

DTN과 MANET이 혼재된 네트워크 환경에서 신뢰성있는 메시지 전송 기법

박종권^o, 이준, 홍충선

경희대학교 컴퓨터공학과

jkpark@networking.khu.ac.kr, junlee@networking.khu.ac.kr, cshong@khu.ac.kr

Reliability Message Transmission Scheme for Mixed DTN and MANET Environment

Jong Kwon Park^o, Jun Lee, Choong Seon Hong

Department of Computer Engineering, Kyung Hee University

요 약

Delay-Disruption Tolerant Network(DTN)는 stored-and-forward를 통한 비대칭 통신을 하며 이러한 통신에 적합한 라우팅을 고려하기 위한 연구가 지속적으로 진행 되고 있다. 이러한 라우팅은 Bundle layer에 스토리지를 두어 종 단간 연결이 끊어지더라도 메시지를 보존 할 수 있으며, 현재 이러한 DTN의 특징을 Mobile Ad-hoc Network(MANET)에 적용하여 노드의 이동성을 보다 다이나믹하게 유지 할 수 있도록 하는 연구가 진행되고 있다. 본 논문에서는 이동성이 잦은 지역에서 메시지를 효과적으로 전송하기 위하여 MANET과 DTN의 특징이 적용된 네트워크 환경에서 AODV 라우팅 테이블 안에 Score를 두어 경로 재탐색이나 메시지 전송이 반복될 경우 Score값을 기준으로 보다 우월한 노드를 찾을 수 있게 하였다.

1. 서 론

Delay-Disruption Tolerant Network(DTN)는 지상에서 우주로 보낸 탐사선 사이의 통신이나 행성간의 통신, 위성 통신과 같이 지연 시간이 분, 시간, 일 이상의 단위인 네트워크와 지연시간의 단위가 초 단위 이하인 지상의 인터넷을 연결하기 위해 설계 되었다[1].

미국 항공우주국(NASA)는 인터넷으로 행성간 데이터를 주고받을 수 있는 프로젝트를 시행하였고, 현재 사용 중인 TCP/IP 를 행성 간 통신에 적용할 수 없기 때문에, 이에 대한 고려 사항으로 DTN 이라는 새로운 개념의 네트워크를 만들게 되었다. 현재 DTN 은 네트워크 인프라가 충분히 갖추어지지 않은 저개발국가 또는 통신 환경이 열악한 재해 지역에 대하여 그 적용 분야가 넓혀져 가고 있으며, 지역적 기술적으로 독립적인 여러 네트워크를 하나로 융합할 수 있는 차세대 인터넷 기술이다.

유비쿼터스 네트워킹을 위해서는 기존의 다양한 네트워크 인프라를 표시 없이 연동할 수 있는 기술이 필수적이므로 이런 측면에서도 DTN 에 관한 연구가 진행되고 있다.

현재 우리가 사용하는 TCP/IP 나 WSN(Wireless Sensor Network) 환경에서는 Source 와 Destination

사이의 종단간 경로가 반드시 존재 하여야 하고, RTT(Round Trip Time) 시간이 길어지면 세션이 끊어지며, 손실이 발생 할 수 있기 때문에 DTN 을 활용하여 노드 사이의 메시지 단절 또는 그에 준하는 지연에 대비 할 수 있다.

Mobile Ad-hoc Network(MANET)는 기반 네트워크가 존재하지 않거나 설치가 용이하지 않은 지역에서, 필요에 따라 고정된 기반 네트워크와는 독립적으로 이동 단말들이 자율적이고 임시적으로 구성하는 네트워킹 기술이며, 이러한 네트워크 토폴로지 변화에 알맞은 라우팅 프로토콜로 Reactive 방식인 Ad hoc On-demand Distance Vector (AODV) routing Protocol 을 사용한다[2].

본 연구는 MANET 과 DTN 이 혼재된 네트워크 환경에서 신뢰성 있는 메시지 전송을 위한 연구 이다[3]. 기존 MANET 환경에서 통신하던 노드는 AODV 특성상 RREQ 패킷이 완전히 Source 까지 전달 되어야만 세션이 맺어지고 통신이 가능하다. 그러나 통신 중간에 연결이 끊어지는 상황이 발생하면, DTN 의 전송 전략을 사용하여, 메시지를 저장하고 연결이 이루어지면 전송 할 수 있도록 한다. 또한 추가적으로 AODV 라우팅 테이블에 Score 를 기록하여, 이전 노드의 신뢰도를 판별하여, 보다 우월한 성능의 노드에게 메시지를 전달 할 수 있도록 하였다. 이러한 시나리오를 검증하기 위하여 OMNet++ 시뮬레이터를 사용하여, MANET 시뮬레이션 모듈을 수정하여 DTN 의 특징을 추가하였다.

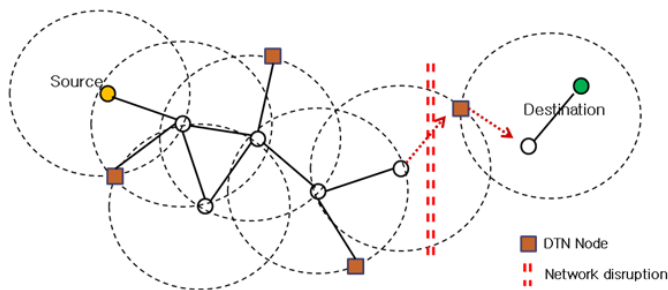
“본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구 센터 지원 사업의 연구결과로 수행되었음”
(NIPA-2010-(C1090-1031-0005)),

Dr. CS Hong is the corresponding author.

본 논문은 다음과 같이 구성되었다. 2 장에서는 현재 DTN 관련하여 진행 중인 관련 연구에 대하여 기술하였고, 3 장에서는 논문의 제안사항을 기술하였고, 4 장에서는 제안 사항을 테스트하기 위한 시뮬레이션 환경에 대하여 작성하였다. 마지막으로, 5 장에서는 결론 및 앞으로의 연구 방향에 대하여 기술하였다.

2. 관련 연구

최근 DTN 의 Bundle 프로토콜을 AODV 라우팅 프로토콜에 적용하여 ad hoc 네트워크상에서 DTN 기반으로 통신이 가능하도록 하였다[3]. 또한 DTN 의 라우팅 방법을 네트워크 토폴로지 변경이 빈번하거나, 메시지 전달 확률이 낮은 환경에 적용하려는 연구가 활발히 이루어지고 있다. 본 논문에서 연구된 환경은 MANET 과 DTN 이 혼재된 환경이며, 각 노드들의 정보를 최신으로 유지하고 주기적으로 테이블 정보를 업데이트해야하는 Proactive 방식은 오버헤드가 매우 크므로, Reactive 방식을 사용한다. 또한, 이러한 모바일 환경에서 이동해 오는 중간 노드와의 연결을 위해서는 멀티 홉 라우팅 기법도 필요하다. 따라서 본 논문에서는 위의 [3]에서 제시한 것과 같은 프레임 워크에서 AODV 라우팅 프로토콜을 기반으로 하였으며, 네트워크 단절 및 지연이 발생할 경우 DTN 포워딩 방식을 사용할 수 있도록 시뮬레이션 환경을 구축 하였다. MANET 과 DTN 과 같은 네트워크 환경에서는 노드 이동성, 불안한 링크 상태, 열악한 기반 시설과 높은 지연시간으로 인해 연결이 불안정할 수 있다.



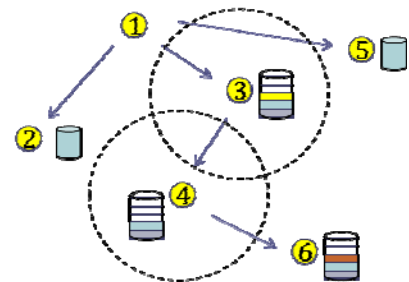
[그림 1] MANET 과 DTN 이 혼재된 환경

위의 그림 1 은 MANET 과 DTN 이 혼재된 환경을 나타낸다. Ad hoc 네트워크는 모바일 노드의 이동으로 인한 토폴로지의 작은 변경에 대처 할 수 있어야 하며,

새로운 노드가 예고 없이 네트워크에 합류하거나 기존의 노드가 삭제되거나 전원이 차단 될 수도 있다. 따라서 이러한 네트워크 토폴로지 특성을 반영하기 위해 DTN 노드를 이용하여 증간에 연결이 끊어질 경우 DTN 노드를 이용하여 패킷을 전송 할 수 있도록 할 수 있다. 현재까지 이러한 네트워크 환경을 고려한 라우팅 기법이 지속적으로 연구되고 있다[3],[4].

3. 제안 사항

본 논문에서는 노드들의 이동성 범위를 제한하였는데, 적어도 한번은 경로가 설정 되었었으며 그 경로들 상에 새로운 노드가 추가 되거나 노드의 파워가 소멸 되었을 경우를 선택하기 위하여, 라우팅 테이블에 Score 를 두었다. 해당 노드에 패킷 수신에 성공할 경우 노드는 자신의 라우팅 테이블에 score 값을 1 씩 증가 시킨다. 따라서 해당 노드의 score 값이 높으면 패킷 전송 성공률도 증가하게 되므로, 패킷을 재전송할 경우 노드의 RSSI 범위에 드는 노드들의 score 값을 비교하여 높은 쪽으로 패킷을 전송하게 한다. AODV 특성상 RREP 가 돌아와야만 경로가 맺어지게 되는데 만약 이러한 노드가 RREQ 를 받지 못하는 경우 해당 노드는 자신의 score 값을 감소시킨다. 따라서 처음 브로드캐스팅에서 자신이 메시지를 받고 score 값을 증가 시켰더라도 RREP 패킷 수신에 되지 않을 경우에는 본래의 값을 유지하게 된다.



[그림 2] RREQ 패킷 전송 절차

위의 그림 2 는 RREQ 패킷 전송 절차를 보여주고 있다. 노드 1 에서 패킷을 노드 2,3,5 에 브로드캐스팅 하면 패킷을 받은 노드들은 자신의 라우팅 테이블에 score 를 1 씩 증가 시킨다. 패킷이 3 에 전달되면 자신의 score 를 증가시키고 4, 6 의 순서로 패킷이 전송된다. 마찬가지로 RREP 패킷이 역경로를 따라서 전달됨에

따라 노드 4,3 는 score 를 감소 할 필요가 없지만, 2,5 번 노드는 RREP 패킷을 받지 못하였으므로, score 값을 감소시킨다. 초기 score 값은 모두 0 으로 설정하였다.

[표 1] 라우팅 테이블 상태

source	Des-	hop	next hop	score++	score--
1	6	2	-	1	1
		5	-	1	1
		3	4	1	
		4	6	1	
		6			

위의 표 1 는 그림 2 에서의 시나리오를 나타내었다. 표에서 와 같이 노드 2 와 노드 5 는 초기 값이 1 로 증가하였다가 다시 0 으로 감소하게 된다. 이러한 매카니즘을 적용하여 노드는 메시지를 재전송 할 경우 라우팅 테이블의 스코어 값을 기본으로 보다 신뢰성을 향상시킬수 있으며, 본 문에서는 OMNet++에서 제공하는 timeout 값을 적용하여 해당 노드의 score 값을 주기적으로 초기화 하도록 설정하였다. 또한, 노드 연결이 끊어졌을 경우에는 DTN 전송 방식을 대입하여 전송 성공률을 높이는 것을 본문에서 제안 하였다.

[표 2] Score 알고리즘

```
DTNModule
if(Recieve Message)
  From Destination
  {
    state=LISTEN;
    if(Packet == RREQ)
    {
      routingtable_score++;
      ForwardMessage(ReceivedMsg);
    }
  }
else// Reverse Path
{
```

```
can't not reveve RREP;
routingtable_score--;
}
}
}
void DTNModule::ForwardMessage
{
SendMessage;

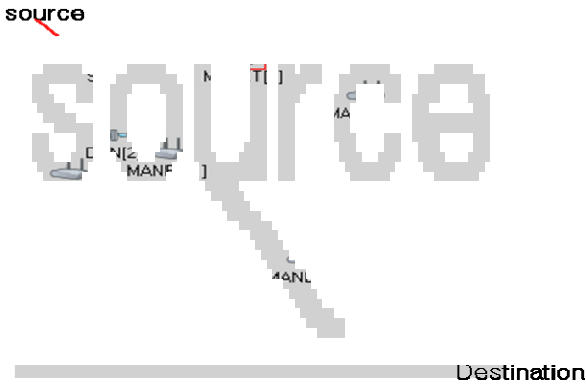
setHopCount( getHopCount()+1 );
neighbor_node=GetNextNdoe();
PathToNextNode=simulation.getModuleByPath(neighbor_node);
if (state==TRANSMIT)
{
state = LISTEN;
cPacket *pkt = new cPacket(SendMeesage);
SetPacketBitLength(pkt);
sendDirect(pkt, radioDelay, duration, PathToNextNdoe->gate("input"));
}
}
```

위의 표 2 는 앞에서 제시한 내용의 세부적인 알고리즘을 코드로 보여주고 있다. 메시지를 수신하였을 경우 패킷을 RREQ 패킷을 판별하여 score 값을 증사시키도록 하였으며, ForwardMessage 함수를 호출 하면 해당 패킷의 홉 수를 1 증가하도록 조정하였다.

4. OMNet++ 시뮬레이션 환경 [4],[5]

우리는 본문에서 제시한 환경을 위하여 MANET 과 DTN 이 혼재된 환경을 그림 3 와 같이 구현하였다. 각각의 MANET 노드들과 DTN 노드들은 임의의 값으로 이동성을 가지고 있으며, 출발지 노드를 MANET 노드로

정하였으며, 목적지 노드 또한 같은 MANET 노드이다. 두 노드는 static으로 설정하였다.



[그림 3] OMNet++에서 test 환경

두 노드 사이에 이동 하는 노드들을 통해 메시지가 전달 되므로 고정된 2개의 소스 노드와 목적지 노드를 설정 하였고, 일정한 패턴이 있다고 가정하였다. 따라서 해당 노드들은 일정한 범위를 두고 이동하게 된다.

[표 3] 시뮬레이션 초기값 설정

**mobility.changeInterval = truncnormal<2s, 0.5s>
**mobility.changeAngleBy = normal<0deg, 30deg>
**mobility.speed = truncnormal<30mps, 10mps>
**mobility.yupdateInterval = 100ms

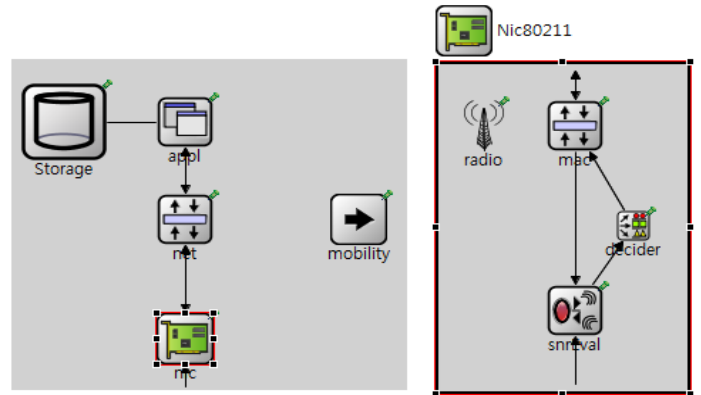
위와 같은 이동성을 위해 표 3 과 같이 노드의 이동 속도 및 전송 간격등을 정의 하였다. *changeInterval* 은 노드의 이동속도를 바꿔주는 값이며, *changeAngleBy* 는 노드의 이동 방향을 바꿔준다. *updateInterval* 은 호스트가 움직일 때 위치를 채널에 보고하는 주기를 나타내며, *speed* 는 노드의 이동 속도를 나타내고 있다.

4.1 OMNet++ 모듈

본문에서 제시하는 환경을 테스트하기 위해 우리는 OMNet++ 상에서 아래의 그림 4 와 같이 모듈을 구현하였다. 현재 까지 제시된 DTN 환경 구현 및 테스트 베드를 구현하기 위해서는 많은 제약이 있기 때문에 아래와 같이 테스트 모듈을 구현 하였다. 정통적인 DTN 테스트 베드 구현을 위해서는 다양한 환경을 요구한다[6].

기본 NIC 모듈은 OMNet++에서 지원하는 모바일 프레임워크 모듈을 사용하였으며, 네트워크 레이어 및 어플리케이션 레이어를 구현하였고, Bundle 레이어의

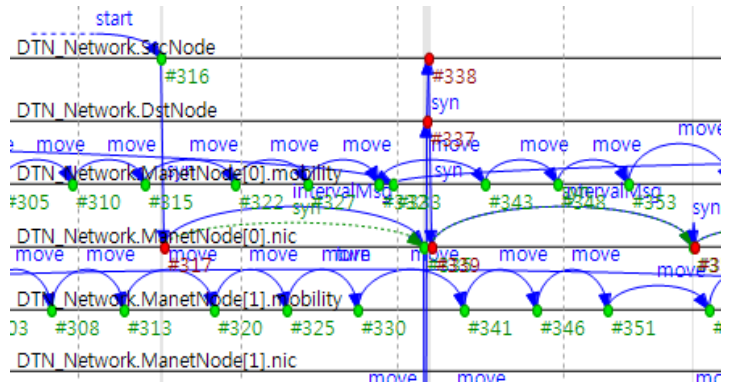
Store-and-Forward 기능을 적용하기위해 storage 를 queue 로 구현하였다[7].



[그림 4] 기본 모듈 설정

4.2 성능 평가

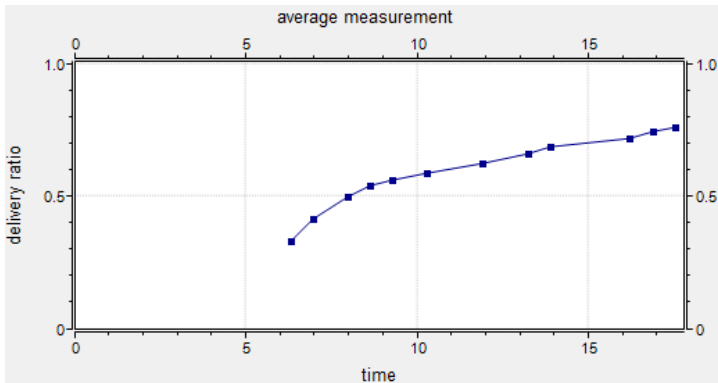
OMNet++에서는 패킷이 전송된 결과를 시뮬레이션 할 수 있는 기능을 제공하고 있으며, 그림 5 와 같이 패킷 전송 시간 별로 패킷이 각 노드에서 점프 되는 과정을 log 로 기록하여 볼 수 있도록 해준다.



[그림 5] 시간 당 패킷 전송 Log

화살표는 패킷이 출발지에서 목적지로 메시지가 전송되는 상황을 그림으로 보여주고 있다. 이러한 log 를 이용하여 패킷이 전송된 경로를 확인할 수 있다. 처음 6s(6 초) 소스 노드(srcNode)로부터 메시지 전송이 시작되었음을 알 수 있으며, 세로축으로 나타난 화살표는 메시지가 전송된 경로를 나타내며, 가로축 화살표는 메시지가 시간 동안 이동된 거리를 나타내고 있다.

그림 6에서는 현재 제안한 사항을 측정된 결과를 그래프로 나타내고 있다. 시뮬레이션 특성상 처음 노드들이 한곳에 모여 있다가 퍼지기 때문에 노드가 이동할 시간을 두기 위하여 처음 6 초 동안 노드가 이동할 수 있도록 시간을 두었다.



[그림 6] 평균 값 측정 결과

처음 전송 후 일정 시간이 지날수록 패킷 전송률이 시간에 대비하여 지속적으로 증가함을 볼 수 있다. 이러한 값을 구하기 위하여 처음 전송 시간 - 마지막 패킷 수신 시간을 계산하여 패킷이 도착할 때 마다 그래프 상에 기록하도록 하였다. 그림에서 보면 알 수 있듯이 처음 출발지에서 목적지까지의 패킷 도착 비율이 시간이 지남에 따라 향상되었다.

5. 결론 및 향후 계획

본 논문에서는 MANET 과 DTN 환경에서 메시지 전송 신뢰성을 향상시키기 위하여 라우팅 테이블을 조정하여 해당 패킷의 전송률을 향상 시켰다.

.OMNet++을 사용하여 DTN 및 MANET 의 세부적인 특징을 완벽히 시뮬레이션 하기에는 많은 어려움이 있다. 특히, DTN 의 경우 구현상에 많은 제약이 따르며, 이동성을 지원하기 위한 라우팅 기법과 번들 프로토콜의 특성을 구현하는데 많은 어려움이 있다. 현재까지 이러한 DTN 및 MANET 의 특성에 관한 연구는 많은 관심을 모으고 있으며, 앞으로도 지속적으로 많은 연구가 이루어질 것이다.

참 고 문 헌

[1]Delay Tolerant Networks(DTNs), A Tutorial, 2003
<http://www.dtnrg.org>
 [2]C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc On-Demand Distance Vector(AODV)Routing. Experimental RFC 3561, July 2003
 [3]J. Ott, D. Kutscher, and C. Dwertmann. Integrating DTN and MANET Routing. In CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks, NY, USA: ACM: 221-228, 2006
 [4]Christian Kretschmer, Stefan Ruhrup, christian Schindeelhauer. DT-DYMO: Delay-tolerant Dynamic MANET On-demand Routing. 29th IEEE International Conference on Distributed Computing Systems Workshops, 2009
 [5]Olafur Ragnar Helgason, Kristjan Valur Jonsson. Opportunistic Networking in OMNet++. OMNet++2008: Proceedings of the 1st International Workshop on OMNet++, ACM978-963, 2008
 [6]Ari Keranen and Jorg Ott, Helsinki University of Technology. Increasing Reality for DTN Protocol Simulations. 2007
 [7]K. L. Scott and S. C. Burleigh. Bundle Protocol Specification. Internet Draft draft-irtf-dtnrg-bundle-spec-02, Work in progress, September 2005