

단일홉 무선 애드혹 네트워크에서 단순 TDMA 시스템을 위한 DESYNC 알고리즘 개선 방안

현상현*, 이제율*, 양동민*

*대전대학교 정보통신공학과

e-mail:winet08@pine.dju.ac.kr, jelee@pine.dju.ac.kr, dmyang@dju.ac.kr

An Enhanced DESYNC Scheme for Simple TDMA Systems in Single-hop Wireless Ad-hoc Networks

Sanghyun Hyun*, Jeyul Lee*, Dongmin Yang*

*Dept of Information & Communications Engineering, Daejeon University

요 약

TDMA(Time Division Multiple Access)는 사용자에게 자신에게 할당된 시간구간에서 모든 대역폭을 이용하여 무선 네트워크에서 통신할 수 있는 다중 접속기술이다. TDMA에서 동기화 방식은 동기화를 이루는 방법이 어렵기 때문에 소규모 네트워크에는 적합하지 않다. 본 논문에서는 DESYNC 알고리즘을 이용하여 소규모 네트워크에 적합한 동기화 방식의 TDMA 기법인 C-DESYNC를 제안하고자 한다. C-DESYNC는 DESYNC의 firing 방식을 이용하고, 주기의 시작 정보를 포함하고 있는 Global Packet을 이용하여 안정화상태까지 걸리는 최대 지연시간을 보장한다.

1. 서론

한정된 채널 자원을 공유하여 통신하는 무선 네트워크에서 효율적으로 통신하기 위해서는 TDMA와 같은 다중 접속(Multiple Access)기술이 필요하다. TDMA란 가용 대역폭을 시간구간으로 나누어 자신에게 할당된 시간구간에서 모든 대역폭을 이용하여 통신하는 다중 접속 기술로, 동기화와 비동기화 방식이 있다. 동기화 방식은 동기를 위한 방법이 복잡하고, 동기화를 위한 추가 장비가 필요하기 때문에 소규모 네트워크에 이용하기에 비효율적이다[1].

본 논문에서는 소규모 네트워크에 적합한 동기화 방식의 TDMA 기법을 제안하기 위해 DESYNC 알고리즘을 이용한다. DESYNC 알고리즘은 간단한 규칙과 주변노드들의 정보를 이용하여 스스로 동기화를 유지하는 알고리즘이다[2][3]. 하지만 DESYNC 알고리즘은 안정화가 되기 까지 지연시간을 보장하지 않는 단점이 있다. 제안하는 알고리즘은 DESYNC 알고리즘의 firing 기법을 이용하여, 전체 노드 개수를 파악하고, 이 정보를 이용하여 안정화에 이르는 최대 지연시간을 보장한다. 그리고 동기화를 이루기 위한 방법도 기존의 동기화 방식의 TDMA 기법에 비해 간단하기 때문에 비용측면에서도 효율적이다.

2. DESYNC

DESYNC 알고리즘은 기존의 TDMA 방식과 달리 동기

화를 위한 신호나 추가 작업이 필요 없고, 네트워크에 참여하는 노드의 수에 관계없이 노드들 스스로 동기를 유지시킬 수 있다[2][3].

DESYNC 알고리즘에서 각 노드들은 주기적이고 공정한 펄스신호를 발생시킨다. 이 펄스신호를 firing이라고 하고, 노드는 이 firing 신호를 주기 T 마다 한번 씩 자신의 전송 범위 내에 있는 주변 노드들에게 전송한다.

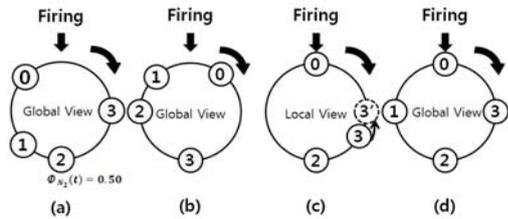
네트워크를 구성하는 노드들이 n 개 존재할 때 각각의 노드를 $N_i(0 \leq i \leq n-1)$ 라 한다. $\Phi_{N_i}(t)$ 는 전체 시간 t 에 주기 T 에서 N_i 이 위치하고 있는 시각을 나타낸다. $\Phi_{N_i}(t) \in [0,1]$ 일 때 $\Phi_{N_i}(t) = 0.65$ 라면 N_i 는 T 에서 65%에 해당하는 지점에 위치한 것을 의미한다. 노드는 $\Phi_{N_i}(t) = 1$ 이 되면 firing하고 $\Phi_{N_i}(t) = 0$ 으로 초기화 한다. T 에서 firing한 노드는 직전에 firing한 노드와 직후에 firing한 노드의 시간차를 이용하여 자신의 다음 firing 시각을 정한다. 노드의 시간차는 $\Delta_{N_i}(t) = \Phi_{N_i}(t) - \Phi_{N_{i-1}}(t)$ 로 표시하며 모든 노드들이 식(1)을 만족하면 안정화 상태가 된다.

$$\Delta_{N_{i-1}}^*(t) = \frac{1}{n} \quad (1)$$

즉, 모든 노드의 firing 간격이 일정해지면 안정화 상태가 된 것이다. 노드의 다음 firing할 시각은 다음 식(2)에 의해 결정된다.

$$\begin{aligned} \tilde{\phi}_{mid}(t) &= \frac{1}{2} [\tilde{\phi}_{N_{i-1}}(t) + \tilde{\phi}_{N_{i+1}}(t)] \\ &= \phi_i(t) + \frac{1}{2} (\tilde{\Delta}_{i+1} - \tilde{\Delta}_i) \end{aligned} \quad (2)$$

그림 1은 DESYNC 알고리즘의 수행 과정을 보여준다. 그림 1에서 큰 원은 T 를 의미하고, 작은 원은 노드를 의미한다. 그림 1-(a)는 DESYNC 알고리즘의 초기 상태를 나타낸다. 노드 N_1, N_2, N_3, N_4 가 초기에 위치하는 시각은 T 상에 임의로 정해진다. 노드들은 시간의 흐름에 따라 큰 원 위에서 시계방향으로 회전한다고 가정하고, 12시에 위치하면 firing한다. 그림 1-(b)에서 다음에 firing할 노드는 N_1 이다. 그림 1-(c)는 노드 N_3 의 Local View를 의미한다. 노드 N_3 는 인접한 노드의 firing 정보를 이용하여 자신이 다음에 firing할 위치를 정하게 된다. 이렇게 여러 주기가 반복되면 모든 노드들의 간격이 그림 1-(d)와 같이 일정해지고, 네트워크는 안정화 상태가 된다.



(그림 1) DESYNC 수행 과정

3. 시스템 모델

DESYNC 알고리즘은 추가적인 동기화 작업이 없고, 간단한 신호를 이용하여 동기를 유지하기 때문에 비용적인 측면에서 효율적이다[2]. 하지만 안정화가 되기까지 지연 시간을 보장하지 않기 때문에 노드의 초기 배치, 노드의 개수에 따라 안정화에 이르는 지연시간이 달라진다. 본 논문에서는 이러한 문제를 해결하기 위한 알고리즘을 제안하고자 한다. <표 1>은 기존의 DESYNC 알고리즘을 수행하기 위한 노드의 기본정보와 제안하는 알고리즘을 수행하기 위한 노드의 기본정보이다. 기존의 DESYNC 알고리즘은 T 에서 인접한 노드의 firing 시간차를 이용하여 동작하기 때문에 이전 노드와의 firing 시간차 $\Delta_{N_i}(t)$, 이후 노드와의 firing 시간차 $\Delta_{N_{i+1}}(t)$, 그리고 노드 N_i 의 firing 시각(Firing Time, FT) 정보를 가지고 있다. 이에 반해 제안하는 알고리즘은 네트워크를 구성하는 전체 노드들의 개수를 파악하여 동작하기 때문에 한 주기 내에서 자신의 firing 시각을 기준으로 이전에 firing한 노드들의 개수(BeforeFiringCount, BFC), 이후에 firing한 노드들의 개수(AfterFiringCount, AFC), 그리고 노드 N_i 의 firing 시각(Firing Time, FT)과 동기를 맞추기 위한 신호인 Global Packet(GP)을 가지고 있다. GP 는 노드의 주기 시

작 시간정보를 가지고 있어 이 신호로 간단히 동기화를 이룰 수 있다.

<표 1> DESYNC와 제안하는 알고리즘의 노드정보

DESYNC : 노드의 기본정보			
$\Delta_{N_i}(t)$	FT	$\Delta_{N_{i+1}}(t)$	
제안하는 알고리즘 : 노드의 기본정보			
BFC	FT	AFC	GP

모든 노드들은 초기에 T 상에 임의로 배치되며 식 (3)의 조건을 만족한다.

$$0 < FT \leq T \quad (3)$$

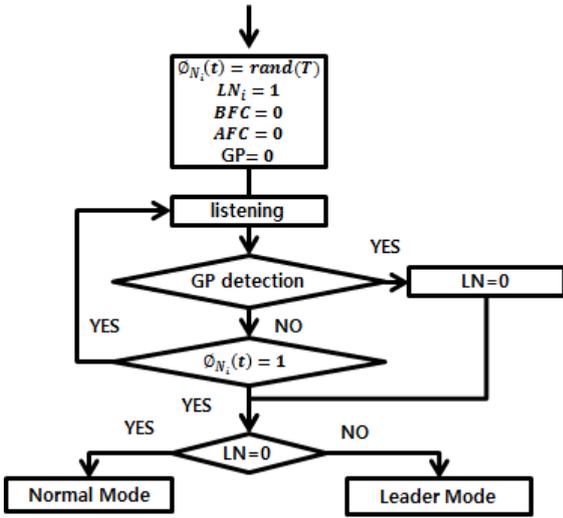
제안하는 알고리즘은 T 내에서 FT 를 기준으로 BFC 와 AFC 를 이용하여 식 (4)를 이용하여 자신의 다음 FT 를 결정한다.

$$\phi'_{N_i}(t) = T - (BFC \times \frac{1}{ACF + BCF + 1}) \quad (4)$$

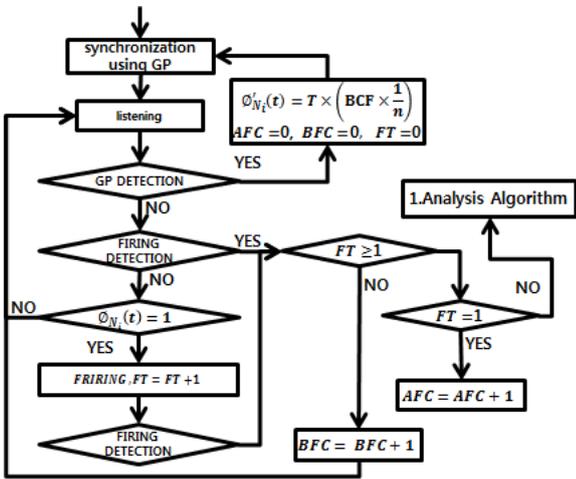
4. C-DESYNC

본 절에서는 DESYNC에서 안정화가 되기까지 걸리는 지연시간을 보장하지 못하는 문제점을 개선하기 위해 C-DESYNC를 제안한다. C-DESYNC는 GP 를 이용하여 동기화를 이루고, DESYNC의 firing 기법을 이용하여 T 에서 자신의 FT 를 기준으로 BFC 와 AFC 정보를 이용해 전체 노드의 개수를 파악하고, 이 정보를 이용하여 노드들의 다음 firing 시각을 정한다. 초기에 배치된 노드들은 T 만큼 대기하며 GP 를 기다린다. GP 를 받은 노드는 일반 노드와 같이 동작하고, GP 를 받지 못한 노드는 자신이 Leader Node(LN)가 되어 T 마다 한번씩 GP 를 전송한다. GP 신호는 노드의 주기 시작 시간정보를 가지고 있어 이 신호를 받은 일반 노드들은 LN 의 주기에 맞춰 동작한다. 그림 2, 그림 3, 그림 4는 C-DESYNC에서 노드들의 동작 알고리즘이다.

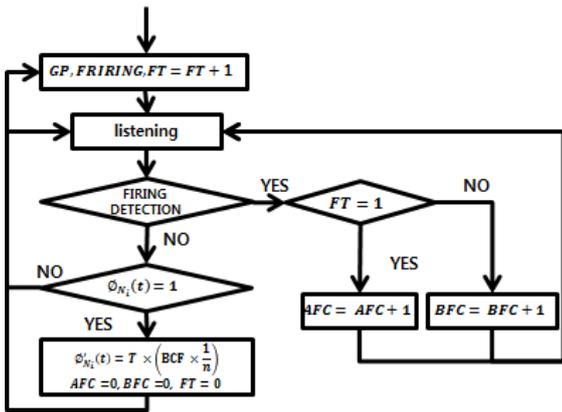
그림 2은 LN 을 선출하기 위한 알고리즘이다. 초기에 노드가 네트워크에 참여 하였을 때 $\phi_{N_i}(t)=1$ 이 까지 기다리며, GP 을 수신했는지 여부에 따라 자신이 일반 노드로 동작할지 LN 으로 동작할지 결정한다. 그림 3는 일반 노드가 자신의 FT 를 기준으로 BFC 와 AFC 의 개수를 구한 후, 자신의 다음 firing 시각을 결정하는 알고리즘이다. 만약 FT 가 1보다 크면 LN 이 제거된 것으로 판단하고 그림 2의 알고리즘을 다시 수행한다. 그림 4은 LN 의 동작 알고리즘을 나타낸다. LN 은 $\phi_{N_i}(t)=1$ 가 되면 GP 를 전송하고, 일반 노드와 같이 firing을 한다.



(그림 2) LN 선출 알고리즘



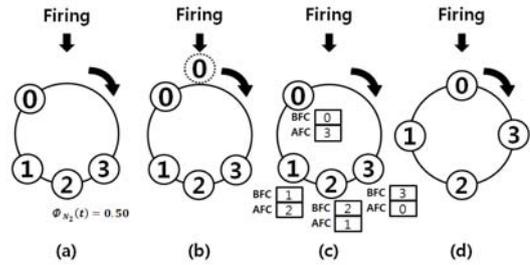
(그림 3) 일반노드 동작 알고리즘



(그림 4) LN 동작 알고리즘

그림 5는 C-DESYNC 알고리즘의 수행 과정을 보여준다. 그림 5에서 큰 원은 T 를 의미하고, 작은 원은 노드를 의미한다. 그림 5-(a)는 C-DESYNC 알고리즘의 초기 상태를 나타낸다. 노드 N_1, N_2, N_3, N_4 는 초기 위치는 T 상에 임의로 정해진다. 모든 노드들은 초기에 배치되었을 때 1 주기 동안 아무런 수행을 하지 않으며 그림 2와 같이 동

작한다. 1주기 이내에 GP가 감지되면 네트워크에 LN이 있는 것으로 판단하고, 그림 3에 따라 일반 노드로 동작하게 된다. 만약 1주기 이내에 GP신호가 감지되지 않았다면 네트워크에 LN이 없는 것으로 판단하고, 그림 4에 따라 LN으로 동작하게 된다. 모든 노드들은 시간의 흐름에 따라 큰 원 위에서 시계 방향으로 회전한다고 가정하고, 12시에 위치하면 firing한다. LN은 firing하기 전에 자신의 주기의 시간정보를 가지고 있는 GP를 전송한다. 그림 5-(a)에서 T 동안 기다린 모든 노드들은 GP를 받지 못하였기 때문에 처음 firing하게 되는 노드인 N_0 가 LN_0 이 된다. 그림 5-(b)는 LN_0 가 보내는 GP에 의해 모든 노드들의 주기 시작 시각이 동기화된 상태이다. LN_0 는 동기를 맞추기 위해 GP를 보냄과 동시에 일반노드의 동작도 수행한다. 그림 5-(c)에서 동기화된 모든 노드들은 T 동안 firing하는 모든 노드들을 BFC와 AFC로 저장한다. 그림 5-(c)에 파악한 전체 노드 개수를 이용하여 식(4)를 이용하여, 모든 노드들은 다음 주기의 firing 시각을 정하고 그 결과 모든 노드들의 간격은 그림 5-(d)와 같이 일정해지고, 네트워크는 안정화 상태가 된다.



(그림 5) C-DESYNC 수행 과정

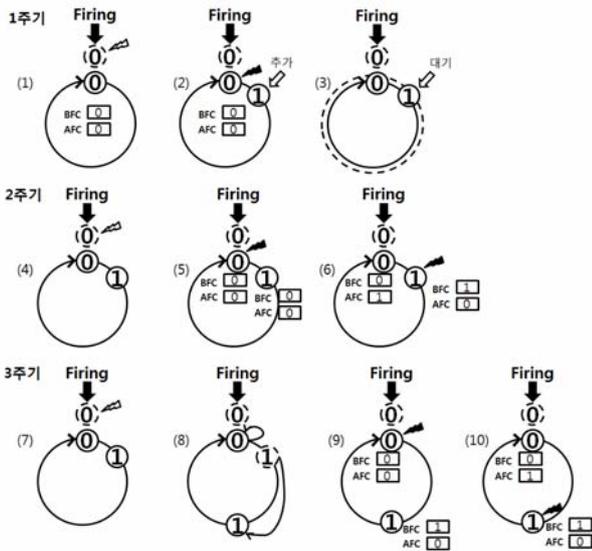
5. C-DESYNC 동작 시나리오

그림 6, 그림7, 그림 8은 C-DESYNC의 상황에 따른 동작 과정이다. 동작 순서는 (1)->(2)->... 로 진행된다.

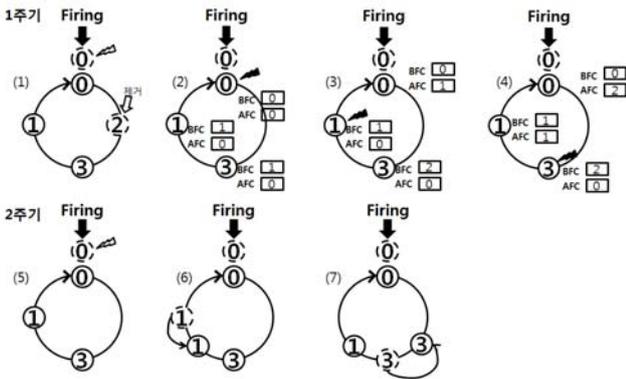
그림 6은 LN_0 가 존재할 때 N_1 이 네트워크에 참여하였을 때를 나타낸다. 그림 6-1주기에서 노드 N_1 이 T 동안 LN이 있는지 확인하기 위해 GP신호를 기다린다. 그림 6-2주기에서 T 내에서 노드 N_1 가 GP신호를 인지하여 동기화되고, 자신의 firing 시각을 기준으로 BFC와 AFC를 파악한다. 그림 6-3주기 이후에는 식 (4)를 이용해 노드 N_0, N_1 가 다음 주기에서 firing할 위치를 정한다. 그림 6-3주기의 (7), (8)은 동시에 일어난다.

그림 7은 LN_0 가 존재하는 네트워크에 참여하고 있던 일반노드 N_0, N_1, N_2, N_3 중 N_2 가 삭제될 때를 나타낸다. 그림 8-1주기에서 N_2 가 삭제되어 나머지 세 개의 노드만 firing한다. 그림 7-2주기에서 남은 노드들은 자신의 firing 시각을 기준으로 BFC와 AFC를 파악하고, 식 (4)를 이용

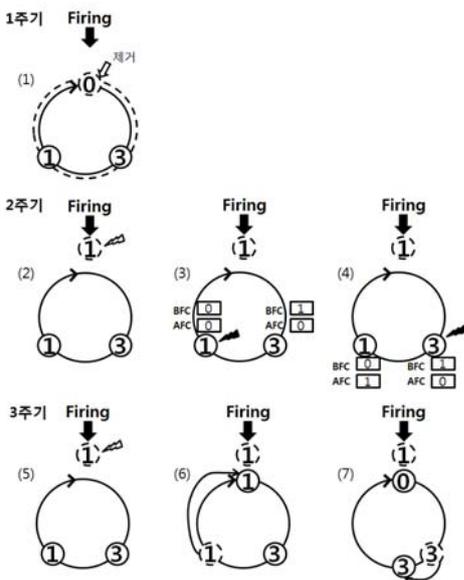
해 다음 주기에서 firing 위치를 정한다. 그림 7-2주기의 (5), (6), (7)은 동시에 일어난다.



(그림 6) C-DESYNC에서 노드 추가



(그림 7) C-DESYNC에서 노드 삭제



(그림 8) C-DESYNC에서 LN 삭제
그림 8은 그림7-2주기 이후에 네트워크에 참여하고 있

던 LN_0 가 삭제될 때를 나타낸다. 그림 8-1주기에서 N_1 노드와 N_3 노드는 T 동안에 GP 를 감지하지 못한 상태이다. 그림 8-2주기는 T 내에서 GP 를 인지하지 못한 노드 N_1 와 N_3 중 먼저 firing하는 N_1 이 GP 를 보내며 N_1 이 LN 을 수행하며 일반노드의 역할도 수행한다. 노드 N_3 는 LN_1 의 GP 신호를 감지하여 동기화되고, 노드들은 자신의 firing 시각을 기준으로 BFC 와 AFC 를 파악한다. 그림 8-3주기에서 노드들은 다음 주기에서 식 (4)를 이용해 firing할 위치를 정한다. 그림 9-3주기의 (5), (6), (7)은 동시에 일어난다.

6. 결론

본 논문에서는 DESYNC 알고리즘을 이용하여 소규모 네트워크에 적합한 동기화 방식의 TDMA 기법인 C-DESYNC를 제안하였다. C-DESYNC는 주기의 시작 정보를 가지고 있는 GP 를 이용하여 동기를 유지하고, DESYNC의 firing 기법을 이용하여 전체 노드의 개수를 파악하고, 이 정보를 이용하여 네트워크가 안정화 상태를 유지하도록 한다. 제안하는 알고리즘은 복잡한 방식으로 동기를 유지해야했던 기존의 동기화 방식과 달리 간단한 정보만으로 동기를 유지시키기 때문에 비용측면에서 효율적이며, 안정화 상태까지 걸리는 최대 지연 시간을 보장한다.

Acknowledgement

“본 논문은 미래창조과학부의 고용계약형 SW석사과정 지원사업으로 수행한 결과임.”

참고문헌

[1] Sivalingam, K.M. ; Agrawal, P. ; Kishore, S., J. "A comparison of MAC protocols for wireless local networks based on battery power consumption", INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 150 - 157 vol.1, 29 Mar-2 Apr 1998

[2] Degesys, J.; Rose, I; Patel, A.; Nagpal, R, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks", Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on, 11-20, 25-27 April 2007

[3] Patel, Ankit, Julius Degesys, and Radhika Nagpal. "Desynchronization: The theory of self-organizing algorithms for round-robin scheduling." ,Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on. IEEE, pp.87-96, 2007.