韓國컴퓨터情報學會 論文誌
第14卷 第5號, 2009. 5.

2009-14-5-3-5

노드의 악의적 행위패턴 및 신뢰수준 기반의 MANET Secure 라우팅 방안

박 성 승*, 박 건 우**, 류 근 호***, 이 상 훈****

A Secure Routing Protocol in MANET based on Malicious behavior Pattern of Node and Trust Level

Seong Seung Park *, Gun Woo Park **, Keun Ho Ryu ***, Sang-Hoon Lee ****

요 약

최근 MANET(Mobile Ad-Hoc Network)에서 보안요소를 추가한 라우팅 연구가 활발히 진행되어왔다. 그러나 기존 연구들은 secure 라우팅 또는 패킷 자체에 대한 악의적인 행위 탐지 중 어느 한 측면에 대해서만 연구가 되어왔다. 본 논문에서는 패킷 자체에 대한 악의적인 행위 및 라우팅 측면에서 보안 요소를 모두 고려한 SRPPnT(A Secure Routing Protocol in MANET based on Malicious Pattern of Node and Trust Level)를 제안한다. SRPPnT는 악의적인 행위가 이루어진 노드를 확인하여 각 노드에 대한 신뢰수준을 측정 후, 획득한 각 노드의 신뢰수준에 따라 라우팅 경로를 설정함으로써 패킷 및 라우팅 경로 설정에 대해 이루어질 수 있는 악의적인 행위를 효율적으로 대응할 수 있다. SRPPnT는 AODV(Ad-Hoc On-Demand Distance Vector Routing)를 기반으로 하였다. NS-2 네트워크 시뮬레이션 결과를 통해, 제안된 SRPPnT는 기존 프로토콜보다 네트워크 부하를 감소시킨 상태에서 악의적인 노드의 보다 정확하고 신속한 식별과 secure한 라우팅이 이루어짐을 확인하였다.

Abstract

In MANET(Mobile Ad-Hoc Network), providing security to routing has been a significant issue recently. Existing studies, however, focused on either of secure routing or packet itself where malicious operations occur. In this paper, we propose SRPPnT(A Secure Routing Protocol in MANET based on Malicious Pattern of Node and Trust Level) that consider both malicious

• 제1저자 : 박성승
• 투고일 : 2009. 03. 18, 심사일 : 2009. 04. 02, 게재확정일 : 2009. 04. 11.
* 충북대학교 전자계산학과 박사과정 ** 국방대학교 관리대학원 전산정보학과 박사과정
*** 충북대학교 전기전자/컴퓨터공학부 교수 **** 국방대학교 관리대학원 전산정보학과 교수
※ 이 논문은 2009년도 교육과학기술부(지역거점연구단 육성사업 / 충북BIT연구중심대학 육성사업단) 지원 및 한국과학재단의 지원을 받아 수행된 연구임(No. R11-2008-014-02002-0)

behavior on packet and secure routing. SRPPnT is identify the node where malicious activities occur for a specific time to compose trust levels for each node, and then to set up a routing path according to the trust level obtained. Therefore, SRPPnT is able to make efficient countermeasures against malicious operations. SRPPnT is based on AODV(Ad-Hoc On-Demand Distance Vector Routing). The proposed SRPPnT, from results of the NS-2 network simulation, shows a more prompt and accurate finding of malicious nodes than previous protocols did, under the condition of decreased load of networks and route more securely.

▸ Keyword : 악의적인 노드(Malicious Node), 악의적 행위패턴(Malicious Behavior Pattern), 신뢰수준(Trust Level), 신뢰수준 임계치(Trust_Threshold), Secure 라우팅(Secure Routing)

Ⅰ. 서 론

MANET은 노드가 신뢰받는 인증기관을 통해 인증 받는 형식이 아니기 때문에, 멀티 홉 방식에 의해 라우팅을 수행할 경우 악의적인 중간노드에 의해 데이터의 무결성 및 기밀성 문제가 발생할 수 있다. 특히, 매체를 신뢰할 수 없는 상황에서 암호를 사용하므로, 암호키에 크게 의존하게 된다. 또한, 기지국이나 AP (Access Point)와 같은 하부구조가 없고 노드들의 이동성과 무선채널의 상태 변화에 따라 네트워크의 토폴로지가 매우 자주 변화한다. 이와 같은 특징 때문에 보안문제가 확실히 해 된다 할지라도 컴퓨팅 문제가 발생된다거나 노드와 네트워크 전체에 심각한 부하를 줄 수도 있게 된다[1]. 따라서 보안측면과 네트워크 효율성 측면에 모두 적합한 알고리즘 구현이 필요하다.

지금까지 MANET에서 라우팅 공격이나 패킷 전송 시 발생 할 수 있는 여러 형태의 악의적인 행위를 효율적으로 탐지하고 대비하기 위한 많은 보안대책이 연구되어 왔다. 하지만 대부분의 연구들은 네트워크의 컴퓨팅 보다는 보안측면에 중점을 두고 있기 때문에 네트워크 오버헤드가 많이 발생한다. 또한 패킷을 버리거나 변경, 거짓 신고 등의 패킷 전달 시 공격에 대한 대책[2, 3, 4, 5] 또는 라우팅 경로에 대한 공격에 대한 대책[6, 7, 8, 9] 중 어느 한 측면에만 중점을 두고 있기 때문에 보다 정확하고 효율적으로 악의적인 행위에 대해 대비 할 수 없었다.

본 논문에서는 악의적인 행위의 빈도수에 의한 각 노드의 신뢰수준을 결정하고 해당 정보를 이용하여 라우팅 경로를 설정함으로써 악의적인 행위에 효율적으로 대응할 수 있는 프로토콜인 SRPPnT(A Secure Routing Protocol based on Malicious Pattern of Node and Trust Level in MANET)를 제안한다.

2장에서는 지금까지 연구된 MANET에서 안전한 라우팅 방법에 대한 관련연구에 대해 살펴보고, 3장에서는 제안하는 프로토콜인 SRPPnT에 대해 자세히 알아보고, 4장에서는 성능평가를 통해 SRPPnT의 효율성을 증명해 보이며, 마지막 5장에서는 결론을 제시한다.

Ⅱ. 관련 연구

앞서 언급한 것처럼 Mobile Ad-Hoc Network에서는 서로 다른 목적을 가진 사용자들이 광범위한 범위에 걸친 연결을 보장하며 그들 장치에 대한 자원을 공유하게 된다. 이러한 네트워크에서는 모든 노드들이 자신의 이익을 우선으로 하기 때문에 이기적인 행동들은 여러 가지 문제점이 발생할 수 있다. 이렇듯 이기적인 노드들로 인해 네트워크 전체에 치명적인 영향을 미칠 수 있기 때문에 이들을 관리하는 메커니즘과 알고리즘이 필요하다. 지금까지 연구된 대표적인 이기적인 노드 관리기법에 대해 알아보고, 장·단점을 분석해 본다.

1. Watchdog

Watchdog(경비견)은 부정한 노드를 탐지하는 방법 중의 하나로 Watchdog은 패킷 전달을 거부하는 노드를 감지하는데 사용되며[10], 메커니즘은 네트워크의 모든 노드들이 주변에 있는 노드들을 감시함으로써 이기적인 노드를 탐지하는 방법에 기반을 두고 있다[11].

즉, 각 노드들은 버퍼를 가지고 있으면서 데이터를 전송할 때마다 전송하기 이전에 관련 정보를 버퍼에 저장해 두고 전송 후 올바르게 다음노드에게 전송했는지 여부를 엿들어 버퍼에 저장해 둔 내용과 일치여부를 확인한다. 버퍼에 저장해 둔 내용과 엿들은 내용이 일치하면, 올바르게 다음 노드로 데이터 패킷을 전달한 것으로 판단하고 버퍼의 내용을 삭제하게

된다. 이와 같은 방법으로 다음 노드의 전송행위를 감시하며 이기적인 노드를 관리하게 된다. 그러나 이 방식은 협조하길 거부하는 악의적인 노드를 처벌하지 않는다는 문제점이 있다. 게다가 오히려 그 악의적인 노드가 다른 노드들을 위해 메시지를 전달해야 하는 부담을 덜어 줄 수 있다.

그림 1은 Watchdog의 이기적 노드 탐지 방법을 보여주고 있는데 소스 노드 S로부터 패킷을 받은 노드 N1은 노드 N4에게 그 패킷을 전달한다. 그러나 여기에서 자신의 역할이 끝나는 것이 아니라, 노드 N4가 목적지 노드 D에게 제대로 패킷을 전달하는지 까지도 감시해야 하는 것이다. 이렇게 모든 노드에 존재하는 Watchdog가 자신이 패킷을 전달한 노드의 다음 행동까지 감시함으로써 이기적인 노드를 탐지하게 된다. 이러한 방법으로 이기적인 노드가 탐지되면 각 노드의 pathrater(경로 안내자)는 그 사실을 반영하여 그 이기적 노드를 제외한 안전한 경로를 제공한다.

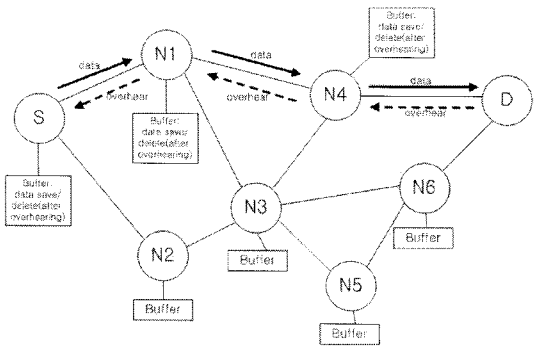


그림 1. Watchdog의 이기적인 노드 관리기법
Fig. 1. Selfish node management method of Watchdog

그림 1은 Watchdog의 이기적 노드 탐지 방법을 보여주고 있는데 소스 노드 S로부터 패킷을 받은 노드 N1은 노드 N4에게 그 패킷을 전달한다. 그러나 여기에서 자신의 역할이 끝나는 것이 아니라, 노드 N4가 목적지 노드 D에게 제대로 패킷을 전달하는지 까지도 감시해야 하는 것이다. 이렇게 모든 노드에 존재하는 Watchdog가 자신이 패킷을 전달한 노드의 다음 행동까지 감시함으로써 이기적인 노드를 탐지하게 된다. 이러한 방법으로 이기적인 노드가 탐지되면 각 노드의 pastorate(경로 안내자)는 그 사실을 반영하여 그 이기적 노드를 제외한 안전한 경로를 제공한다.

하지만 이 메커니즘에는 몇 가지 문제점이 있다. 우선 노드 N1이 노드 N4가 정상노드임에도 불구하고 이기적 노드라고 거짓으로 신고를 해도 소스노드 S로서는 노드 N1의 말을 무조건 믿을 수밖에 없다는 점이다. 또 다른 경우, 노드 N1과 노드 N4가 모두 악의적인 노드인 경우, 노드 N4가 목적지 노드 D에게 패킷을 전달하지 않아도 노드 N1은 소스노드에게 신고하지 않을 것이다. 즉, 네트워크에 참여하는 모든 노드가 최소한 악의적인 노드는 아니라는 가정 하에서만 메커니즘이 재대로 작동할 수 있는 것이다. 또 하나의 문제점은 이기적 노드라고 판명된 노드가 어떠한 징계도 받지 않는다는 사실이다. Watchdog을 통해 이기적 노드를 탐지하면 경로안내자에 따라 그 노드를 제외한 경로가 다시 제공된다. 그러나 이기적 노드의 관점에서 보면 그 노드는 여전히 네트워크에 참여할 수 있고 자신이 생성한 데이터를 보내는데 어떠한 제약도 받지 않는다. 결과적으로 다른 노드를 위해 데이터를 전달해 주지 않아도 되기 때문에 오히려 이기적 노드에게 더욱 유리하게 작용할 수도 있는 메커니즘인 것이다.

2. Pathrater

Pathrater(경로안내자)는 Watchdog과 유사한 방법으로 이기적인 노드를 관리한다. 즉 각 노드들은 버퍼를 가지고 있고 데이터를 전송 할 때마다 전송 여부를 확인시키기 위한 인증서(certificate)와 목적지 노드의 정상적인 데이터 수신에 대한 ACK를 소스노드까지 전송하는 과정을 수행함으로써 next 노드의 악의적인 행위를 감시 및 확인한다. 각각의 노드는 next 노드에 데이터를 전송하고 각 노드들은 자신의 버퍼에 전송한 데이터의 복사본을 저장해 둔다. 이전 노드는 데이터 전송 후 next 노드로부터 인증서를 수신함으로써 데이터가 정상적으로 next 노드에 전송되었음을 확인한다. 마지막으로 목적지 노드가 데이터를 수신하게 되면 목적지 노드는 소스 노드에게 ACK를 전송한다.

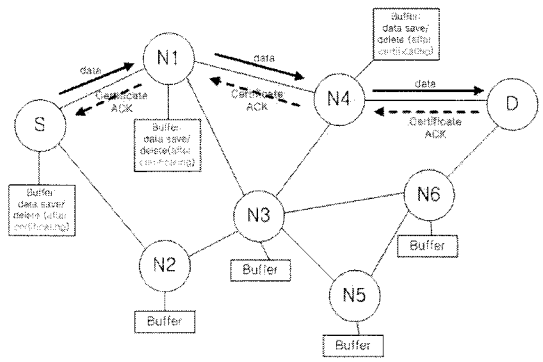


그림 2. Pathrater의 이기적인 노드 관리기법
Fig. 2. Selfish node management method of Pathrater

그림 2는 Pathrater의 이기적 노드 탐지방법을 보여주고 있다. 노드 N1은 노드 N4에게 데이터를 전송하고 노드

N4는 자신의 개인키를 사용하여 암호화된 인증서를 노드 N1에 전송한다. 그러나 노드 N4가 목적지 노드 D에 데이터를 정상적으로 전송하지 않으면 노드 N1은 목적지 노드 D로부터 ACK를 수신하지 못하게 된다. 따라서 노드 N1은 이 같은 사실을 노드 N4로부터 수신한 인증서와 함께 소스노드 D에게 보고한다. 해당 인증서는 노드 N4의 개인키로 암호화된 인증서이며 다른 노드들은 임의로 동일한 인증서를 발급할 수 없다. 따라서 거짓 보고를 방지할 수 있다. 이때 소스노드 S는 노드 N4의 공개키를 사용하여 인증서를 복호화하게 되고 이때 보고된 사항이 거짓인지 진실인지를 결정하게 된다.

여기서 노드 N1을 악의적인 노드로 가정해 보자. 노드 N1은 노드 N4에게 데이터를 전송하고 노드 N4로부터 인증서를 수신한다. 이때 노드 N1은 노드 N4로부터 발급된 인증서와 함께 노드 N4가 데이터를 정상적으로 전송하여도 소스노드에게 거짓 신고를 할 수 있다. 또한 목적지 노드로부터 전송되는 ACK를 버릴 수 있으며 이와 같은 악의적인 행위를 탐지할 방법이 없다. 즉 Pathrater도 Watchdog과 같이 정상적인 노드를 악의적인 노드로 거짓 신고하거나 인접하는 두 노드가 모두 악의적인 노드일 경우 악의적인 행위에 대한 확인이 불가능한 문제점을 갖고 있다[10].

3. CONFIDANT

CONFIDANT(Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks)는 Watchdog과 비슷하게 각 노드들이 서로를 감시하면서 이기적인 노드를 탐지하는 메커니즘이다[12]. 그러나 Watchdog을 이용한 메커니즘과는 달리, 악의적인 노드를 감지하고 그런 노드들을 네트워크에서 고립시켜 네트워크의 서비스를 이용하지 못하도록 하는 것이 이 메커니즘의 목적이다.

CONFIDANT에서는 노드에 대한 정보를 두 가지로 구분하여, 이기적 행동을 직접 경험한 노드에서 온 정보와 간접적으로 얻은 노드에게서 온 정보로 구별하여 서로 다른 가중치를 둔다. CONFIDANT는 각각의 노드에 그림 3과 같은 요소들이 설치되어 작동된다.

3.1 이웃 감시자(neighborhood monitor)

정상적인 라우팅 행동에서 벗어나는 일탈 행위를 감시하고, 만약 특정 노드가 비정상적인 행동을 하게 되면 신뢰관리자에게 알람 메시지를 보내게 된다.

3.2 The Trust Manager

이웃감시자로부터 들어온 알람메시지를 필터링하고, 다른 신뢰 관리자들과 알람 메시지를 교환한다. 이때, ALARM 메시지는 다른 노드들의 악의적인 행위를 경고하기 위해 노드의 trust manager에 의해 전송된다.

Outgoing ALARM은 악의적인 행위에 대한 보고내용의 수신, 관찰, 경험 후 노드 자신에 의해 생성된다. Incoming ALARM은 친근하거나 그 외의 다른 이웃노드들에 의해 발생하며 그에 따라 ALARM의 소스는 적용되기 전에 신뢰정도가 확인되어져야 한다. 따라서 보고하는 노드의 신뢰수준에 따라 incoming ALARM 메시지가 생성된다.

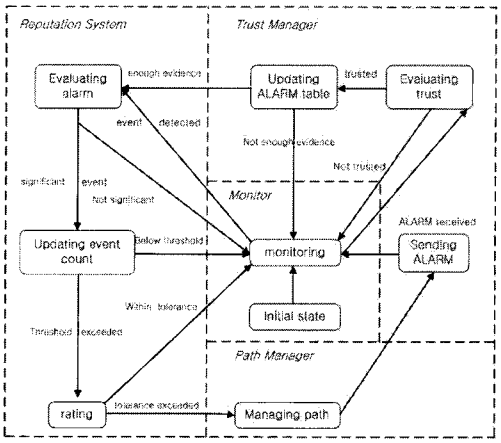


그림 3. CONFIDANT에서 제안하는 메커니즘 구조
Fig. 3. Proposed Mechanism of CONFIDANT

3.3 평가시스템(reputation system : Node Rating)

다른 노드들에 관해서 관찰된 혹은 보고된 행동에 의해 그 노드에 대한 등급을 매긴다. 등급은 부합되는 임계치(threshold)를 초과하거나 노드의 악의적 행위에 대한 근거가 충분하면 변경된다. 등급은 행위 탐지유형마다 서로 다른 가중치를 부여하는 등급 함수(rate function)에 의해 변경된다. 즉 자신의 경험 요소에는 최대의 가중치를 부여하고 이웃 노드들의 관찰결과에 대해서는 좀 더 작은 가중치를 부여한다.

3.4 경로관리자(The Path Manager)

경로 상에 존재하는 노드의 평가결과 및 security metric에 따라 경로의 재순위화를 시키는 경로에 대한 등급을 관리한다. 즉 악의적인 노드를 포함하는 경로를 삭제하거나 악의적인 노드를 포함하는 라우팅 메시지가 들어왔을 때 적절한 행동을 한다.

이 프로토콜은 시뮬레이션 결과 전체 네트워크의 절반정도가 악의적인 노드인 경우 잘 대처할 수 있는 것으로 밝혀졌다. 그러나 이 메커니즘을 네트워크의 모든 이동노드에서 구현하기에는 그 부담이 너무 크다는 문제점이 있다.

4. Secure Mechanism to manage selfish nodes

Secure Mechanism to manage selfish nodes는 신고자와 혐의자 목록으로 구성되는 신고테이블을 이용하여 악의적인 노드에 대한 효율적인 검출방법을 제안하고 있다[13, 14]. 네트워크 내의 각 노드는 데이터를 전송 후 복사본을 버퍼에 저장하고 있으면서 다음 노드의 데이터 전송여부를 확인하기 위해 엿듣는다. 만약 일정시간 내에 엿듣지 않으면 다음 노드에 대한 failure tally를 증가시키고 tally가 임계치(threshold)를 초과하면 misbehavior라고 판단하는 것까지는 Watchdog과 동일하다. 다른 점은 Watchdog에서는 소스 노드에게 유니캐스트로 신고를 하였지만, 여기서는 misbehavior를 네트워크 전체에 신고함으로써 악의적인 노드를 검출한다. 그리고 Watchdog은 소스 라우팅 방식인 DSR에서만 적용 가능하지만, Secure Mechanism to manage selfish nodes는 Self-Organized Network Layer Security in Mobile Ad Hoc Network에서 제안한 RREP 메시지에 다음 홉(next hop)의 주소 필드를 추가시키는 알고리즘을 활용함으로써 AODV에서도 다음 노드의 패킷 전달행위를 엿듣는 것이 가능하도록 하였다. 신고를 받은 모든 노드는 자신의 신고 테이블에 동일한 신고자, 혐의자 목록이 있는지 여부를 검사해서 있다면 신고를 무시하고 버린다. 그림 4는 다른 노드를 임의로 거짓 신고하는 경우에 대해 동작하는 과정을 나타낸다. 악의적인 노드 M이 경로설정 및 데이터 전달과 무관하게 임의의 노드 X를 거짓으로 신고하는 경우에는 각 노드의 신고테이블에 ①과 같은 목록이 생성된다. 악의적인 노드 M이 현재 위치에 그대로 있거나 아니면 새로운 위치로 이동한 후에도 임의의 다른 노드를 거짓으로 신고하는 행위를 계속 할 경우 각 노드의 신고테이블에는 ②, ③과 같은 신고목록이 계속 추가된다.

결국 각 노드의 신고테이블에 신고자 노드 M의 목록이 임계치(k)개 이상 기록되면 노드 M이 거짓 신고자로 식별되어 더 이상 네트워크의 동작에 참여 할 수 없게 된다. 하지만 이러한 방법은 누적된 신고 횟수에 근거하여 악의적인 노드를 검출해내는 방식이므로 네트워크의 생명주기 동안 임계치를 초과하지 않는 악의적인 행위가 발생하면 실제 악의적인 노드임에도 불구하고 효과적으로 탐지하지 못하는 경우가 발생 한다. 또한 별도의 신고테이블을 유지해야하고 주기적으로 갱신이 발생하므로 네트워크의 부하가 증가하는 단점이 있다.

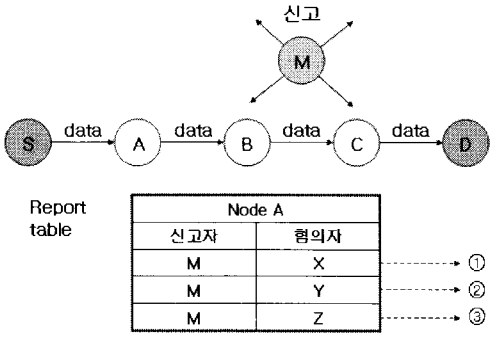


그림 4. 다른 노드를 임의로 거짓 신고하는 경우
Fig. 4. A Case of false report for any other node arbitrarily

III. 제안 스킴 : SRPPnT (A Secure Routing Protocol in MANET based on Malicious behavior Pattern of Node and Trust Level)

SRPPnT는 네트워크 내에서 보안성을 높이기 위해 두 단계의 과정을 거친다[15]. 첫 번째 단계는 네트워크 내에 존재하는 각 노드들의 신뢰정도를 판단하기 위한 단계로써 각 노드들의 악의적인 행위를 확인하여 각 노드마다 신뢰수준을 결정하는 단계이다. 두 번째 단계는 정해진 신뢰수준을 바탕으로 경로를 설정하여 데이터 패킷을 보낸다. SRPPnT는 AODV를 기반으로 하였으며 동작과정은 각각 그림 5와 같다[16].

MANET에 속한 노드는 데이터를 전송 후 전송한 데이터를 버퍼에 저장하고 next노드의 전달과정을 엿들어 자신이 보낸 메시지와 비교한다. 만약 어떠한 노드를 악의적인 노드라고 판단하게 되면 신고하는 노드는 자신의 개인키(private key)로 신고(report) 메시지를 암호화하여 네트워크 에러메시지와 함께 broadcast 한다. 신고 메시지는 거짓 신고를 당하는 노드는 이에 대응하는 반박(retort) 메시지를 broadcast 한다.

이때, 악의적인 노드를 신고하는 메시지와 반박 메시지를 수신한 목적지 노드 또는 반박 메시지를 보낸 노드의 이웃 노드들은 신고와 반박 메시지를 증명하기 위한 증명(proof) 메시지를 broadcast 한다. 이와 같은 과정에서 소스노드는 신고, 반박, 증명 메시지를 모두 수신하게 되며 수신한 모든 메시지를 비교하여 악의적인 노드를 결정한다. 악의적인 노드가 결정되면 소스노드는 별도의 캐시에 해당 노드의 신뢰수준을 감소시키고 갱신된 신뢰수준을 broadcast 한다.

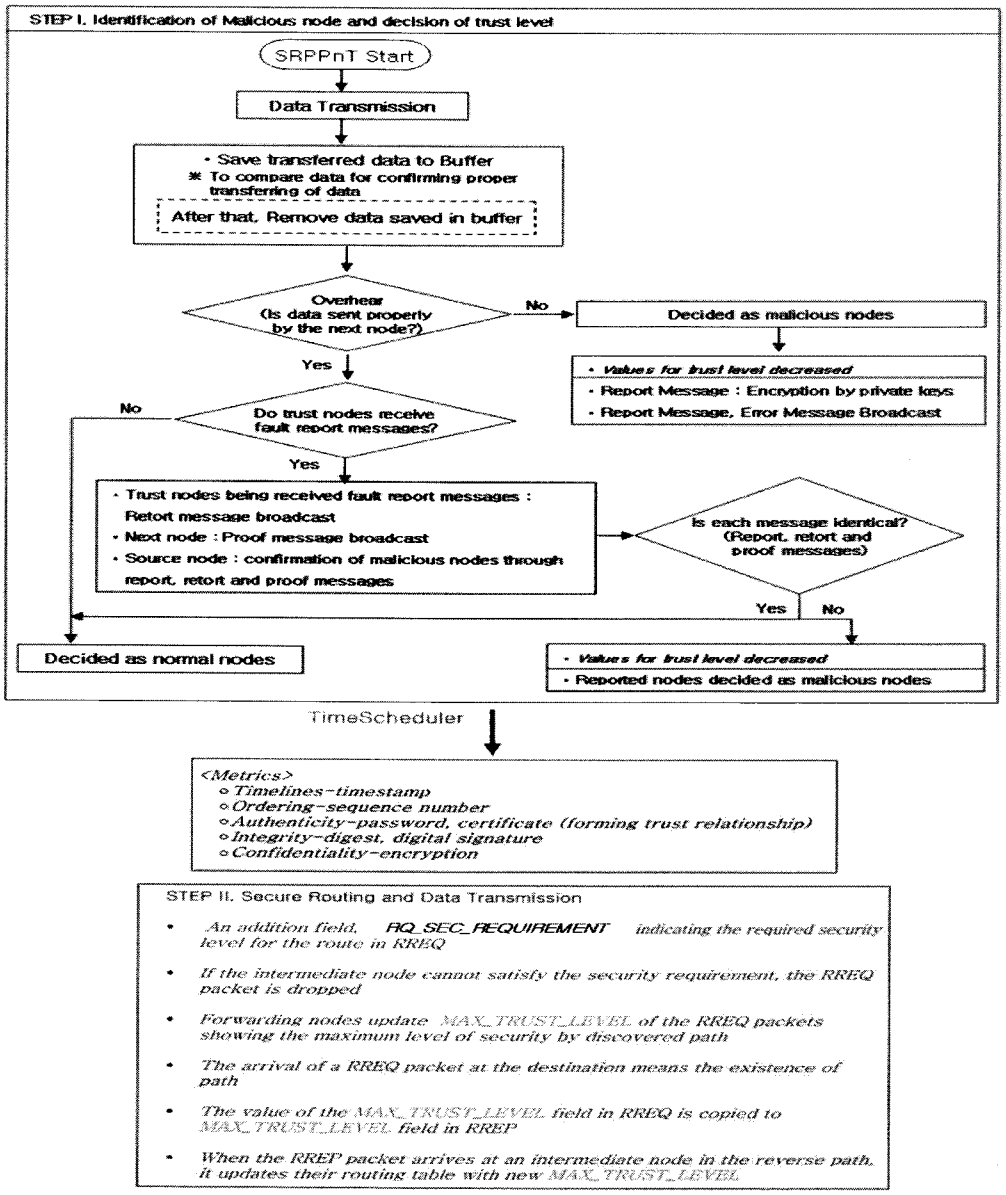


그림 5. SRPPnT의 동작 과정(1단계, 2단계)
Fig. 5. Action Process of SRPPnT (Step 1 and 2)

1. Assumptions

각 노드들은 양방향 링크를 경유하여 연결되어 있다. 각 노드들은 promiscuous 모드로 동작하며 따라서 이웃노드들의 데이터 전송을 엿들을 수 있다.

노드의 거짓 보고를 방지하기위해 비대칭 암호화 방식이 사용되며, 네트워크에 존재하는 노드 i의 개인키 Ki- 는 노드 i를 제외하고는 어떤 노드에게도 공개되지 않는다. 노드 i의 공개키 Ki+는 네트워크에 존재하는 모든 노드들에게 공개된다. 따라서 노드 K를 악의적인 노드로 보고하기 위해서는 노드 i는 개인키 Ki-를 사용하여 보고 메시지를 암호화하여 broadcast(보고자 : node I, suspect 노드 k)한다. 이때 보고 메시지를 수신한 모든 노드들은 공개키 Ki+를 사용하여 복호화된 보고 메시지를 읽을 수 있다. 개인키 Ki-가 공개되지 않기 때문에 다른 노드들은 거짓 메시지를 만들 수 없다.

2. 악의적인 노드 확인

2.1 〈case 1〉 : 다른 노드로 위장하여 거짓 신고

SRPPnT는 개인키와 공개키를 이용한 비대칭 암호방식 통해 다른 노드의 ID를 이용하여 다른 노드인 것처럼 위장하여 거짓 신고하는 악의적인 노드를 확인한다. 그림 6과 같이 만약 노드 i가 노드 X인 것처럼 위장하여 거짓 신고를 하더라도 노드 i는 노드 X의 개인키 KX-를 모르기 때문에 자신의 개인키 Ki-로 암호화를 한다. 거짓 신고를 수신한 각 노드들은 노드 X의 신고인 것으로 판단하고 노드 X의 공개키 KX+로 신고 메시지를 복호화 한다. 하지만, 노드 X의 개인키 KX-로 암호화되지 않았기 때문에 정상적으로 복호화 되지 않고 오류를 발생시키므로 각 노드들은 이 신고가 잘못된 신고임을 알게 되므로 노드 i는 다른 노드로 위장하여 거짓 신고를 할 수가 없게 된다.

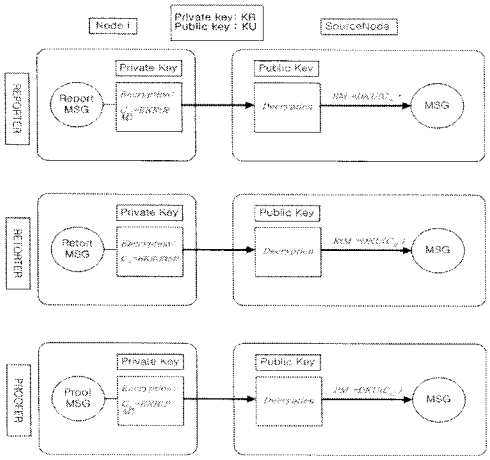


그림 6. 다른 노드로 위장하여 거짓 신고하는 노드 확인
Fig. 6. Validation of a node which falsely reports covering as an other node

2.2 〈case 2〉 : 악의적인 노드의 데이터 버림 및 변경

악의적인 노드가 데이터를 버리거나 변경하는 경우 악의적인 노드의 전송을 엿들은 노드가 악의적인 노드를 신고하게 된다. 이때, 신고(report) 메시지를 받은 목적지 노드는 증명 메시지를 소스 노드로 보내 신고를 증명한다.

그림 7의 경우 신고 메시지와 증명(proof) 메시지를 받은 소스노드 S는 노드 N3이 보낸 신고(report)메시지와 인접 노드가 보낸 증명(proof) 메시지를 비교하여 사실 여부를 확인 후 노드N6이 목적지 노드 D에게 메시지를 보내지 않은 것을 확인하고 노드 N6을 악의적인 노드로 판단한다.

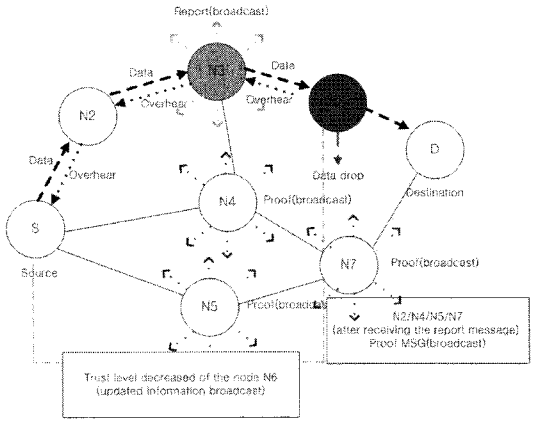


그림 7. 데이터를 버리는 악의적인 노드 확인
Fig. 7. Validation of a malicious node which dumps data

따라서 소스 노드 S는 노드 N6에 대한 신뢰수준(trust level)을 감소시키고 갱신된 신뢰수준(trust level)을 이웃노드에게 broadcast 하여 알린다.

2.3 〈case 3〉 : 정상적인 노드를 거짓 신고하는 경우

그림 8의 경우 노드 N6은 정상적으로 목적지 노드에게 데이터를 전송 했지만 악의적인 노드 N3은 목적지 노드에게서부터 전송된 ACK 메시지를 버리고 노드 N6을 악의적인 노드라고 거짓 신고한다. 이때, 노드 N6의 신고 (report) 메시지를 수신한 노드 N6은 자신이 악의적인 노드가 아님을 반박하는 반박(retort) 메시지를 네트워크에 존재하는 이웃 노드들에게 broadcast 하게 된다.

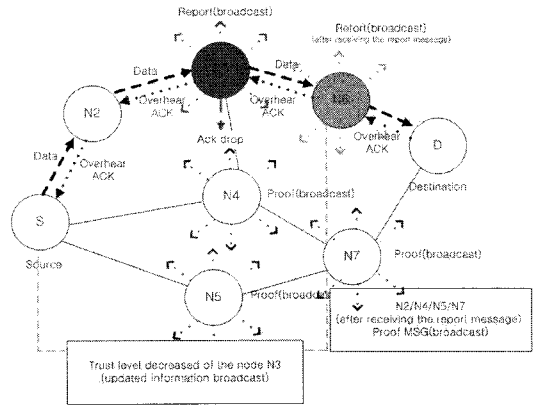


그림 8. 정상적인 노드를 거짓 신고하는 악의적인 노드 확인
Fig. 8. Validation of a malicious node which falsely reports a normal node

또한, 신고(report) 메시지를 수신한 목적지 노드 역시 증명(proof) 메시지를 broadcast 하게 된다. 신고(report), 반박(retort), 증명(proof) 메시지를 모두 수신한 소스노드 S는 신고(report) 메시지에 신고 된 메시지와 증명(proof) 메시지가 첨부되어 있다면 노드 N6은 목적지 노드 D에게 메시지를 전송하였다고 판단한다. 따라서 노드 N3은 노드 N6이 정상적인 노드임에도 불구하고 악의적인 노드로 거짓 신고한 것으로 확인되어 소스 노드 S는 노드 N3에 대한 신뢰수준(trust level)을 감소시키고 업데이트된 정보를 이웃 노드들에게 알린다.

2.4 〈case 4〉 : 악의적인 노드의 거짓 반박

악의적인 노드가 자신의 악의적인 행위에 대해 신고를 당했을 경우 이를 거짓으로 반박 할 수 있다. 이 경우 그림 9에서 볼 수 있듯이 수신한 메시지가 다르게 된다. 따라서 노드 N6의 반박은 거짓으로 확인되고 노드 N6이 악의적인 노드로 판단되어진다. 이후 소스노드 S는 노드 N6에 대한 신뢰수준(trust level)을 감소시키고 업데이트 된 정보를 이웃노드들에게 알린다.

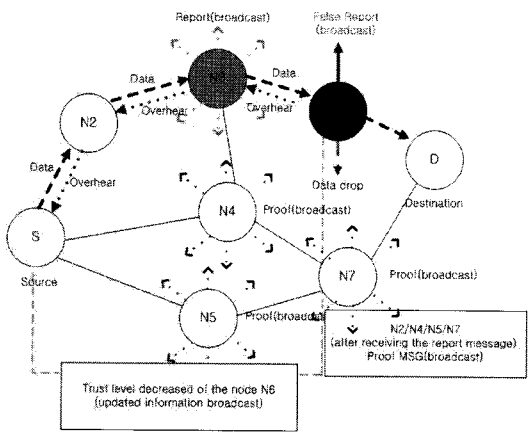


그림 9. 거짓 반박하는 악의적인 노드 확인
Fig. 9. Validation of a malicious node which falsely retorts on

3. Secure 라우팅 경로를 통한 데이터 전송

소스노드는 악의적인 노드를 확인하는 단계에서 획득 된 인접노드들의 신뢰수준을 확인하고 신뢰수준에 따라 설정된 임계치(threshold)를 적용하여 경로를 설정 한다. 즉, 신뢰수준에 대한 정보는 소스 노드가 RREQ 패킷을 전송 할 때 확인되어지며 이때 각 노드의 신뢰수준이 일정 요구 수준에 미치지 못하면 경로 설정에서 제외시키고 보다 높은 신뢰수준을 보유하고 있는 인접노드를 경로 설정에 포함시키면서 목적지까지의 경로를 설정한다.

SRPPnT는 RREQ 패킷에 RQ_SEC_REQUIREMENT, MAX_TRUST_LEVEL 필드를 추가하였다. RQ_SEC_REQUIREMENT는 송신하는 노드에서 경로 설정에 포함시킬 노드의 요구되는 security 레벨 값을 갖는다. 이 필드는 송신자에 의해서 최초 설정되며 경로 설정 단계가 끝날 때까지 변경되지 않는다. 단, 모든 노드들이 일정 수준의 security 레벨을 보유하고 있지 않을 경우 RQ_SEC_REQUIREMENT 값은 감소시킨 후 경로 재탐색과정을 수행한다. 따라서 중간노드들은 RREQ 패킷을 수신하게 되면 가장 먼저 각 노드들은 해당 노드가 RQ_SEC_REQUIREMENT를 충족하는 지를 확인한다.

만약 노드가 경로 설정에 참여 할 수 있을 만큼 충분한 secure 레벨을 보유하고 있다면 SRPPnT는 RREQ 패킷을 이웃노드로 forward하여 ADOV에 기반하여 경로설정 과정을 수행한다. 하지만 중간 노드가 요구되는 secure 레벨을 충족하고 있지 못하면 RREQ 패킷은 버려지고 더 이상 forward 되지 않는다. MAX_TRUST_LEVEL 필드는 경로 설정 과정 중에 각 노드들이 보유하고 있는 신뢰수준의 최대 값을 의미하며 RREQ가 forward 되기 전 기존 보유하고 있는 MAX_TRUST_ LEVEL 보다 더 큰 값을 보유하고 있는 노드가 존재하면 값을 갱신 후 forward 한다.

RREQ 패킷이 최종 목적지에 도착하면 목적지 노드는 RREQ 패킷을 AODV에 기반하여 전송한다. 이때 RREQ 패킷과 동일하게 MAX_TRUST_LEVEL 필드가 추가되며 목적지 노드는 RREP 패킷을 이전 노드에 유니캐스트하며 목적지 노드로부터 소스 노드까지 최종 단일 경로를 설정한다. 즉 RREQ를 전송 할 때 신뢰수준이 어느 정도 되어야 하는지에 대한 정보를 실어 보냄으로써 해당 수준 이하인 신뢰도를 갖는 노드는 경로설정 시 제외하고 일정 수준의 신뢰수준으로 이루어진 노드들로 구성된 경로를 설정하여 데이터 패킷을 전송한다.

그림 10은 각 노드들의 신뢰수준에 대한 정보를 기반으로 가장 security 한 경로를 설정 하여 데이터 패킷을 보내는 과정을 나타낸다. 소스 노드 S는 RQ_SEC_ REQUIREMENT를 7로 설정하여 RREQ를 forward 한다. 이때 한 홉 떨어진 인접 노드들 중 임계치 이상의 값을 갖고 있는 노드는 N1(value : 12)이므로 다른 노드에 대한 RREQ 패킷은 버려지고 노드 N1에게만 forward 된다. 소스 노드 S는 노드 N1의 RQ_SEC_REQUIREMENT 값이 자신이 보유하고 있는 값보다 크기 때문에 MAX_TRUST_LEVEL 필의 값을 12로 갱신한 후 forward 한다.

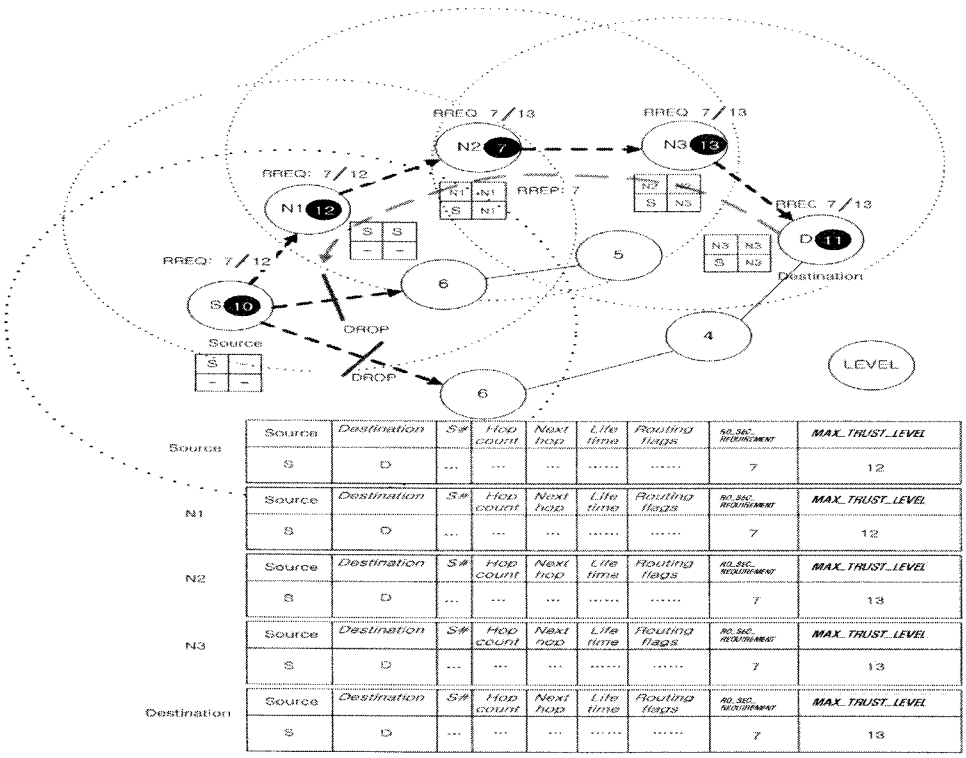


그림 10. 신뢰수준이 높은 경로 설정 과정
Fig. 10. Process of setting up a path with high trust level

소스 노드 S로부터 RREQ를 수신한 노드 N1은 next 노드의 신뢰수준과 수신한 RREQ 패킷의 MAX_TRUST_LEVEL 값을 비교하여 기존 값보다 next 노드의 값이 크면 갱신하고 작거나 같으면 갱신하지 않고 RREQ 패킷을 노드 N2로 forward한다. 노드 N2의 신뢰수준이 7이므로 기존 RREQ 패킷의MAX_TRUST_LEVEL 값(12) 보다 작기 때문에 RREQ 패킷의 MAX_TRUST_LEVEL 필드는 갱신되지 않고 노드 N2로 forward 된다. 이때 라우팅 테이블은 노드 N1로부터 소스 노드 S까지 가는 역경로가 저장된다. 이후 해당되는 중간 노드들(노드 N2, 노드 N3)은 노드 N1과 동일한 과정을 수행한다. 노드 N1로부터 RREQ를 수신한 노드 N2는 노드 N3의 신뢰수준과 수신한 RREQ 패킷의 MAX_TRUST_ LEVEL 값을 비교한다. 노드 N3의 신뢰수준이 13이므로 기존 RREQ 패킷의 MAX_ TRUST_LEVEL 값(12)보다 크기 때문에 RREQ 패킷의 MAX_TRUST_LEVEL 필드는 13으로 갱신되어 노드 N3으로 forward된다. 이때 라우팅 테이블은 노드 N2로부터 소스노드 S까지 가는 역경로가 저장된다.

노드 N2로부터 RREQ를 수신한 노드 N3은 목적지 노드 D의 신뢰수준과 수신한 RREQ 패킷의MAX_TRUST_LEVEL 값을 비교한다. 목적지 노드 D의 신뢰수준이 11이므로 기존 RREQ 패킷의MAX_TRUST_ LEVEL 값(13) 보다 작기 때문에 RREQ 패킷의 MAX_TRUST_LEVEL 필드는 갱신되지 않고 목적지 노드 D로 forward 된다. 이때, 라우팅 테이블은 노드 N3로부터 소스노드 S까지 가는 역경로가 저장된다.

노드 N3으로부터 RREQ를 수신한 목적지 노드 D는 RREP 패킷을 유니캐스트하고 라우팅 테이블이 저장되어 있는 목적지 노드로부터 소스 노드까지의 역경로를 추적하면서 최적의 경로를 설정하게 된다. 이 같은 과정을 통해 선정된 경로는 신뢰수준이 양호한 경로로써 데이터 패킷을 소스 노드로부터 목적지 노드까지 보다 안전하게 전송 할 수 있다.

IV. 성능 분석 및 평가

제안 알고리즘의 성능을 분석하기 위한 시뮬레이션 결과를 제시한다. 성능 분석 및 평가를 위한 서버의 소프트웨어와 하드웨어 환경은 그림 11과 같다. 즉, 성능 평가를

위한 시뮬레이션은 NS-2 및 자체 개발한 에뮬레이터(Emulator)를 이용하여 실시하였다.

- CPU : Pentium core 2 duo 2.166 GHz
- Memory : 2 Gigabyte
- Disk : 150 Gigabyte 10000rpm raptor
- Server : Linux
- UI Tool : NS-2, Emulator(자체 개발)

※ Emulator 개발 언어 : 비주얼 C++

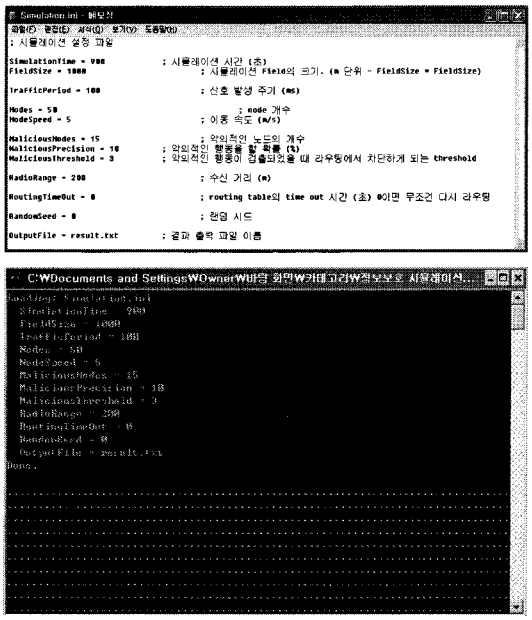
그림 11. 서버의 소프트웨어 및 하드웨어 환경
Fig. 11. Environment for Analysis of performance and Evaluation

표 1. 이동성 및 트래픽 모델
Table 1. Mobility and Traffic Model

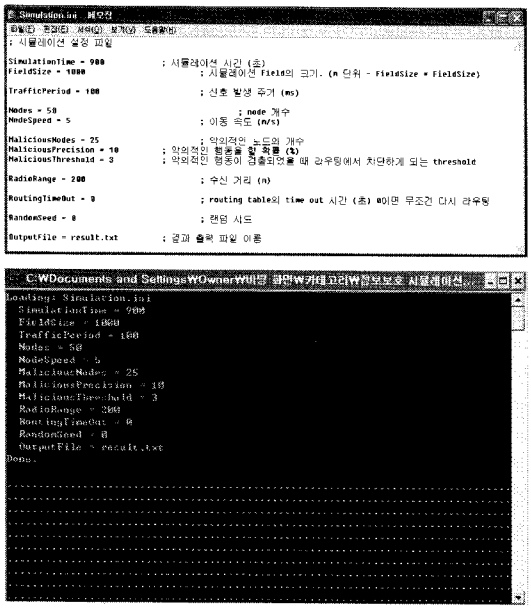
Terrain Dimensions	1000 * 1000(m)
Number of Nodes	N(사용자 지정)
No. of Malicious Nodes	0 ~ N(사용자 지정)
Simulation Time	900 sec
Routing Protocol	AODV
Traffic	- CBR - Packet Size : 512Kbytes - Packet Interval : 5 packet/s
Movement	- Random Waypoint - Pause Time : 0 ~ 900 sec - Speed : min 0, max Nm/s

표 1은 시뮬레이션에 적용한 노드의 이동성 및 트래픽 모델을 나타낸다. 시뮬레이션 결과에서 정지시간(pause time)이 0초일 경우는 모바일 노드가 시뮬레이션 시간 동안 항상 움직이는 것을 의미하고, 900초일 경우는 이동성이 없는 것을 의미한다. 정상 및 악의 적인 노드 수는 사용자가 임의로 지정 할 수 있으며 악의적인 행위 수에 대한 정상 및 악의적인 노드 분류를 위한 임계치를 설정할 수 있도록 하였다. 악의적인 노드 탐지 시간은 1/1000초(sec) 단위로 설정 하였다. 악의적인 노드는 특정 패턴(시간, 횟수 등)을 가지고 악의적인 행위를 하지 않기 때문에 악의적인 행위를 할 확률을 사용자가 지정할 수 있도록 하였다.

(a) 악의적인 노드 설정 : 5, 트래픽 발생주기 : 100ms
(a) Setting up malicious node :5, Traffic period : 100ms



(b) 악의적인 노드 설정 : 15, 트래픽 발생주기 : 100ms
(b) Setting up malicious node : 15, Traffic period : 100ms



(c) 악의적인 노드 설정 : 25, 트래픽 발생주기 : 100ms
(c) Setting up malicious node : 25, Traffic period : 100ms

그림 12. 시뮬레이션을 위한 환경 설정 및 에뮬레이터 동작
Fig. 12. Setting up simulation environment and Operation of Emulator environment

그림 12는 시뮬레이션을 위해 에뮬레이터가 읽어 들이는 환경 설정파일(Simulation.ini) 및 에뮬레이터 동작 화면을 나타내는 것으로써 사용자는 환경 설정파일의 직접 수정이 가능하며 다양한 시뮬레이션 환경에 따른 결과를 확인할 수 있다.

1. 악의적인 노드 판별 시간

설정한 실험 환경과 같이 악의적인 노드는 1/000초 간격으로 확인되며 총 900초 동안 시뮬레이션을 실시하였다. 악의적인 노드가 실제 악의적인 행위를 하는 확률은 25% 내로 설정하였으며, 임계치는 0으로 설정하여 악의적인 행위 즉시 악의적인 노드로 분류 될 수 있도록 하였다. 그림 13, 14에서 볼 수 있듯이 기존 연구 방법들은 악의적인 행위가 계속 이루어지지 않는다는 사실을 간과했고, 임계치(threshold)를 적용함으로써 일정시간 지속되는 네트워크 수명주기 동안 악의적인 노드검출이 제대로 이루어지지 않는 것을 확인할 수 있다. 하지만 SRPPnT의 경우 악의적인 행위가 발생하는 즉시 관련 노드를 악의적인 노드로 선별 관리함으로써 네트워크 수명주기 동안 가능한 많은 수의 악의적인 노드가 탐지됨을 알 수 있다.

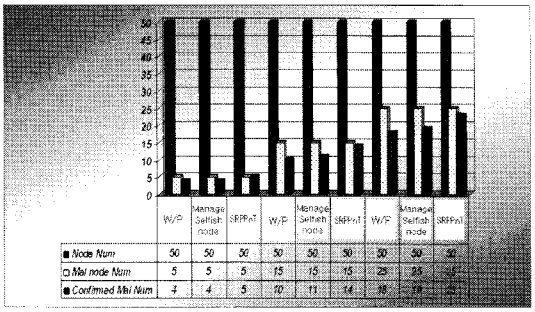


그림 13. 악의적인 노드 탐지 수
Fig. 13. The number of Malicious nodes Detected

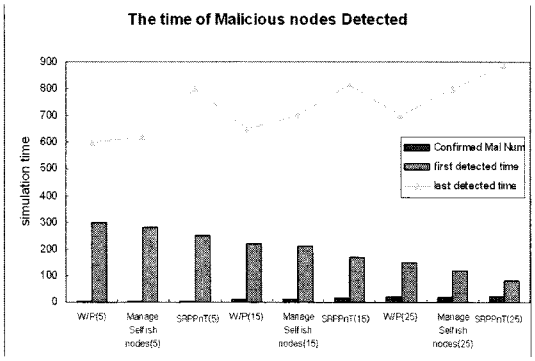


그림 14. 악의적인 노드 탐지 시간
Fig. 14. The time of Malicious nodes Detected

2. 라우팅 측면의 신속성

기존 연구 방법들은 악의적인 행위가 이루어질 때마다 해당 횟수를 테이블에 저장해두고 저장된 값이 임계치를 초과하였을 경우 해당되는 노드를 악의적 노드로 최종 판단한다. 따라서 임계치를 초과하기 전 까지는 악의적인 노드에게 수차례의 메시지를 계속 보내게 됨으로써 라우팅의 신속함이 떨어지게 된다. 하지만 SRPPnT는 신고, 반박, 증명 메시지를 통해 악의적인 행위에 대해 발생 즉시 탐지함으로써 기존 프로토콜들에 의해 발생하는 단점을 보완 하였다.

2.1 End-to-End Delay

그림 15, 16은 end-to-end 지연을 나타낸 것으로 SRPPnT는 정적인 환경과 동적인 환경 모두 Watchdog and Pathrater, Secure Mechanism to Manage Selfish Nodes, CONFIDENT 보다 전반적으로 양호한 결과를 나타낸다.

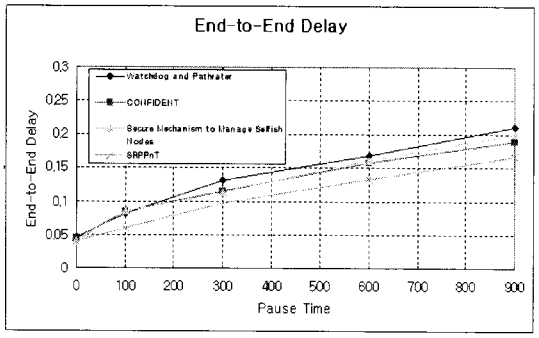


그림 15. 네트워크 내 존재하는 악의적인 노드 수 : 5
Fig. 15. Number of Malicious nodes existing in Net : 5

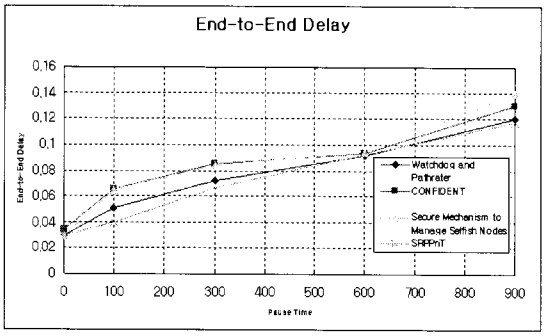


그림 16. 네트워크 내 존재하는 악의적인 노드 수 : 25
Fig. 16. Number of Malicious nodes existing in Net : 25

특히 정적인 환경에서 동적인 환경으로 변화하면서 SRPPnT와 기존 연구된 프로토콜들 간의 end-to-end 지연 폭이 점점 더 커지는 것을 확인 할 수 있다. 기존 연구들은 일정시간 동안 악의적인 행위가 발생함에도 불구하고 악의적인 노드로 판별하지 못하는 경우가 다수 발생하게 되지만, SRPPnT는 신고, 반박, 증명 메시지를 통해 좀 더 신속하고 정확하게 악의적인 행위를 탐지함으로써 여러 번 중복되는 소스노드의 전송과정을 최소화하여 end-to-end delay를 감소시킴을 확인할 수 있다.

2.2 Overhead Ratio

그림 17, 18은 overhead ratio를 나타낸 것으로 SRPPnT는 정적인 환경과 동적인 환경 모두 Watchdog and Pathrater, Secure Mechanism to Manage Selfish Nodes, CONFIDENT보다 전반적으로 양호한 결과를 나타낸다.

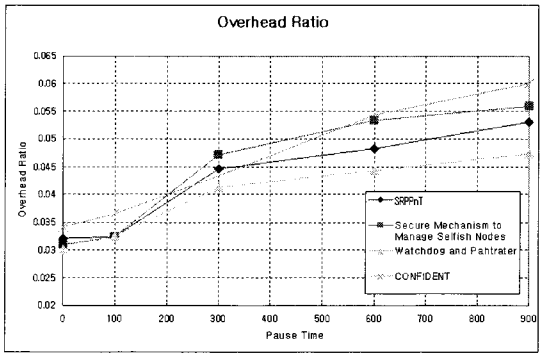


그림 17. 네트워크 내 존재하는 악의적인 노드 수 : 5
Fig. 17. Number of Malicious nodes existing in Net : 5

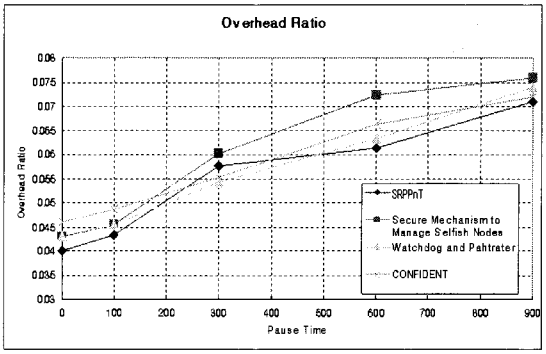


그림 18. 네트워크 내 존재하는 악의적인 노드 수 : 25
Fig. 18. Number of Malicious nodes existing in Net : 25

Watchdog and Pathrater는 경로 설정 시 각 노드의 라우팅 테이블에 소스로부터 자신의 위치까지 모든 경로정보를 누적하여 저장하기 때문에 오버헤드가 많이 발생하며, 악의적인 행위를 효율적으로 탐지하지 못하여 경로탐색 및 설정이 반복적으로 발생함으로써 오버헤드가 크게 발생한다. Secure Mechanism to Manage Selfish Nodes의 경우에는 각 노드들이 악의적인 행위에 대한 정보를 확인하기위해 활용할 별도의 테이블을 유지하기 때문에 네트워크 사이즈가 증가하게 되면 오버헤드가 많이 발생하게 된다. 또한 임계치를 적용함으로써 악의적인 노드로 분류되지 않음에도 불구하고 각 노드는 별도의 테이블을 지속 유지하고 있어야 하며 정기적으로 갱신이 이루어지기 때문에 그에 따르는 오버헤드가 많이 발생한다. CONFIDENT는 별도의 이웃 감시자, trust manager, 평가시스템, 경로관리자를 가지고 있기 때문에 이와 같은 모듈을 통해 주고 받은 제어 패킷 및 악의적인 노드탐지 실패에 의해 발생하는 반복되는 경로탐색 및 설정 과정에서 오버헤드가 많이 발생한다.

3. 정확한 악의적인 노드 판별에 의한 패킷 전송

3.1 Packet Deliver Ratio

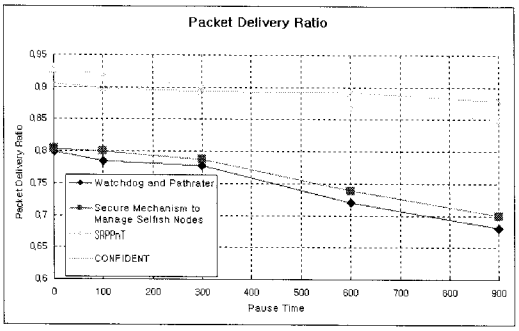


그림 19. 네트워크 내 존재하는 악의적인 노드 수 : 5
Fig. 19. Number of Malicious nodes existing in Net : 5

그림 19에서와 같이 악의적인 노드가 네트워크 내에 10%(5개) 존재한다는 가정 하에 데이터 전송률에 대한 성능평가 결과를 살펴보면 Watchdog and Pathrater, Secure Mechanism to Manage Selfish Nodes, CONFIDENT, SRPPnT 모두 65% 이상의 전송률을 나타낸다. SRPPnT는 정적인 환경에서 90% 이상의 전송률을 나타내고 동적인 환경에서는 85%의 전송률을 나타낸다.

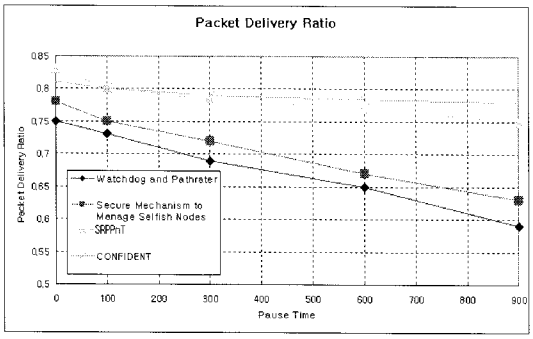


그림 20. 네트워크 내 존재하는 악의적인 노드 수 : 25
Fig. 20. Number of Malicious nodes existing in Net : 25

그림 20에서와 같이 악의적인 노드가 네트워크 내에 50%(25개)가 존재한다는 가정하에 데이터 전송률에 대한 성능 평가 결과를 살펴보면 Watchdog and Pathrater, Secure Mechanism to Manage Selfish Nodes, CONFIDENT, SRPPnT 모두 60% 이상의 데이터 전송률을 나타낸다. 특히 SRPPnT가 75%로 가장 높은 데이터 전송률을 나타낸다. 그래프에서 살펴 볼 수 있듯이 점점 동적인 환경으로 갈수록 SRPPnT의 데이터 전송률은 Watchdog and Pathrater, Secure Mechanism to Manage Selfish Nodes, CONFIDENT 보다 점점 더 성능의 차이가 커짐을 확인 할 수 있다.

MANET에서 기존의 악의적인 노드 판별 및 차단하는 알고리즘들은 데이터 버림, 데이터 변조 또는 거짓 신고 등과 같은 악의적인 행위가 지속적으로 이루어짐에 따라 임계치를 초과할 경우 그 노드를 악의적인 노드라고 판단하여 네트워크의 동작에서 제외시킨다. 하지만 실제 네트워크에서 적용 할 때 악의적인 노드는 악의적인 행동을 연속적으로 계속하지는 않을 것이다. 따라서 SRPPnT는 한 번의 악의적인 노드의 올바르지 않은 행동을 차단함으로써 더욱 정확하게 악의적인 노드를 판별하여 Watchdog and Pathrater, Secure Mechanism to Manage Selfish Nodes, CONFIDENT보다 효과적인 라우팅 경로 설정 후 패킷을 정상적으로 전달할 수 있다.

V. 결 론

MANET은 그 특성상 유선 네트워크에 비해 공격에 매우 취약하다. 따라서 악의적인 노드의 공격으로 데이터가 변조, 손실되거나 라우팅 경로가 훼손되었을 경우에 그 피해가 심각하다. 따라서 적극적이고 안전한 보안대책이 강구되어야 한다. 제안한 프로토콜은 악의적인 노드 확인 과정에서는 신고, 반박, 증명 메시지를 통해 기존의 신고 테이블 이용 시 증가할 수 있는 오버헤드와 threshold 사용 시 발생할 수 있는 악의적인 노드 검출시간 지연 및 오류율을 감소시킴을 확인하였다. 또한 악의적인 행동을 하는 노드의 확인을 통해 각 노드의 신뢰수준을 결정하고 신뢰수준에 대한 정보 활용을 통해 보안 측면이 양호한 노드가 존재하는 경로를 설정하여 데이터를 전송함으로써 MANET에서 발생할 수 있는 라우팅 공격 양상에 효율적으로 대처할 수 있음을 확인하였다.

참고문헌

[1] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang, "Security in Mobile Ad Hoc Networks: Challenges and Solutions", IEEE Wireless Communications, 2004.

[2] Djamel Djenouri, Othmane Mahmoudi, Mohamed Bouamama, David Llewellyn-Jones, and Madjid Merabti, "On Securing MANET Routing Protocol Against Control Packet Dropping", IEEE International Conference, pp. 100-108, July 2007.

[3] Liu Hongxun, J.G. Delgado-Frias, and S. Medidi, "Using a Cache Scheme to Detect Misbehaving Nodes in Mobile Ad-Hoc Networks", Networks, 2007. ICON 2007. 15th IEEE International Conference, pp. 7-12, Nov. 2007.

[4] Claude Crepeau, Carlton R. Davis, and Muthucumaru Maheswaran, "A Secure MANET Routing Protocol with Resilience against Byzantine Behaviours of Malicious or Selfish Nodes", Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshop - Vol. 2, pp. 19-26, 2007.

[5] 나가진, 도인실, 편혜진, 채기준, "Secure Mechanism to manage selfish nodes in Ad hoc Network", JCCI, 2004.

[6] Yu Ming, Zhou Mengchu, and Su Wei, "A Secure Routing Protocol Against Byzantine Attacks for MANETs in Adversarial Environments", Vehicular Technology, IEEE Transactions on Vol. 58, No. 1, pp. 449-460, Jan. 2009.

[7] 김한식, 박현희, 강병석, 백상헌, 강철희, "모바일 애드혹 네트워크에서 안전한 라우팅을 위한 경로조사 기법", 한국통신학회 하계학술대회, 2007년 7월.

[8] 서대열, 김진철, 김경목, 오영환, "MANET 환경에서 Zone Routing Protocol을 이용한 안전한 경로설정 보안 알고리즘 S-ZRP", 전자공학회논문지 제43권 TC편 제4호, 13-21쪽, 2006년 4월.

[9] 이재영, 최승권, 이병록, 김선철, 신병곤, 조용환, "모바일 Ad Hoc 네트워크에서 AODV 프로토콜의 성능향상을 위한 라우팅 공격탐지", 한국통신학회 논문지, 제32권 제6호(무선통신), 2007년 6월.

[10] S. Marti et al., "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", ACM MOBICOM, 2000.

[11] Y. Hu, D. Johnson, and A. Perring, "Sead: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks", IEEE WMCSA, 2002.

[12] Sonja Buchegger, and Jean-Yves Le Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes, fairness in dynamic ad-hoc networks", In Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, June 2002.

[13] JongOh Choi, "Energy-Efficient Secure Routing Protocol and WSN Management Framework In Mobile Ad Hoc and Sensor Network", a doctoral thesis, Yonsei University, Feb. 2007.

[14] 이강석, 최종오, 지종복, 송주석, "MANET에서 악의적인 노드의 안전하고 효율적인 검출 방안", 한국정보처리학회 논문지, 제12권, 제5호, 617-622쪽, 2005년 10월.

[15] 박건우, 변용성, 이승찬, 마용재, 송주석, "MANET에서 악의적인 노드 확인에 기반한 Secure 라우팅 방안", 한국정보보호학회, 하계 학술대회, 749-753쪽, 2006년.

[16] S. R. Das, and C. E. Perkins, "Ad Hoc On-Demand Distance Vector(AODV) Routing for Mobile Ad Hoc networks", *http://www.ietf.org/rfc/rfc3561.txt*, July 2003.

저 자 소 개

박 성 승

1982:경북대학교 전자공학과 학사
1990:미국 해군대학원 통신체계 석사
1982~2007:육군 방공/정보통신장교
2001:충북대학교 전자계산학 박사수료
현 재:국방대학교 전산실장
관심분야:유비쿼터스 컴퓨팅, 센서 네트워크, 시공간 추론, 불확실성 처리 등

박 건 우

1997:충남대학교 컴퓨터과학과 학사
2007:연세대학교 컴퓨터과학과 석사
1997~2007:육군 정보통신/전산장교
현 재:국방대학교 전산정보학과 박사과정
관심분야:유비쿼터스 컴퓨팅, 센서 네트워크, 정보보호, 정보검색, 소셜 네트워크

류 근 호

1976:숭실대학교 전산학과 이학사
1980:연세대학교 대학원 전산전공 공학석사
1988:연세대학교 대학원 전산전공 공학박사
1976~1980:육군군수지원사 전산실 ROTC장교
1980~1983:한국전자통신연구원 연구원
1983~1986:한국방송통신대 전산학과 조교수
1989~1991:미국 Univ. of Arizona 연구원
1986~현 재: 충북대학교 전기/컴퓨터공학부 교수
관심분야: 시공간 DB, 데이터마이닝, Temporal GIS, 지식기반 정보검색, 유비쿼터스 컴퓨팅/스트림 데이터처리, DB 보안, 바이오 인포매틱스 등

이 상 훈

1978:성균관대학교 전자공학과 학사
1989:연세대학교 전산학과 석사
1997:일본 교토대학교 정보공학과 박사
1998~2000:충남산업대학교 멀티미디어학과 교수
2000~현재:국방대학교 전산정보학과 교수
관심분야:정보검색, 정보보호, 데이터베이스, 데이터마이닝 등