

DYMO 라우팅 프로토콜의 이동 속도와 트래픽 부하에 따른 경로 축적의 효율성

나권문
 고려대학교 컴퓨터정보통신대학원
 e-mail : nawkm@korea.ac.kr

Efficiency of Path Accumulation with DYMO Routing Protocol in Mobility and Load Environment

Kwon-Moon Naw
 Graduate School of Computer Information & Communication,
 Korea University

요 약

MANET에서의 전통적인 라우팅 프로토콜은 일반적으로 두 종류로 나뉘는데 proactive 라우팅 프로토콜과 on-demand 라우팅 프로토콜이다. 라우팅 정보의 proactive 전파와 reactive 발견 사이에는 필수적인 trade-off 가 존재한다. 모든 시나리오들을 충족하는 라우팅 프로토콜이 존재하지 않는 것은 분명하다. 그래서 최적의 라우팅 프로토콜을 발견하기 위한 연구가 계속되고 있으며 IETF의 MANET 워킹그룹은 여러 다른 프로토콜을 제안 중에 있고 그 중에 하나가 on-demand 라우팅 프로토콜의 AODV를 계승한 DYMO(Dynamic MANET On-demand)이다. DYMO는 경로 축적 메커니즘을 채택하였다. 경로 발견 과정에서 노드들은 라우팅 메시지를 포워딩하기 전에 그들 자신의 라우팅 정보도 패킷에 추가할 수 있다. 결국 소스와 목적지 사이의 모든 노드들의 라우팅 정보가 교환되는 것이다. 이 논문에서는 이 경로 축적 메커니즘의 사용 유무에 따라 다양한 속도와 트래픽 부하에서 DYMO의 성능이 어떻게 변하고 최적의 환경은 무엇인지 연구해보고자 한다.

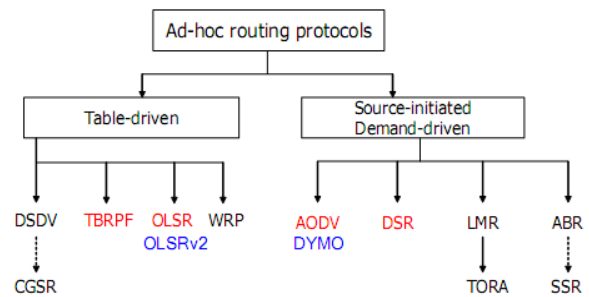
1. 서 론

MANET에서의 전통적인 라우팅 프로토콜은 일반적으로 두 종류로 나뉜다. Proactive 라우팅 프로토콜과 on-demand 라우팅 프로토콜이다.

DSDV [1], OLSR [2], TBRPF [3] 등과 같은 proactive 라우팅 프로토콜은 노드들 사이에 주기적으로 정보를 교환하고 네트워크 안에서 모든 노드들의 유효한 경로의 집합을 계속적으로 유지한다. proactive 프로토콜은 경로 정보를 미리 교환함으로써 낮은 지연시간과 높은 신뢰성을 갖지만 참여하는 노드의 수가 증가할수록 높은 오버헤드를 수반하게 된다. 반대로 AODV [4], DSR [5] 같은 on-demand 라우팅 프로토콜은 데이터 패킷이 네트워크에 발생할 때만 경로 발견 절차를 수행하기 때문에 오버헤드를 줄이는 장점을 가지고 있다. 그러나 노드의 트래픽 부하가 커지게 되면 좋은 성능을 내지 못하게 된다. [6] 라우팅 정보의 proactive 전파와 reactive 발견 사이에는 필수적인 trade-off 가 존재한다.

모든 시나리오들을 충족하는 라우팅 프로토콜이 존재하지 않는 것은 분명하다. 그래서 최적의 라우팅 프로토콜을 발견하기 위한 연구가 세계 여러 곳에서 계속되고 있으며 IETF의 MANET 워킹그룹은 여러 다른 프로토콜을 제안 중에 있다. 그 중에 하나가 on-demand 라우팅 프로토콜의 AODV를 계승하여 작고 간단하며 확장성 있

는 프로토콜을 목표로 연구되고 있는 DYMO (Dynamic MANET On-demand)이다 [7].



(그림 1) IETF MANET Working Group

DYMO는 경로 축적 메커니즘을 채택하였다. 이 논문에서는 경로 축적 메커니즘의 사용 유무에 따라 다양한 속도와 트래픽 환경에서 DYMO의 성능이 어떻게 변하고 최적의 환경은 무엇인지 알아보려고 한다.

2장에서는 DYMO 프로토콜에 대해 간략히 설명하였고 3장에서는 시뮬레이션을 통해 경로 축적의 성능을 평가 분석하였다.

2. DYMO (Dynamic MANET On-demand)

2.1 DYMO 개요

On-demand 프로토콜로서 DYMO는 작고 간단하며 확장성 있는 유니캐스트 라우팅을 목표로 표준화되고 있다. DYMO는 노드들이 요청할 때만 목적지까지의 경로를 설정한다. 결과적으로 최소한의 라우팅 정보만을 교환하는 것이다. 하나의 노드가 목적지까지의 경로를 필요로 할 때 RREQ 메시지를 전체 네트워크에 브로드캐스트한다. RREQ가 목적지에 도착하면 목적지 노드는 RREP를 소스 노드로 유니캐스트로 보냄으로써 경로가 설정되게 된다. 경로를 설정하는 동안 노드들은 자신의 경로 정보를 패킷에 덧붙임으로써 경로를 추적할 수 있게 된다. 그래서 소스와 목적지 경로뿐만 아니라 더욱 많은 경로 정보들이 경로 설정 패킷에 포함되게 된다.

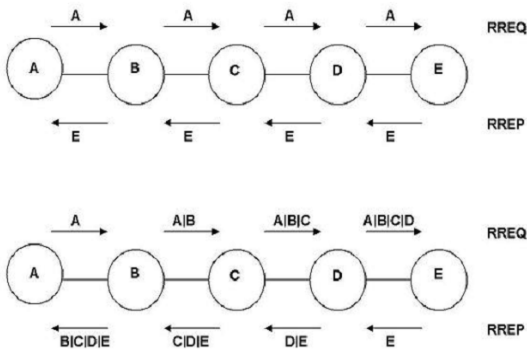
경로가 오랫동안 사용되지 않으면 그 경로는 삭제되게 된다. 만약 삭제된 경로로 패킷 포워딩이 요청되면 노드는 RERR 메시지를 소스노드로 보내고 경로를 더 이상 사용하지 못하게 한다. 또한 DYMO는 순차번호와 홉 카운터를 사용함으로써 경로의 유효성을 체크한다.

2.2 경로 발견과 경로 추적

라우팅 메시지라고 불리는 RREQ와 RREP가 경로 발견 절차동안 교환된다. 이 메시지에는 소스와 목적지의 주소뿐만 아니라 루프를 방지하기 위한 홉 카운터와 순차번호도 포함된다.

노드가 라우팅 메시지를 받았을 때 순차번호와 홉 수를 기초로 하여 이미 알고 있는 정보보다 최신정보인지 확인하고 그들의 라우팅 테이블을 갱신한다. 유효하지 않은 라우팅 메시지는 삭제하고 더 이상 전파하지 않는다.

노드들은 라우팅 메시지를 포워딩하기 전에 그들 자신의 라우팅 정보도 패킷에 추가할 수 있다. 결국 소스와 목적지 사이의 모든 노드들의 라우팅 정보가 교환되는 것이다. 이것이 바로 경로 축적(Path Accumulation)이다. (그림 2)는 경로 축적을 설명하고 있다 [8].



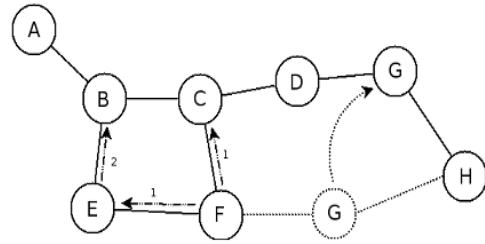
(그림 2) Without and With Path Accumulation

2.3 경로 유지

DYMO는 동적인 네트워크 토폴로지에 적용되기 때문에

경로가 설정된 이후에 지속적인 모니터링이 필요하다. 특별한 모니터링 메커니즘을 강요하지는 않지만 타이머를 이용해서 오래되고 안전하지 못한 경로들을 삭제함으로써 최신의 경로 정보를 유지한다.

만약 단절된 경로가 발견되면 RERR 메시지가 네트워크에 브로드캐스트된다. RERR 메시지 안에는 접근할 수 없는 노드와 이 노드를 통해 이전에 접근할 수 있었던 노드 정보도 포함된다.



(그림 3) RERR 브로드캐스트

(그림 3)에서처럼 F가 G의 이동을 발견하게 되면 더 이상 G와 H가 F를 통해 접근할 수 없다고 RERR 메시지를 브로드캐스트한다. 만약 C가 D를 통한 G로의 경로를 알고 있다면 C는 더 이상 RERR를 전파하지 않는다. 그러나 RERR이 최신의 정보라면 RERR은 모든 이웃노드들에게 전파된다.

3. 경로 축적의 효율성

경로 축적을 사용하면 소스와 목적지 사이의 모든 노드들의 라우팅 정보가 교환되므로 경로를 설정할 때 RREQ를 감소시키고 더 빠른 RREP를 가능하게 한다. 게다가 RREQ를 받았을 때 노드가 목적지까지의 유효한 경로를 알고 있다면 이것을 이웃 노드로 전파하는 대신에 intermediate RREP를 보내게 된다. 결과적으로 노드는 RREP를 더 빠르게 받고 RREQ를 더 이상 전파하지 않아도 됨으로써 트래픽 오버헤드를 감소시킨다. 그러나 노드의 이동속도가 낮고 트래픽 부하가 적어서 경로 단절과 재설정이 많이 일어나지 않는 환경에서도 경로 축적이 좋은 효율을 보일지 의문이다. 이러한 환경에서는 경로 축적 메커니즘의 사용이 경로 설정 메시지의 감소로 인한 이득보다는 경로 설정 패킷의 크기가 커지는 단점으로 인해 오히려 비효율적이지는 않을까. 이 논문에서는 다양한 환경에서의 실험을 통해 이를 알아보려고 한다.

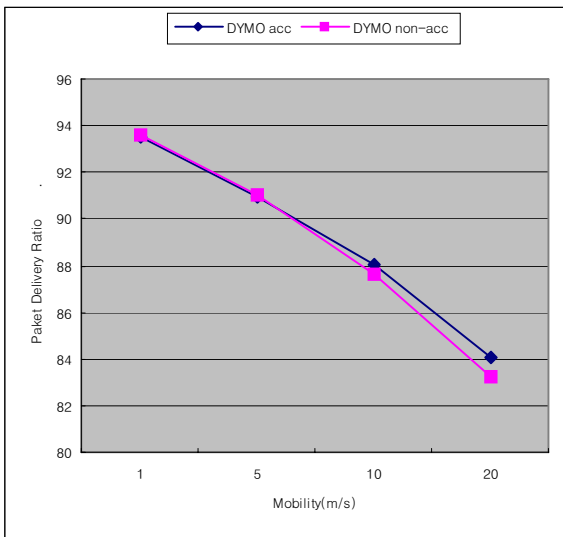
4. 성능 평가

이 논문에서는 DYMO 라우팅 프로토콜의 다양한 이동속도와 트래픽 부하에 따른 경로 축적의 효율성을 평가하기 위해 패킷 레벨 시뮬레이터인 ns-2 (ver. 2.29)를 사용하였다 [9]. DYMO 프로토콜은 ns-2에 포함되어 있지 않기 때문에 공개된 소스코드인 DYMOUM(ver. 0.3)을 사용

하였고 소스코드 내의 no_path_acc 파라미터의 설정을 변경함으로써 경로 축적의 유무를 달리하면서 결과를 측정하였다 [10][11].

시뮬레이션은 표준 IEEE 802.11 라디오 주파수의 2Mbps의 대역폭과 250미터의 전송범위로 설정했고 모든 실험은 100초 동안 실행했다. 데이터 패킷 크기는 512바이트이고 4kbps의 전송속도로 보낸다. 랜덤하게 생성되는 UDP를 사용하고 소스와 목적지의 커넥션 수는 실험1에서는 20쌍으로 고정했으나 실험2에서는 트래픽 부하의 변화를 주기위해 5, 10, 15, 20쌍으로 달리하였다.

네트워크의 크기는 1500x300미터이고 ns-2에서 제공하는 랜덤 시나리오 생성 툴을 이용하여 자유롭게 움직이는 25노드가 사용되었다. 노드들의 이동 속도는 실험1에서는 1, 5, 10, 20m/s로 변화를 주면서 성능을 측정하였고 실험2에서는 20m/s로 동일한 속도에서 실험을 하였다. 그리고 실험을 통해 나온 값들의 오차를 줄이기 위해 20번의 반복적인 실험으로 평균값을 얻었다.

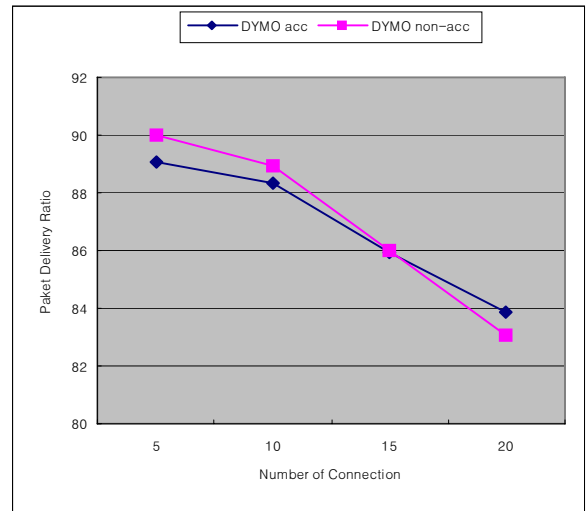


(그림 4) 이동속도에 따른 패킷 전송률 (1500x300m, 25nod, 20connection)

(그림 4)는 실험1의 결과로 일정한 트래픽 부하 (20connections)에서의 속도에 따른 패킷 전송률을 보여준다. 속도가 낮을 때는 패킷 전송률이 거의 비슷한 결과를 보였다. 그 이유는 속도가 낮을 때는 경로 축적 유무에 상관없이 높은 패킷 전송률을 보이는데 이럴 경우 경로 축적 메커니즘의 사용은 라우팅 메시지 패킷을 커지게 하여 트래픽에 부담을 주기 때문에 추가적인 라우팅 정보 교환의 효과를 상쇄시킨다. 하지만 속도가 높아짐으로써 경로 축적의 효율성이 나타나기 시작한다.

노드들의 이동 속도가 높아짐으로써 경로 단절로 인해 새로운 경로 발견을 위한 라우팅 메시지들이 생성되게 되는데 경로 축적을 하는 경우 추가적인 라우팅 정보로 인해 이러한 라우팅 메시지들의 생성을 감소시키게 되고 패

킷 전송률을 높일 수 있게 되는 것이다. 경로 축적은 속도가 20m/s일대 약 1%의 패킷 전송 효율성을 나타냈다.



(그림 5) 트래픽 부하에 따른 패킷 전송률 (1500x300m, 25nod, 20m/s)

(그림 5)는 실험2의 결과로 일정한 속도(20m/s)에서의 커넥션 수에 따른 패킷 전송률을 보여준다. 커넥션 수가 적을수록 경로 축적을 하지 않는 것이 더 좋은 성능을 나타냈다. 이는 낮은 커넥션 수와 낮은 트래픽 부하를 보이는 환경에서는 오히려 경로 축적이 라우팅 메시지의 패킷 크기를 증가시켜 전송률에 부정적인 영향을 미치기 때문이다. 그러나 커넥션 수가 증가하면서 트래픽이 증가하고 전체적으로 전송률이 낮아지는 환경에서는 경로 축적이 좋은 성능을 나타냈다. 그 이유는 경로 축적이 높은 트래픽 부하에서 전체 네트워크에 전파되어 많은 오버헤드를 발생시키는 라우팅 메시지를 줄이기 때문이다. 경로 축적은 커넥션 수가 20일 때 약 1%의 성능 향상을 보였다.

5. 결론 및 향후 연구 계획

DYMO의 경로 축적 유무에 따른 여러 가지 환경에서 패킷 전송률을 알아보았다. 낮은 속도와 낮은 트래픽 환경에서는 경로 축적은 오히려 성능을 약간 감소시키지만 높은 속도와 높은 트래픽 환경에서는 성능 향상을 보인다는 것을 알 수 있었다.

위의 결과를 볼 때 모든 시나리오에서 최적의 성능을 내는 라우팅 프로토콜이 없듯이 경로 축적 또한 특정한 환경에서 최적의 효과를 볼 수 있다는 것을 알 수 있었다. 하지만 아직 DYMO는 많은 연구가 진행 중이고 경로 축적 메커니즘도 현재에 머물러 있지 않고 계속 향상된 메커니즘으로 발전할 것으로 보인다.

가장 이상적인 DYMO 프로토콜은 노드의 이동속도와 커넥션 수에 따라 경로 축적 사용 유무를 동적으로 바꾸는 것이다. 향후 이러한 동적 메커니즘 사용을 실제로 프로토콜에 적용할 수 있도록 연구해 보고자 한다.

참고문헌

- [1] C. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," Proceeding of the ACM SIGCOMM'94, Sep. 1994.
- [2] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, Oct. 2003.
- [3] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," Internet Draft, draft-ietf-manet-tbrpf-11.txt, Oct. 2003.
- [4] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. <http://www.ietf.org/rfc/rfc3561.txt>, July 2003.
- [5] D. Johnson, Y. Hu, and D. Maltz. The dynamic source routing protocol (DSR). <http://www.ietf.org/rfc/rfc4728.txt>, February 2007.
- [6] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols," Proceedings of IEEE INFOCOM'00, Mar. 2000.
- [7] I. Chakeres and C. Perkins. Dynamic MANET On-demand (DYMO) routing (work in progress). <http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-10.txt>, July, 2007.
- [8] Charles E. Perkins "Better Plumbing for Reduced Flooding" - www3.ietf.org/proceedings/07mar/slides/manet-10.pdf, March, 2007.
- [9] The network simulator - ns2," Available at <http://www.isi.edu/nsnam/ns/>.
- [10] Pedro M. Ruiz, Francisco J. Ros "DYMOUM: A DYMO implementation for Linux and ns-2" August 4th, 2005.
- [11] DYMOUM "<http://sourceforge.net/projects/dymoum/> Accessed on July 2007.