

상황인식 이동성 예측 모델에서의 효율적인 포워딩 경로 산출 기법

정래진 · 오영준 · 이강환*

*한국기술교육대학교

Efficient Forwarding Path Computing Method for Context-Awareness Mobility Prediction Model

Rae-jin Jeong · Young-jun Oh · Kang-whan Lee*

*Korea University of Technology and Education

E-mail : jrj8819@koreatech.ac.kr

요 약

본 논문은 계층적 클러스터 구조에서, 상황인식 이동성 예측모델을 통해 예측된 이동성으로 효율적인 포워딩 경로를 산출하는 기법을 제안한다. 이동성으로 인하여 노드 간 연결이 쉽게 변화하는 환경에서 통신의 불안정성을 극복하기 위해, 제안하는 알고리즘은 상황인식 이동성 예측모델을 통해 얻은 클러스터의 예측된 속성정보를 활용한다. 예측정보를 통해 클러스터간의 연결성을 매트릭스의 형태로 정리하며, 마스킹 기법을 응용한 포워딩 경로 산출 기법을 통해 효율적인 경로를 도출한다. 모의실험 결과, 제안하는 알고리즘으로 클러스터의 이동성을 고려하여 노드 간의 연결이 오래 지속될 수 있는 포워딩 경로를 선택 후에 전송함으로써 지연시간이 줄어드는 결과가 기대된다.

ABSTRACT

In this paper, we proposed efficient forwarding path computing method using Context-Awareness Mobility Prediction Model. Context-Awareness Mobility Prediction Model is storing and classifying node's previous velocity and direction according to time in the hierarchical cluster structure. To overcome environment which node-to-node connection is broken off easily, the proposed algorithm calculate the connectivity formed matrix structure by comparing predicted velocity and direction, and use masking operation for selecting relay moving to destination. The proposed algorithm identified to show short delay by utilizing forwarding path which is continue node-to-node connection in the unstable situation.

키워드

Context-awareness, Connectivity, Masking, Mobility, Prediction

I. 서 론

이동노드들이 다양한 이동성을 보이는 MANET (Mobile Ad-hoc Network) 환경에서는 필연적으로 네트워크 토폴로지가 계속 변화한다.[1] 따라서 종단간의 연결이 불안정적인 단점이 나타났다. 이를 극복하기 위해 종단 간의 경로 선정 없이 이동성을 가진 중계노드를 통해 데이터를 전달하는 DTN (Delay Tolerant Networks)가 제시되었다.[2]

DTN에서 소스노드는 자신의 이웃노드들에게

데이터를 전송하여 이웃노드의 버퍼에 저장한다. 저장된 데이터는 이웃노드의 이동성으로 인하여 네트워크 영역을 이동하게 되며, 도중에 확립되는 다른 노드들에 대하여 데이터를 전달한다. 데이터를 전달받은 노드들은 자신이 목적노드일 경우 데이터를 수용하며, 아닐 경우 중계노드가 되어 전체 네트워크에 데이터를 확산시킨다.[3]

하지만 통신의 지연시간이 길어졌으며, 제한된 저장 용량으로 오버플로우 및 오버헤드가 나타났다. 따라서 본 논문은 클러스터의 속도와 방향을

고려하여 선택적인 포워딩 경로를 확립함으로써 짧은 지연시간 동안에 데이터를 전달하였다.

II. 상황인식 이동성 예측 기법에서 포워딩 경로 산출 기법

상황인식 이동성 예측모델의 통해 얻어진 클러스터의 속성정보[4]를 활용하여 해당 클러스터와 베이스스테이션과 연결성을 판단할 수 있다. 연결성은 네트워크 영역에서 베이스스테이션을 향하고 있으며, 속도가 빠를수록 더 큰 값을 가지며 수식 1을 활용하여 구할 수 있다.

$$CN_{BS,i}^v = \begin{cases} \frac{(1 - \sin(Dir_{quad} - Dir_i)) \times v_i}{\sum_{i' \in NH} v_{i'}}, & (|Dir_{quad} - Dir_i| < \frac{\pi}{2}) \\ 0, & otherwise \end{cases} \quad (1)$$

위 수식에서 NH는 헤드노드 i 의 이웃 헤드노드들의 집합을 이르며, $\sum_{i' \in NH} v_{i'}$ 은 이웃한 헤드노드들의 속도의 합을 나타낸다. Dir_{quad} 는 베이스스테이션을 기준으로 4등분으로 나누어진 각각의 영역에서 베이스스테이션을 향하는 벡터의 방향이다.

제안하는 포워딩 경로 산출 기법은 헤드노드마다 자신이 이웃하는 헤드노드들 간의 연결을 인접행렬로 나타낸다. 데이터 전송 시 헤드노드는 자신의 인접행렬을 참조하여 이웃 헤드노드들을 확인하며, 이웃 헤드노드 중 연결성이 높은 헤드노드에게 그림 1과 같이 데이터를 포워딩한다.

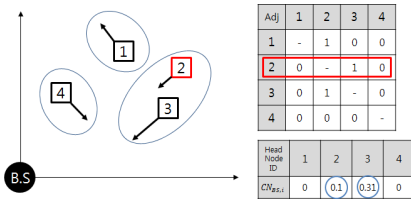


그림 1. 이웃 클러스터 헤드노드의 연결성에 따른 포워딩

그림 1에서 데이터를 전송하려는 헤드노드 2는 자신의 인접행렬로 헤드노드 1과 3이 연결되어 있음을 확인한다. 확립된 연결을 통해 각 헤드노드와 베이스스테이션 간의 연결성을 참조하여 연결성이 높은 헤드노드 1에게 데이터를 포워딩한다.

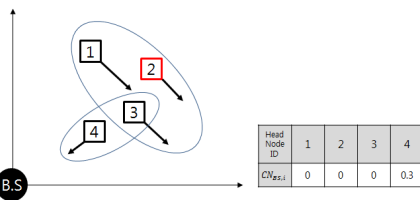


그림 2. 이웃 헤드노드가 베이스스테이션을 지향하지 않는 상황

그림 2는 헤드노드 2의 이웃 헤드노드들의 연결성이 없는 경우를 나타내고 있다. 이러한 경우에 헤드노드 2는 그림 3과 같이 자신의 인접행렬과 이웃 헤드노드의 인접행렬들을 병합한다,

Adj of merged	1	2	3	4
1	-	1	0	0
2	1	-	1	0
3	0	1	-	1
4	0	0	1	-

Adj of CH2	1	2	3	4
1	-	1	0	0
2	1	-	1	0
3	0	1	-	0
4	0	0	0	-

Adj of CH3	1	2	3	4
1	-	0	0	0
2	0	-	1	0
3	0	1	-	1
4	0	0	1	-

Adj of CH1	1	2	3	4
1	-	1	0	0
2	1	-	0	0
3	0	0	-	0
4	0	0	0	-

그림 3. 병합된 이웃 헤드노드의 인접행렬

이웃 헤드노드들의 인접행렬을 병합함으로써 포워딩할 헤드노드를 더 넓은 범위에서 찾을 수 있게 된다. 그림 2에서 헤드노드 2의 이웃 헤드노드들 중 연결성을 가지는 헤드노드가 없으므로 연결성이 높은 제3의 헤드노드와 연결이 확립되어 있는 이웃 헤드노드를 선택한다.

이 과정에서 경로의 중복을 방지하기 위해 마스킹 연산을 활용하여 이미 포워딩을 거친 헤드노드에게 전송하지 않도록 그림 4와 같이 인접행렬을 수정한다.

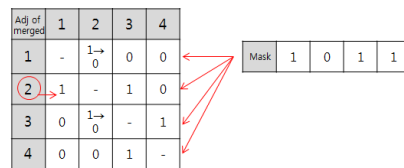


그림 4. 경로의 중복을 방지하기 위한 마스킹 연산

위의 그림 4에서 헤드노드 2는 연결을 확립하고 있는 헤드노드 1에 대해서 임시 경로를 정하고 자신에게 경로가 돌아오지 않도록 인접행렬에서 자신이 속한 열을 0으로 처리한 마스크를 인접행렬의 각 행들과 마스킹 연산하여 값을 0으로 변환한다. 마스킹 연산 후 인접행렬에서 헤드노드 1과 연결되어 있는 헤드노드가 없으므로 이전의 헤드노드 2에 대해 경로 산출 과정을 이어간다. 다음으로 헤드노드 2와 연결을 확립하고 있는 헤드노드 3에 대하여 그림 4와 같은 방식으로 마스킹 연산을 처리하면 그림 5와 같다.

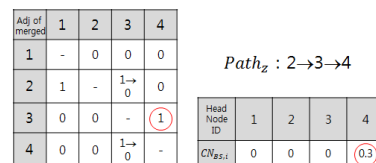


그림 5. 산출된 포워딩 경로

그림 5에서 헤드노드 2는 헤드노드 3에 대해서 임시경로를 선정하였으며 마스킹 연산을 통해 헤드노드 3으로 전송하는 경로를 인접행렬에서 제거하였다. 이어서 헤드노드 3과 연결된 이웃노드들 중에 헤드노드 4가 베이스스테이션과 연결성을 가지고 있으므로 헤드노드 2와 3과 4를 거치는 포워딩 경로를 산출하였다.

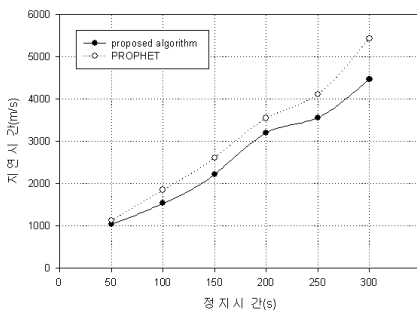
III. 실험 및 분석

본 논문에서 제안하는 포워딩 경로 산출 기법의 성능을 비교하기 위해 PROPHET 알고리즘[5]을 대상으로 하여 일정한 정지시간에서 패킷이 베이스스테이션까지 도달하는 시간을 측정하였다.

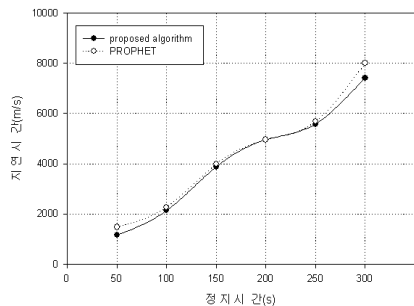
표 1. 모의실험 환경

항목	내용
헤드노드의 수	16 EA
노드의 속도	10, 15 m/s
네트워크 영역	1000×1000 (m)

모의 실험 결과를 살펴보면 제안하는 알고리즘이 PROPHET 알고리즘보다 더 빠르게 데이터를 전달한 것을 확인할 수 있다. 하지만 노드의 이동 속도가 빨라짐에 따라 전체적인 지연 시간은 증가하였으며, 두 알고리즘에서 지연 시간 차이가 줄어들었다.



(a)



(b)

그림 6. 속도에 따른 지연시간 (a)10m/s, (b)15m/s

그림 6의 (a)와 (b)를 통해 노드의 속도가 느릴 때 토폴로지 구조의 변화가 비교적 적어 중단간의 전송이 빠르게 이루어 졌으며, 포워딩 경로 산출에서 이동성을 고려하였을 때 더 빠르게 데이터를 전달할 수 있음을 알 수 있다.

IV. 결 론

본 논문에서는 상황인식 이동성 예측 모델을 통해 예측된 이동성을 포워딩 경로 선정에 적용하는 기법을 제안한다. 이동성으로부터 기인한 네트워크 토폴로지의 변화로 인해 중단간의 전송경로가 쉽게 달라지는 불안정한 네트워크 환경이 나타났다. 이를 극복하기 위해 포워딩과정에서 속도와 방향과 같은 속성정보를 활용하여 연결성을 산출하며, 노드의 연결 상태를 매트릭스의 형태로 정리하여 마스킹 연산을 응용한 포워딩 경로를 구하였고, 산출된 포워딩 경로는 더 짧은 지연시간을 보여주었다. 효율적인 포워딩 경로를 산출하는 기법은 모의실험에서 긍정적인 성능을 보였지만 네트워크의 확장 및 여러 가지 요인으로 인해 발생할 수 있는 문제를 극복하기 위해 추가적인 연구가 필요하다.

참고문헌

- [1] P. Santi, Topology Control in Wireless Ad Hoc and Sensor Networks, pp. 8-11, 2005.
- [2] Mary R. Schurgot, Cristina Comaniciu and Katia Jaffres-Runser, "Beyond Traditional DTN Routing : Social Networks for Opportunistic Communication", Communications Magazine IEEE, vol 50, no. 7, pp. 155 - 162, July, 2012.
- [3] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks", Duke Univ., tech. rep. CS-2000-06, 2000.
- [4] Young-jun Oh, Kang-whan Lee, "Energy conserving routing algorithm based on the direction for Mobile Ad-hoc network", J. Korea Inst. inf. Commun, Eng, vol.17, no.11, page 2699-2707, nov 2013.
- [5] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks", Service Assurance with Partial and Intermittent Resources, 239-254.