

Geographic Ad hoc 라우팅을 위한 두 단계 위치 발견 알고리즘

길홍열^o 김중권^o
서울대학교 컴퓨터공학부
{hrgil^o, ckim}^o@popeye.snu.ac.kr

A Two-step Location Discovery Algorithm for Geographic Ad hoc Routing

Hong-ryeol Gil^o Chong-kwon Kim^o
School of Computer Science & Engineering, Seoul National University

요 약

대부분의 Ad hoc 라우팅 프로토콜은 네트워크의 규모에 비례하여 경로 설정에 따른 오버헤드가 증가한다. 이에 반해 Geographic 라우팅 프로토콜은 노드의 지리적인 위치만을 이용하여 패킷을 forwarding하기 때문에 각각의 노드는 자신의 이웃 노드에 대한 정보만 유지하면 된다. 따라서 Geographic 라우팅 프로토콜이 큰 규모의 Ad hoc 네트워크 환경에 적합하다. 이러한 Geographic 라우팅은 목적지의 위치를 얻기 위해 Location service를 필요로 한다. 본 논문에서는 Geographic 라우팅 프로토콜을 큰 규모의 네트워크 환경에 적용할 수 있도록, 효율적인 Location service를 제공하는 TLD를 제안한다.

1. 서론

Ad hoc 통신망은 기지국과 같은 고정적인 통신 인프라의 도움 없이 이동성을 갖는 무선단말기만으로 구성되는 네트워크이다. Ad hoc 네트워크는 기존의 통신망과는 전혀 다른 환경으로 데이터 전송을 위해 모든 노드들이 호스트의 역할뿐만 아니라 라우터의 역할도 수행해야 한다.

Ad hoc 네트워크의 특징으로 노드의 이동성에 따른 네트워크 토폴로지의 동적인 변화를 들 수 있다. 이러한 새로운 네트워크 환경에, 기존 유선망에서 사용되던 라우팅 프로토콜을 그대로 적용하기에는 여러 어려운 점들이 있기 때문에 Ad hoc 네트워크 환경에 맞는 라우팅 프로토콜의 연구가 활발히 진행되었다. 그 결과, DSDV [1], DSR [2], AODV [3], GPRS[4] 등과 같은 새로운 라우팅 프로토콜이 제안되었다.

DSDV는 pro-active 라우팅 프로토콜로서 노드간의 주기적인 업데이트 패킷을 주고 받음으로써 네트워크상의 모든 노드에 대한 경로 정보를 라우팅 테이블에 유지한다.

DSR, AODV와 같은 on-demand 라우팅 프로토콜은 필요에 따라 RREQ(Route Request) 패킷을 네트워크 전체에 flooding하여 목적지까지의 경로를 찾는다. DSR에서는 RREQ 패킷에 지나온 노드마다의 ID를 패킷에 기록했다가 RREP(Route Reply) 패킷을 통해 송신지 노드에게 알려준다. ADOV에서는 RREQ 패킷이 지나온 노드마다 reverse-path에 대한 정보를 라우팅 테이블에

추가하고, RREP 패킷이 reverse-path를 통해 송신자에게 전달될 때, 송신자에서 목적지 사이의 경로가 중간 노드들의 라우팅 테이블에 추가된다.

이와 같은 DSDV, DSR, AODV는 모두 네트워크 토폴로지에 토대를 둔 라우팅 프로토콜로서, 규모가 큰 Ad hoc 네트워크에는 적합하지 않다. 먼저, DSDV의 경우 네트워크의 규모가 커지면 주고 받는 라우팅 정보의 양이 커지게 되고, 노드마다 유지하는 라우팅 테이블의 크기가 커지는 문제가 있다. 그리고 DSR과 AODV의 경우, 경로를 찾기 위해 RREQ 패킷을 전체 네트워크에 flooding해야 하므로 네트워크의 규모가 커질수록 경로 설정을 위한 오버헤드도 커지게 된다. 또한 DSR의 경우, 네트워크의 규모가 커지면 소스 라우팅을 위한 ID의 리스트 길이가 늘어나기 때문에 RREQ 패킷과 RREP 패킷, 그리고 경로 캐시(Route cache)의 크기가 지나치게 커질 수 있다.

GPRS와 같은 Geographic 라우팅 프로토콜은 일반적인 라우팅 프로토콜과는 달리 노드의 위치 정보를 이용하여 패킷을 전달한다. 노드의 위치 정보는 GPS와 같은 시스템을 이용하여 구할 수 있다고 가정한다. 그리고 각각의 노드는 주기적인 Hello 패킷을 주고 받음으로써 이웃 노드의 위치 정보를 유지한다. 송신자가 목적지 노드에게 패킷을 전송하고자 할 경우, 송신자는 자신의 이웃 노드 중 목적지와 가장 가까운 노드에게 패킷을 전달한다. 그러면 이를 수신한 노드는 자신이 목적지 노드인지 검사한다. 만약 그렇지 않다면, 송신자와 마찬가지로 자신의 이웃 노드 중 목적지 노드와 가장 가까운 노드에게

패킷을 전달한다. 이러한 과정을 반복하면 패킷은 목적지 노드에게 전달된다.

Geographic 라우팅 프로토콜에서 각각의 노드는 이웃 노드의 위치 정보만 유지하면 되고, 경로를 찾기 위해 on-demand 라우팅 프로토콜처럼 전체 네트워크에 flooding할 필요가 없으므로 큰 규모의 Ad hoc 네트워크 환경에 적합하다.

그러나 Geographic 라우팅에도 문제점은 있다. 송신자가 목적지의 위치를 알기 위한 Location service가 필요하다는 것이다. 가장 간단한 방법으로 하나의 Location server를 두고 모든 노드들이 자신의 위치 변화를 업데이트하는 방법이 있을 수 있으나, Server에 너무 많은 트래픽이 몰리는 문제가 있다. 이에 따라 Location server를 분산적으로 유지하는 GLS [5]가 제안되었다. 이 방법이 비록 트래픽이 몰리는 문제는 해결했지만, 위치 정보를 중복해서 기록해 둘뿐만 아니라 목적지로 사용될지 안될지도 모르는 네트워크상의 모든 노드의 위치 정보를 유지함으로써 불필요한 오버헤드를 야기한다.

본 논문에서는 Location server를 사용하지 않으면서, 필요에 따라 목적지의 위치를 알아내는 TLD(Two-step Location Discovery) 알고리즘을 제안한다. TLD는 필요에 따라 목적지 주소를 적은 오버헤드로 찾음으로써 효율적인 Location service를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 Location service와 관련된 선행 연구에 대해 알아본다. 그리고, 3장에서 TLD의 구성 요소에 대해 설명한다. 마지막으로 4장에서는 TLD의 향후 과제와 결론을 맺는다.

2. 관련 연구

Ad hoc 네트워크에 대한 활발한 연구의 결과로 큰 규모의 Ad hoc 네트워크 구축도 가능해지고 있다. 이와 같은 큰 규모의 네트워크 환경에는 Geographic 라우팅이 가장 적합하다고 여겨진다. 이러한 Geographic 라우팅은 목적지 노드의 위치를 알기 위해 Location service를 요구한다.

GLS(Grid Location Service)에서는 각각의 노드가 Location server를 분산시킴으로써 Scalability를 높이고자 하였다. GLS에서 모든 노드는 동일한 Grid 구성을 알고 있다고 가정한다. Grid는 여러 order를 갖는 사각형으로 이루어지는데, order i 의 사각형이 4개가 모여 order $i+1$ 의 사각형을 구성한다.

각각의 노드는 자신과 인접한 order 1의 사각형에서 자신의 ID와 가장 근접한 ID를 갖는 노드를 Location server로 지정한다. 그리고, 반복적으로 order의 크기를 늘려가며 Location server를 지정한다.

어떤 노드가 목적지의 위치가 필요할 때는 자신이 속한 order 1의 사각형에서 목적지 ID와 가장 근접한 node에게 Location query를 보낸다. Location query를 수신한 노드는 자신이 해당 목적지의 Location server라면 송신자에게 응답하고, 그렇지 않다면 order를 늘려가며 같은 동작을 반복해 나간다. 그러면 결국, Location

query는 목적지의 Location server에 전달된다.

GLS가 Location server를 분산적으로 두는 방법으로도 단일 Location server를 사용하는 것에 비해 효율적이지만, 네트워크상의 모든 노드가 Location server를 선정하고 자신의 위치가 변할 때마다 업데이트 해야 하기 때문에 불필요한 오버헤드를 야기한다. 어떤 노드는 한동안 목적지로 사용되지 않을 수 있기 때문이다. 따라서, GLS는 큰 규모의 Ad hoc 네트워크 환경에는 적합하지 않다.

3. TLD

3.1 Parting Time Table

Geographic routing을 위해서는 모든 노드가 일정 주기로 Hello 패킷을 주고 받아 이웃 노드의 위치 정보를 유지해야 한다. TLD에서는 이러한 Hello 패킷을 이용해 Parting time table에 추가적인 정보를 유지한다. Table의 entry는 노드의 ID와 Parting time이 쌍을 이룬다.

각각의 노드는 일정 주기마다 Parting time table에서 현재 이웃 노드가 아닌 entry의 Parting time 값을 1씩 증가시킨다. 만약, Parting time의 값이 P_{Max} 값을 초과하면 table에서 해당 entry를 삭제한다.

그리고 Hello 패킷을 수신하게 되면, Table에서 Hello 패킷을 송신한 노드의 Parting time을 0으로 설정한다. Table에 존재하지 않는다면 새로 추가하면 된다. 결국, Parting time은 과거 이웃 노드였다가 헤어진 뒤에 흐른 시간을 나타낸다.

GRID QUERY
Source ID
Source location
Destination ID
Cross node location

그림 1. Grid query packet fields

3.2 Grid query

송신자 노드는 동서남북의 4방향으로 자신과 d만큼 떨어진 위치에 Cross 노드의 위치를 정한다. 그리고 해당 Cross 노드의 위치로 GQ(Grid Query) 패킷을 Geographic forwarding을 통해 전달한다.

GQ 패킷에 적힌 Cross 노드 위치와 가장 근접한 노드가 Cross 노드가 되고, 해당 Cross 노드는 수신된 방향을 제외한 나머지 3방향으로 Cross 노드를 정한다. 그리고 GQ 패킷을 각각의 Cross 노드로 Geographic forwarding을 한다. 이러한 작업을 반복하면 결국 격자 모양으로 GQ가 전파된다. 이때, GQ 패킷을 중복해서 수신하거나, 이미 목적지에 해당하는 ZQ(Zone Query) 패킷을 수신한 노드는 수신된 GQ 패킷을 무시한다.

GQ를 수신한 노드가 목적지 노드이면, Grid query를 중지하고, 자신의 위치를 Geographic forwarding을 통해 송신자 노드에게 알려준다. GQ를 수신한 노드가 목적지 노드는 아니지만 목적지 노드의 ID가 Parting time table에 존재한다면, Grid query를 중지하고 Zone

query를 시작한다.

ZONE QUERY
Source ID
Source location
Destination ID
Cross node location
Time to live

그림 2. Zone query packet fields

3.3 Zone query

목적지 노드가 Parting time table에 존재한다는 것은 이 노드 근방에 목적지 노드가 존재한다는 것을 의미한다. Zone query를 시작한 노드는 목적지 노드에 해당하는 Parting time에 1을 더한 값으로 TTL을 정해 ZQ 패킷을 브로드캐스팅한다. TTL을 정해서 브로드캐스팅하기 때문에 작은 범위 내의 노드들만 ZQ 패킷을 수신하게 된다.

목적지 노드가 ZQ 패킷을 수신하게 되면 자신의 위치를 송신지 노드에게 Geographic forwarding를 통해 알려준다.

이상의 Query 과정에 대한 예가 그림 3에 나와있다.

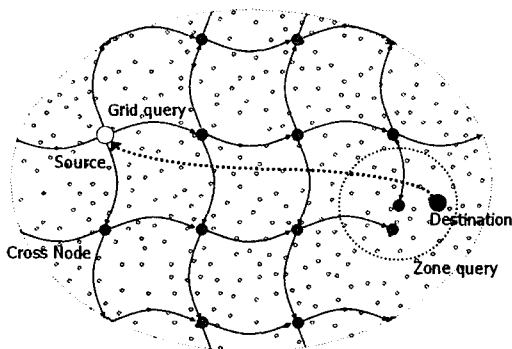


그림 3. TLD Example

3.4 Query optimization

송신자와 목적지간의 경로가 설정되면, 서로 주고 받는 패킷에 각자의 위치를 기록함으로써 추가적인 Query 발생을 감소시킬 수 있다.

그리고 위치 캐시(Location cache)를 사용하여 Query 패킷의 발생을 줄일 수 있다. 캐시를 통해 대략적인 목적지 위치를 알 경우, 해당 위치를 Cross 노드로 하여 Grid query를 그 방향으로만 보낸다. 그러면 해당 지역에서만 Zone query가 일어난다.

4. 결론 및 향후 과제

본 논문에서는 Geographic 라우팅에서 효율적인 Location service를 제공하기 위한 TLD를 제안하였다. TLD는 특정한 Location server를 두지 않고, 필요에 따

라 목적지의 주소를 찾는 방법을 사용한다. 각각의 노드는 현재 이웃 노드의 위치와 과거 이웃이었던 노드와의 Parting time만을 유지하면 된다. 목적지의 위치를 알고자 하는 송신자는 4방향으로 Cross 노드를 정하고 GQ를 전파하면 된다. Parting time table에 해당 목적지 ID를 갖고 있는 노드가 GQ를 수신하게 되면, ZQ를 좁은 지역 내에서만 브로드캐스팅하여 목적지의 위치를 찾아내게 된다. 또한 위치 캐시를 이용할 경우 과거의 목적지 위치를 Cross 노드로 하여 그 근방에서만 ZQ를 브로드캐스팅함으로써 목적지의 위치를 찾을 수도 있다.

이처럼 TLD는 필요에 따라 목적지의 위치를 찾아내기 때문에 기존의 Location service와 달리 Location server를 유지하는데 필요한 오버헤드와 Location update에 따른 오버헤드가 전혀 없다. 따라서 TLD는 큰 규모의 Ad hoc 네트워크 환경에 적합하다.

TLD는 이동성을 갖는 Ad hoc 네트워크를 가정했으나, 이동성이 적을 경우 Grid query시에 Parting time table에 해당 목적지의 정보가 없어서 Zone query를 못하는 경우가 발생할 수 있다. 이 경우 단계적으로 d의 값을 줄여가며 Grid query를 재시도 하는 방법을 사용할 수 있다. 다른 방법으로는 각각의 노드가 자신의 이동성 정도를 판단하여 사전에 4방향으로 d만큼 떨어진 노드에게 자신의 존재를 알리는 방법이 있다. 해당 노드를 중심으로 십자가선상에 위치하는 노드들의 Parting time table에 해당 노드의 정보를 추가시키는 것이다. 이렇게 되면 Grid query시에 Zone query가 반드시 실행된다.

저자는 앞으로 모의 실험을 통해 TLD의 성능을 평가하고 각각의 파라미터 P_{Max} 와 d의 최적값을 찾아낼 계획이다.

5. 참고문헌

- [1] C. Perkins, P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers", In Proceedings of ACM SIGCOMM, August 1994.
- [2] D. Johnson, D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, Kluwer Academic Publishers, 1996, pp.154-181.
- [3] C. Perkins, E. Royer, "Ad-hoc On-Demand Distance Vector Routing", In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999.
- [4] B. Karp, H. T. Kung, "GPSR:Greedy Perimeter Stateless Routing for Wireless Networks", In Proceedings of ACM/IEEE MobiCom 2000, August 2000.
- [5] J. Li, J. Jannotti, D. DeCouto, D. Karger and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing", In Proceedings of MobiCom 2000.