

A Study of Minimization of Network Overhead Using local Repair in Global-MAHN

Global-MAHN에서 지역경로복구를 이용한 네트워크 오버헤드 최소화에 대한 연구

Yang-Geun Jang*, Jae-Ho Kim*, Jin-Seung Bae*, Jae-Seung Ha*, Choong-Yeul You*,
Hyun-Wook Kim*, Kwang-Bae Lee*
장 양근*, 김재호*, 배진승*, 하재승*, 유충렬*, 김현욱*, 이광배*

Abstract

Recently, The importance of Ubiquitous network that can be supported useful information to MANET mobile device user through wired internet network in anywhere, anytime is growing rapidly. This proposed algorithm is the research on the algorithm, minimizing network overhead by preventing unnecessary network overhead previously and adapting Local repair mechanism to efficient data transmission when route error happens in network, in GMAHN environment.

요약

최근들어 신속히 네트워크를 구성할 수 있는 MANET 환경하에서 MANET 단말기 이용자가 언제, 어디서나 유선 기반망에 접속하여 원하는 정보를 제공 받을 수 있는 유비쿼터스 네트워크의 중요성이 부각되고 있다. 제안한 알고리즘은 자치분산네트워크인 MANET과 유선 인터넷 기간망과의 연결을 위해 필요한 효율적인 게이트웨이 설정방식을 제안한 GMAHN환경에서 경로상에 에러가 발생하였을 경우 경로 재설정시 생성되는 불필요한 네트워크 부하를 사전에 방지하여 네트워크 부하를 최소화 하며 효율적인 데이터 전송을 위해 지역복구 기법을 적용한 알고리즘에 관한 연구이다.

Key words : Mobile Ad Hoc, MANET, GMAHN, Global MANET

I. 서론

MANET은 기반 망 및 어떠한 중심이 되는 접근점 (Access Point)의 중재 없이 상호 망을 구성하는 이동노드들의 집합으로, 네트워크 안에서 상호 통신을 하기 위해 단말의 역할뿐만 아니라 라우터의 역할도 하여 네트워크의 다른 노드를 통한 멀티 홉 (multi-hop) 라우팅에 의해 통신이 이루어지기 때문에, 재해, 재난, 군사용이나 대체(Backup) 네트워크용으로 활용에 중점을 두어 연구가 진행되어 왔다.¹⁾

기존의 AODV라우팅 프로토콜에서 경로손실이 발생

하면 에러정보 패킷인 RERR을 소스노드로 보내고, 소스노드에서는 다시 경로 설정을 하게 된다. 이러한 경우에는 전체 네트워크에서 라우팅 오버헤드가 발생되어 네트워크 부하의 원인이 된다.

본 논문에서는 MANET과 유선망의 연결을 위한 효율적인 게이트웨이 설정방식을 제안한 GMAHN 환경을 사용하여 경로상에 에러가 발생하였을 경우 경로 재설정시 생성되는 불필요한 네트워크 부하를 사전에 방지하고 네트워크 부하를 최소화 하며 효율적인 데이터 전송을 위해 지역복구 기법을 제안하였다.

제안한 알고리즘의 성능평가를 위해 게이트웨이 네트워크 영역, 이동노드의 수, flow수를 변경시킴으로써 각 환경에서의 성능을 분석하였다.

본 논문 2장에서는 기본적인 MANET에 대한 전반적인 이해를 돕기 위해 기본개념을 설명하였다. 3장에서는 Ad-Hoc 망에서 사용되는 Routing Protocol 중 본 논문에 이용되고 있는 Reactive 방식의 AODV

* 明知大學校 電子工學科

(Dept. of Electronics Engineering, Myongji University)

接受日:2007年 10月 22日, 修正完了日: 2007年 12月 20日

Routing Protocol 과 이동 애드 혹 네트워크와 유선 망과 연결에 대해 살펴보고 4장에서는 제한한 알고리즘에 대해 설명하였으며 5장에서는 성능 평가를 위한 제한한 방식을 다양한 실험환경에 따라 분석 및 평가 하였다. 마지막으로 6장에서는 본 논문이 분석 및 평가된 자료를 근거로 종합하여 결론을 맺는다.

II. 애드 혹 네트워크

MANET(Mobile Ad hoc Network)은 Mobile IP와 같이 반드시 AP나 기지국을 통해 이동 서비스를 지원 하는 형태의 네트워크가 아니라 단말기 자체가 Ad-Hoc 형태로 네트워크 라우팅 인프라 구조를 형성한 자치분산 네트워크이다.

‘Ad-hoc’은 우리말로 ‘임시용변의’ 또는 ‘특별히 실행되는’ 이라는 뜻으로서 유선망을 구성하기 어렵거나 망을 구성한 후 단기간 사용되는 경우에 적합하며 Ad-Hoc 네트워크에서는 호스트의 이동에 제약이 없고 유선망과 기지국(Base Station)이 필요 없다.²⁾

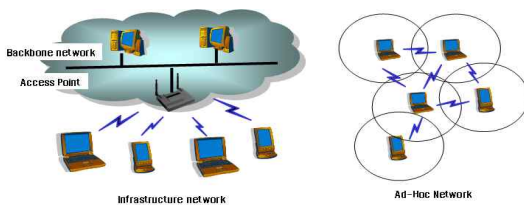


Fig. 1. Ad-hoc Network and Infrastructure Network
그림 1. 애드 혹 망과 기간망

Ad-hoc 네트워크는 그림 1과 같이 기간 네트워크의 도움 없이 무선 인터페이스를 가진 이동노드간의 자율적으로 구성되는 임시적인 네트워크와 중앙 집중적인 제어를 필요로 하는 기간 네트워크와의 상대적인 개념을 가진 네트워크이며 완전히 독립적인 형태로 존재하거나, 또는 게이트웨이에 의해 기간 네트워크에 접속하는 형태로 존재할 수 있다.

III. AODV 라우팅 프로토콜

3.1 라우팅 프로토콜 패킷

AODV 라우팅 프로토콜에서의 패킷은 RREQ (Route Request), RREP(Route Reply), RERR(Route Error), Hello message로 구분된다.

RREQ 패킷은 경로설정요구패킷으로 한 소스노드가

해당 패킷을 목적지까지 전달하려고 할 때, 자신의 경로 라우터 테이블에 목적지까지 가는 경로가 없을 경우 RREQ를 네트워크 환경 내에 있는 모든 노드에게 브로드캐스트 한다.

RREP 패킷은 경로설정패킷에 대한 경로응답패킷으로, 통신을 원하는 모바일 소스노드의 RREQ패킷이 정상적으로 목적지노드에 도달되었거나, 중간 노드가 라우팅 테이블 내에 목적지까지의 경로를 저장하고 있을 때에 경로설정패킷에 대한 응답메시지인 RREP 패킷을 소스노드를 향하여 유니캐스트 방식으로 전송한다.

RERR는 에러발생에 대한 경로에러패킷이며, 경로에 에러가 발견되었을 때, 다른 이웃노드들에게 경로에러를 알려주기 위해 사용된다. RERR패킷은 경로에러 때문에 현재의 목적지까지 이 경로를 사용하여 데이터의 전송을 할 수 없다는 정보를 담고 있으며, 각 노드들은 이웃노드들과의 precursor list를 생성하여 이와 같은 링크상의 변화정보를 이웃노드들에게 전달한다.

Hello message는 경로 유지를 위한 패킷이며, RREQ 패킷과 RREP 패킷을 사용하여 설정된 경로상의 모바일 노드들이 자신의 이웃노드와 Hello message를 주고받음으로써 이웃노드와의 경로가 유효한지를 판단한다. 만약 일정시간 내에 Hello message를 주고받지 못한다면 이웃노드와의 경로가 유효하지 못하다고 판단하여 경로에러패킷인 RERR 패킷을 전송한다.³⁾⁵⁾

3.2 경로 설정

그림 2 에서 보는 것과 같이 소스노드 S가 목적지 노드 D로 데이터를 전송하려고 할 때, 다음과 같은 과정을 수행하게 된다.

우선, 자신이 가지고 있는 라우팅 테이블의 경로정보를 확인하여 목적지 노드까지의 경로가 없다면 이웃노드에게 RREQ 패킷을 브로드캐스트 방식으로 전송한다.

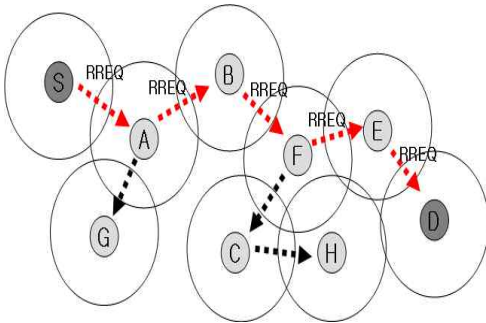


Fig. 2. RREQ Packet Transmission
그림 2. RREQ 패킷의 전송

소스노드의 이웃노드인 노드 A는 RREQ 패킷을 수신한 다음 자신의 라우팅 테이블의 정보와 비교하여 목적지 노드의 경로를 알고 있으면 바로 소스노드에게 RREP 패킷을 보낸다. 만약 목적지 노드의 경로를 알고 있지 않으면, RREQ 패킷을 자신의 이웃 노드인 B와 G에게 전송을 하며, 이와 같은 과정을 반복하여 최종적으로 목적지 노드까지 전달되게 된다.

목적지 노드에서 경로응답패킷인 RREP 패킷을 생성하여 유니캐스트 방식으로 소스노드로 전송하게 되고, 소스노드에서는 생성된 경로를 통해 데이터를 보내게 된다.

다음의 표 1은 RREQ 패킷포맷 구조이다.

Table 1. Format of RREQ message
표 1. RREQ 메시지 포맷

0	1	2	3
01234567890123456789012345678901			
Type = 1	J R G D U	Reserved	Hop Counter
RREQ ID(Broadcast ID)			
Destination IP Address			
Destination Sequence Number			
Source IP Address			
Source Sequence Number			

소스노드가 RREQ를 발생시켜 플로딩을 통해 이웃노드들에게 브로드캐스트 하기 전에 다음과 같은 절차를 수행해야 한다.

- 1) 소스노드는 RREQ 패킷에 Type을 1로 선택하고 홉 카운트를 0으로 변경
- 2) 브로드캐스트 ID는 현재의 브로드캐스트 ID에 1을 증가시키면 새로운 RREQ가 생성될 때마다 증가
- 3) RREQ 패킷을 받은 노드들의 경로 테이블 정보를

확인하여, RREQ패킷을 받은 노드들이 자신이 목적지 노드이거나 중간노드로서 목적지 노드의 경로를 알고 있는 경우 RREP 패킷을 소스노드로 보내고, 목적지 노드에 대한 어떠한 정보도 갖고 있지 않다면 홉카운트를 하나 증가 시키고 RREQ를 이웃 노드에게 전송한다.

- 4) RREQ를 수신했을 때, 브로드캐스트 ID와 소스주소를 이용해 중복된 패킷을 제거하고 이웃노드로 전송하지 않는다.

다음의 표 2는 RREP 패킷포맷 구조이다.

Table 2. Format of RREP message
표 2. RREP 메시지 포맷

0	1	2	3
01234567890123456789012345678901			
Type = 2	R A	Reserved	Prefix H o p
Size Counter			
Destination IP Address			
Destination Sequence Number			
Source IP Address			
Lifetime			

RREQ 패킷을 받은 목적지 노드나 목적지 노드까지의 경로를 알고있는 중간노드는 이에 대한 응답패킷인 RREP 패킷을 생성하여 전송한다.

다음의 그림 3은 RREQ패킷과 RREP패킷을 사용하여 최종경로를 설정하는 그림이다.

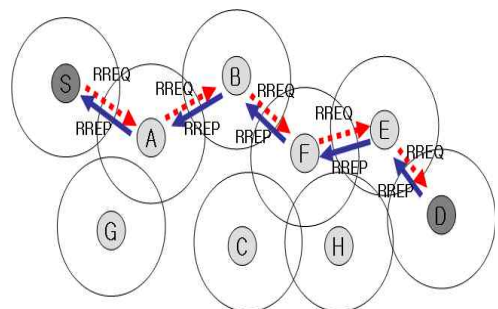


Fig. 3. RREP Packet Transmission
그림 3. RREP 패킷 전송

3.3 경로 유지

노드들의 경로유지는 Hello 메시지를 전송함으로써 경로 유효성을 확인하는 방법이 있다. 이는 Hello 메시지의 TTL(Time to Live)값을 1로 하여 자신의 이웃노드 외에는 전달하지 않도록 한다. 만약, 데이터 전송 중에 목적지노드나 중간 노드가 이동하여 Allowed_Hello_Loss * Hello_Interval 동안 Hello 메시지를 수신하지 못하면 해당 노드와의 연결이 불가능한 것으로 판단하여 경로에러를 인식하고, 에러를 탐지한 노드는 RERR 패킷을 소스노드로 전달한다. RERR 패킷을 받은 소스노드는 경로 설정과정을 재실행하여 새로운 경로를 설정하고 다시 데이터를 전송한다. 또한 유효한 경로라는 것을 알리기 위해 이전의 경로에 대한 목적지 시퀀스번호보다 하나 더 큰 목적지 시퀀스번호와 Broadcast ID로 경로 발견단계를 시작한다.

그림 4는 Hello Message 전송을 나타내고 있다.

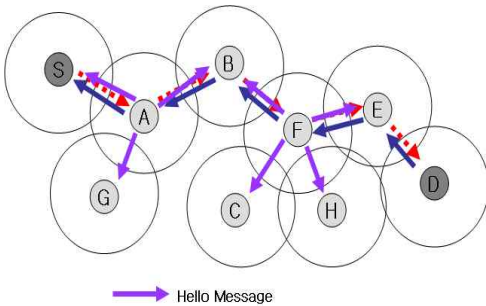


Fig. 4. Hello Message Transmission
그림 4. Hello 메시지 전송

3.4 애드 혹 망과 유선망의 연결

그림 5는 이동 애드 혹 망과 유선망을 연결했을 때의 그림이다.

이동 애드 혹 망과 유선망의 연결을 위해서 중간 노드로서의 게이트웨이(gateway)를 사용하였는데, 게이트웨이 자체가 하나의 노드 역할을 담당하면서 이동 애드 혹 망과 유선망을 서로 연결하는 역할을 한다.

게이트웨이에서는 게이트웨이 Advertisement 메시지를 관리반경 범위에 있는 노드들에게 전송하고, 이를 받은 노드들은 게이트웨이 주소를 업데이트함으로써 게이트웨이와 애드 혹 노드들이 연결된다.

이동 애드 혹 망과 유선망을 연결하여 서로간의 원활한 통신을 위해서는 이동 노드들의 입장에서 게이트웨이를 발견하는 알고리즘이 가장 중요하며, 또한 게이

트웨이 입장에서 주변 이동 노드들에게 정확하고 신속한 방법으로 자신의 존재를 알리는 것이 중요한 알고리즘이 된다. 또한 게이트웨이는 유선망과 이동 애드 혹 망의 중간 매개체로서, 하나의 노드로서 동작해야한다.⁽⁵⁾

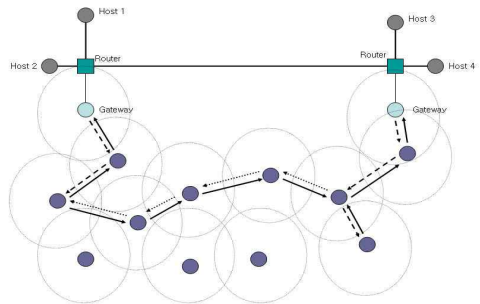


Fig. 5. Access between MANET and Internet
그림 5. 이동 애드 혹 망과 인터넷 액세스

IV. 제안한 AODV 라우팅 프로토콜

본 논문에서는 유·무선통합 망에서 링크 손실이 발생하였을 때의 대처 방법에 대한 알고리즘을 제안한다.

4.1 기존방식

기존에는 에러가 발생하면 에러를 감지한 노드가 에러가 발생하였음을 알리기 위해 RERR패킷을 생성하여 소스노드로 전송하고 또한 이웃노드들에게도 보내어 각각의 Routing Table에 정보를 갱신한다.

소스노드에서 에러패킷인 RERR패킷을 받았을 경우 다시 경로 설정을 위하여 RREQ패킷을 네트워크 안에 브로드캐스팅 하여 경로를 재설정한다.

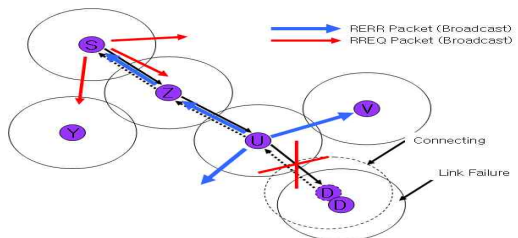


Fig. 6. Link Failure
그림 6. 링크 손실

그림 6은 기존의 방법에서 에러가 발생했을 때의 상황으로, 노드 D의 이동으로 인해 Link Failure가 발생하여 노드 U는 RERR패킷 생성하여 이웃 노드에 브로드캐스트 한다. 이 RERR패킷이 노드 Z를 거쳐 소스노드가 수신하면 다시 RREQ를 네트워크에 브로드캐스트 하여 경로를 재설정 하는 것이 기존 방식이다.

4.2 지역경로 복구

지역 경로 복구는 링크 손실이 발생 하였을 경우 먼저 손실된 위치를 파악하게 되는데, 손실링크 위치가 소스노드 보다 목적지노드에 가까울 경우 지역 경로 복구 알고리즘을 실행한다.

```
if(ch->num_forwards() > rt_hops)
```

num_forwards()는 소스노드까지의 홉 수이고, rt_hops는 목적지까지의 홉 수이다. 위 조건을 만족할 경우 지역경로복구 알고리즘을 실행한다.

이러한 이유는 경로 손실이 소스노드 근처에서 발생하였을 경우 지역경로복구 알고리즘의 효과가 나타나지 않기 때문이다.

단절상위 노드가 경로복구를 실행하는 동안 소스노드로부터 도착한 모든 데이터 패킷은 단절 상위노드의 버퍼(Buffer)에 저장되며, 경로가 다시 복구된 후에 다시 목적지 노드에게 전송된다.

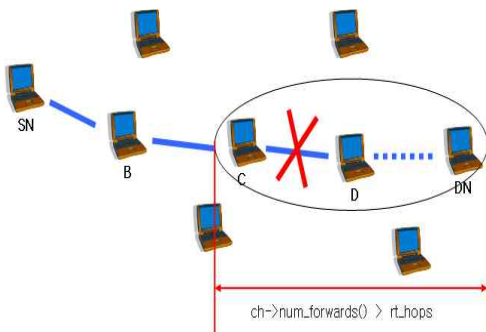


Fig. 7. Local Repair
그림 7. 지역경로 복구

그림 7을 보면, 노드 C와 D가 단절되면 단절상위노드인 노드 C는 먼저 위치를 파악하고 조건을 만족하면 노드 C는 소스노드가 되어 목적지노드 DN의 경로를 발견하기 위해 경로탐색과정을 시작하여 RREQ를 생성하여 브로드캐스트 한다.

RREQ를 수신한 노드들은 먼저 소스노드 SN에 대한 역 경로 정보를 자신의 라우팅테이블에 갱신하며 자신이 목적지노드이던지 또는 목적지노드까지의 라우팅 정보를 가지고 있다면 RREP패킷을 생성하여 응답하게 된다. 만일 그렇지 않다면 단순히 RREQ의 TTL 값을 감소시켜서 이웃노드에게 전달한다.

이러한 일련의 과정으로 중간노드들 중에 경로에러가 발생했을 경우 자체 지역적으로 재 경로설정이 되어 기존의 방법인 경로에러가 발생하였을 경우 소스노드에서부터 다시 경로 설정을 하지않기 때문에 제어메시지가 줄어들고 신속한 경로 설정으로 네트워크의 부하를 줄일 수 있다.

본 논문에서는 지역 경로복구 알고리즘을 Global-MAHN에 적용하여 네트워크내의 Overhead의 감소와 데이터수신율 등으로 네트워크의 부하의 감소를 확인 하여 성능평가를 하였다.

V. 성능 평가

5.1 시뮬레이션 환경

Global-MAHN에서 지역 경로복구 알고리즘에 대한 성능평가를 위해 NS-2.1b9a 시뮬레이터를 사용하였다. NS-2는 UC Berkeley에서 개발한 Event 구동 네트워크 시뮬레이터로 다양한 IP 네트워크를 시뮬레이션하며, C++ 과 스크립트언어인 OTcl(object tool command language)로 구성되어 있다.⁷⁾

표 3과 표 4는 각각 시뮬레이션 환경과 기본 시뮬레이션 설정요소에 대한 설명이다.

Table 3. Environment of Simulation

표 3. 시뮬레이션 환경

Software	Simulator	NS-v2.1b9a
	Language	C++, Tcl
Hardware	Pentium4 1.7GHz CPU	
O/S	WOW Linux 7.3	

Table 4. Simulation parameter

표 4. 시뮬레이션 파라미터

Parameter	Value
Transmission range	250 m
Simulation time	900 s
Topology size	800 m × 500 m
Number of mobile nodes	15
Number of sources[Flow수]	5
Number of gateways	2
Traffic type	Constant Bit Rate
Packet rate	4 packets/s
Packet size	512 bytes
Node Pause time	5 s
Maximum speed	10 m/s

5.2 성능평가 알고리즘

본 논문의 성능평가 알고리즘은 표 5와 같다.

Table 5. Performance evaluation algorithm

표 5. 성능평가 알고리즘

순번	알고리즘	설명
1	GMAHN	무선 애드혹 네트워크와 유선망을 연결한 유무선 통합망
2	LR (Local Repair)	GMAHN에서 Local Repair가 제공되는 알고리즘

5.3 시뮬레이션 성능평가

평가 항목으로는 다음 표 6과 같이 Gateway Distance, Mobile Nodes, Flows의 파라미터에 변화를 주어 각각의 데이터 수신율과 라우팅 오버헤드를 비교하였다.

Table 6. The Contents of Evaluation

표 6. 평가 항목

평가 항목	변수
Gateway_Distance [Hops]	1, 2, 3, 4, 5, 6
Mobile Nodes [개]	15, 20, 25, 30, 35, 40
Flows [개]	5, 6, 7, 8, 9, 10

가. Gateway_Distance의 영향

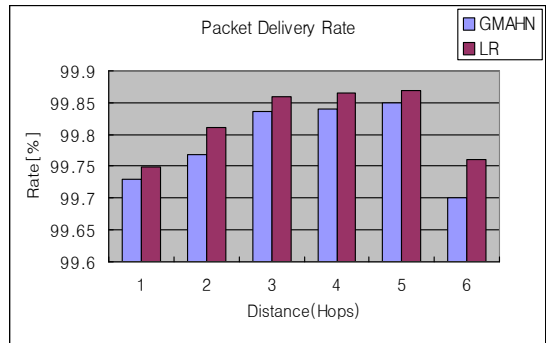


Fig. 8. Packet delivery rate - Gateway Distance

그림 8. 데이터 수신율 - 게이트웨이 거리

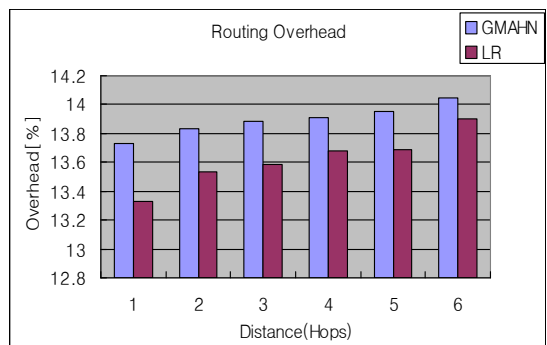


Fig. 9. Routing Overhead - Gateway Distance

그림 9. 라우팅 오버헤드 - 게이트웨이 거리

그림 8, 9의 X축은 멀티 홉(multi hop)을 의미하며, 1홉에서 6홉까지 1홉 간격으로 Gateway가 이웃노드에게 전송하는 Advertisement Message의 멀티 홉을 나타낸다.

Advertisement Message는 Gateway에서 이동노드의 관리를 위한 메시지로 홉 수가 1이면 Gateway에서 홉 수가 1인 이동노드만을 관리한다. 이동노드에서

글로벌 망 (호스트)으로 데이터 전송 요청이 있을 경우, 위 관리 홉 수가 클수록 빠른 경로 설정이 이루어짐을 예상할 수 있었지만, 결과를 확인해 보면 1 홉에서 5홉까지는 데이터 전송율이 예상대로 상승하지만 6홉에서 감소하는 현상이 나타났다. 이는 Gateway에서 관리하는 노드수가 많아서 경로 정보는 많이 보유하고 있지만 그로 인해 오버헤드가 증가하여 오히려 네트워크에 부하를 가져오기 때문이다.

나. 이동 노드의 영향

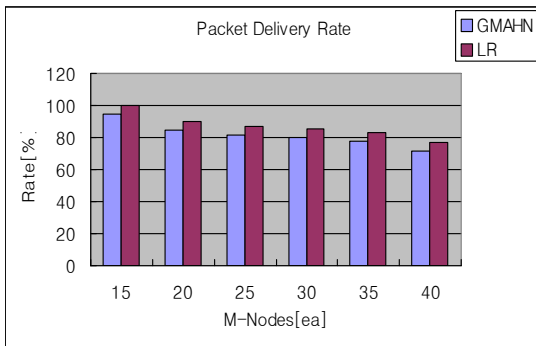


Fig. 10. Packet delivery rate - Mobile Node
그림 10. 데이터 수신율 - 이동노드 수

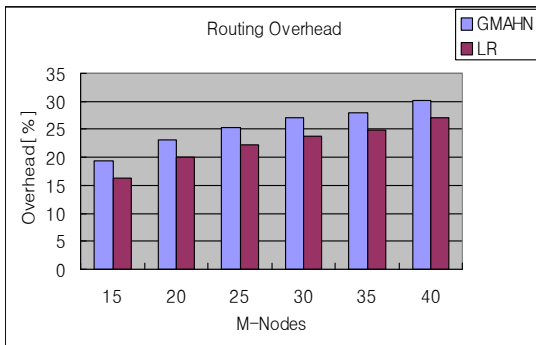


Fig. 11. Routing Overhead - Mobile Node
그림 11. 라우팅 오버헤드 - 이동노드 수

네트워크 안에서 이동 노드의 수가 증가하면 이웃노드들의 채널 공유로 인해 패킷충돌 및 채널감지를 위해 필요한 노드자체내의 네트워크 리소스 사용 등 실제 네트워크 성능에 많은 영향을 미치게 된다.

그림 10과 11은 이동노드 수에 따른 데이터 수신율과 라우팅 오버헤드를 보여주고 있으며, 노드 수가 많아짐에 따라 데이터 수신율은 낮아지고 오버헤드는 증가함을 알 수 있다. 또한 지역경로 복구시에도 노

드가 많아질수록 잦은 패킷충돌로 인해 제 성능을 발휘하기 어렵기 때문에 다른 실험환경보다 성능차이가 적음을 알 수 있다.

다. Flow 수

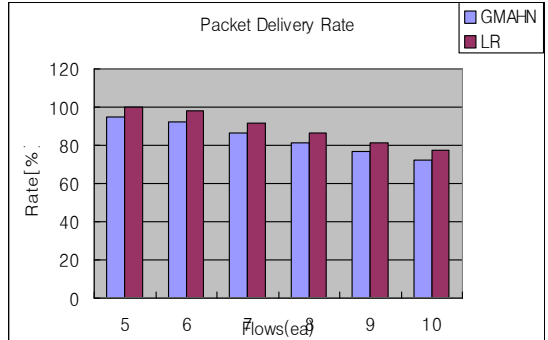


Fig. 12. Packet delivery rate - Flows
그림 12. 데이터 수신율 - Flow 수

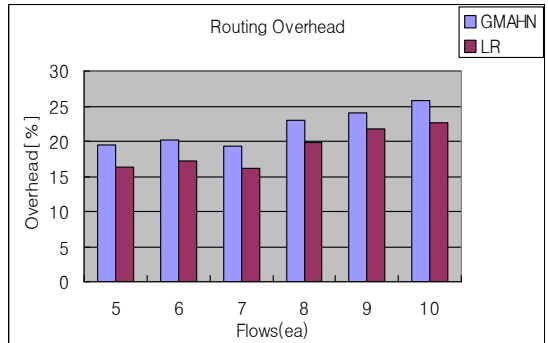


Fig. 13. Routing Overhead - Flows
그림 13. 라우팅 오버헤드 - Flow 수

그림 12와 13은 네트워크 내에서 이동노드 중에 소스노드와 목적지 노드의 데이터 통신을 이루는 Flow의 증가에 따른 데이터 수신율과 라우팅 오버헤드를 보여주고 있다.

전체적으로 Flow의 증가에 따라 네트워크 내에 많은 오버헤드와 데이터 증가로 인한 데이터 수신율이 감소하고 있다.

이는 Flow 수가 많아질수록 각각의 Flow에서 발생하는 경로단절과 경로복구과정으로 전체 라우팅 오버헤드가 증가하는 결과를 낳기 때문이다.

VI. 결론

본 논문에서는 MANET과 GW를 이용하여 유선 기반망과의 연결을 하여 GMAHN환경에서 소스 이동노

드와 유선 기반 망 호스트 노드사이에서 데이터 통신을 기본 환경으로 구성하여, 경로유지 시 보다 안정적인 성능 향상을 위해 지역경로 복구 알고리즘을 적용한 AODV 라우팅 프로토콜을 다양한 환경변화에서 실험 하였다. 본 실험은 리눅스 환경에서 NS2 시뮬레이터를 사용하여 데이터 전송율과 Routing Overhead를 기준으로 기본 환경인 GMAHN와 LR을 비교·분석 하였다. 결과는 다음과 같다.

Gateway_Distance는 관리범위(멀티 홉)를 정하는 것을 뜻하는데 전체적으로 살펴보면 멀티 홉이 증가함에 따라 관리 영역이 넓어지므로 네트워크성능은 향상되었다. 하지만 5홉을 넘어 가면서 성능의 감소를 나타내는데 그 이유는 넓은 지역의 관리로 인해 Overhead의 증가로 오히려 네트워크 성능저하 현상이 나타났다. 따라서 본 실험을 통하여 5홉까지의 범위를 관리 하는 것이 가장 효율적이라는 것을 확인할 수 있다.

네트워크 안에서 이동노드 수의 증가는 각각의 노드들은 이웃노드들의 채널 공유로 인해 패킷충돌 및 채널감지를 위해 필요한 노드자체내의 네트워크 리소스 사용 등 실제 네트워크 성능에 많은 영향을 미치게 된다. 실험에서는 이러한 환경에서 기존 환경인 GMAHN와 지역경로 복구 알고리즘을 적용하여 비교한 결과 성능이 약간 향상되었다.

Flows의 증가는 기본 환경의 실험 결과 전체적으로 Flow의 증가에 따라 네트워크 성능이 감소되었지만 기존 방식보다 데이터 수신율의 증가와 오버헤드 감소를 확인할 수 있었다.

이것으로 MANET 환경에 기반 망을 연결한 GMAHN에서 지역경로 복구 알고리즘을 적용하여 다양한 실험을 해본 결과, 데이터 전송률, Routing Overhead 부분에서 전체적인 성능 향상을 확인하여 본 알고리즘이 네트워크의 오버헤드 최소화에 접근한 것을 알 수 있었다.

앞으로 연구되어야 하는 부분은 지역경로복구 알고리즘을 최적화 하여 성능 향상과 다양하고 실제와 비슷한 환경에서의 연구가 필요하다.

참고문헌

[1] Elizabeth M.Royer and et. al., "A Review of

Current Routing Protocols for Ad Hoc Mobile Wireless Networks,"IEEE Personal Communication, pp.46-55, Apr, 1999

[2] S.Corson, J.Macker, and et. al., "Architectural considerations for Mobile mesh Networking," "http://tonnant.itd.nrl.navy.mil/mmnet/mmnetRFC.txt, 1996.

[3] David B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," IEEE Workshop on Mobile Computing Systems and Application, Dec, 1994.

[4] C. E. Perkins, E. M. Royer, and et. al., "Ad Hoc On-Demand Distance Vector (AODV) Routing," Internet Draft draft-ietf-manet-aodv-03.txt, Jun, 1999.

[5] J. Broch, David B. Johnson, and et. al., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft draft-ietf-manet-dsr-03.txt, Oct, 1999.

[6] Ryuji Wakikawa and Jari T. Malinen and Anders Nilsson and Antti J. Tuominen, "Global connectivity for IPv6 Mobile Ad Hoc Networks" draft-wakikawa-manet-globalv6-03.txt, IETF Internet draft, Oct, 2003, Work in progress.

[7] Clif Flynt, "Tcl/TK for Real Programmers," AP PROFESSIONAL, Academic Press, 1994.

저 자 소 개

장 양 근 (정회원)



2001년 : 세명대학교 전자공학과 졸업 (공학사)
2007년 : 명지대학교 대학원 전자공학과 (공학석사)
<주관심분야> Ad-Hoc Network, Embedded System

김 재 호 (학생회원)



2006년 : 명지대학교 전자공학과 졸업 (공학사)
2006년 3월 ~ 현재 : 명지대학교 대학원 전자공학과 (석사과정)
<주관심분야> GMAHN, Automatic Robot Computer Network

배 진 승 (학생회원)



2002년 : 명지대학교 전자공학과 졸업 (공학사)
2004년 : 명지대학교 대학원 전자공학과 (공학석사)
2004년 3월 ~ 현재: 명지대학교 대학원 전자공학과 (박사과정)
<주관심분야> Embedded system, Automatic Robot

Automatic Robot

하 재 승 (정회원)



1991년 : 명지대학교 전자공학과 졸업 (공학사)
1993년 : 명지대학교 대학원 전자공학과 (공학석사)
2001년 : 명지대학교 대학원 전자공학과 (공학박사)
2005년 ~ 현재 : 명지전문대학 정보통신과 부교수

<주관심분야> SDR(Software Defined Radio), Mobile IP, Ad-Hoc

유 충 렬 (정회원)



1988년 : 명지대학교 전자공학과 졸업 (공학사)
1991년 : 명지대학교 대학원 전자공학과 (공학석사)
1996년 : 명지대학교 대학원 전자공학과 (공학박사)
2003년 ~ 현재 : 명지대 방목기초 교육대학 조교수

<주관심분야> Microprocessor, Computer Architecture

김 현 옥 (정회원)



1978년 : 고려대 전자공학과 졸업 (공학사)
1980년 : 고려대 대학원 전자공학과 (공학석사)
1987년 : 고려대 대학원 전자공학과 (공학박사)
1980년 ~ 1981년 : 동양전문대 전자공학과 전임강사

1981년 ~ 1988년 : 명지대 전자공학과 전임강사
1990년 ~ 현재 : 명지대 전자공학과 정교수
<주관심분야> Embedded system, Ad Hoc Network, MPEG-4

이 광 배 (정회원)



1979년 : 고려대학교 전자공학과 졸업 (공학사)
1981년 : 고려대학교 대학원 전자공학과 (공학석사)
1984년 ~ 1986년 : Univ. of Southern California, 컴퓨터공학 공학석사
1986년 ~ 1991년 : Arizona state Univ. 컴퓨터공학 공학박사

1992년 ~ 현재 : 명지대학교 전자공학과 정교수
<주관심분야> Embedded system, Ad Hoc Network, Computer Network