

무선 Ad-hoc망에서 N-hop 기반의 효율적 Flooding 기법

최종혁[^], 한승진^{**}, 오임걸^{***}, 이균하^{**}

^{*}인하대학교 정보통신공학과

^{**}인하대학교 컴퓨터 공학부

^{***}한서대학교 인터넷공학과

e-mail:c2001011@inhavision.inha.ac.kr

An Efficient Flooding Scheme of N-hop base in Mobile Ad hoc Networks

Jong-Hyeok Choi[^], Seung-Jin Han^{**}, Im-Geol Oh^{***}, Kyoon-Ha Lee^{**}

[^]Dept. of Information Communication and Engineering, Inha Univ.

^{**}School of Computer Science and Engineering, Inha Univ.

^{***}Dept. of Internet technology and Engineering, Hanseo Univ.

요 약

Ad-hoc망은 기존 유선 및 무선망과 같이 안정된(고정된) 하부구조를 필요로 하지 않는 무선 단말들의 그룹으로 구성된 네트워크이다. 현재 라우팅을 위해 제시된 on-demand 방식의 라우팅 프로토콜은 이동성에 따른 신뢰도를 높이기 위해 flooding을 수행하는데, 이동성이 커질 경우 경로발견과 이를 유지하기 위한 오버 헤드가 크다. 따라서 본 논문에서는 각 노드의 다양한 이동성향이 설정된 노드간의 경로유지에 영향을 적게 미치도록 하고 오버헤드가 적은 경로설정을 위한 N-hop기반의 효율적인 flooding 기법을 제안하고 노드 이동성에 따른 최적의 N값을 제시한다.

1. 서론

Ad-hoc은 네트워크 하부구조를 필요로 하지 않는 무선 단말들의 그룹으로 구성된 네트워크이다. 이러한 특성은 네트워크의 배치(Deployment)를 용이하게 하여 군 작전 시 및 수색과 구조 작업 시의 통신 환경 등과 같은 분야에서 실제적으로 활용이 가능하다. 또한 무선에서의 이동성 지원에 관한 기술이 쏟아지면서 Ad-hoc은 더욱 주목을 받는 연구 분야가 되어가고 있다[1,2].

그러나, 그와 더불어 Ad-hoc이 무선 네트워크의 근본적인 구조가 되기 위해서 거론되어야 할 많은 기술적인 문제들이 있는데, 그 문제들 중에 보다 중요한 것은 라우팅이다. 왜냐하면 기존 유선 및 무선망의 라우팅에서 고려하지 않았던 문제들을 갖고 있기 때문이다. 망을 구성하는 모든 노드들은 라우터로서 동작을 하여 이동노드로부터 패킷을 수신하여 다른 이동노드로 전달하여야 하는데, Ad-hoc 망에서 노드들의 빈번한 이동과 제한된 전력소모 등으로 인하여 이동노드들간에 경로발견과 이를 유지하기 위한 효율적인 라우팅 기법이 요구된다.

현재까지 Ad-hoc망을 위해 많은 라우팅 프로토콜들이 제안되었으며 그 중에서 on-demand방식의 프로토콜들은 경로발견을 위해서 blind flooding을 수

행함으로써 라우팅 오버헤드가 커지는 문제점을 갖고 있다. 본 논문에서는 각 노드에서 N-hop 거리의 이웃노드들에 대한 정보를 관리하게 함으로써 쿼리가 발생했을 때 선택적으로 flooding하는 기법을 위해 노드 이동성에 따른 최적의 N값을 제시한다. 시뮬레이션을 통해 제시한 N값이 on-demand 라우팅 프로토콜의 경로재설정에 대해 라우팅의 효율성이 높아짐을 보인다.

2. 기존 Ad-hoc 라우팅 프로토콜

기존의 유선망에서 라우팅 프로토콜은 Ad-hoc 네트워크에 존재하는 빈번한 네트워크 형상의 변화를 고려하지 않았다. 기존의 유선 네트워크에서 라우터 사이의 경로는 고장으로 인해 동작하지 않거나 통신 과부하로 인하여 새로운 경로설정이 필요하지만 라우터의 위치가 고정적이기 때문에 경로변화에 따라 능동적으로 대처하기가 힘들다. 이동 호스트가 라우터 역할을 대신하는 Ad-hoc 네트워크 환경에서 기존의 알고리즘 특히 거리-벡터(Distance-Vector)알고리즘도 통신망의 동적 변화 후 새로운 통신 경로 설정에 많은 통신 지연 현상이 발생한다[3].

Ad-hoc 네트워크를 위한 라우팅 프로토콜은 호스트의 이동과 같은 변화를 빠르게 수용할 수 있으면

서 변화가 없는 상태에서도 오버헤드가 거의 없는 장점을 제공해야한다. Ad-hoc 네트워크 라우팅 프로토콜은 경로를 결정하는 시기에 따라 크게 Table driven(테이블 구동)방식 또는 Proactive(순항적)설정 방식 또는 사전결정방식과 On-demand(송신자-주도 주문식)방식 또는 Reactive(반응적)설정 방식 또는 요구기반 결정방식이 있으며, 두 가지가 혼합된 Hybrid 설정 방식으로 분류된다[4].

3. N-hop기반의 선택적 flooding 기법

제안하는 N-hop기반의 선택적 flooding 기법은 기본적으로 노드는 급격하게 이동하지 않는다는 공간적 지역성(spatial locality)에 기반을 두고 있으며 변경된 경로는 기존의 경로에 비해 크게 다르지 않다고 가정한다.

3.1. N-hop거리의 이웃노드에 대한 정보관리

우선 각 노드들은 일정시간 간격으로 HELLO패킷을 주고받음으로써 이웃노드에 대한 정보를 유지한다. 이웃노드에 대한 정보는 HELLO패킷 뿐만 아니라 각 노드에서 이웃노드에서 발생하는 데이터 전송을 지속적으로 청취함으로써도 알 수 있다. 또한 HELLO패킷을 이웃노드로 전송을 할 때 자신이 가지고 있는 이웃노드에 대한 정보를 같이 포함해서 보내게 되면 각 노드에서는 자신으로부터 최대 N-hop 거리에 대한 노드의 정보를 유지할 수가 있게 되며 N-hop 이내에서의 노드이동에 의한 경로변경은 따로 쿼리를 발생시키지 않기 때문에 라우팅 오버헤드를 줄일 수 있게 된다.

이 방법의 문제점은 주기적으로 이웃노드에 대한 정보를 같이 실어서 보내주어야 하므로 패킷의 크기가 커진다는 점인데 이는 각 노드에서 이웃노드 테이블을 유지하여 HELLO패킷을 보낼 때 기존의 노드 리스트와 차이가 없으면 이웃노드 리스트를 실어 보내지 않고 변경된 리스트만을 보내게 되면 패킷의 크기를 크게 줄일 수 있다.

3.2. N-hop look-ahead를 이용한 선택적 flooding

노드의 이동으로 인해 기존경로가 변경이 되었을 지라도 노드가 급격하게 이동하지 않는 한 기존의 경로와 큰 차이를 보이지 않으며 일정한 거리내의 노드이동으로 인한 경로 변경은 N-hop거리의 이웃노드에 대한 정보를 이용해서 복구가능하며 N-hop 이상의 경로차가 발생했을 경우에는 N-hop look-ahead(사전조사)된 것을 이용해서 쿼리를 고품화시키게 된다.

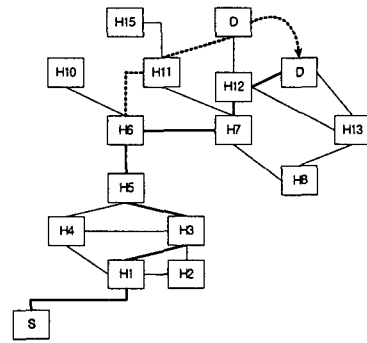


그림 1. N-hop Look-ahead를 이용한 선택적 flooding

그림 1은 2-hop기반의 선택적 flooding의 경우로, 목적지 노드 D가 이동을 하여 기존의 경로 [S, H1, H3, H5, H6, H11, D]로 전달되는 패킷은 노드 H6에서 경로가 변경이 되었음을 알게 되고 2-hop 거리의 이웃 노드상에 기존 경로가 존재하지 않음을 알게 되어 노드 H6에서 기존의 경로의 주변에서 변경된 경로 이후의 노드를 찾게 된다.

이 때 쿼리 패킷을 수신한 노드가 기존의 경로상에 존재하지 않는 노드일 경우에는 N의 값을 하나씩 증가시키게 되며 N값이 미리 설정한 최대값보다 크게 되면 그 노드는 패킷을 더 이상 flooding시키지 않게 되어 기존경로에서 일정범위내의 노드에게 선택적으로 flooding을 시키게 된다. 위의 예에서는 N의 최대값이 1인 경우로 노드 H6으로부터 시작된 쿼리는 노드 H7에서의 이웃노드테이블에 의해서 새로운 경로를 발견하게 된다.

4. 시뮬레이션 및 결과

앞에서 제시한 방법의 성능평가를 위해서 ns-2의 CMU Monarch extension을 사용하여 시뮬레이션을 수행하였다[5].

4.1. 시뮬레이션 환경

표 1. Simulation parameters

Mobility Model	Number of Nodes	50
	Environment Size	1000m×1000m
	Simulation Time	100sec
Traffic Model	v_{max}	20m/sec
	Traffic Type	CBR
	Packet Rate	4 packets/sec
	Packet Size	512bytes

시뮬레이션에서 사용한 프로토콜은 기존의 AODV로서 AODV는 경로관리를 위해서 내부적으로

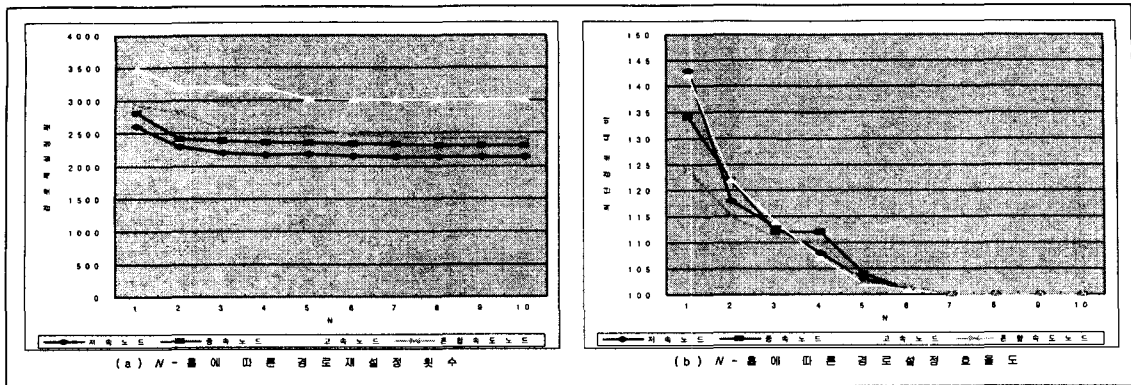


그림 2. 시뮬레이션 결과

HELLO패킷을 주기적으로 보내주고 있어 여기에 N-hop거리의 이웃노드정보를 관리하게 하고 선택적 flooding 이 가능하도록 변경하고 노드들의 구성은 저·중·고·혼합속도의 네 가지 부류로 자유롭게 움직이는 상황을 가정하여 실시하였다. 실험은 N값에 따른 변화를 관찰하기 위해 이를 노드들의 움직이는 상황을 동일하게 해서 비교·관찰을 하였고 이와 같이 처리한 목적은 노드의 움직임과 속도를 완전히 랜덤하게 해서 N값에 따른 변화를 측정하면서 나타날 수 있는 비교의 불확실성을 제거하기 위해서이다.

4.2. 결과 및 분석

결과 및 분석은 두 가지 기준에 의해 실시한다. 첫 번째는 경로재설정 횟수이다. N-값이 증가함에 따라 전체적으로 경로재설정 횟수가 감소하는 결과가 그림 2의 (a)와 같이 나옴을 알 수 있고, 속도가 빨라지면 경로재설정 횟수가 증가하는 것으로 나타났다. 이는 빠르게 움직이는 노드일수록 빈번하게 경로를 재설정해야 한다는 것을 의미한다. 그리고 속도가 혼합된 노드들에서는 경로재설정 횟수가 고속노드들 보다 적고 중속노드들 보다는 많음을 알 수 있다.

두 번째로 그림 2의 (b)는 경로효율도를 나타낸다. N-값이 증가할수록 효율적인 경로를 생성함을 알 수 있다. 이는 N-hop이내의 링크연결상태를 파악할 수 있기 때문에 N-hop에 따라 선택적으로 flooding 되어짐을 보여주고 있다. 특히 N값이 3이상이 되면 blind flooding방법에 비해 경로효율도가 기존의 AODV보다 최대 15%까지 차이가 나는 것을 알 수 있다. N값이 7이상이 되면 수렴하는데 이는 7-hop 이상의 이웃 상태에 대한 Look-ahead(사전조사)는 무의미하다는 것을 보여주고 있다.

5. 결론

On-demand 라우팅 프로토콜은 Ad-hoc망에서 효과적인 프로토콜이나 경로발견을 위한 쿼리의 blind flooding이 프로토콜의 효율성을 제한하여 왔다. 본 논문에서는 이러한 on-demand 라우팅 프로토콜이 갖는 문제점을 N-hop거리의 이웃노드에 대한 정보를 관리하게 함으로써 쿼리를 선택적으로 flooding 시키는 기법을 위해 네 가지 부류의 노드 이동성에 따라 경로재설정 횟수와 경로설정 효율도를 통해 최적의 N값을 보였다.

향후 연구과제로서는 AODV외의 on-demand 라우팅 프로토콜에서의 수행능력 및 선택적 flooding 이 패킷전송율에 미치는 영향과 네트워크 이동성에 따른 최적의 N값을 정하는 것이다.

참고문헌

[1] M. Chatterjee, Sk. and D. Turgut, "An On-Demand Weighted Clustering Algorithm for Mobile Ad-Hoc Networks," in *proc. of the IEEE GLOBECOM*, 2000.
 [2] D. Johnson, D. Maltz, Y. Hu, J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad-hoc Network," *IETF Internet Draft-ietf-manet-dsr-05.txt*, March, 2001.
 [3] G.Resta, P.Santi, "An Analysis of the Node Spatial Distribution of the Random Waypoint Model for Ad-hoc Network", in *Proc. of ACM Workshop on Principles of Mobile Computing 2002*, Toulouse, pp.44-50, October, 2002.
 [4] Elizabeth M. Royer, Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hc Mobile Wireless Networks," *IEEE Personal Communications*, Vol.6, No.2, pp46~55, 1999.
 [5] Kevin Fall, Kannan Varadhan, "ns notes and documentation," The VINT project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, 1999.