

이동 애드혹 네트워크에서 주소 공간의 지역적 분배

김해용⁰, 최낙중, 최양희

서울대학교 컴퓨터공학부

{hykim⁰, fomula}@mmlab.snu.ac.kr, yhchoi@snu.ac.kr

The Local Distribution of Address Pool in Mobile Ad Hoc Networks

Haeyong Kim⁰, Nakjung Choi, Yanghee Choi

School of Computer Science and Engineering, Seoul National University

요 약

노드가 네트워크에 참여하고 다른 노드와 통신을 하기 위해서는 관리자가 수동으로 해당 노드를 설정하거나 네트워크 상의 DHCP 서버로부터 자동 설정이 가능해야 한다. 그러나 이러한 방법은 인프라 구조가 없는 이동 애드혹 네트워크에서는 실용적이지 못하다. 본 논문은 이동 애드혹 네트워크에 참여하려는 새로운 노드가 충돌 없이 효율적으로 IP 주소를 획득할 수 있는 LDAP 알고리즘을 제시한다. LDAP 알고리즘에서는 모든 노드가 모든 IP 주소에 대한 주소 풀(address pool)을 가지고 있으며 새롭게 네트워크에 참여하는 노드에게 주위 노드가 주소 풀을 분배하는 방식으로 동작한다. 모의 실험을 통하여 제안된 기법이 기존에 제시된 알고리즘에 비하여 주소 설정 시간과 통신 오버헤드 측면에서 우수한 성능을 보임을 증명한다.

1. 서 론

이동 애드혹 네트워크는 어떤 인프라 구조도 없이 동적으로 자동 네트워크를 형성하는 이동 노드의 집합을 의미한다. 이동 노드들은 기지국이나 AP 같은 중계기가 없어도 서로 간에 멀리 흩어져 통신이 가능하다. 지금까지 이 분야 대한 대부분의 연구는 라우팅 문제에 초점이 맞추어져 있는데, 이동 애드혹 네트워크에 참여하는 노드에게 이미 IP 주소가 할당되었다는 것을 가정하고 있다.

IP 주소 할당을 위한 방법에는 미리 IP 주소를 설정하는 정적인 방법, DHCP를 사용한 동적 할당 방법, 자동 설정 방법을 통한 동적 할당 방법으로 나눌 수 있다. 하지만 각 이동 노드가 이동 애드혹 네트워크에 참여하기 전에 수동으로 IP 주소를 설정한다는 가정은 궁극적으로 자동 동작이 가능해야 하는 이동 애드혹 네트워크에는 알맞지 않다. 따라서 이동 네트워크에 참여하는 노드의 주소를 자동으로 설정하는 방법이 필요하다. 본 저자들은 이를 위해 local IP 주소를 사용하는 자동 설정 방법인 LDAP(Local Distribution of Address Pool) 알고리즘을 제안한다.

2. 관련 연구

주소를 자동 설정하는 방법은 크게 두 가지로 나누어 볼 수 있다. 첫 번째로는 특정 노드를 선출하여 주소 할당을 위한 서버로 두는 방법이 있다. 중앙 집중형 방법에서 선출된 노드는 주소 리스트를 관리하고 할당해 주는 역할을 하게 되며, 새로운 노드는 주소를 얻기 위해 특정한 방법으로 선출된 서버 노드와 통신을 하게 된다. 이러한 방법은 [1, 2]에서 볼 수 있는데, 특정 노드를 선출하는 방법과 선출된 노드의 관리에서만 차이를 보일 뿐 일단 특정 노드를 선출한 후 주소를 할당하는 방법은 큰 차이를 보이지 않는다. 네트워크의 통합, 분리에 의한 주소 관리가 쉽다는 점

이 장점이지만, 서버 노드가 네트워크 장애의 병목지점이 될 수 있다는 단점이 있다.

두 번째로는 네트워크 내의 모든 노드가 동등한 입장을 유지하는 분산형 방법이 있다. 새로운 노드는 보편적으로 네트워크의 모든 노드와 통신을 함으로써 주소를 할당 받을 수 있게 된다. [3]에서는 MANETconf라는 임의로 선택된 주소가 다른 모든 네트워크 노드들로부터 주소의 중복 여부를 확인 받은 후에 사용될 수 있도록 하는 방법을 제안하였다. 하지만 이것은 주소 할당을 위해 트래픽이 순간적으로 급증하는 문제가 있다. 또한 동의를 얻는데 실패하는 것에 비례하여 주소 할당 시간이 길어지고 리스트를 관리해야 하는 추가적인 비용이 요구된다. [4]에서는 동일한 주소가 생성될 확률이 거의 없는 함수를 고안하여 네트워크의 오직 한 노드와 통신하는 것만으로 주소를 할당할 수 있는 Prophet 방법을 제안하였다. 이 방법은 주소 할당 시간과 통신 오버헤드 측면에서 명백한 장점을 가진다. 하지만 이용 가능한 주소 공간을 심하게 낭비하며 어디까지나 확률에 의존하는 방법이므로 결국 좀 더 견고한 알고리즘을 위해서 부가적인 방법을 필요로 한다. 그리고 이런 부가적인 방법은 결국 Prophet의 가장 큰 장점인 한 노드와의 통신만으로도 주소 할당이 가능하다는 이득을 상실하게 만든다.

3. 자동 주소 설정 알고리즘

LDAP 알고리즘은 몇 가지 기능을 전제로 하고 있다. 첫째, 각 노드는 모든 주소 범위에 대한 비트맵을 가지고 있다. 둘째, 동일한 네트워크에 속한 노드들은 동일한 netID를 가지고 있다. 마지막으로 주기적으로 주고 받는 헬로 메시지는 '자신의 주소 풀의 크기, netID'를 포함한다. 알고리즘의 기본 동작은 각 노드가 Start, Normal, Starve, Exhaust, Monopoly의 5가지 상태 중 하

* 본 논문은 2004년도 두뇌한국21, 국가지정연구실 프로젝트 지원을 받아서 수행되었음.

나의 상태로 머무르면서 발생하는 이벤트에 따라 적절한 동작과 함께 현재 상태를 변화시킴으로써 이루어진다.

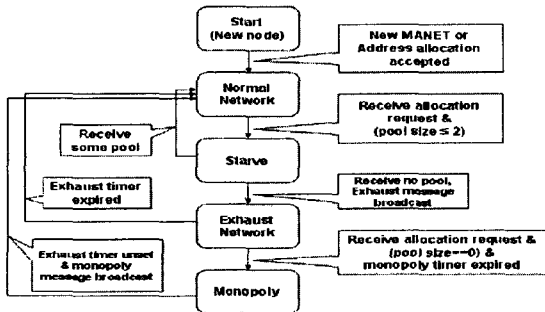


그림 1. LDAP 알고리즘의 순서도

Start 상태에서는 IP 주소를 갖지 못한 새로운 노드가 주소 할당 요청을 하고 이웃 노드로부터 주소 할당 응답을 받게 된다. 3회의 주소 할당 요청에 대하여 아무런 응답을 받지 못하면 임의로 IP를 선택하고 자신의 IP를 초기값(seed)으로 사용하여 netID를 만들어 스스로 독립적인 네트워크를 형성하게 된다. 이 때 애드혹 네트워크에서 사용 가능한 모든 주소 범위에 해당하는 주소를 풀로 지닌다. 이웃 노드들로부터 주소 할당 응답을 받았을 경우에 메시지에 포함된 netID 정보와 주소 풀의 크기 정보를 이용하여 적절한 노드를 선택하여 주소 풀 할당을 다시 요청하게 된다. 이 요청을 받은 노드는 자신의 주소 풀의 반을 전해주게 되고, 새로운 노드는 획득한 주소 풀에서 하나의 주소를 선택하여 자신의 주소로 사용하게 된다.

IP 주소가 할당된 노드들은 Normal 상태가 된다. 이 상태는 안정된 네트워크 상황으로 주기적으로 주고 받는 헬로 메시지를 통하여 자신의 주소 풀 크기와 이웃 노드의 주소 풀 크기를 비교하게 된다. 이웃 노드의 주소 풀 크기가 자신의 주소 풀 크기의 4배 이상인 경우, 상대의 주소 풀의 반을 넘겨 받는 풀 교환 과정을 통해 네트워크의 모든 노드들이 어느 정도 균등한 주소 풀을 갖도록 한다. 이를 통해 주소 풀이 전체 네트워크에 어느 정도 여유 있게 남아 있는 경우에는 특정 메시지를 전체 네트워크에 플래딩하는 오버헤드 없이 기존의 노드가 새로운 노드에게 주소를 빠르게 할당해 줄 수 있다.

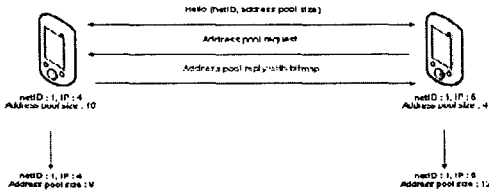


그림 2. Normal 상태

새로운 노드로부터 주소 할당 요청을 메시지를 받았을 때 자신의 주소 풀의 크기가 2 이하라면 Starve 상태가 된다. Starve 상태가 된 노드는 전체 네트워크에 starve 메시지를 플래딩해서 일정 임계값(자신의 주소 풀의 크기+1)x2+2) 이상의 주소 풀의 크기를 지닌 노드가 있다면 이 중 일부(1/4)를 받도록 한다. 만약 임계값 이상의 주소 풀을 가지고 있는 노드가 없어서 주소 풀을 얻지 못할 경우, 전체 네트워크에 exhaust 메시지를 플래딩하여 현재 애드혹 네트워크의 주소풀이 거의 균등하게 고갈되어

있음을 알린다.

Exhaust 메시지를 받은 네트워크의 모든 노드들은 Exhaust 상태가 된다. 이 상태가 되면 먼저 exhaust 타이머를 작동시켜 이후에 새로운 노드에게 주소 풀을 할당하고 나서 자신의 주소 풀이 2 이하가 되더라도 starve 메시지를 플래딩하지 않는다. 이를 통해 네트워크에서 exhaust 메시지가 반복적으로 플래딩되는 것을 막는다. Exhaust 상태가 되면 이웃 노드의 주소 풀의 크기가 자신의 주소 풀 크기의 3배 이상인 경우, 1/2을 넘겨 받는다. Exhaust 타이머가 종료되면 Normal 상태로 돌아간다.

새로운 노드로부터 주소 할당 요청 메시지를 받았을 때 자신이 가진 주소 풀이 0이면 그 노드는 monopoly 상태가 되어 모든 다른 노드들에게 새로운 netID가 포함된 monopoly 메시지를 보낸다. Monopoly 메시지를 받은 노드들은 자신의 netID를 바꾸고, 자신의 모든 주소 풀을 삭제한 뒤 응답을 한다. 이러한 과정을 통해 해당 애드혹 네트워크의 netID를 변경하여 이후에 언급될 네트워크 분리, 통합 과정에 유용하게 사용될 수 있도록 하고, monopoly 상태의 노드는 네트워크에서 이용 중인 주소를 제외한 모든 주소에 대하여 주소 풀을 획득함으로써 노드 고장 등으로 인해 발생한 주소 풀의 누수 현상을 해결한다. 이렇게 수집된 주소 풀은 다시 임의로 몇 개의 노드에게 유니캐스트를 통하여 동등하게 할당하여 네트워크 내의 노드들이 빠르게 주소 풀을 균등하게 나누어 가질 수 있도록 한다. 또한 동시에 여러 노드가 monopoly 상태가 될 경우에는 IP 주소가 낮은 노드에게 우선 순위를 주어 경쟁을 해결하도록 한다.

네트워크 분리와 통합 측면에서 LDAP 알고리즘은 일시적인 네트워크 분리 후 다시 통합하는 경우에는 다른 특별한 처리가 필요하지 않다. 일시적인 네트워크 분리가 발생했다고 하더라도 주소 풀이 남아 있다면 가능한 이것을 먼저 사용하고 해당 네트워크 주소 풀의 크기가 매우 작다면 Monopoly 상태가 되어 네트워크 분리나 노드 고장으로 인하여 발생한 주소 풀의 누수를 찾아 이용하기 때문이다. 그러나 네트워크 분리된 후에 Monopoly 상태를 거쳐 주소 풀에 누수가 생긴 부분을 찾아 새롭게 주소 풀을 분배했을 때 독립적으로 형성된 netID가 다른 두 네트워크 통합하는 경우에는 특별한 동작이 필요하다. Monopoly 상태에서 네트워크의 모든 노드들에게 새로운 netID가 할당되며, 독립적으로 형성된 다른 네트워크는 당연히 netID가 다르므로 헬로 메시지에 포함된 netID를 통하여 통합을 인식할 수 있다. 통합을 인식한 각 네트워크의 두 노드는 Monopoly 상태가 되어 monopoly 메시지를 각 네트워크에 플래딩한다. 이 때 두 노드는 자신들의 IP 주소를 활용하여 하나의 새로운 netID를 생성하고 이것을 monopoly 메시지를 통하여 알려주게 된다. 두 노드는 수집된 주소 풀을 이용하여 중복된 주소의 노드를 알아내어 이 노드들에게 새로운 IP 주소를 할당해 주고, 동시에 남아 있는 주소 풀을 몇 개의 노드에게 균등하게 분배한다. 만약 동시에 여러 노드들이 네트워크 통합을 인식하는 경우에는 더 낮은 netID를 생성하는 두 개의 노드에게 우선권이 주어져 경쟁을 해결한다. 이처럼 잠시 분리가 되었던, 이전에 동일한 netID로 네트워크를 형성하던 서브 네트워크의 통합의 경우에 Monopoly 상태를 거쳐 통합이 되었을 때는 새로운 netID를 선택함으로써 주소 풀이 겹치지 않도록 재분배 되도록 할 수 있다.

현재 LDAP 알고리즘은 다른 노드의 주소 풀까지 알아내어 주소 풀의 총용까지는 검사하지 않는다. 따라서 단지 사용 중인 IP

주소의 충돌만을 찾아내기 위하여 기존 연구를 통하여 제안된 방법들을 사용할 수 있다. 다른 노드의 IP 주소에 대한 정보를 받았을 때 이 IP 주소가 자신의 주소 풀에 속한다면 자신의 주소 풀에서 이 IP 주소를 삭제한다. 만약 중복되는 IP 주소가 자신이 사용하는 IP 주소라면 자신의 IP 주소를 자신의 주소 풀의 다른 IP 주소로 변경한다. 자신의 주소 풀이 없다면 Start 상태로 돌아가 새로운 노드가 참여하는 과정을 거쳐 IP 주소를 할당 받는다.

다음 장에서는 실험을 통하여 LDAP 알고리즘이 기존 연구들에서 제안된 방법들에 비하여 우수한 성능을 보임을 보인다.

4. 성능 평가

기존 연구에서 제안되었던 알고리즘 중 RADA (Random ADdress Allocation) [5] 방법과 비교를 통해 LDAP의 성능을 평가해 보았다. 성능 평가의 척도로는 주소 할당 비율에 따른 주소 할당 지연 시간, 평균 주소 할당 오버헤드(메시지 수)가 사용되었다. 각 노드는 네트워크 분리가 일어나지 않도록 실험 토폴로지 전체에 걸쳐서 고르게 분포하도록 위치시켰고, 주소 풀의 크기는 256, 네트워크에 참여하는 노드의 수를 1에서 245까지 증가시키면서 진행되었다.

그림 3은 주소 할당 비율에 따른 주소 할당 지연 시간을 보여주는 그래프이다. RADA의 경우에 주소 할당 비율이 증가할수록 주소 충돌이 빈번하게 발생하게 되고 이에 따라 지연 시간이 3번의 타임 아웃에 추가되므로 평균 지연 시간이 급격히 증가하는 추세를 보인다. LDAP의 경우, 기본적인 지연 시간은 이웃 노드들의 주소 풀이 모두 0이 아닌 이상, 단지 이웃 노드와의 통신에 걸리는 시간뿐이다. 주소 할당 비율이 작은 경우는 모든 이웃 노드의 주소 풀이 0인 확률은 극히 작으며, 따라서 평균 지연 시간은 $2 * (\text{통신 딜레이})$ 의 상수로 유지된다. 주소 할당 비율이 증가함에 따라 새로운 노드의 모든 이웃 노드가 가진 주소 풀이 0인 경우가 발생하게 되면, 주소를 두 홉 이상으로부터 빌려오는 지연 시간이 추가되어 평균 지연 시간도 증가하게 된다. 하지만 이는 RADA의 지연 시간에 비해서 상당히 작고 선형적으로 증가하는 경향을 보인다.

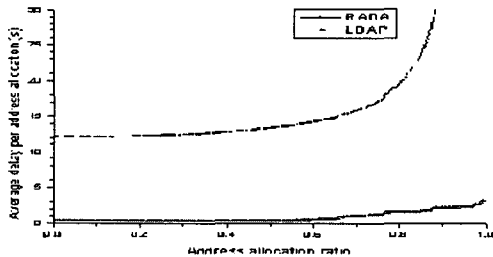


그림 3. 주소 할당 비율에 따른 주소 할당 시간

그림 4는 주소 할당 비율에 따른 단위 주소 할당 시 평균 메시지 교환 수를 보여주는 그래프이다. RADA의 경우 기본적으로 $3 * (\text{전체 네트워크 노드 수})$ 해당하는 만큼의 메시지 교환이 필요하다. 또한 주소 할당 비율이 높아질수록 위의 과정을 반복하게 되므로 RADA의 메시지 오버헤드는 급격히 증가하게 된다. 반면에 본 논문에서 제안한 LDAP 알고리즘의 경우, 새로운 노드가 임의의 위치에 들어왔을 때 단지 이웃 노드들과 메시지를 교환하므로 아주 작은 오버헤드가 든다. 주소 할당 비율이 증가하여 약

0.56이상이 되면 새로운 노드의 모든 이웃 노드 주소 풀이 0인 경우가 발생하여 임의의 한 노드가 Monopoly 상태가 되어 전체 네트워크로부터 주소를 얻어오는 과정을 거쳐야 하기 때문에 각 노드의 수만큼의 브로드캐스트 메시지와 각 노드에서 보내는 응답의 유니캐스트 메시지가 추가의 오버헤드가 된다. 하지만 이러한 오버헤드도 RADA에 비하면 상당히 작은 값이다.

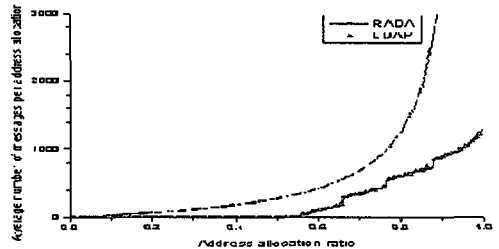


그림 4. 주소 할당 비율에 따른 주소 할당 오버헤드

5. 결론

본 논문에서는 이동 애드혹 네트워크에서 IP 주소를 자동으로 설정하는 LDAP 알고리즘을 제안하였다. 기본적으로 LDAP 알고리즘의 노드들은 5가지 상태 중에 하나의 상태에 머물면서 특정이벤트가 발생하였을 경우, 지정된 동작을 취한다. 이와 같은 LDAP 알고리즘은 기존의 대부분의 다른 연구들과는 달리 주소 할당 과정이 전체 네트워크를 통하여 이루어지는 것이 아니라 이웃 노드 중 한 노드에 의해 지역적으로 이루어지게 되며, 오버헤드도 적고 지연 시간도 매우 작아지게 된다. 또한 네트워크 분리 및 통합에 있어서도 주소 풀이 고갈되어 새로운 노드에게 지역적으로 주소를 할당해 줄 수 없는 상황을 제외한다면 큰 오버헤드 없이 네트워크 분리 및 통합을 처리할 수 있다. 따라서 새로운 노드가 네트워크에 참여하려는 경우에 발생하는 메시지의 오버헤드, 지연 시간의 관점에서 매우 좋은 성능을 보인다.

6. 참고 문헌

- [1] Mesut Gunes and Jorg Reibel, "An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-hoc Networks," In Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services, Sophia Antipolis, France, September 2002.
- [2] Yuan Sun and Elizabeth M. Belding-Royer, "Dynamic Address Configuration in Mobile Ad hoc Networks," UCSB Technical Report 2003-11, June 2003.
- [3] S. Mesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," In Proceedings of the IEEE conference on Computer Communications (INFOCOM), New York, NY, 2002.
- [4] Hongbo Zhou, Lionel Ni and Matt Mutka, "Prophet Address Allocation for Large Scale MANETs," In Proceedings of the IEEE conference on Computer Communications (INFOCOM), San Francisco, 2003.
- [5] C. E. Perkins, J. T. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun, "Ad hoc Address Autoconfiguration," draft-ietf-manet-autoconf-01.txt, November 2001.