수행된 후 변경된 경로 정보를 알리기 위해서 사용된다.

그림 5에서는 MHCR 프로토콜을 적용하기 위해서 ad-hoc 네트워크 내에 존재하는 모든 이동 호스트들이 유지해야 하는 테이블들을 보인다.

RT(Region Table)은 CFP 패킷을 통해서 획득한 각 호스트에 대한 경로 정보(2홉 이하)를 유지하는 테이블로서 Dest ID는 목적지 ID, S는 목적지 호스트의 상대, Next ID는 목적지 호스트로 가기 위한 이웃 호스트 ID, Hop count는 목적지 호스트까지의 거리를 나타낸다. NHRT(Next Hop Routing Table)는 RT에 없는 경로 정보(3홉 이상)를 유지하기 위한 테이블로 경로 정보 획득을 위한 경로 설정 과정을 필요로 한다. 이 테이블에서의 필드는 위의 RT 테이블의 필드 값과 같다. OT(Optimization Table)는 경로 설정 과정 중에서 응답 패킷(RRP)이 송신 호스트로 전송되는 도중 경로 최적화를 위해서 사용되는 테이블로서 CFP의 OE 확장 필드 정보를 저장한다. 여기서 Opt ID는 OE를 발생시킨 목적지 호스트의 ID, Opt hop은 Opt ID까지의 거리, Dest ID는 RRP의 Dest ID, Dest hop은 RRP를 생성시킨 호스트에서 Opt ID까지의 거리를 나타낸다. RPT(Received Packet Table)는 중복 수신된 패킷을 제거하기 위해서 사용되는 테이블이다.

각 클러스터 헤드는 그림 5의 테이블 외에 그림 6의 테이블들을 추가로 유지해야 한다. NCHT(Neighbor Cluster Head Table)는 각 클러스터 헤드들이 이웃 클러스터 헤드 정보를 유지하기 위해서 사용하는 테이블로, 선출된 브릿지 호스트가 방송하는 CFP의 BE 필드

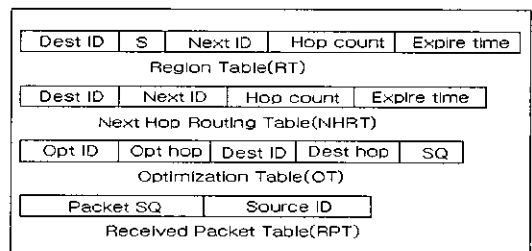


그림 5 각 이동 호스트가 유지하는 테이블들

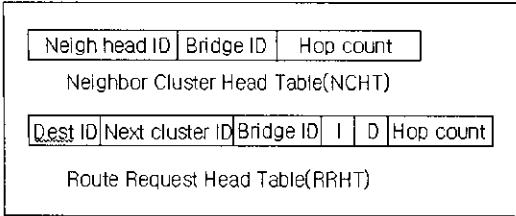


그림 6 클러스터 헤드가 추가로 유지하는 테이블

를 통해서 얻어진다. 이 테이블의 Neigh head ID는 이웃 클러스터 헤드 ID, Bridge ID는 이웃 클러스터 헤드로 가기 위한 Bridge 호스트 ID, Hop count는 이웃 클러스터 헤드로의 거리를 나타낸다.

RRHT(Route Request Head Table)는 경로 설정 과정에서 RQP와 RRP 패킷을 수신한 클러스터 헤드들에 의해서 기록되는 테이블로서 RRP를 송신 호스트로 전송하기 위해 RQP의 경로를 저장하거나 경로 실패가 발생한 경우 경로 복구를 위하여 RRP가 전송된 경로의 역 경로로 RQP를 전송하기 위해서 사용되는 테이블이다. RRHT의 Dest ID는 RQP나 RRP의 목적지 주소, Next cluster ID는 목적지로의 다음 클러스터 헤드 ID, Bridge ID는 Next cluster ID로 가기 위한 브릿지 호스트 ID, I는 RQP인 경우 1, RRP인 경우 2로 설정되며, D는 목적지 호스트를 발견한 클러스터 헤드에게만 1로 설정되고 나머지는 0이 된다.

3.3 MHCR 프로토콜의 동작과정

3.3.1 클러스터의 형성

처음 무선 ad-hoc 네트워크가 형성될 때 각 호스트는 자신의 이동성을 바탕으로 우선순위를 결정하고, 클러스터를 구성하기 위하여 CFP를 이웃 호스트들에게 방송한다. CFP를 전송한 호스트는 클러스터 형성 시간(cluster formation time) 동안 이웃 호스트로부터 전송되는 CFP를 수신하게 된다. 그림 7은 클러스터의 형성 과정을 나타낸다.

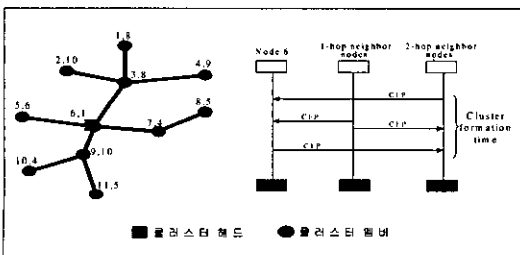


그림 7 클러스터의 형성 과정

그림 7의 왼쪽 그림에서 X,Y의 값은 X가 호스트 ID, Y가 호스트의 우선순위 값(낮은 값이 우선)을 나타내고 있다. 클러스터 형성 시간이 끝나면 각 호스트는 클러스터의 크기(2홉)의 거리에 있는 호스트 중에서 자신보다 우선순위가 높은 호스트가 있는지를 조사하여 자신을 헤드 또는 멤버로 결정하게 된다. 6번 호스트가 주위 2홉 거리의 호스트들 보다 높은 우선 순위를 가졌으므로 헤드로 결정되고 나머지 호스트는 멤버로 선택된다. 클러스터 헤드로 선출된 이동 호스트는 CFP 패킷의 S 필드의 값을 1로 설정하고 맴버인 경우는 2로 설정하여 CFP 패킷을 주기적으로 방송하게 된다. 여기서 사용되는 CFP의 PV 필드의 값은 각 호스트의 우선순위이며, S 필드는 각 호스트의 상태를 나타내는데 클러스터를 형성하지 않은 초기 호스트들의 상태는 0, 클러스터 헤드인 경우는 1, 클러스터 맴버인 경우는 2, 브릿지 호스트인 경우는 3의 값을 갖게 된다. SQ는 패킷의 일련번호, Dest ID는 CFP 패킷을 생성한 호스트의 ID가 된다. 이 값들은 중복 수신된 패킷의 처리를 위해서 사용된다.

CFP의 Hop 필드는 두 가지 역할을 수행하는데, 형성되는 클러스터의 크기를 결정해 주며, 각 호스트가 Hop 거리만큼의 지역적인 경로 정보를 얻을 수 있게 해준다. 이 Hop의 값은 각 호스트를 거치면서 1씩 감소하게 되고 0보다 작은 패킷은 제거된다. 현재 이 Hop의 값은 2로 설정되어 있고 이 값은 각 호스트의 이동성을 기반으로 변경되어질 수 있는데, 이 값에 의해서 동적으로 클러스터의 크기를 변경한다.

3.3.2 브릿지 호스트의 선출

헤드와 맴버를 갖는 클러스터가 형성되면 클러스터 헤드간 패킷 전송을 담당할 브릿지 호스트를 선출해야 한다. 브릿지 호스트의 기능은 클러스터 헤드간의 패킷 전송을 전담하여 네트워크 내에 불필요한 패킷의 발생을 억제하고 두 클러스터 헤드간의 최소 경로를 항상 제공하는 것이다. 브릿지 호스트가 되기 위해서는 두 개 이상의 클러스터 헤드로부터 유효한 CFP 패킷을 수신할 수 있어야 하는데, 그림 8에서 c1, c2, c3가 브릿지 후보 호스트가 된다. 브릿지 후보 호스트들은 BE를 생성하여 CFP 패킷과 함께 이웃 호스트들에게 전송하고 브릿지 호스트가 되기 위해서 서로 경쟁하게 된다.

각 브릿지 후보 호스트는 이웃 후보 호스트로부터 수신된 BE필드의 정보와 자신이 속한 클러스터 헤드 쌍의 거리를 비교하게 된다. 거리가 짧은 호스트가 브릿지 호스트가 되며, 거리가 같은 경우 CFP의 우선순위를 비교하여 자신을 브릿지 호스트로 선출한 것인지를 결정한다. 브릿지 호스트는 CFP의 S 필드 값을 3으로 설정하여 BE 확

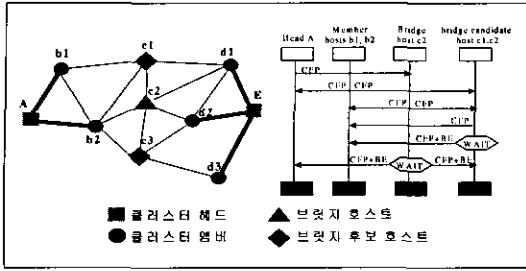


그림 8 브릿지 호스트의 선출 과정

장 필드와 함께 CFP를 주기적으로 이웃 호스트에게 전송한다. 브릿지 호스트는 항상 브릿지 후보 호스트들과 경쟁하며, 자신보다 클러스터 헤드간의 홉 거리가 짧거나 우선 순위가 높은 후보 호스트가 나타나면 그 호스트를 브릿지 호스트로 선출한다. BE에 있는 Cluster head ID와 Hop은 브릿지 호스트가 속해있는 클러스터 헤드의 ID와 거리로써 그림 c2 브릿지 호스트는 BE 필드의 값으로 (Head A, 2홉), (Head E, 2홉)의 정보를 갖게 된다.

3.3.3 경로 발견

브릿지 호스트를 포함한 클러스터 형성 작업이 이루어지면 각 이동 호스트는 무선 ad-hoc 네트워크 내에 경로 정보를 획득하기 위한 경로 설정 작업을 수행한다.

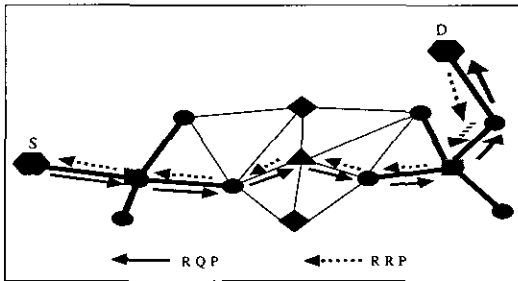


그림 9 경로 설정 과정

그림 9는 경로 설정 과정을 보여준다. 목적지(member D)에 대한 경로 정보가 필요한 송신 호스트(member S)는 자신의 영역 안(RT)에 자신이 원하는 목적지가 있는지를 확인한다. 만약 원하는 목적지 정보가 없으면 RQP를 생성하여 자신의 클러스터 헤드에게 전송하게 된다. RQP를 전송 받은 클러스터 헤드는 자신의 멤버 호스트 중에 목적지 호스트가 있는지 확인하고 없는 경우, 이웃 클러스터 헤드로 RQP를 전송하기 위해 자신의 브릿지 호스트로 RQP를 전송한다. RQP의 정보는 클러스터 헤드에 의해서 RRHT에 저장되게 되는데, 이 정보는

목적지 호스트가 송신 호스트로 응답 패킷을 보내는 경우 RQP가 거처온 클러스터 헤드들의 역 경로를 나타내게 된다. 따라서 RRHT에 저장된 정보를 통하여 RRP가 송신 호스트로 전송될 수가 있다. RQP를 수신한 클러스터 헤드는 RRHT의 I 필드에 1을 설정한다. 이때 사용되는 RQP의 Neig head ID는 이웃클러스터의 ID, Source ID는 RQP를 생성한 호스트의 주소, Dest ID는 RQP의 목적지 주소를 나타낸다.

RQP를 수신한 목적지 호스트나 목적지에 대한 경로 정보를 가지고 있는 중간 호스트들은 RRP를 작성하여 송신 호스트로 전송하게 된다. RRP는 위에서 설명한 대로 RRHT에 의해서 송신 호스트로의 역 경로를 알 수 있다. RRP를 수신한 클러스터 헤드는 RRHT의 I 필드를 2로 설정하고 목적지 호스트를 발견한 클러스터 헤드는 RRHT의 D 필드에 1을 설정하여 송신 노드로 RRP를 전송한다. RRP의 Neig head ID는 RRHT에 의해서 알아낸 RQP를 송신한 이웃 클러스터 헤드의 주소로, RRP가 전송되어야할 역 경로이다. Q는 일반적으로 0으로 설정되고 1인 경우는 경로 재 설정을 위해서 사용되는데 이 경우는 위에서 설명한다. Sour ID는 송신 호스트의 주소, hop은 목적지 호스트에서 송신 호스트로의 거리로써 각 호스트에 의해 1씩 증가된다.

3.3.4 경로 최적화

RQP가 송신 호스트로 전송되는 동안 RRP를 수신한 호스트들은 경로 최적화 작업을 수행하는데, 경로 발견으로 설정된 경로는 최적화된 경로가 아니기 때문이다. 아래 그림 10은MHCR의 경로 최적화 과정을 보여준다.

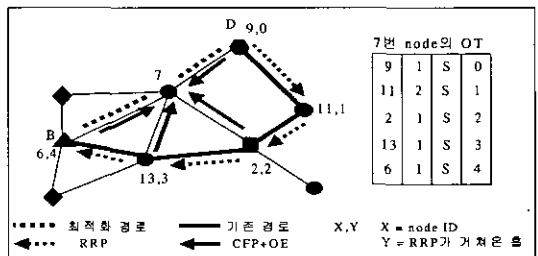


그림 10 경로 최적화 과정

그림 10은 9번 목적지 호스트에서 6번 브릿지 호스트로의 최적화 과정을 보여준다. 위의 경우 9번 호스트에서 6번 호스트로 가는 경로는 9, 11, 2, 13, 6거처는 경로 외에 9, 7, 6을 거치는 최단 경로가 존재하지만 클러스터 기반의 경로 설정 방식으로 인해 9, 11, 2, 13, 6을 거치는 경로가 선택된다. 이 경로는 최적화 된 경로가 아니기 때문에 경

로에 대한 최적화 작업이 이루어져야 한다. 경로 최적화를 위해서 사용되는 것이 CFP의 OE 필드이다. RRP를 전송 받은 각 호스트들은 OE를 생성하여 RRP의 Hop, SQ, Dest ID의 값을 OE에 복사하고 CFP와 함께 OE를 전송한다. OE를 전송받은 호스트들은 OT(optimization table)에 수신된 OE 정보를 기록하여 최적화 과정을 수행한다. 그림 10에서 7번 호스트의 OT 모습을 보여주고 있는데, 7번 호스트는 최적화 과정이 필요한 RRP의 목적지 S에 대해서 9번 호스트와 6번 호스트가 자신으로부터 1홉씩 떨어져 있다는 것을 알 수 있다. 또한 6번 호스트에서 9번 호스트까지의 RRP의 거리가 4홉(4-0)이라는 것을 알 수 있다. 따라서 7번 노드는 RRP의 원래 경로 4홉 보다 자신을 통하여 가는 2홉의 거리가 더 짧다는 것을 알고, 9번 호스트에게 NHRT 변경을 요청하고 6번 노드에게는 RUP를 전송하여 경로 최적화를 마무리한다. 이 RUP 패킷의 Dest ID는 목적지 호스트의 주소, Opt-Source ID는 RUP 패킷을 생성한 호스트로 여기서는 7번 호스트가 된다. Hop은 새로 적용될 목적지까지의 거리로 7번 호스트에서 목적지 호스트까지의 거리가 된다. RUP 패킷을 전송 받은 6번 호스트는 자신의 NHRT를 수정하고 RUP를 송신지 호스트로 전송하게 된다. RUP는 각 호스트를 거치면서 NHRT의 Dest ID에 대한 홉 정보를 수정한다.

3.3.5 경로 유지

이동 호스트의 특성상 설정된 경로는 계속해서 변하게 된다. 경로가 더욱 짧아지기도 하고 길어지기도 하며 경로가 실패하기도 한다. 다음에서 이러한 경우 MHCR에서 사용하는 경로 유지 방법들을 설명한다.

MHCR에서 사용되는 데이터 패킷은 RP이며 패킷의 형식은 다음과 같다. U는 지역적인 경로 복구가 이루어졌을 때 1로 설정되고 그 외의 경우는 0이 된다. Past r-hop list는 RP 패킷이 지나온 경로 중 부분 호스트의 리스트로 현재 호스트로부터 클러스터 크기의 홉 수만큼 지나온 리스트의 목록이 된다. 위의 그림 10에서 클러스터의 크기가 2홉이고 데이터가 9, 11, 2, 13, 6의 경로를 따라서 전송된다고 가정하면 6번 호스트는 2와 13, 13번 호스트는 11과 2를 Past r-hop list로 가지게 된다. Dest ID는 목적지 호스트의 주소가 되고 DATA는 실제 전송되는 데이터 정보이다.

데이터가 전송되는 도중 이동 호스트의 이동으로 인해 각 경로는 더욱 짧아 질 수 있다. Past r-hop list는 지역 정보(RT)와 비교되어 데이터가 전송되는 도중의 경로 최적화 작업을 수행하게 된다. 아래의 그림 11은 지역적인 경로 최적화를 보여준다.

목적지 9번 호스트는 RP의 Past r-hop list를 통해서

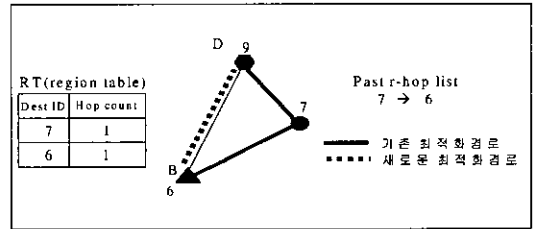


그림 11 지역적인 경로 최적화

RP가 6번 호스트와 7번 호스트를 거쳐왔다는 것을 안다. 여기서 RT는 6번 노드가 1홉의 거리에 있다는 것을 알려 주므로 9번 호스트는 6번 호스트에게 RUP를 전송해 9번으로 직접 데이터를 전송하도록 설정하게 된다.

이동 호스트의 이동은 위에서 설명한 것 같이 경로에 대한 변화와 함께 경로 실패를 발생시키게 된다. MHCR에서는 새로운 경로 발견 절차를 수행하기 전에 경로 복구를 위해서 두 단계의 사전 경로 복구 과정을 수행하는데 그림 12는 이 과정을 보여준다.

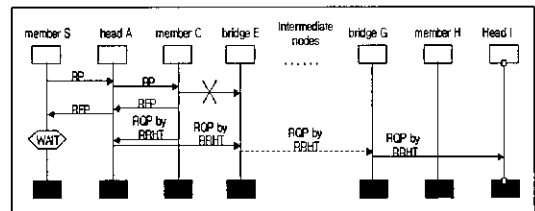


그림 12 경로 복구 과정

먼저, 경로 실패를 발견한 호스트는 경로 실패를 발생시킨 next hop 호스트를 자신의 RT에서 찾는다. 다음의 그림 13은 이 과정을 설명한다. 그림에서 브릿지 호스트 6은 7번 호스트로 데이터를 전송하려 하지만 7번 호스트로의 경로가 실패되어 데이터를 전송할 수 없다. 이때 6번 호스트는 RT 테이블 정보를 통해서 7번 호스트를 찾게 되는데 7번 호스트가 RT에 기록되어 있으면 7번 호스트에게

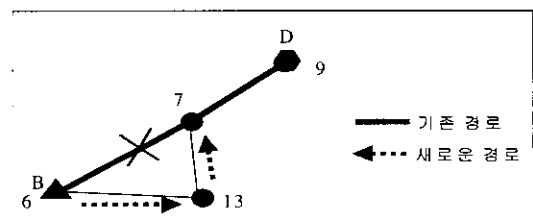


그림 13 지역적인 경로 복구

데이터를 전송할 수가 있다.

RT에서 next hop 호스트를 발견하면, RP의 U 필드를 1로 설정하여 next hop 호스트에게 전송한다. U 필드의 값은 목적지 호스트가 패킷을 전송 받은 경우 새로이 RRP를 송신 호스트로 전송하여 경로 최적화 작업을 수행하게 한다.

next hop 호스트를 찾지 못한 경우 경로 실패를 발견한 호스트는 송신 호스트에게 REP를 전송한다. REP를 전송 받은 송신 호스트는 경로 복구 시간(route restore time) 동안 새로운 경로 발견 절차를 수행하지 않는다.

여기서 REP의 Hop은 경로 실패가 발생한 호스트와 송신 호스트까지의 거리, Dest ID는 목적지 주소, Pre link fail ID와 Next link fail ID는 경로가 실패된 링크의 호스트를 나타낸다. 또한 경로 실패를 발견한 호스트는 자신의 클러스터 헤드에게 자신을 송신지로 하는 RQP 패킷을 전송한다. 이때 RQP 패킷의 Q 필드의 값을 1로 설정하게 되는데 이 RQP 패킷은 초기 목적지 호스트를 검색하기 위해 모든 이웃 클러스터 헤드에게 전송되었던 RQP 패킷과는 다르게 특정 클러스터 헤드들에게만 전송되게 된다. 여기서의 특정 클러스터 헤드는 RQP 패킷의 목적지 호스트에 대해서 처음 경로 설정 과정에서 RRP를 전송했던 클러스터 헤드들이 된다. RRP를 전송했던 클러스터 헤드는 각 클러스터 헤드들이 유지하게 되는 RRHT에 기록되게 된다. 따라서 새로 생성된 RQP는 RRP가 전송되었던 역 경로 즉, 목적지 호스트의 방향으로 전송되게 되고 결국에는 목적지 호스트가 있었던 클러스터 헤드로 전송되게 되어 경로 복구를 수행하게 된다.

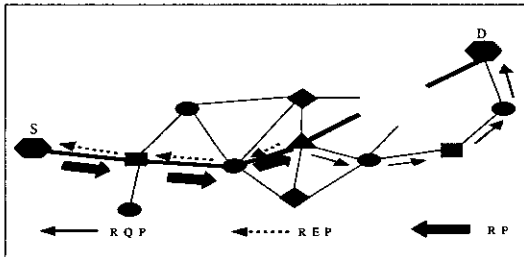


그림 14 경로 실패로 인한 경로 재 복구

이러한 경로 복구 방식은 이동 호스트들의 이동이 지역에 기반하여 이동하기 때문에 새로운 경로 설정 보다 적은 네트워크 오버헤드를 발생시키게 된다. 위에서의 경로 복구가 실패하면 송신 호스트는 경로 설정 과정을 재 수행하게 된다.

3.4 MHCR 알고리즘

MHCR 프로토콜에서 클러스터를 형성하기 위해 사용

되는 우선순위 값은 각 이동 호스트의 CMR(Call to Mobility Ratio)을 통한 이동성을 바탕으로 결정하게 된다. 알고리즘 1은 CFP를 처리하기 위한 각 이동 호스트의 동작과정으로 무선 ad-hoc 네트워크 내에 클러스터를 형성하고 지역 정보를 얻을 수 있게 한다.

형성된 클러스터의 변화는 이동 호스트에게 추가적인 오버헤드를 발생시키는 요인이 되므로 변화되는 클러스터 수를 최소화하기 위한 방법이 제시되어야 한다. 아래의 알고리즘 2는 클러스터 유지를 위하여 각 호스트가 수행하는 알고리즘이다.

```

node A broadcasts CFP;
while (cluster formation time) {
  if (a node A receives other host's CFP) {
    if (packet's hop < 0 || previous received packet)
      discard CFP;
    else {
      if (CFP's Dest ID exists in RT) {
        if (packet's hop < hop in RT) {
          update hop for Dest ID;
          rebroadcast CFP;
        }
      }
      else
        rebroadcast CFP;
    }
  }
  else
    record CFP's route information in RT;
  if (node A priority > CFP's PV)
    node A becomes a cluster head;
  else
    node A becomes a cluster member;
}
}
if (node A doesn't receive other host's CFP)
  node A becomes a cluster head;
    
```

알고리즘 1 클러스터의 형성과 지역 정보 획득 알고리즘

```

if (a cluster head A finds a new host in his region by CFP) {
  if (a new host is a cluster head) {
    compare its PV to the new cluster head's PV;
    if (its own PV > the new cluster head's PV) {
      continue to act a cluster head;
      record CFP's route information in RT;
    }
  }
  else {
    give up a cluster head;
    record CFP's route information in RT;
    trigger cluster formation procedure;
  }
}
else
  record CFP's route information in RT;
if (a cluster head loses its member by expire time in RT)
  clear the entry of the member;
if (a cluster member loses all its own cluster heads)
  become a cluster head;
if (cluster member loses a host in his region by expire time in RT)
  clear the entry of the host;
    
```

알고리즘 2 클러스터 유지 알고리즘

클러스터가 형성되고 나면 클러스터와 클러스터를 연결해 주어야 할 브릿지 호스트를 선출해야 한다. 알고리즘 3은 브릿지 호스트의 선출과정이다.

```

candidate bridge host A broadcasts CFP with BE;
if (candidate bridge host A receives CFP with BE for the same
cluster heads pair){
if (its own hop < other candidate bridge host hop for the
same cluster heads) {
set CFP's S field to 3;
continue to broadcast CFP with BE;
}
else if (its own hop == other candidate bridge host hop
for the same cluster heads) {
if (its own PV > other candidate bridge host's PV) {
set CFP's S field to 3;
continue to broadcast CFP with BE;
else
continue to broadcast CFP without BE;
}
else
continue to broadcast CFP without BE;
}
else
set CFP's S field to 3;

```

알고리즘 3 브릿지 호스트 선출 알고리즘

알고리즘 4는 브릿지 호스트가 방송하는 BE 필드에 대하여 클러스터 헤드가 동작하는 알고리즘이다. BE 필드는 각 클러스터 헤드에게 이웃 클러스터 헤드의 정보를 제공하게 된다.

```

if (a cluster head receives CFP with BE) {
if (packet's S field equals 3) {
if (neighbor cluster ID in BE exists in NCHT) {
if (CFP packet's dest ID equals bridge ID in NCHT) {
if (BE's hop < hop in NCHT)
update hop in NCHT to BE's hop;
else
none;
}
else
update hop and bridge host ID for neighbor cluster ID;
}
else
record BE's information in NCHT;
}
else
none;
}
else
none;

```

알고리즘 4 BE가 추가된 CFP 패킷을 받은 클러스터 헤드의 동작 알고리즘

목적지 호스트에 대한 경로 정보를 획득하고 싶은 호

스트는 RQP를 작성하여 경로 설정 절차를 수행한다. 알고리즘 5는 RQP를 수신한 각 호스트들의 동작 과정을 보인다.

```

a source node checks whether the destination is within its
region by RT;
if (the path to the destination exists in RT)
unicast RP;
else
send RQP to its cluster heads;
if (a intermediate host receives RQP) {
check whether the destination is within NHRT;
if (the destination exists in NHRT)
send RRP to the source node;
else
send RQP to the next host or the bridge host;
}
if (a cluster head receives RQP) {
check whether the destination is within its region by RT;
if (the path to the destination exists in RT) {
record RQP's information in RRHT;
unicast RQP to destination;
}
else {
record RQP's information in RRHT;
send RQP to its neighbor cluster head;
}
}
if (destination node receives RQP)
send RRP to the source node;

```

알고리즘 5 각 호스트에서 RQP 패킷의 처리 알고리즘

RQP를 수신한 목적지 호스트나 중간 호스트는 자신의 경로 정보를 송신 노드에게 알리기 위해 RRP를 생성하여 전송한다. 알고리즘 6은 RRP를 수신한 호스트의 동작 과정이다.

```

a destination host send RRP to neighbor cluster head
that RQP is received from;
if (a intermediate node receives RRP) {
record RRP's information in NHRT;
increment hop in RRP;
send CFP with OE;
send RRP to the next hop host for reaching
neighbor cluster head;
}
if (a cluster head receives RRP packet) {
record RRP's information in NHRT;
record RRP's information in RRHT;
increment hop in RRP;
if (source node exists in its region)
send RRP to the source node;
else
send RRP to its neighbor cluster head;
}
if (a source node receives RRP)
send RP to the destination;

```

알고리즘 6 각 호스트에서의 RRP 패킷의 처리 알고리즘

알고리즘 7은 경로 실패가 발생한 경우 경로 실패를 복구하기 위해 사용되는 RQP의 생성과 RQP에 대한 처리 과정을 보여주는 알고리즘이다.

```

if (a node discovers route failure) {
  find the next hop node by RT;
  if (the next hop node exists in RT)
    send RP with U field set 1 to the next hop node;
  else {
    send REP to the source node;
    send RQP to it's cluster head;
  }
}
if (a intermediate host receives RQP) {
  check whether the destination is within NHRT;
  if (the destination exists in NHRT)
    send RRP to the the node that find route failure;
  else
    send RQP to the next host or the bridge host;
}
if (cluster head receives RQP) {
  check whether the destination is within its region by RT;
  if (the path to the destination exists in RT)
    unicast RQP to the destination;
  else if (D field for destination in RRHT == 1)
    discard RQP;
  else {
    record RQP's information in RRHT;
    send RQP to its neighbor cluster head by RRHT;
  }
}
if (destination node receives RQP)
  send RRP to the source node;
    
```

알고리즘 7 Q필드가 1로 설정된 RQP를 수신한 호스트의 처리 알고리즘

알고리즘 8은 CFP의 OE 필드를 통하여 각 호스트가 수행하는 최적화 과정이다.

```

if (a node receives CFP with OE) {
  record CFP and OE information in OT;
  if (OT's hop for the path to the same destination < RRP's hop)
    send RUP to the source node;
}
    
```

알고리즘 8 OE를 통한 각 호스트의 최적화 알고리즘

4. 성능평가

본 논문에서는 MHCR 프로토콜의 성능 평가를 위하여 기존의 클러스터 기반 라우팅 프로토콜인 CBRP와의 비교를 통해서 시스템의 성능을 평가한다. 또한 MHCR 모델의 성능 평가를 위해서 최적화 알고리즘을 사용한 경우와 최적화 알고리즘을 사용하지 않은 경우 경로 실패 회수와 데이터 전송 시간을 비교하였다. 본 논문에서는 DEVS (Discrete EVent Simulation) 이론에 기반한 시뮬레이터

DEVS Scheme을 이용하여 시뮬레이션을 수행하였다.

4.1 시뮬레이션 환경

본 논문에서 MHCR 프로토콜의 시뮬레이션을 위해, 무선 ad-hoc 네트워크의 크기는 1.5km의 정방향 구조를 가지고 있으며, 각 호스트의 패킷 전송 거리는 100m로 가정한다. 또한 1.5km의 정방향 구조에는 100개의 이동 호스트가 존재하는 것으로 가정하였다. 각 이동 호스트의 이동 속도는 $\alpha \cdot (10m/s - 50m/s)$ 로 α 는 이동 속도에 대한 가중치이다. 각 호스트의 평균 데이터 발생 주기는 10ms로 하였다. 인접 호스트간의 데이터 전송은 전파 지연을 갖게 되는데 시스템 자원이나 전송 장치 대역폭과 같은 제한에서 발생한다. 다음은 전파 지연에 따른 노드간의 데이터 전송시간을 계산한 값이다.

$$DT = T_{data} + D_{delay}$$

$$D_{delay} = \sum_{k=1}^n ((T_s \cdot P_s + T_r \cdot P_r) P_k^r + P_{su})$$

DT=데이터 전송시간
 Tdata=데이터 전송시의 총 지연 시간
 Ddelay=각 호스트의 시스템 자원이나 대역폭으로 인한 총 지연 시간
 Ts=시스템 자원으로 인한 전송지연 시간
 Ps=시스템 자원의 전송 지연 확률
 Tr=전송장치로 인한 전송지연 시간
 Pr=전송장치의 전송지연 확률
 Pk=1-n번째 까지 전송 지연을 가질 확률
 Psu=n+1부터 전송 지연을 가지지 않을 확률

클러스터를 구성하기 위해 CBRP에서 사용되는 HELLO 패킷과 MHCR의 CFP 패킷의 전송 주기는 200ms초로 가정하였으며, 경로 설정에 드는 시간 지연은 데이터 패킷의 전송 지연 시간과 같고 각 송신 노드에서 목적 노드까지의 클러스터 헤드의 개수를 평균하여 비용을 산출하였다. 경로 실패로 인한 경로 복구에서 지역적인 경로 복구는 경로 복구 호스트와 경로 복구 목적지 호스트의 거리를 고려하여 다음과 같이 정의한다.

$$RDT = DT \cdot RH + NT$$

RDT=검색된 next-hop 노드로 데이터 전송시간
 RH=검색된 next-hop 노드의 홉 거리
 NT=next-hop 노드로 데이터를 전송하기 위한 추가 지연 시간

next-hop 노드를 찾지 못하고 RRHT에 의한 경로 복구를 필요로 할 때 경로 복구 시간은 다음과 같다.

$$RRDT = 2 \cdot (DT \cdot RH + 2 \cdot NT + \sum_{k=1}^n HSk)$$

RRDT=RRHT에 의한 목적지 호스트로의 경로 검색 시간
 RH=검색된 목적지 노드로의 홉 거리
 NT=목적지 노드를 검색하기 위한 추가 지연 시간
 HS=클러스터 헤드들의 목적지 검색을 위한 추가 지연 시간

경로 실패로 인하여 경로 복구가 불가능한 경우 송신 노드의 경로 재 설정 이전에 드는 지연 시간은 경로 복구 시간(route restore time)과 경로 실패 발견 호스트까지의 데이터 전송 시간 합으로 계산하였고, 여기서 경로 복구 시간(route restore time)의 값은 무선 ad-hoc 네트워크

의 최대 지름 홉 수*전파지연의 시간+ γ 의 값으로 계산 하였다.

4.2 시뮬레이션 결과

본 논문에서는 먼저 CBRP와 MHCR을 비교하기 위해 각 이동 호스트들의 유지 정보를 비교한다. 이동 호스트의 이동성이 큰 경우 MHCR이 클러스터 크기를 1로 설정 하면, RT에 유지되는 호스트 정보와 경로 설정을 통해서 얻어진 호스트 정보는 CBRP와 같게 된다. MHCR이 토폴로지 변화가 거의 없는 네트워크에 대해서 네트워크의 지름에 해당하는 홉 수만큼의 클러스터를 형성하면 네트워크 내의 모든 호스트 경로 정보를 RT에 유지할 수 있다. 이 때 RT의 크기는 이동호스트 N에 대해서 O(N)의 크기를 가지지만, 이 경우 경로 정보는 모두 사용 가능하므로 경로 설정에 드는 지연 시간과 오버헤드를 방지 할 수 있다. 위의 예에서 2홉의 클러스터 크기는 1홉의 CBRP 보다 2배 이상의 경로 정보를 유지해야 하지만 2 홉 거리의 호스트에 대해서는 앞에서와 같이 경로 설정에 드는 비용을 줄일 수 있다. 따라서 MHCR은 경로 유지 비용과 경로 설정 비용을 비교하여 보다 유리한 경로 정보 획득 방법을 선택할 수 있게 된다.

다음은 CBRP와 MHCR 간의 이동성 변화에 따른 경로 실패 회수, 데이터 전송 시간, 네트워크 트래픽을 비교한다. 시뮬레이션을 위해 먼저 송신 호스트와 목적지 호스트를 임의로 선택하여 데이터 전송을 시작하고, 모든 통신이 끝난 경우에 새로운 호스트들을 선택하여 새로운 통신을 시작한다. 여기서 전송되어야할 데이터 패킷의 개수는 1000개로 설정하였고 데이터가 모두 전송되었을 때의 성능을 측정하여 측정된 값을 평균하였다. 테스트는 이동 호스트의 이동성 가중치의 값에 따라 30회 반복 수행하였고, 이동성은 상수 값(0.2)으로 증가시키면서 테스트하였다.

각 이동 호스트의 이동 범위는 가중치 값의 최대치와 최소치 값에서만 이동 가능하고, 이동 속도는 이동 값의 중간 값의 지수 분포를 따르며, 이동 방향은 랜덤 함수의 지수 승 값을 mod 하여 방향 값을 결정하였다.

그림 15에서 1000개의 데이터 패킷을 전송하는 경우 경로 실패의 회수를 비교하였다. 두 경우 이동 호스트의 이동성이 증가함에 따라서 경로 실패 회수가 증가하고 있지만 MHCR의 경우 경로 실패 회수가 CBRP보다 낮아지게 된다. 이것은 클러스터의 형성이 이동성에 기반하여 설정된 경로가 보다 안정적이기 때문이다.

그림 16은 경로 실패와 관련하여 CBRP와 MHCR 데이터 전송 시간을 비교한 것이다. 이동 호스트의 이동성이 증가할수록 데이터 전송 시간이 큰 차이를 나타내게 되는데, 이것은 경로 실패에 따르는 경로 복구와 유지에

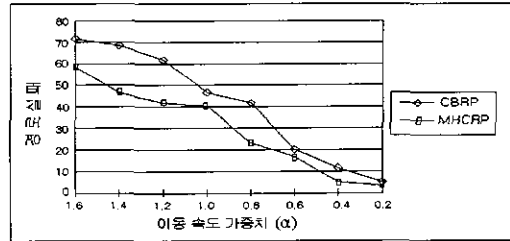


그림 15 MHCR 프로토콜과 CBRP의 경로 실패 회수

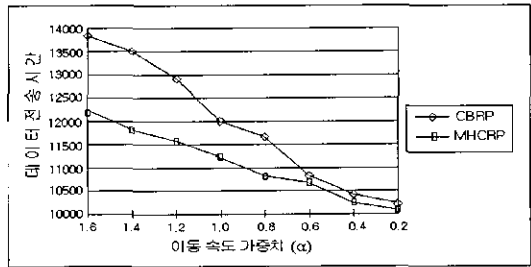


그림 16 MHCR와 CBRP의 총 데이터 전송시간

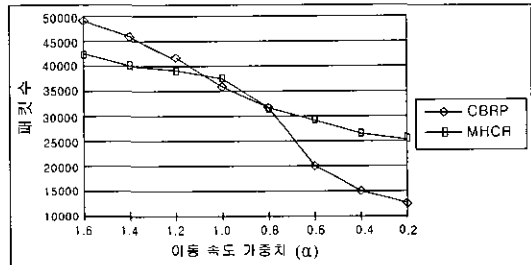


그림 17 MHCR 프로토콜과 CBRP의 네트워크 트래픽 발생 수

MHCR이 보다 효과적으로 대처하기 때문이다.

그림 17은 이동 속도 가중치에 따른 네트워크 트래픽을 측정한 그림이다. 위에서 설명한 환경에 기반하여 전체 호스트의 5%가 지수 분포의 형태로 항상 데이터를 전송한다고 가정하였으며, 패킷들은 타임에 상관없이 노드간의 링크를 통해 전송되는 경우 1단위로 측정하였다. 그림에서 이동 속도 가중치가 1.0 이하인 경우는 CBRP의 네트워크 트래픽이 MHCR보다 적는데 CBRP의 HELLO 패킷의 1홉 전파와 경로 설정에 드는 비용이 CFP의 2홉 전파와 경로 설정에 드는 비용보다 적기 때문이다. 그러나 1.0이상인 경우는 이와 반대가 되는데 상대적으로 많은 클러스터 헤드와 빈번한 경로 설정으로 인해 CBRP의 추가 네트워크 오버헤드가 발생되기 때문이다. 따라서 토폴로지 변화

가 심한 네트워크일수록 MHCR의 성능이 더욱 좋아지며, 네트워크 변화에 따라서 클러스터의 크기를 적절히 조절할 수 있다면 더 적은 네트워크 오버헤드와 지연시간으로 데이터를 전송할 수 있게 된다.

그림 18과 19는 MHCR 프로토콜에 최적화 과정이 이루어진 경우와 최적화 과정이 이루어지지 않은 경우 경로 실패와 데이터 전송 시간을 비교한 그래프이다.

최적화 과정은 경로의 단축을 통해서 데이터가 전송되는 hops를 줄일 수 있지만 설정된 경로가 안정된 경로가 아니기 때문에 최적화 과정이 수행되지 않은 경로 보다 많은 경로 실패와 지연 시간을 발생하게 된다. 그림 19의 결과에서 데이터 전송시간이 이동 속도 가중치가 0.6이하인 경우, 이동 호스트의 이동성이 낮은 경우는 최적화 과정이 이루어진 경우가 데이터 전송 시간이 향상되게 되는데 이것은 경로 재 설정에 드는 지연 시간 보다 경로 최적화를 통해서 얻어진 경로가 보다 효율적으로 데이터를 전송하는 결과를 나타내 주기 때문이다.

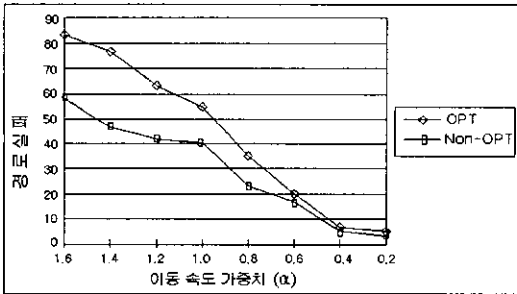


그림 18 최적화된 경우와 그렇지 않은 경우 경로 실패 회수

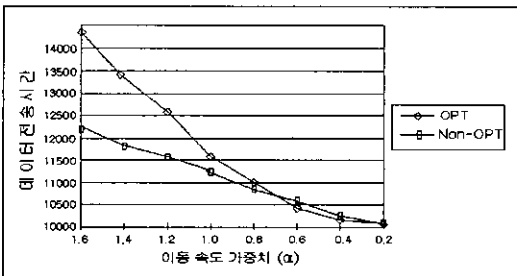


그림 19 최적화된 경우와 그렇지 않은 경우 데이터 전송시간

5. 결 론

본 논문에서 제안하는 MHCR 프로토콜은 무선 ad-hoc

네트워크를 구성하는 이동 호스트들의 이동성을 고려하여 적은 네트워크 오버헤드와 지연시간으로 최적화 된 경로 정보를 획득할 수 있는 방법을 제시한다. 기존 클러스터 기반의 무선 ad-hoc 네트워크 라우팅 프로토콜은 클러스터의 크기를 고정된 크기로 설정함으로써 네트워크 변화에 따른 오버헤드와 지연시간에 대해서 능동적으로 대처할 수 없었다. 따라서 이 논문에서는 멀티 홉을 갖는 클러스터의 크기를 동적으로 변화시켜 네트워크 변화에 잘 적용할 수 있게 하였고, 경로 복구와 경로 최적화를 통해서 경로 유지비용을 최소화 할 수 있는 방법들을 제시하였다. 또한 각 호스트를 기준으로 지역 경로 정보를 획득할 수 있게 하여 지역에 인접한 호스트에 대해서는 별도의 경로 설정 절차 없이 경로 정보를 사용할 수 있게 하였다.

향후 연구 과제로는 무선 ad-hoc 네트워크 라우팅 프로토콜을 설계하는데 있어서 영향을 미칠 수 있는 여러 요소들에 대한 연구와 그러한 요소들의 변화에 따라 클러스터 크기를 적절히 조절할 수 있는 방법에 대한 연구가 요구된다. 또한 QoS 서비스에 대비한 기능 확장과 프로토콜의 재 정의가 필요할 것으로 여겨진다.

참 고 문 헌

- [1] S. Corson, and J. Macker, "Mobile Ad hoc Networking(MANET)," Internet Draft, Oct. 1998.
- [2] Z. J. Hass, and S. Tabrizi, "On Some Challenges and Design Choices in ad-hoc Communication," MILCOM'98, 1998.
- [3] C. E. Perkins, "Mobile Ad hoc Networking Terminology," Internet Draft, Nov. 1998.
- [4] S. Corson, J. Macker, and S. Batsell, "Architectural Consideration for Mobile Mesh Networking," RFC XXXX, May 1996.
- [5] D. B. Jonson, "Routing in Ad hoc Networks of Mobile Hosts," In Proceeding of The IEEE Workshop on Mobile Computing Systems and Application, Dec. 1994.
- [6] M. Jiang, J. Li, and Y. C. Tay, "Cluster Based Routing Protocol(CBRP) Functional Specification," Internet Draft, Aug. 1999.
- [7] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance- Vector Routing (DSDV) for Mobile Computers," ACM SIGCOMM, vol. 24, no. 4, Oct. 1994.
- [8] P. Jacquet, P. Muhlethaler and A. Qayyum, "Optimized Link State Routing Protocol(OLSRP)," Internet Draft, Nov. 1998.
- [9] J. Broch, D. B. Johnson and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft, Dec. 1999.

- [10] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc On Demand Distance Vector(AODV) Routing," Internet Draft, Nov. 1999.
- [11] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm(TORA) Functional Specification," Internet Draft, Aug. 1998.
- [12] G. Aggelou, and R. Tafazolli, "Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) Protocol," Internet Draft, Sep. 1999.
- [13] J. J. Garcia-Luna-Aceves, M. Spohn, and D. Beyer, "Source Tree Adaptive Routing (STAR) Protocol," Internet Draft Oct. 1999.
- [14] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol(ZRP) for Ad Hoc Network," Internet Draft, Aug. 1999.
- [15] M. S. Corson, S. Papademetriou, and P. Papadopoulos, V. Park, A Qayyum, "An Internet MANET Encapsulation Protocol(IMEP) Specification," Internet Draft, Aug. 1999.
- [16] R. Sivakumar, P. Sinha, and V. Bharghavan, "Core Extraction Distributed ad-hoc Routing (CEDAR) Specification," Internet Draft, Nov. 1999.



전 형 국

1999년 성균관대학교 정보공학과 학사.
2001년 성균관대학교 전기전자컴퓨터 석사. 2001년 ~ 현재 한국전자통신연구원.
인터넷정보기전연구부 연구원



김 문 정

1998년 2월 성균관대학교 정보공학과 졸업(학사). 1998년 3월 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부 석사과정. 관심분야는 이동 컴퓨팅, 분산 시스템



엄 영 익

1983년 서울대학교 계산통계학과 졸업(학사). 1985년 서울대학교 대학원 전산과학전공(석사). 1991년 서울대학교 대학원 전산과학전공(박사). 1983년 ~ 1986년 서울대학교 도서관 전산화준비실. 1986년 ~ 1993년 단국대학교 전자계산학과 부교수. 1993년 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부 교수. 관심분야는 분산 시스템, 이동 컴퓨팅, 분산 객체 시스템, 운영체제