

AODV 기반의 제어 메시지 Overhearing을 통한 다중경로 설정 메커니즘

서돈철^o 김화성
광운대학교 전자통신공학과
don@kw.ac.kr, hwkim@daisy.kw.ac.kr

An AODV-based Multi-path Routing using Control Packet Overhearing

Doncheol SEO, Hwasung KIM
Dept. of Electronic and Communications Engineering, Kwangwoon Uni.

요 약

애드 hoc 네트워크는 기존의 하부 기반구조를 필요로 하지 않고도 이동 노드들 사이의 통신이 가능한 네트워크를 의미한다. 현재 애드 hoc 라우팅 프로토콜에 관련하여 IETF MANET에서는 단일 경로를 제공하는 라우팅 프로토콜들이 제안되어 있으나 애드 hoc 네트워크의 특성상 노드의 잦은 이동으로 인해 빈번한 경로 실패가 발생함으로 인하여 큰 오버헤드와 지연시간을 가지고 있다. 이러한 문제를 해결하기 위해 다양한 다중경로 설정 메커니즘들이 제안 되었으나 추가적인 제어 메시지와 복잡한 메시지 처리 과정으로 인하여 애드 hoc 노드의 제한된 처리 능력과 네트워크 자원을 소모를 가져왔다. 본 연구에서는 대표적인 요구기반(on demand) 방식의 단일경로 라우팅 프로토콜인 AODV와 다중경로 라우팅 프로토콜 AOMDV의 단점을 보완한 저비용 다중경로 메커니즘을 제안한다.

1. 서 론

이동 애드 hoc 네트워크는 노드의 이동성으로 인한 동적인 네트워크 위상(topology)과 제한된 채널 대역폭과 노드의 전원으로 특징지을 수 있다.

이러한 애드 hoc 네트워크에서 가장 중요한 문제는 동적인 네트워크 위상에서 적은 오버헤드로 경로설정을 할 수 있도록 하는 것이다. 여기에서 오버헤드는 경로 설정을 위한 제어 메시지를 말하는데 이런 제어 메시지는 통신을 하는데 있어 채널 대역폭뿐만 아니라 이를 처리하기 위해서 노드의 전원을 소비하게 된다.

이런 오버헤드를 줄이기 위해서 데이터를 전송하기 위한 경로가 필요할 때에만 제어 메시지를 이용하는 On-Demand 방식의 라우팅 프로토콜이 제안 되었다. 이런 방식이 라우팅 프로토콜의 예로서는 DSR(Dynamic Source Routing), TORA(Temporally Ordered Routing Algorithm), AODV(Ad-hoc On-demand Distance Vector) 등이 있다[1]. 앞의 라우팅 프로토콜은 모든 노드들 간의 경로를 주기적으로 업데이트 하여 유지하는 Proactive 방식의 프로토콜과 대조된다.

최근 애드 hoc 네트워크의 성능에 관한 연구에서 On-demand 라우팅 프로토콜이 Proactive 라우팅 프로토콜에 비해서 낮은 라우팅 오버헤드를 요구함을 보여 주었다[2]. 하지만 On-demand 방식의 라우팅 프로토콜은 높은 경로발견 지연(Delay)과 동적인 네트워크 환경에서 잦은 경로발견 과정을 거침으로서 라우팅 프로토콜의 성능의 저하라는 문제점을 가지고 있다.

이를 해결하기 위해 본 논문에서는 대표적인 On-demand 방식의 라우팅 프로토콜인 AODV에 다중경로 설정 메커니즘을 적용하여 한번의 경로발견 과정을 통해 다중경로를 설정할 수 있도록 하였다[3,4]. 이때 다중경로는 소스 노드뿐만 아니라 중간 노드에서도 설정이 가능하다. 이때 재 경로설정은 오직 모든 경로가 사용가능 하지 않을 때에만 일어나게 된다.

이러한 방법은 경로설정 지연뿐만 아니라 라우팅 오버헤드를 줄이는데 있어서도 효과적인 방법으로 적용될 수 있다. 또한 다중경로 설정은 애드 hoc 노드의 부하를 분산 시키는데 있어서도 이용될 수 있다.

본 논문에서는 기존 다중경로 설정 라우팅 프로토콜을 개선하여 보다 빠르고 효과적인 장애 극복 능력과

동적인 네트워크 상황에서 신뢰성 있는 경로 제공 방법을 제시한다.

2. AODV 와 AOMDV 라우팅 프로토콜 동작 메커니즘

AODV 는 기존의 Proactive 방식의 애드 혹 라우팅 프로토콜인 DSDV의 Destination Sequence Number(목적지 순서번호)와 On-demand 방식의 프로토콜 DSR의 On-demand 경로발견 방식을 결합한 Loop-free, On-demand, 단일경로, distance vector 의 특성을 제공하는 프로토콜이다. AODV는 크게 경로발견(Route Discovery)과 경로유지(Route Maintenance)의 메커니즘으로 동작을 하게 된다[3]. 이 절에서는 AODV와 AOMDV의 동작 메커니즘에 대해서 설명한다.

2.1 Route Discovery and Route Maintenance

경로발견 과정은 소스 노드로 데이터를 전송하기 위해 목적지까지의 경로를 필요로 하게 되면 수행되는데 이는 목적지 노드로 향하는 RREQ(Route Request) 패킷을 Flooding 하고 목적지 노드로부터 RREP(Route Reply)패킷을 받는 과정을 포함한다. 이때 중간 노드는 RREQ패킷을 송신한 노드로 향하는 소스 노드까지의 역 경로를 설정하게 된다. 이 과정에서 중간 노드가 중복된 RREQ 패킷을 수신하는 경우가 발생하는데 이러한 중복된 RREQ 패킷은 삭제되게 된다. 중간 노드에서 목적지 노드로 향하는 유효한 경로를 유지하고 있는 경우 RREQ 패킷을 다시 Flooding 하는 대신 중간 노드에서 직접 소스 노드로 역 경로를 따라서 RREP 패킷을 전송하게 된다. 그러나 RREQ 패킷이 목적지 노드까지 전송이 이루어진 경우는 RREP 패킷을 RREQ 패킷의 Flooding 과정을 통해 설정된 역 경로를 통해 소스 노드로 전송하게 된다. 이렇게 소스 노드가 RREP 패킷을 수신함으로써 데이터 전송을 위한 경로설정이 이루어지게 된다. AOMDV의 경로 발견과정에서의 차이점은 다중경로 설정을 위해 중복된 RREQ 패킷의 수신과 목적지 노드에서 도착한 모든 RREQ 패킷에 대하여 RREP 패킷을 전송한다는 점이다[4]. 본 논문에서는 오직 순수 Flooding 기법을 가정하고 애드 혹 네트워크의 모든 링크는 양방향성(Bidirectional)을 가진다고 생각한다.

경로 유지는 RERR(Route Error) 패킷을 이용하여 이루어지며 AODV와 AOMDV 모두 동일한 방식으로 중간 노드가 이동하여 목적지 노드로 데이터를 전달할 수 없게 되면, 상황 스트림이 파괴된 이전의 노드는 새로운

순서번호와 ∞ 의 홑 카운트를 가지는 RERR 패킷을 전송하게 된다. 소스 노드에서 이런 RERR 패킷을 수신하게 되면 데이터 전송을 위해 경로가 계속 필요한 경우 소스 노드는 새로운 경로발견 과정으로 초기화 한다. 각 노드에서 사용되지 않는 경로는 타이머를 이용하여 삭제 되게 된다[3,4].

2.2 Sequence Number and Loop Freedom

순서번호(Sequence Number)는 AODV와 AOMDV 라우팅 프로토콜에 있어서 Loop-free 한 경로설정이 중요한 요소인데 이를 위해 경로 상의 모든 노드는 동일하게 증가하는 순서번호를 유지한다.

모든 노드는 각 목적지 노드에 대해 가장 높은 순서번호를 라우팅 테이블에 유지하는데 이를 목적지 순서번호(Destination Sequence Number)라 한다. 목적지 순서번호는 모든 라우팅 메시지에 포함이 되는데 이를 통해서 같은 목적지 노드에 대한 여러 개의 라우팅 메시지 중에서 가장 최근의 경로설정 정보를 선택할 수 있도록 해주는 메커니즘을 제공한다.

따라서 AODV와 AOMDV 라우팅 프로토콜은 유효한 경로에 대해 동일하게 증가하는 목적지 순서번호를 통하여 라우팅 Loop 경로를 방지할 수 있다[3,4].

3. 역 경로 정보와 Overhearing을 통한 다중경로 설정

본 논문에서는 AODV와는 달리 목적지 단말은 수신된 모든 RREQ에 대해 RREP 메시지를 전송하여 RREQ Flooding에 의해 생성된 다중의 역 경로가 버려지지 않고 효과적으로 다중경로 설정에 사용될 수 있도록 하고자 한다. 반면에 중간 단말들이 중복된 RREQ 메시지를 수신하고 처리하는 AOMDV와는 다르게 기존 AODV 처럼 처음 도착한 RREQ 메시지만을 처리하도록 함으로써 저비용 다중경로 설정이 수행될 수 있도록 한다.

그러나 역 경로 정보만을 이용하여 다중경로 설정시 발생하는 경로 수의 제한을 해결하기 위해 그림 1에서와 같이 목적지 노드에서 보내게 되는 RREP 패킷을 중간 노드에서 수신 하도록 하여 추가적인 경로를 획득할 수 있도록 한다. 그림 1에서와 같이 목적지 노드에서 중복된 RREQ 패킷을 수신하여 이에 대한 RREP 패킷을 보내게 되고 중간 노드에서는 RREP 패킷을 Overhearing 하도록 함으로써 다중 경로를 설정할 수 있게 된다. 또한 소스 노드에서는 중복된 RREP 패킷을 수신하여 현재 사용 가능한 경로의 수를 알 수 있게

된다. 이를 통하여 기존의 AODV에서 생성되는 역 경로 정보와 RREP 패킷 Overhearing을 통하여 추가적인 오버헤드를 필요로 하지 않고 다중경로를 설정할 수 있다.

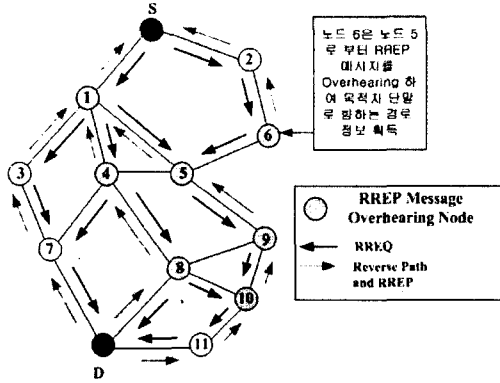


그림 1. RREP 메시지의 Overhearing

본 실험을 통하여 그림2와 같이 노드의 이동성의 커질수록 본 연구에서 제안하는 다중경로 설정 메커니즘이 더 작은 라우팅 오버헤드를 필요로 함을 보여준다.

6. 결론

본 연구에서는 기존의 다중경로 설정 라우팅 알고리즘에 비하여 기존 단일 경로 설정 라우팅 알고리즘의 메커니즘을 최소한으로 수정하여 저비용으로 다중경로 설정을 수행할 수 있음을 보여준다. 따라서 저비용 다중경로 설정 메커니즘을 통하여 애드 혹 네트워크에서 중간 노드의 이동으로 인한 잦은 경로 실패와 경로 설정 과정으로의 초기화를 줄임으로서 저비용의 신뢰성 있는 경로를 제공한다.

앞으로의 과제는 AODV와 AOMDV에서의 지역 경로 복구 방법을 개선하여 RREQ 패킷의 Flooding 범위를 제한함으로써 이로 인한 네트워크 자원의 낭비를 최소화하는 것을 앞으로의 해결 과제로서 고려하고 있다.

7. 참고 문헌

5. 실험 및 결과

5.1 실험 환경

본 논문에서는 제안된 메커니즘을 검증하기 위해 NS-2 시뮬레이터를 이용하여 비교 실험을 수행하였다. 실험에서 MAC 계층은 802.11을 사용하며 노드의 전송 범위는 250m 이고 네트워크 대역폭은 2 Mbps 로 설정 하였다. 그리고 노드의 수를 30개로 고정한 상태에서 600x600(m²)의 영역에서 400s 동안 시뮬레이션을 수행 하였다[5].

5.2 실험 결과

아래의 그림2는 본 연구에서 제안하는 다중경로 설정 메커니즘과 기존 다중경로 설정 라우팅 프로토콜인 AOMDV의 라우팅 오버헤드를 측정 한 것이다. 여기에서 라우팅 오버헤드는 데이터 패킷을 전송하기 위해서 어느 정도의 제어 메시지를 필요로 하는지를 나타내며 Pause Time은 노드의 이동성을 나타내며 그 값이 작을수록 노드의 이동성이 더욱 크다.

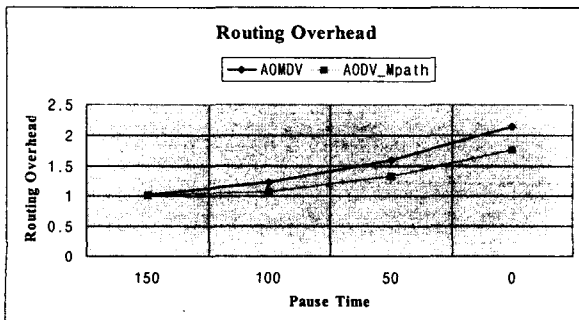


그림2 데이터 전송시 라우팅 오버헤드

[1] C.K Toh, "Ad Hoc Mobile Wireless Networks Protocols and Systems", Prentice Hall, 2002.
 [2] Smir R. Das, Charles E. Perkins, Elizabeth M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", IEEE Personal Communications, Volume:8 Issue:1, Feb 2001, page:16~28.
 [3] C. E. Perkins and E. M. Royer. "Ad Hoc On-Demand Distance Vector Routing.", In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pages 90-100, 1999
 [4] Mahesh K.Marina, Samir R.Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks," In procedins of the International Conference for Network Protocols(ICNP), 2001
 [5] K. Fall and K. Varadhan(Eds.), The ns Manual <http://www.isi.edu/nsnam/ns/ns-documentations.html>, 2002