

Ad-hoc Network에서 Dual-Path와 Local-Repair를 이용한 On-demand Routing에 관한 연구

고관옥* 송주석
연세대학교 컴퓨터과학 산업시스템공학과
{kwanok*, jssong}@emerald.yonsei.ac.kr

A Study of On-demand Routing with using Dual-Path and Local-Repair in Ad-hoc Networks

Kwanok Koh* Jooseok Song
Department of Computer Science, Yonsei University

요 약

Ad-hoc Network는 Mobile Node들 간에 Multi-hop 무선 링크로 구성되는 Network을 일컫는 말이며, 동시에 Network를 통제하는 Infrastructure없이 Node들 간의 상호 통신에 의해서 Network이 구성되기도 하고, Node들이 이동하거나 환경적인 장애에 의해서 일시적으로 Network이 구성되지 않기도 한다. 이 논문에서는 대표적인 On-demand Routing Algorithm인 DSR를 이용하여, 두 개의 경로를 유지하여 Primary Path에 문제가 발생하는 경우, Secondary Path가 Primary Path로 전환되어 Data를 전송하고, 이 전 Primary Path에 대하여 지역적으로 복구(Repair)를 수행하고, 설정된 Secondary Path에 대해서도 특정한 조건에서 복구작업을 수행함으로써 Ad-hoc Network에서 경로를 찾고 설정하는데 필요한 Routing Overhead를 줄이고 Ad-hoc Network의 특성상 반드시 보완하여야 하는 전송 Route에 대한 Robustness를 강화하는 방법이다.

1. 서 론

Ad-hoc Network는 Mobile Node들 간에 Multi-hop 무선 링크로 구성되는 Network을 일컫는 말이며, 동시에 중앙에서 Network를 통제하는 Infrastructure없이 Node들 간의 상호 통신에 의해서 Network이 구성되기도 하고, Node들이 이동하거나 환경적인 장애에 의해서 일시적으로 Network이 구성되지 않기도 하는 등 Network Topology가 시간에 따라서 계속해서 변화하는 Network을 말한다. 이러한 Network 환경에서 Source Node가 Data를 Destination Node로 전송하기 위해서는 Network 상에서 Source에서 Destination까지의 경로가 존재하는가를 우선적으로 찾아야 한다. 이 경우 가장 일반적인 방법이 Flooding을 하는 것으로 Source Node는 자신 주변의 전 Node들에게 Destination Node까지의 경로 정보가 있는지 Query하고, 중간 Node에서 Destination 정보를 Source Node에게 알려 주거나, 그렇지 않은 경우는 Destination Node까지 Query 정보가 전달되고, Destination Node가 이 Query에 답변함으로써 경로가 구성된다. 가장 일반적으로 연구되고 있는 방법에는 모든 경로 정보를 Table을 유지하여 관리하는 Table-Driven 방식과 경로 정보가 필요한 경우에 경로를 찾는 On-demand 방식이 있다.

이 논문에서는 Source Routing 방식 중에서 DSR [1] (Dynamic Source Routing)을 이용하여 두 개의 경로를

유지하여 Primary Path에 문제가 발생하는 경우, Secondary Path를 이용하여 전송하면서 Primary Path에 대하여 지역적으로 복구를 수행하여 Ad-hoc Network의 Routing Overhead를 줄이고 Robustness를 강화하는 방법이다.

이 논문의 2장 1절에서는 DSR Algorithm을 간단히 소개하고, 2절에서는 개선된 형태의 Algorithm을 설명하고 3절에서는 개선된 Algorithm을 수식을 통하여 원래의 DSR Algorithm과의 성능을 비교 설명하고, 3장 결론으로 구성되었다.

2. 본 론

2.1 Dynamic Source Routing(DSR)의 개요

DSR은 Source Routing을 사용하는데, 이 기법은 데이터를 보내고자 하는 Source Node에서 어떠한 경로를 통하여 Destination Node까지 데이터를 전송할 것인가를 결정하는 방법이다. DSR에는 기본적으로 Route Discovery와 Route Maintenance 과정을 거친다. 여기서는 기본 개념만을 설명하고자 한다.

2.1.1 Route Discovery

Route Recovery과정은 Destination Node까지의 경로를 찾고자 하는 Source Node는 Query Packet을 주변의 모든 Node에게 Flooding한다. 이 Query Packet을 수신한 Node들은 Destination Node의 정보를 알고 있는 경우는

그 정보를 Reply Packet에 실어서 Source Node에게 전송하고 그렇지 않은 경우 다시 자신의 주변 Node에게 Flooding한다. 또한 Query Packet에는 Sequence Number가 있어서 중복 수신되는 경우는 Drop된다. Query Packet이 Destination Node에 도착하면, Destination Node는 Reply Packet을 Query Packet의 경로 정보를 이용하여 Source Node로 전송한다. Reply Packet을 수신한 Source Node는 이 경로에 대한 정보를 Route Cache에 저장하여 Data전송에 사용한다. 이 과정에서 Ad-hoc Network의 계속적인 Topology 변화로 인하여 경로가 유실되는 경우에 계속적으로 경로 탐색을 위한 Query Packet을 Flooding해야 하기 때문에, Routing Overhead를 발생시키는 요인이 된다.

2.1.2 Route Maintenance

Packet을 전송할 때 각 Node는 Packet 전송에 대한 책임을 진다. 다음 Node의 문제를 판별하는 방법은 ACK를 이용하는데, 첫째로는 IEEE 802.11의 MAC에 구현된 방법을 이용하는 것과 둘째는 데이터를 보낸 Node가 다음 Node에서 그 다음 Node 전송되는 Packet을 수신하는 Passive acknowledgement 방법이 있고, 위의 두 가지 방법을 사용할 수 없는 경우는 DSR의 기능으로 Packet을 수신하는 Node에게 수신여부를 물어봐서 응답을 기다리게 된다. 위의 세가지 방법으로도 ACK를 수신하지 못하는 경우 Link에 문제가 있다고 판단하게 된다. 그 다음 각 Node는 Route Cache에서 문제가 있는 Node가 포함된 경로를 삭제하고, 이전 Node들에게 Route Error Packet을 전송한다. 그리고 Source Node는 Route Cache에서 Destination Node에 대한 다른 경로를 이용하여 전송할 수 있는 경우는 그 경로를 이용하여 전송하고 그렇지 않은 경우는 Route Discovery를 다시 수행해야 한다.

2.2 이중경로와 지역수리를 이용한 Algorithm

2.2.1 이중경로 설정

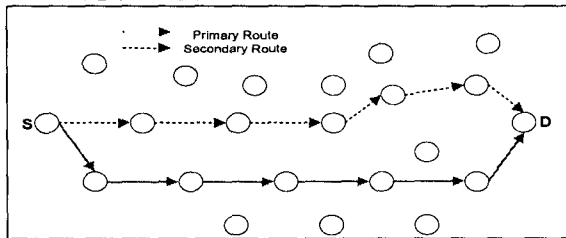


그림 1 Dual-path 형성 개념도

Source Node와 Destination Node간에 여러 개의 경로를 유지하는 Algorithm[2, 3]에서는 3개 이상의 다중 경로를 Route Cache에 유지하고 관리한다. 하지만 이 논문에서는 단지 두 개의 Primary Path와 Secondary Path만을 Route Cache에 유지한다. 두 개의 경로를 설정하기 위해서는 Destination Node는 자신에게 도착한 Query Packet 중에서 서로 Disjoint한 두 개의 경로를 선택하고 이 경로를 통하여 Reply Packet을 Source Node로 전송한다. Disjoint한 경로만을 선택하는 이유는 하나의 Link에 문제가 발생하는 경우 다른 Link에 영향을 주지 않기 위해서이

다. 또한 중간 Node들이 Route Reply를 하는 경우는 Source에 도착하는 순서로 두 개의 Route를 선택하게 된다.

그래서 Primary Path에 문제가 발생하는 경우, Route Error Packet을 수신한 Source Node는 Secondary Path를 이용하여 계속해서 전송하게 된다. 두 개의 Route에 모두 문제가 발생하여 데이터를 더 이상 전송할 수가 없는 경우는 DSR의 Route Discovery를 다시 수행하게 된다.

2.2.2 Secondary Path 관리

Primary Path로 통신하는 동안에도 Network Topology는 계속해서 변하기 때문에, 일정 시간이 경과 후, 처음에 설정된 Secondary Path가 유효한지를 Source Node에서는 알 수가 없다. 이러한 단점을 보완하기 위하여, Primary Path로 통신하는 중에도 Secondary Path를 유지하기 위한 관리기법이 필요하다.

이 논문에서는 Secondary Path를 관리하기 위해서 Ping을 사용한다. Primary Path로 통신하는 동안 주기적으로 Ping 신호를 Secondary Path를 통하여 전송한다. Ping을 수행하기 위한 조건은 Route Cache에 있는 다음 Node의 활동이 일정 시간 이상 감지 되지 않을 때 시작한다. 즉, 다음 Node가 이동하여 전파 도달 거리 이상으로 멀어진 경우와 전송할 데이터가 없는 경우로 아무런 동작을 하지 않는 경우가 해당된다. 이 방법을 통하여 Control Message의 수를 줄일 수 있다. Ping에 대한 응답은 명시적으로 전달되지는 않는다. 하지만 다음 Node에서 그 다음 Node로 Ping을 보내는 것을 이전 Node가 수신함으로써 다음 Node까지의 경로가 설정되어 있음을 묵시적으로 알게 된다. 다음 Node가 전송하는 Ping을 수신하지 못하는 경우 잠정적으로 해당 Node에 문제가 있다고 판단하고, 연속해서 세 번의 Ping 신호를 수신하지 못하는 경우는 Secondary Path에 대한 Local Repair를 수행한다.

2.2.3 Local Repair

Local Repair는 기본적으로 Node의 Spatial Locality를 가정한다. 즉, 어떤 Mobile Node든지 일반적인 이동환경을 고려한다면, 매우 빠르게 이동하지 않는 한, 이 Node를 포함하고 있는 Path에 문제가 발생한 경우라도, 문제가 되는 Node는 공간적으로 매우 근접한 위치에 Node가 있을 가능성이 높기 때문에, 문제가 되는 Node 주변 Node들에 한하여 Flooding을 수행함으로써 Routing Overhead를 감소시킬 수 있다. 즉 DSR에서는 Route Error Packet을 수신한 경우 문제가 있는 Node가 포함된 Routing 정보를 Route Cache에서 삭제하는 반면에, 이 논문에 사용된 방법에서는 Routing Cache의 정보를 이용하여 Data Packet이 정상적으로 전달되었던 마지막 Node, 즉, Route Error Packet을 전송한 Node가 Local Repair를 수행하게 된다. 이러한 Local Repair는 문제가 생긴 Path가 복구되면 종료된다. 문제의 Path가 복구되지 않은 상태에서 Secondary Path에도 문제가 발생한 경우에는 Local Repair를 중단하고 Route Discovery 단계를 다시 시작한다. 여기서 Query를 수행할 Hop의 범위를 제한함으로써 불필요한 Routing Overhead를 줄일 수 있다. Local

Repair에서 Hop-limit = k라고 한다면, 초기값은 k=1에서 시작하고, Query 수행 후에도 Path가 복구되지 않으면, k를 1씩 증가시키고 최대 값은 k=4로 제한한다. 그리고 동일한 Node에서 Link에 문제가 발생하는 경우에는 k-1 값을 이용한다. 예를 들면, 이전 단계에서 사용된 k=3인 경우, k=1에서 증가하는 것이 아니라 k=2를 사용한다. 이 경우 Spatial Locality를 이용할 수 있으며, Routing Overhead가 증가하는 것을 방지할 수 있다.

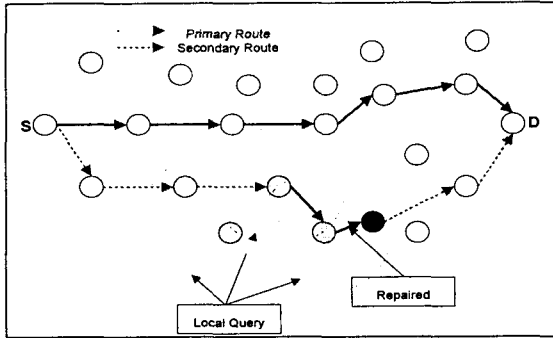


그림 2 Local Repair 과정(k=2)

$$Flooding = \sum_{i=1}^n \frac{i-1}{2} \quad (4)$$

의 Flooding으로 문제가 생긴 Path를 복원하게 된다. 하지만 n이 4보다 큰 경우에는 그림 6에서처럼 Local Repair에서 발생하는 Routing Packet의 수가 기존 DSR에서 발생하는 수보다 증가하게 되어 효율성이 떨어진다. 즉 Spatial Locality를 이용하기 위해서는 Hop_limit(k)를 4로 제한할 필요가 있다.

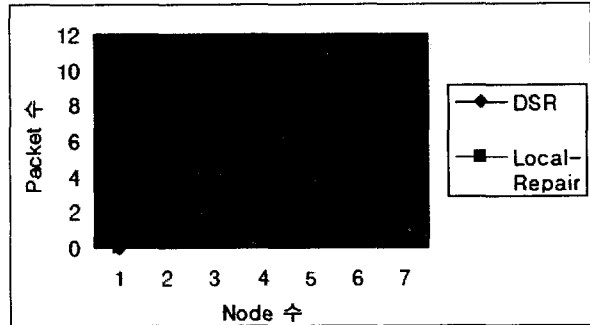


그림 3 DSR과 Local-Repair Routing Packet 수 비교

2.3 성능분석

2.3.1 Dual-path Analysis

기본적인 가정사항으로 이미 Route Discovery 단계가 종료되고, 경로가 설정되었다고 할 때 특정 시간 동안 Path에 문제가 발생할 가능성은 Poisson Random Variable[4]로 볼 수 있다.

$$P_k = \frac{\alpha^k}{k!} e^{-\alpha}, k = 0, 1, 2, \dots \quad (1)$$

여기서 k는 발생 빈도를 나타내며, α 는 해당 시간 간격 동안의 평균 발생 빈도를 의미한다. 여기서 Data Packet의 전송될 확률은

$$p_{data} = 1 - p_k \quad (2)$$

가 된다.

기존의 DSR Algorithm에서는 Path에 문제가 발생하는 경우 Route Discovery를 수행해야 하기 때문에 식 (1)의 확률로 발생하지만, 이 논문에서 제안된 방식으로 두 개의 Path를 유지하는 경우, 각각의 Path에 문제가 생길 확률은 동일하지만, 주어진 시간에서 Data Packet이 전송될 확률은

$$p_{data} = 1 - \frac{1}{2} p_k \quad (3)$$

이 된다.

2.3.2 Local Repair Analysis

DSR에서는 경로에 문제가 발생할 때 Flooding을 수행하는데 아래 하나의 Path가 n개의 Node로 구성되고 가정할 때 Source Node가 Destination Node를 찾기 위해서는 최악의 경우 n-1번의 Flooding이 필요하다. 하지만 이 논문에서 제안된 Local Repair 방법을 사용하면

2.4 향후 계획

이 논문에서 제안된 방법의 적절성을 검증하기 위해서는 ns-2를 이용하여, Routing Overhead, End-to-end Delay, Packet Delivered Ratio 등을 통하여 두 방법 간의 성능비교를 위한 시뮬레이션을 수행하고자 한다.

3. 결론

Ad-hoc Network은 중앙 통제적인 Network이 아니며 Mobile Node의 이동성에 따라 시시각각으로 변화하는 Network Topology로 인하여, Routing Overhead가 많이 발생하고, 또한 Network의 Robustness를 유지하기 어렵다. 하지만, Dual-path와 Local-Repair를 이용하는 경우 Routing Overhead를 감소시킬 수 있으며, Network의 Robustness를 보장할 수 있다.

4. 참고문헌

- [1] Josh Broch, David Johnson, and David Maltz. The dynamic source routing protocol for mobile ad hoc networks. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>, Apr 2003. IETF Internet Draft.
- [2] Asis Nasipuri and Samir R. Das, On-Demand Multipath Routing for Mobile Ad Hoc Networks.
- [3] Marc Pearlman, Zygmunt Haas, Peter Sholander, Siamak S. Tabrizi, Alternative Path Routing in Mobile Ad Hoc Networks, In proceedings of IEEE MILCOM, 2002.
- [4] Alberto Leon-Garcia, Probability and Random Processes for Electrical Engineering, 2nd ed, pages 106-110, 1994