

엿듣기를 이용한 Ad-hoc 네트워크 라우팅에서의 오버헤드 절감 아이디어

이동현
국가보안기술연구소
donghyun2u@etri.re.kr

Overhead Reduction Ideas based on Overhear in Ad-hoc Networks Routing

Dong-Hyun Lee
National Security Research Institute

요 약

Ad-hoc 네트워크의 효율적인 라우팅을 위해서는 라우팅 각 과정의 오버헤드를 최소화하여야 한다. 본 논문에서는 기존의 연구물들을 바탕으로 Ad-hoc 네트워크 라우팅에서 흔히 일어나는 주요 오버헤드 요소들을 살펴보고, 그 오버헤드를 절감시킬 수 있는 간단하고 효율적인 아이디어를 제시한다. 제안한 방법은 무선 네트워크의 특성을 활용하여 별도의 메시지 교환 없이 엿듣기를 통하여 라우팅 정보를 수집하는 특징을 가지고 있다. 이 엿듣기로 수집된 라우팅 정보를 바탕으로 플러딩 메시지 감소, 멀티 패스, 중앙 병목현상 분산을 구현하고, 이를 통해 라우팅 오버헤드를 크게 경감시킬 수 있을 것으로 기대한다.

1. 서론

Ad-hoc 네트워크는 이동성을 가진 단말들만으로 구성되는 네트워크로, 각 단말들은 데이터를 주고 받는 주체이면서 또한 라우터의 역할도 수행한다. Ad-hoc 네트워크는 일반적인 유/무선 네트워크와 비교하여 한정된 자원(배터리, 연산능력, 대역폭 등)을 사용하고, 일시적이며 동적으로 변화하는 네트워크 토폴로지를 가지며, 기본적으로 인프라 없이 구성된다는 주요 특징이 있다.

Ad-hoc 네트워크가 가지는 이러한 여러 가지 특수한 조건들은 네트워크의 효율적인 통신을 방해하며, 기존의 네트워크에서는 볼 수 없었던 새로운 문제점들을 발생시킨다. 그 결과 중 하나가 라우팅이 매우 까다롭다라는 것이고, 이에 대한 활발하고 다양한 연구가 진행 중이다.

본 논문에서는 Ad-hoc 네트워크 라우팅에서 오버헤드가 되는 주요 요소에는 무엇이 있는지 살펴보고, 이를 해결하기 위한 아이디어를 제시한다. 제안한 방법은 엿듣기[1]를 바탕으로 하고 있다. 엿듣기는 무선 네트워크의 특성을 활용하여 별도의 메시지 교환 없이 이웃 단말들의 송수신 메시지를 도청하는 것을 말한다. 이를 활용하면 각 단말들은 주위의 라우팅 정보를 수집할 수 있으며, 여기서는 플러딩 메시지 감소, 멀티 패스 구현, 중앙 병목현상 분산을 수행하는 간단하면서도 효율적인 아이디어를 제시한다.

각 장의 순서는 다음과 같다. 2장에서는 Ad-hoc 네트워크 라우팅에서 발생하는 오버헤드 요소에 대해서 살펴보고, 3장에서는 엿듣기 정보를 바탕으로

오버헤드를 적절히 감소시키는 아이디어를 제시한다. 끝으로 4장에서 결론을 맺는다.

2. 오버헤드 요소

효율적인 라우팅을 위해서는 라우팅 각 과정에서 생기는 오버헤드를 최소화하여야 한다. 이번 장에서는 기존의 연구물들을 바탕으로 Ad-hoc 네트워크 라우팅에서 일어나는 오버헤드 요소들을 살펴본다.

2.1. 플러딩

Ad-hoc 네트워크에서는 On-demand 방식의 라우팅이 선호된다. 이 방식의 라우팅이 가지는 문제점 중 대표적인 것은 경로 탐색 과정의 플러딩(flooding)으로 인한 오버헤드이다 [2]. 플러딩은 소스 단말이 목적지 단말까지의 패스를 찾기 위해서 이웃 단말들에게 RREQ(route request) 메시지를 멀티캐스트하고, 이를 전달 받은 이웃 단말들은 목적지 단말에 RREQ가 전달될 때까지 또다시 그 이웃 단말들에게 멀티캐스트하는 일을 반복하는 메커니즘을 말한다. 이는 중복적인 자원 낭비가 많은 방법이다. 따라서 경로 탐색 과정의 플러딩 메시지를 최적화하는 노력이 요구된다.

2.2. 빈번한 경로 재설정

앞서 2.1. 장의 플러딩 오버헤드는 Ad-hoc 네트워크의 동적인 네트워크 토폴로지 특성에 의해 더 심화된다. 즉, 경로가 자주 끊어짐으로 인해서 경로를 재설정해 주기

위한 플러딩이 반복되게 된다. 이것은 매우 비효율적이다. 멀티 패스 라우팅 [3,4,5]은 이에 대한 대안 중 하나이다. 싱글 패스 라우팅과는 달리 여러 개의 대안 패스를 가짐으로써, 끊어진 패스를 복구하는 것이 가능하여 경로를 재설정하기 위해 경로 탐색 과정의 플러딩이 반복되는 것을 막아준다. 또한 멀티 패스 라우팅은 로드를 분산하거나 동시에 여러 패스를 활용하는 등 좀 더 유연성을 가지는 장점이 있다.

2.3. 중앙 병목현상

Ad-hoc 네트워크의 동적인 토폴로지 변화에 대응하기 위해 정적인 라우팅 알고리즘은 한계를 가진다. 따라서 라우트 패스도 동적으로 변화할 필요가 있다. 그러나 동적인 라우트 패스 관리는 그에 상응하는 오버헤드를 수반하므로 매우 신중히 접근할 필요가 있다.

토폴로지에 민감한 오버헤드 중 하나는 Ad-hoc 네트워크 덩어리들의 중심부에 생기는 오버헤드 병목현상이다. 일반적으로 라우팅 프로토콜은 거리나 대기시간이 짧은 경로를 선택하기 때문에, 네트워크 덩어리의 중심에 라우팅 패스가 집중되는 경향이 있다 [6]. 그 결과 중앙의 단말들에게 오버헤드가 집중되고 자원을 급격히 소모시키며, 최종적으로는 관계된 네트워크 성능을 동시에 떨어뜨리게 된다. 그러므로 한쪽에 몰리는 라우팅 경로를 분산시켜줄 필요가 있다.

- [단계3] 단말 B는 단말 A로부터의 거리 X를 얻는다.
- [단계3] 단말 B는 $2 \cdot (R-X) / (\text{전파속도}) + (\text{연산시간})$ 만큼의 시간동안 이 RREQ에 대한 엿듣기를 수행한다.
- [단계4] 단말 B는 엿듣기로 플러딩 메시지가 감지되면 자신은 플러딩을 수행하지 아니하고, 시간이 초과될 때까지 플러딩 메시지가 없으면 자신이 플러딩에 참여한다.

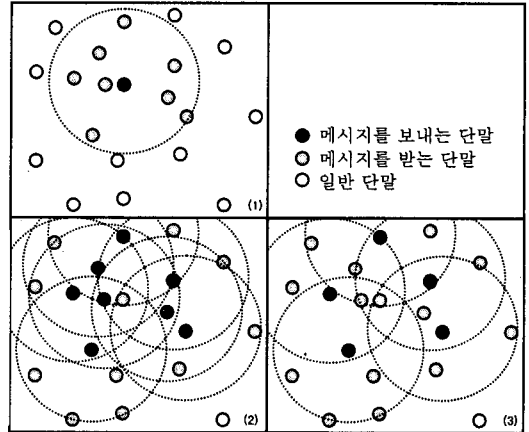


그림 1. 플러딩 오버헤드 감소 아이디어

이를 통해 외곽의 단말들만이 플러딩에 참여하는 것이 가능하게 된다. 하지만 이 방법은 아직 구체화가 필요하다. 통신의 시간차 계산, 각 단말 별 통신 환경 등을 고려한 일반화된 모델이 요구된다. 그리고 RREQ 메시지가 도달되는 단말의 수를 최대화하기 위해서 단말이 얼마나 기다려야 되는지에 관한 최적화도 더 고려할 대상이다. 또 이 방법은 사이사이에 RREQ 메시지를 받지 못하는 단말이 생기기 때문에, 원래의 플러딩을 통해서 발견되었던 목적지 단말이 이 방법을 통해서 RREQ를 받을 수 없는 경우도 발생할 수 있다. 따라서 첫 번째 시도 실패 시 두 번째 시도에서는 기존 플러딩 방식으로 전환할 수 있게 하는 등 보완도 생각해 볼 수 있다.

3. 엿듣기를 이용한 오버헤드 절감 아이디어

무선 통신의 특성상 각 단말은 이웃 단말이 전송시킨 메시지를 메시지 전달 유효반경 안에 있다면 무차별적으로 엿들을 수 있다. 그러므로 별도의 메시지 교환 없이 각 단말은 주변에서 일어나는 라우팅 정보를 수집할 수 있게 된다. 제안한 방법은 바로 이 엿듣기 [1]에 기반한다.

엿들은 라우팅 정보를 활용하면 많은 오버헤드 감소를 유도할 수 있다. 본 장에서는 2장에서 소개한 각 오버헤드를 경감시킬 수 있는 엿듣기를 활용한 오버헤드 절감 아이디어를 제시하고 고찰한다.

3.1. 플러딩 오버헤드 감소

기본 아이디어는 그림1-(1)과 같은 RREQ 멀티캐스트가 있을 때 그림 1-(2)와 같이 RREQ 메시지를 받은 모든 이웃 단말들이 플러딩에 참여하는 대신에 그림 1-(3)과 같이 메시지 도달 반경의 가장자리에 있는 단말들만이 플러딩을 이어나감으로써 메시지를 줄이는 것이다.

그 과정은 다음과 같다.

[가정1] RREQ 메시지가 도착했을 때 그 메시지를 전달한 단말로부터의 시간차(즉, 거리)를 계산할 수 있다.

[가정2] 각 단말의 전파반경은 R로 동일하다.

[단계1] 단말 A가 RREQ 메시지를 송신한다.

[단계2] 단말 B가 이 메시지를 수신한다.

3.2. 멀티 패스 라우팅

엿듣기를 통해 이웃의 라우팅 메시지를 도청함으로써 각 단말은 메인 패스 외에 다른 대안 패스에 대한 라우팅 정보를 수집하게 된다. 이를 통해 멀티 패스 라우팅을 구현한다.

라우팅 경로 탐색 과정에서 다음을 수행한다.

[단계1] 경로 탐색에 참여하는 각 단말은 경로 탐색 과정 중 엿듣기로 얻은 라우팅에 대한 정보 중 자신과 n개의 홉만큼 떨어진 단말까지의 패스 정보를 임시로 저장한다.

[단계2] 메인 패스가 설정된다.

[단계3] 메인 패스에 속하는 각 단말은 임시로 저장한

경로 정보 중에서 메인 패스에서 자신과 n개의 홉만큼 떨어진 단말에 대한 대안 패스를 보관한다.

만일 메인 패스가 손상되었을 경우 끊어짐을 탐지한 단말은 다음을 수행한다.

[단계4] 자신이 가진 다른 대안 패스를 통해 n 개의 홉만큼 떨어진 단말과 통신한다.

[단계5] 통신에 성공하면 패스가 바뀌었음을 통보하고 통신을 이어간다. 통신에 실패하면 바로 이전 단말에 패스가 끊어짐을 알린다.

[단계6] 이전 단말은 [단계4]부터 같은 과정을 수행하며, 이는 대안 패스를 찾거나 더 이상 대안 패스를 찾을 수 없을 때까지 반복된다.

각각의 중간 단말이 패스를 기억하는 것이 좋다는 것은 [3]의 결과를 이용하였다. 이 방법은 n의 의미와 값을 결정하기 위한 실험이 필요하다. 또한 패스가 재설정된 이후에 재설정예 새로 참여한 단말은 멀티 패스를 보유하고 있지 않다는 단점을 보완해야 한다.

3.3. 중앙 병목현상 분산

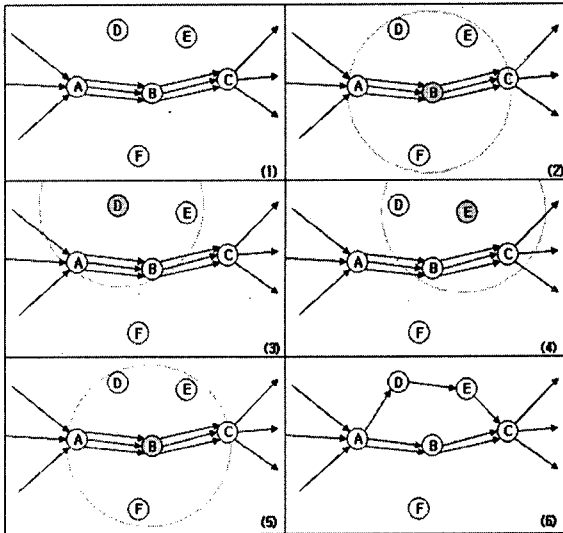


그림 2. 중앙 병목현상 분산 아이디어

아이디어는 후처리 방법과 전처리 방법 두 가지이다. 후처리 방법은 단말의 자원량에 근거하여 시간당 문턱치(T1) 이상의 라우팅 자원을 소모하게 되면 이후의 라우팅 요청을 거부하는 방법이다. 그리고 선처리 방법은 마찬가지로 문턱치(T2)를 두어 시간당 라우팅 자원 소모량이 이 문턱치(T2)를 넘으면 주변 이웃 노드에 도움을 요청해서, 라우팅 패스를 넘기는 방법이다.

선처리 방법을 좀 더 자세히 살펴보면 다음과 같다.

[단계1] 단말 A, B, C가 메인 경로로써 통신하고 있다. 단말 B가 병목지점이다. 그림 2-(1).

[단계2] 단말 B가 문턱치(T2)에 도달하여 주위 이웃 단말들에 도움 요청 메시지를 발송한다. 이 메시지는 메인 경로 A-B-C에 대한 정보를 담고

있다. 그림 2-(2).

[단계3] 메인 경로에 참여하지 않는 이웃 단말 D, E, F는 자신이 평소에 A 또는 C의 메시지를 엿듣기로 감지하고 있었던 경우 이를 주위에 알린다. 그림 2-(3)의 경우 D가 먼저 이를 수행하였다. 이 메시지는 경로 A-D에 대한 정보를 담고 있다. 마찬가지로 단말 E도 그림 2-(4)와 같이 경로 E-C에 대한 정보를 주위에 알린다.

[단계4] 단말 B는 이 정보를 수집하여 대안 패스를 결정하고, 메시지를 통해 이를 설정한다. 그림 2-(5),(6)

이를 통해 병목 현상이 생긴 경로를 분산시킬 수 있다. 하지만 이 알고리즘 역시 문턱치와 자원량 계산에 대한 정확한 정의와 메시지에 담기는 정보의 포맷 등 좀더 구체화가 필요하다.

4. 결론

본 논문에서는 이동 Ad-hoc 네트워크에서의 주요 오버헤드 요소를 살펴보고, 이를 감소시키기 위한 엿듣기에 기반한 몇 가지 아이디어를 제시하였다. 제안한 아이디어를 통해 라우팅 오버헤드를 효율적으로 경감시킬 수 있을 것으로 기대하지만, 제안한 아이디어는 아직 완전한 알고리즘이 아니며 실험을 통한 검증, 보완이 필요하다. 향후 연구를 통해 구현 및 그 유효성을 검증해 볼 생각이다.

5. 참고문헌

[1] Sung-Ju Lee and Mario Gerla, "AODV-BR: Backup Routing in Ad hoc Networks", Wireless Communications and Networking Conference, Vol.3, p.1311-1316, 2000.9.
 [2] Mahesh K. Marina, Samir R. Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks", International Conference on Network Protocols, p.14-23, 2001.11.
 [3] Asis Nasipuri, Robert Castaneda, Samir R. Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks", Mobile Networks and Applications 6, 339-349, 2001
 [4] I-Shyan Hwang, Cheng-Ching Yeh, Chiung-Ying Wang, "Link Stability, Loading Balance and Power Control based Multi-Path Routing(SBPMR) Algorithm in Ad Hoc Wireless Networks", International Conference on Telecommunications, Vol.1, p.406-413, 2003.3.
 [5] Sangkyung Kim, Wonjong Noh, Sunshin An, "Multi-path Ad Hoc Routing Considering Path Redundancy", IEEE International Symposium on Computers and Communication (ISCC'03), Vol.1, p.45-50, 2003.6.
 [6] Ganjali, Y., Keshavarzian, A., "Load balancing in ad hoc networks: single-path routing vs. multi-path routing", INFOCOM 2004. Vol. 2, 7-11 p.1120-1125, 2004.5.