

애드 혹 네트워크를 위한 전력 기반 동적 클러스터 구성 방법

이종호*, 윤희용*, 이형수**, 전기만**

*성균관대학교 정보통신공학부,

**전자부품연구원 유비쿼터스컴퓨팅 연구센터

{glizid, youn}@ece.skku.ac.kr*, {hslee, kmjeon}@keti.re.kr**

Power Based Dynamic Clustering Scheme for Ad Hoc Networks

Jong Ho Lee*, Hee Yong Youn*, Hyungsu Lee**, Ki-man Jeon**

*School of Information and Communication, Sungkyunkwan University

**Ubiquitous Computing Research Center, Korea Electronics Technology Institute

요 약

애드 혹 네트워크(Ad Hoc Networks)은 유선 기반 망을 배제한 완전한 모바일 네트워크이다. 이동성이 큰 애드 혹 네트워크 노드간에는 경로 설정을 위해서 주기적으로 라우팅 패킷이 전송된다. 라우팅 방법은 소스 전송 요구에 따른 요구 기반 라우팅과 주기적으로 라우팅 테이블을 갱신하는 테이블 기반 라우팅으로 나누어진다. 또한, 이 두가지 방법을 혼합한 라우팅을 사용할 수 있는데, 기존의 연구에서 동등한 노드의 자원과 역할을 가정하기 때문에, 실제 네트워크에 적용하는데 한계가 있다. 본 논문에서는 클러스터를 이용한 라우팅 방법을 제안한다. 단순한 클러스터를 이용한 노드 관리는 클러스터 구성에 따른 관리 오버헤드가 발생하게 된다. 따라서 클러스터 구성 노드의 적정 수준을 유지하고, 전파 자원이 충분한 클러스터 선출 방법을 제안함으로써, 네트워크 전체의 관리 오버헤드를 줄일 수 있다. 시뮬레이션은 NS2 [12]를 이용하여, 시뮬레이션 시간동안 노드가 잔여 전력에 따라 클러스터 멤버의 수와 Hello Packet의 주기를 변화 시킴으로써 노드의 생존성을 보장하고, 최적을 클러스터 구성할 수 있는 노드 수를 측정하였다.

1. 서론

애드 혹 네트워크는 고정된 기반 망의 도움 없이 무선 인터페이스를 가진 이동 노드들 간에 구성되는 동적 네트워크이다. 애드 혹 네트워크는 망 형태의 비예측성, 제한된 대역폭, 에너지 그리고 메모리 등의 성격을 가진다. 이러한 무선 동적 네트워크는 매 순간마다 망의 형태가 변화하고 높은 이동성을 가지므로, 각 무선 노드 간의 데이터 교환을 위한 라우팅 기법이 매우 중요하다.

애드 혹 네트워크에서 제안된 라우팅 기법은 크게 테이블 기반(Table driven)방식과 요구 기반(Demand driven)방식으로 분류된다. 테이블 기반 방식은 모든 노드들의 라우팅 정보를 라우팅 테이블에서

상시 유지함으로써 전송 필요 시 별도의 경로 설정 없이 라우팅 테이블의 정보를 이용 전송함으로써 경로 구성의 지연이 짧다. 그러나 주기적인 라우팅 플러딩(flooding)을 통해서 무선 대역을 낭비하고, 애드 혹 네트워크의 높은 이동성에 기인한 라우팅 패킷이 증가해 네트워크의 성능이 떨어질 수 있다. 반면, 요구 기반 방식은 데이터 전송 필요 시에만 경로 설정을 수행한다. 이는 주기적인 라우팅 패킷 전송이 필요 없어 경로 설정의 오버헤드는 줄일 수 있으나, 데이터 전송 시 경로 설정 후 데이터를 전송하기 때문에 즉각적인 실시간 통신이 가능하지 않다는 단점이 있다. 두 방법 모두 장단점을 가지고 있지만, 현재로서는 노드의 이동성과 애드 혹 네트워크의 갖가지 제한을 해결하기 위해서 요구 기반 방식이 많이 사용된다.

기존의 요구 기반 라우팅 프로토콜은 각각의 노드를 동일한 전파반경과 임의의 이동 속도와 좌표 네트워크 위치를 기준으로 설계되었다. 애드 혹 네트워크 라우팅 설계에서 중요한 부분이 송수신을 위한 경로 설정과 노드의 이동 후 바뀌는 경로 관리이다. 따라서 속도와 효율적인 네트워크 관리를 위해서 클러스터 헤더를 선정하여 요구 기반 라우팅과 테이블 기반 라우팅의 혼합형이 제안되어졌다[8]. 그러나 혼합 라우팅 방법은 멤버 노드의 위치 변경에 따른 노드 정보 관리가 어렵고, 이에 따른 클러스터 헤더의 자원낭비가 발생할 수 있다. 균등한 클러스터 헤더의 자원 분배를 위해서는 클러스터 헤더가 관리하는 멤버 수를 제한할 필요가 있다.

본 논문에서는 모바일 노드의 자원과 서비스 요구에 기반 한 클러스터 관리 방법을 제안하고 클러스터 헤더에 집중된 관리 오버헤드를 인접 클러스터에 분산 시킴으로써 네트워크 전체의 자원 균등을 유지하는 방법을 제안한다. 시뮬레이션은 NS2를 이용하여, 노드의 잔여전력에 따라 동적으로 클러스터를 구성할 수 있는 노드 수를 제한하고, 주기적인 *Hello Packet*의 간격을 조절함으로써 기존 프로토콜에서 보장되지 않는 노드의 생존성을 시뮬레이션 시간 동안 보장됨을 확인하였다.

2. 관련 연구

애드 혹 네트워크에서는 각 노드 간의 높은 이동성 때문에 노드간의 통신을 위해서 적절한 라우팅이 필수적이다. 각 노드는 기본적으로 작은 자원을 가지고 있는 반면에 이동성은 높아, 유선 망에 비해 에너지 소모가 많다. 또한 라우팅 시에 불필요한 플러딩 패킷의 발생으로 네트워크 오버헤드가 증가한다. 경로 설정을 위해서 주기적으로 발생하는 패킷 오버헤드를 감소시키기 위해서 최근 들어 테이블 기반 방식 보다는 요구 기반 방식의 라우팅 기법이 많이 사용된다. 요구 기반 방식의 라우팅 기법에는 DSR(Dynamic Source Routing)[2], TORA(Temporally Ordered Routing Algorithm)[7], AODV(Ad Hoc On-Demand Distance Vector Routing) [6] 등이 있다.

클러스터를 이용한 계층적 라우팅 방법은 대부분 플러딩 기반으로 관리 패킷을 전송한다. 플러딩 방법의 경우, 소스 노드와 목적지 노드간에 노드의 위치 변경을 고려하지 않고 연결 설정을 할 수 있는 이점이 있다. 그러나 모바일 노드 수가 증가하게 되면 동시에 관리 패킷의 양도 증가하게 된다.

전통적인 클러스터 라우팅은 클러스터헤더가 1-홉 거리에 있는 이웃노드의 정보를 수신하여 관리 테이블을 구성하여 노드를 관리하고 인접 클러스터 헤더 간에는 게이트웨이를 이용하여 클러스터간에 통신하게 된다. 클러스터 기반 라우팅의 문제점은 첫째, 클러스터 헤더나 멤버 노드가 이동 시에 새로운 클러스터의 멤버나 헤더로 변경된다. 모바일 노드의 이동 속도가 증가할수록 클러스터는 자주 변경된다. 두 번째, 모바일 노드의 밀집도가 증가할 수록 클러스터

헤더가 가지는 오버헤드는 증가하게 된다. 즉, 멤버 노드 관리를 위한 Hello 패킷에 대한 응답 처리에 따른 자원 소모가 증가하게 된다.

따라서 클러스터 헤더에서 발생하는 오버헤드를 인접 클러스터 헤더와 동등하게 하기 위해서 클러스터 멤버의 수를 기준으로 클러스터를 합병/분할하는 방법을 생각해 볼 수 있다[9].

네트워크의 각 노드들은 자율적으로 클러스터 구성에 참여하게 된다. 각 노드는 클러스터 헤더를 중심으로 클러스터를 구성하게 되는데, 클러스터간에는 게이트웨이 노드를 통해서 정보 교환을 한다.

각 클러스터는 그림 1 a)의 *b, d, e, f, j, k, m, o*의 게이트웨이를 통해서 주기적으로 인접 클러스터의 정보(인접 클러스터 헤더와 멤버 수에 관한 정보)를 획득한다. 만약 현재 구성된 클러스터의 멤버 수가 하한 값(Lower bound)보다 작으면, 클러스터는 인접 클러스터와 합병하게 된다. 그림 1 a)에서 클러스터 헤더가 *c*인 클러스터는 게이트웨이 *b, d, e*를 통해서 인접 클러스터의 노드 수를 파악한다. 현재 클러스터 하한 값보다 노드 수가 적은 클러스터 헤더 *c*는 합병 메시지를 인접 노드인 *g*에 전달하게 되고 두 클러스터는 멤버 수 상한 값(Upper bound)에서 그림 1 b)와 같이 합병하게 된다.

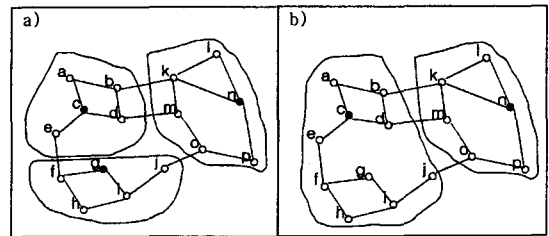


그림 1. 클러스터 구성

기존 연구에서는 클러스터의 분할과 합병을 결정하는 멤버 수의 상한 값과 하한 값에 선정 방법이 결정되어져 있지 않다. 모든 노드는 네트워크가 활성화되는 시간동안 자신의 전력자원을 이용하여 노드 유지 및 통신을 수행한다. 따라서 노드의 전파 자원에 따른 클러스터 헤더의 선정과 클러스터에 대한 분할/합병 알고리즘을 제시할 수 있다.

3. 클러스터 구성

네트워크의 형태를 $G=(V, E)$ 으로 정의 할 때 모바일 노드는 $v_i \in V$ 로 표시한다. 노드간의 링크는 $(v_i, v_j) \in E$ 로 표시하며, v_j 는 노드 v_i 의 이웃 노드이다. 임의의 클러스터 C_i 는 V 의 부분 집합으로, 클러스터의 멤버 수는 $L \leq |C_i| \leq U$ 로 정의 된다.

각 노드는 주기적으로 *Hello Packet*를 전송하여, 이웃 노드와 정보를 교환한다. 각 노드는 수신된 이

웃 노드의 정보를 이용하여 이웃 노드 테이블을 구성한다. v_i 의 이웃 노드의 집합은 NN_i 이고 v_i 가 클러스터 헤더 k 의 멤버라면 v_i 의 이웃 노드 집합은 식 1과 같이 정의할 수 있다.

$$NN_i^k = \{v_j | v_j \in NN_i \wedge k = Cluster(v_j)\} \quad (1)$$

만약 식 2와 같이 v_i 가 현재 어떤 클러스터 헤더에 속해 있지 않고, 이웃 노드들도 클러스터 헤더 정보를 갖지 않는다면, v_i 의 클러스터 헤더는 v_i 자신이 된다.

$$NN_i = \{v_j | v_j \in NN_i \wedge NULL = Cluster(v_j)\} \quad (2)$$

노드 v_i 의 클러스터 헤더가 없고, 수신된 Hello Packet의 이웃 노드가 클러스터 헤더 k 의 멤버라면 노드 v_i 는 자신의 클러스터 헤더 정보를 업데이트하고 클러스터 헤더에 멤버로 가입한다. 멤버 가입 후 주기적으로 Hello Packet을 전송할 때 이웃 노드들 중 노드 v_i 와 같이 클러스터 헤더 k 의 멤버가 없다면 v_i 는 이웃 노드의 클러스터 헤더 정보로 다시 업데이트하고 새로운 클러스터에 가입하게 된다.

4. 클러스터 분할 / 합병

네트워크가 활성화된 상태에서 모바일 노드는 효과적인 라우팅과 데이터 전송으로 전력 소모를 최소화할 필요가 있다. 특히 안정된 전력의 유선 네트워크와는 달리 모바일 노드 자체에서 경로 설정 과정과 데이터 전송이 동시에 이루어지는 애드 혹에서는 적절한 전력 제어가 네트워크 성능 보장에 관건이다. 네트워크 활성화 시간 동안에 통신을 보장하기 위해서 노드는 식 3과 같이 최소한의 P_{total} 의 초기 전력을 가져야 한다.

$$P_{total} \geq P_{idle}T + mP_{Tx} + nP_{Rx} \quad (3)$$

T = 네트워크 활성화 시간
 m = 전송시간
 n = 수신시간

그러나 실제 모바일 노드가 T 을 유지할 수 있는 전력을 가지는 것이 한정되어 있으므로, 잔여 전력량을 기준으로 노드의 역할을 구분할 필요가 있다. 각 클러스터 헤더는 자신의 잔여 전력량을 이용하여 클러스터 멤버의 최대 상한 값을 구할 수 있다. 노드의 잔여 전력량을 P_{remain} , Hello Packet의 시간 간격을 H_{period} , 남은 네트워크 시간을 T_{remain} , 전송 소비 전력을 P_{Tx} , 수신 소비 전력을 P_{Rx} 라 두자. 남은 네트워크 시간 동안 노드의 전력을 보장하기 위해서는 주기 적

으로 전송하는 Hello Packet에 대한 수신 패킷의 양을 줄일 필요가 있다. 모든 전송 Hello Packet에 대해서 n 개의 노드로부터 응답 메시지가 수신된다. 식 4는 Hello Packet 주기와 남은 네트워크 시간과의 관계를 나타낸다.

$$P_{remain} = \frac{(P_{Tx} + nP_{Rx})T_{remain}}{H_{period}} \quad (4)$$

식 4를 이용해서 노드의 수신 가능한 패킷의 수를 구할 수 있는데, 이는 클러스터 헤더에 의해서 전송된 Hello Packet의 응답한 노드 수와 동일하다. 따라서 클러스터 헤더가 네트워크 시간동안 노드간의 통신을 보장하기 위한 클러스터 멤버 수의 상한은 식 5와 같이 나타낼 수 있다.

$$U = \frac{P_{remain}H_{period} - T_{remain}P_{Tx}}{T_{remain}P_{Rx}} \quad (5)$$

클러스터 C_i 에 대해서 C_i 의 이웃 클러스터의 집합을 $NA_i = \{C_1, L, C_m, L, C_n\} (m \neq i)$ 로 정의할 수 있다. 클러스터의 노드 수가 하한 값 보다 적은 $|C_i| < L$ 인 경우, C_i 는 이웃 클러스터와 합병하게 된다. 이 경우 이웃 클러스터의 집합 중 $|C_i| + |C_k| \leq U$ 인 경우에는 노드 수가 가장 적은 C_k 와 합병하고, $|C_i| + |C_k| > U$ 인 경우에는 노드 수가 가장 많은 C_k 와 합병한다. 클러스터 간의 합병에서 새로운 클러스터 헤더의 선정은 클러스터 ID가 높은 순으로 결정한다. 만약 클러스터의 노드 수가 상한 값 보다 높은 (즉, $|C_i| > U$) 경우에는 클러스터 멤버 수가 상한 값 이하인 클러스터로 나뉘게 된다. 이 때 새로운 클러스터 헤더를 선정해야 하는데, 기존 클러스터 헤더가 클러스터 분할 메시지(Cluster Partition Packet)를 브로드캐스트하면, 멤버 노드는 자신의 클러스터 상한 값 U 를 계산해서 클러스터 헤더에 응답한다. 클러스터 헤더는 수신된 멤버의 상한 값 중 가장 높은 노드에 대해서 분할 허락 메시지(Partition Allow Packet)를 전송하게 된다. 클러스터 상한 값이 같은 노드가 생기면 클러스터 헤더는 클러스터 ID가 높은 노드에 분할 허락 메시지를 전송한다.

5. 성능 평가

성능 평가를 위한 시뮬레이터는 U. C. Berkeley에서 개발된 NS2를 사용하였다. 그리고 애드 혹 네트워크 라우팅은 Carnegie Mellon University에서 개발한 NS2 Extension을 사용하였다. NS2는 이벤트 구동 네트워크 시뮬레이터로 다양한 네트워크 메커니즘과 알고리즘을 제공한다 [12].

네트워크 노드는 200개 노드를 750m × 750m에 Random Waypoint Model로 이동성을 부여했다. 노드의 방향각은 Hello Packet의 전송 간격마다 변경하였다. 각 노드의 평균속도는 20m/s로, 그리고 패킷 전송과 수신에 사용되는 전력은 각각 1.33W와 967mW [13]으로, 클러스터에서 사용될 하한 값 L 은 10으로 가정하고 실험하였다.

실험을 크게 노드의 잔여 전력량을 측정하는 방법과, 잔여 전력량을 고정하고 제어 패킷 생산을 제어함으로써 노드의 생존성을 측정하는 두 가지 실험을 하였다. 노드의 잔여 전력량은 노드의 초기 전력량을 500W으로 두고, 5초 간격마다 고정적으로 Hello Packet의 발생과 노드의 속도 및 방향을 변화 시켰다.

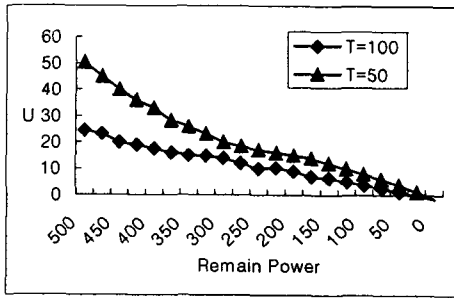


그림 2. 잔여 전력량에 따른 클러스터 멤버 상한 값.

그림 2는 시뮬레이션 시간 100초와 50초 동안 클러스터 헤더가 자신의 전력을 보장하면서 가질 수 있는 멤버 수의 상한 값이다. 시뮬레이션 시간 동안의 노드는 자신의 Hello 주기마다 자신의 남은 전력량을 확인하여 자신의 클러스터 멤버 가능 상한 값을 계산한다. 노드는 자율적으로 자신의 생존성을 보장하기 위해서 클러스터 멤버와의 송수신을 제한하기 위해서 멤버의 수를 제한한다. 클러스터 헤더는 주기적으로 자신의 정보를 플러딩하고, 멤버 수 상한 값 이내에서만 응답 메시지를 확인하고, 나머지는 응답에 대해서는 무시하게 된다.

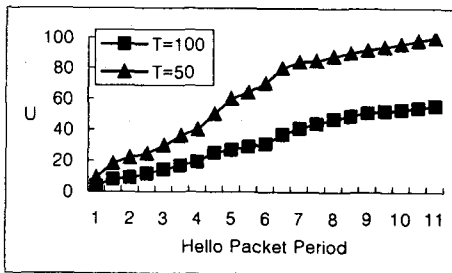


그림 3. Hello Packet 전송 주기에 따른 클러스터 멤버 상한 값.

만약 노드가 고정된 전력량을 가지고 서비스 시 간의 변화만 있다고 가정하면, 노드는 그림 3과 같이

Hello Packet의 전송 주기를 조정하여 전력 소모를 줄이게 된다. Hello Packet의 전송 주기를 길게 하고 그에 따른 클러스터 멤버 상한 값을 재조정함으로써 시뮬레이션 시간동안의 노드 생존성을 보장할 수 있다.

6. 결론 및 향후과제

본 논문은 애드 혹 네트워크에서 계층적 라우팅 프로토콜을 위한 클러스터 구성을 구성함에 있어서 발생하는 클러스터 헤더의 관리 패킷의 짐중으로부터 노드를 멤버 노드의 수를 분산하여 헤더의 수신 오버헤드를 줄일 수 있었다. 또한 클러스터 헤더의 잔여 전력을 중심으로 클러스터 멤버 수의 제한을 얻어 클러스터 헤더가 네트워크 시간 동안의 생존성을 보장받을 수 있었다. 그러나 노드의 생존성 보장을 위한 상한 값을 제시하였으나, 클러스터 간의 합병을 위한 하한 값은 앞으로 연구가 더 진행되어야 할 것이다.

참고문헌

- [1] C-K Toh, "Ad Hoc Mobile Wireless Networks", Prentice Hall PTR 2002.
- [2] David B. Johnson, David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing 1996.
- [3] G. Park, "In Mobile Ad-hoc Network for Reliable Data Transmission based Active Network Approach", 2001
- [4] Jochen H. Schiller, "Mobile Communications", Addison-Wesley 2000.
- [5] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, "A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", MobiCom 1998.
- [6] C. Perkins, E. Royer, "Ad-Hoc On-Demand Distance Vector Routing", Workshop on Mobile Computing Systems and Applications, 1996.
- [7] V. Park, M. Scott Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", INFOCOM 1997.
- [8] Z. Haas, M. Pearlman, "The Zone Routing Protocol for Ad Hoc Network", MobiCom 1998.
- [9] T. Ohta, S. Inoue, Y. Kakuda, "Ad Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks", ISADS 2003.
- [10] M. J. Suh, H. Y. Youn, J. H. Lee, H. Choo, "Multiple Zoning with Multiple Posts for Efficient Routing in MANET", 2th International Conference on Computer and Information Science (ICIS 2002), pp. 831-836, August 2002.
- [11] A. Nasipuri, R. Burlinson, B. Hughes, J. Roberts "Performance of a hybrid routing protocol for mobile ad hoc networks", Computer Communications and Networks, 10th International Conference, pp. 296-302, 2001.
- [12] The CMU Monarch Project's Wireless and Mobility Extensions to ns, The CMU Monarch Project, August 1999. Available from <http://www.monarch.cs.cmu.edu/>
- [14] M. Ilyas, "The Handbook of Ad Hoc Wireless Networks", pp. 22-1~22.10, CRC Press, 2003