

# 전투지역에서의 이동통신을 위한 분산제어 알고리즘 설계

正會員 李 京 鉉\* 正會員 宋 周 錫\*\*

## The Design of Distributed Control Algorithm for Mobile Communication Network in the Battle Field

Kyung Hyun LEE\*, Joo Seok SONG\*\* *Regular Members*

**要 約** 전투지역 환경의 특성을 조사하고 요구되는 네트워크의 성질을 고려하여 그 성질에 적합한 네트워크 제어구조를 설계한다. 또한 네트워크의 제어구조를 구성하며 제어구조에서 형성된 근간 네트워크에서의 전송계획을 작성하기 위한 분산 제어 알고리즘을 설계한다. 전투지역에서 변화하는 네트워크의 연결성 유지와 노드의 파괴로 인한 네트워크의 즉각적인 재구성능을 신뢰성 있게 하기 위한 분산 알고리즘과 또한 공통 채널을 이용하는 네트워크에서 노드간에 전송충돌이 없는 전송 계획을 작성하기 위한 분산 알고리즘을 제안한다. 특히 중요한 노드가 파괴 및 사라질 때 되도록 적은 지연을 통해 네트워크가 동작할 수 있는 방법을 제안한다. 그리고 제안된 분산 알고리즘을 시뮬레이션을 통해 검증하고 네트워크 동작에 관해 설명한다.

**ABSTRACT** In this paper, we review the characteristics of battle field, consider required properties of mobile radio network, and design network control architecture proper to battle field network. Also we design distributed algorithm which constructs network control architecture and transmission schedule. For the information of network connectivity and the immediate reconfiguration caused by failure, loss, or movement of nodes, configuration and transmission schedule algorithms are to be executed periodically and for entire network reliability, algorithms must to be executed fully distributedly. At the time of loss or failure of primary local control node, we suggest the method which diminishes transmission delay. We explain the operation of network with examples. Finally, designed algorithms are verified and analyzed by computer simulation.

### I. 서 론

현재 사용되고 있는 네트워크들은 대부분 유선을 중심으로 개발되어 왔고 사용자들 역시 유선을 중심으로한 네트워크 이용자가 대부분이었다. 그러나 최근 인공위성의 발달 및 높은 실용성, 그리고 마이크로 웨이브를 이용한 무선통신의 발전 등으로 무선을 중심으로하는 편리한 통신이 세계적으로 붐을 이루고 있다. 본 논문은 이러한 이동통신이 극히 필요한 전투지역의 환경에 대해 설명하고 그에 적합한 네트워크 구조와

그 구조를 형성하기 위한 알고리즘과 전송계획을 작성하는 알고리즘을 제시한다.

그동안 효율적이고 신뢰성있는 이동무선 통신 네트워크에 대한 연구가 다각도로 이루어져 왔다. 이동무선 통신 네트워크의 제어 구조에 대해서는 중앙집중형, 계층형, 완전 분산형, 그리고 분산 클러스터 제어 구조가 연구대상에 고려되었다. 또한 채널 접근방법으로는 동기식 시분할 다중 액세스, ALOHA, R-ALOHA 등에 대해 연구가 진행 되었다. 그리고 통신 네트워크를 구성하는 알고리즘에 대해서도 M.J.Post<sup>(1)</sup>, 그리고 D.J.Baker<sup>(2)</sup> 등 많은 사람들에 의해서 연구되었다.

특히 전투지역에서는 비교적 좁은 영역에 많은

\* 三星電子 情報通信 中央研究所  
 \*\* 延世大學校 電算科學科  
 論文番號 : 91 - 110 (接受1991. 8. 12)

사용자 노드들이 있고, 심한 잡음과 적의 공격으로부터의 보호를 위해 노드의 이동이 필수적이며, 또한 적의 공격으로 인해서 노드가 파괴될 수도 있다. 그래서 전투지역에서는 노드의 이동에 따라 노드들 사이의 연결성이 변하므로 이동 네트워크의 변화된 근간 네트워크를 계속 재구성해야 한다. 안전한 통신 링크의 필요성으로 인하여 self-organizing 네트워크와 결합된 확산 대역 기법(Spread Spectrum Technique)<sup>6)</sup>들이 사용될 수도 있다. 그리고 전투지역의 특성상 마이크로 웨이브를 이용한 무선 통신을 해야 한다. 또한 각 노드는 네트워크를 구성하기 위한 알고리즘을 완전히 분산수행해야 한다.

이동 무선 시스템에는 여러가지 제어구조가 있지만 중앙 제어 구조에서의 취약성<sup>7)</sup>을 줄이고 완전 분산 제어 구조에서의 통신 부담분제<sup>8)</sup>를 줄이기 위해 이용한 제어구조가 분산 클러스터(cluster) 제어구조이다. 이 구조는 네트워크가 여러개의 클러스터들로 구성되며 각 클러스터는 여러개의 노드들로 구성된다. 그래서 한 클러스터는 하나의 주지역 제어 노드(primary local control node)가 있어서 해당 클러스터를 제어한다. 즉 클러스터만 볼 때 중앙제어구조이다. 그러나 클러스터간에는 완전한 분산제어 구조이다<sup>6,7)</sup>.

여기서 제시되는 알고리즘은 기존에 연구된 알고리즘들에서 발생하는 과도한 주지역 제어노드와 링크의 생성, 주지역 제어 노드를 만들 때의 복잡성, 클러스터 간을 연결하는 게이트웨이 선정의 복잡성 등의 문제점들을 해결한다.

본 논문의 구성은 다음과 같다. II 장에서는 이동 무선 네트워크의 일반적인 동작에 관한 사항을 언급하고 III 장에서는 분산 클러스터 제어 구조 및 전송 계획을 위한 알고리즘들을 기술한다. IV 장에서는 알고리즘들을 이용한 시뮬레이션 결과를 보여주고 V 장에서 결론을 맺는다.

## II. 이동무선 네트워크의 동작

### 2.1 이동무선 통신 시스템의 채널 접근방법과 제어구조

이동 무선 시스템에서 이용하는 채널 접근방법은 두 가지로 생각해 볼 수 있다. 동기식 채널 접근방법은 외부의 간섭에 강하며<sup>9)</sup> 대표적인 방법으로 시분할 다중 액세스(Time Division Multiple Access)가 있다. 이 방식은 사용자의 수가 고정되고 통신량이 많을때만 효율적인 단점이 있는가 하면 다른 모든 접근방법에 비해 간섭에 강한 것이 장점이다<sup>10)</sup>. 그래서 기존의 전투지역에서 사용된 이동 무선 네트워크인 HFITF 네트워크와 BID 네트워크는 채널 접근 방법으로 시분할 다중 액세스를 이용하고 있고 여기서 제한하는 네트워크 역시 이 방식을 이용하고 있다.

네트워크 내의 각 노드는 공통채널을 이용해 제약된 분산 클러스터 제어 구조를 구성하고 전송계획을 작성해서 모든 노드들이 전송 충돌없이 통신할 수 있다.

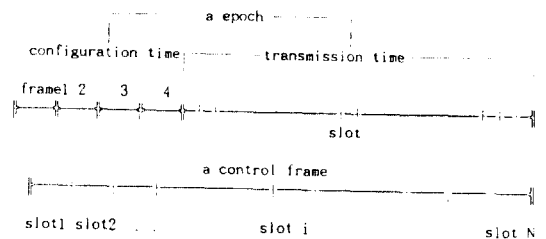


그림 1. 이동 무선 네트워크의 동기식 TDMA 구조  
Fig. 1. Synchronous TDMA structure for mobile radio network.

공통 채널은 그림 1에서 보는 것과 같이 크게 구성 기간과 전송 기간으로 이루어진 주기들로 구성된다. 공통채널은 동일한 형태의 주기가 반복된다. 한 주기는 네트워크 재구성 기간과 노드간 통신 기간으로 구성되며 재구성 기간 동안 노드들은 서로의 연결정보를 주고 받으며 근간 네트워크를 형성하고 전송계획을 작성한다. 그리고 전송 기간 동안에는 작성된 전송계획에 따라 모든 노드가 통신을 한다. 만약 네트워

크 연결성이 변화되면 다음 주기의 재구성 기간에 새로운 근간 네트워크와 전송계획이 작성되어 통신을 계속할 수 있게 된다. 구성기간은 다시 4개의 제어 프레임으로 구성된다. 그리고 각 제어 프레임은 N개의 슬롯을 갖게 된다. N은 이동 네트워크의 최대 노드 수이다. 이 제어 프레임의 슬롯을 통해 각 노드는 연결 및 제어 정보를 주고 받으며 제어구조와 전송계획을 작성한다. 전송 기간은 고정된 수의 슬롯들로 구성되고 재구성 기간에서 작성된 각 노드의 전송계획에 의해 이용된다. 이렇게 제어 프레임에서 제어 정보를 주고 받으며 근간 네트워크와 전송계획을 작성하게 된다.

## 2.2 재구성 시간간격(reconfiguration interval)에 대한 고려

재구성 시간간격이 짧을 경우 일단 네트워크를 구성하는데 많은 대역폭을 제어 정보를 주고받기 위해 소모하게 된다. 즉 전송 지연이 커지게 된다. 그러나 네트워크 연결성을 즉각적으로 반영한다. 그러므로 전송기간중 네트워크 연결성 변화에 잘 적응하고 전송기간중 전송실패가 적어진다. 이 기간이 길어질 경우 반대의 효과가 발생한다. 이 논문에서는 되도록이면 전송 지연을 적게 하기 위해 이 간격을 길게한다. 그리고 네트워크 구성의 초기 연결성을 변화시키는 노드가 전체 노드수의 4%~6%가 되도록 결정한다. 위와 같이 하므로써 적은 수의 전송실패를 감수하면서 네트워크 연결성에 즉각적인 적응을 하며 전송 지연을 줄인다.

## 2.3 노드의 사라짐과 나타남의 인식과 네트워크 링크 동작의 변화

네트워크에 새로이 들어갈 노드 혹은 이미 동작하는 노드가 전체 네트워크에 어떤 노드들이 있는가를 알 수 있는 가장 쉬운 방법으로 방송을 이용한다. 각 노드는 전송 기간중 전송기본단위인 슬롯 안에 일부분을 제어 영역(즉 IdBitPattern)으로 사용한다. 각 노드는 전송기간이 시작될 때부터 현재 전송할 때까지 수신해서

수정되왔던 IdBitPattern을 자료와 함께 전송한다. 만약 TDMA에 할당된 슬롯에 전송할 자료가 없어도 IdBitPattern은 전송한다. 그리고 병소에 수신하는 슬롯에서 IdBitPattern을 들어서 자신의 IdBitPattern을 수정한다.

IdBitPattern은 전송 기간중 모든 슬롯에 포함되기 때문에 수신 예측했던 노드로부터의 슬롯안에 최소한 IdBitPattern은 있어야 한다. 각 노드는 자신의 링크 활성화 스케줄(link activation schedule)을 가지고 있다. 뿐만 아니라 이웃노드와 이 전송계획을 제어 프레임에서 서로 교환하므로 이웃노드의 전송계획 역시 알 수 있다. 따라서 만약 이 IdBitPattern이 없으면 수신예측했던 노드는 전송 범위 밖으로 벗어 났는지 또는 노드가 파괴된 것이다.

사라진 노드가 일반 노드일 때는 링크 동작에 아무런 변화도 필요없다. 단지 사라진 노드를 인식한 노드들은 이 노드에 전송을 그 주기의 나머지 동안 중지하면 된다. 그리고 사라진 노드가 게이트웨이인 경우 높은 사용자 밀도로 인하여 다른 게이트웨이가 있을 확률이 높다. 만약 클러스터에 다른 게이트웨이가 없을 경우에는 그 주기의 나머지 동안은 클러스터 안의 노드들간의 통신만 하게된다. 그러나 사라진 노드가 주지역제어 노드일 경우에는 문제는 심각해진다. 이것은 노드 자신이 주지역제어의 클러스터 범위를 벗어났거나 주지역제어 노드가 고장났다는 것을 의미한다. 그러면 그 주기의 나머지 동안은 보조지역제어 노드를 이용하여 통신하므로써 전송 지연을 줄일 수 있다. 여기서 보조지역제어 노드는 자신이 제어 노드로서 활성화되었음을 알 수 있어야 한다. 방법은 주지역제어 노드의 고장을 인식한 노드가 자신의 보조 지역제어 노드로의 전송 슬롯에서 ABF(activation bit field)를 1로 해서 전송하므로써 보조지역제어 노드는 이 슬롯을 수신한 후 제어 노드로서 역할을 한다. 그러나 제어 노드의 역할은 ABF를 1로 한 노드에 대해서만 한다. 요청하지 않은 노드들은 주지역제어 노드를 유지하고 있기 때문이다. 만약 보조지역 제어 노드가 없을 때는

주지역 제어 노드에 속해있던 일반 노드들은 통신이 다음 주기까지는 완전히 불가능하다.

범위밖의 노드가 범위안으로 들어오는 경우에는 거의 즉시 수신 슬롯에 충돌이 발생한다. 왜냐하면 범위안의 노드들의 전송계획과 충돌이 발생하기 때문이다. 그러므로 이 충돌을 감지하자마자 송신을 다음 주기까지 일시적으로 멈춘다. 이 동작은 네트워크에 막 들어온 노드 역시 똑같은 반응을 한다. 범위안에 새로운 노드의 출현으로 전송 계획을 다시 작성하기는 부담이 너무 심하다. 거의 모든 노드가 전송계획을 새로 작성해야 하기 때문이다. 나머지 전송기간이 길다 할지라도 전송을 멈추는 것이 네트워크 전체의 효율성에 도움이 된다.

### Ⅲ. 분산 클러스터 제어구조 및 전송계획을 위한 알고리즘들

분산 클러스터 제어구조 및 전송계획을 작성하기 위한 알고리즘의 수행기간은 그림 2와 같다. 이와 같이 수행기간이 다른 것은 네트워크의 연결성 정보를 축적하고 그것을 바탕으로 알고리즘들이 수행되기 때문이다. 각 노드가 구성기간 동안 구성 알고리즘을 수행하면서 유지해야할 자료구조를 알아본다.

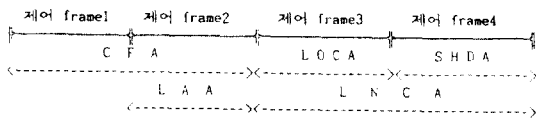


그림 2. 알고리즘들의 수행기간  
Fig. 2. Execution times for algorithms.

- PNodeStatus : 근간 네트워크에서 설정된 노드의 역할
- SNodeStatus : 보조지역 제어 노드 혹은 일반노드
- OwnPHead : 자신의 주지역제어 노드의 Id
- OwnSHead : 자신의 보조지역제어 노드의 Id

- OnehopNodes : 제어 프레임 1에서 얻은 한 홉 노드의 Id들
- TwohopNodes : 제어 프레임 2에서 얻은 두 홉 노드의 Id들
- NodeHead : 노드 id와 헤드(head)의 쌍의 연결 리스트 구조
- HeadsOnehopAway : 한 홉 안의 주지역 제어 노드들의 Id들
- SHeadsOnehopAway : 한 홉 안의 보조지역 제어 노드들의 Id들
- HeadsTwohopAway : 두 홉 안의 보조지역 제어 노드들의 Id들
- IdBitPattern : 네트워크내에서 동작하는 노드들의 Id를 표현한 비트 형태
- gateway1(H1, H2) : 헤드 1과 헤드 2를 연결하는 게이트웨이1
- gateway1 list : 한 홉 안의 게이트웨이 1들을 저장
- head pair(H1, H2) : 중복되지 않은 클러스터들의 헤드 쌍
- gateway2 pair(gA, gB) : 중복되지 않은 클러스터 쌍을 연결하기위한 게이트웨이 쌍으로 헤드 쌍(H1, H2)에 연결됨

#### 3.1 클러스터 형성 알고리즘-CFA(cluster formation algorithm)

클러스터 형성 알고리즘은 주어진 지역 범위에 산재해 있는 노드들중 주지역제어 노드가 될 노드를 설정하는 알고리즘이며 완전히 분산수행된다.

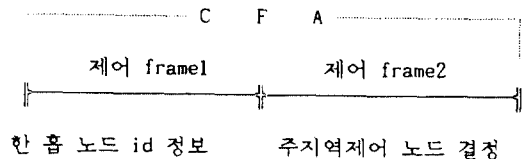


그림 3. 클러스터 형성 알고리즘의 수행기간  
Fig. 3. CFA execution time.

그림 3에서 보는것과 같이 제어 프레임 1 동안 송수신 메카니즘에 의해 각 노드는 자신의 노드 id를 전송하고 한 홉 노드 id를 들으므로 한 홉 노드 id를 알 수 있다.

실제로 클러스터 형성 알고리즘은 분리된 몇 개의 주파수 대역을 갖는 네트워크를 위해 만들어졌고, 각 주파수 대역은 서로 다른 전파(propagation) 특성을 갖는다. 한번에 한 개의 주파수 대역에 대한 클러스터 형성 알고리즘의 재구성 작업이 수행되며 다른 주파수 대역과는 독립적으로 수행된다.

클러스터 형성 알고리즘에서는 모든 노드들의 연결성을 양방향이라고 가정했지만 실제로는 노드 i와 노드 j간의 연결성이 단방향인지 양방향인지를 결정할 수 있다(제어 프레임 2 후에).

이제 제어 프레임 2에서 자신의 유일한 주지역 제어 노드(PLCN)를 결정하고 주지역 제어 노드 역시 자신을 주지역 제어 노드로 결정한 노드들을 알 수 있어야 한다. 주지역 제어 노드를 결정할 때 노드 id를 이용한다. 즉 범위안의 노드 id들 중 가장 작은 id를 갖는 노드를 주지역 제어 노드로 결정한다. 이 알고리즘은 아래와 같이 간단하게 기술할 수 있다.

알고리즘 CFA / \* 가정 : 전송하는 슬롯=i,  
수신하는 슬롯=j  
\*/

프레임 1 :

전송 : -NodeId<sub>i</sub> 전송 / \* 자신의 노드 Id 전송  
\*/

-OnehopNode<sub>i</sub> 전송 / \* i번째 슬롯 이전  
에서 들었던 노드Id들 전송 \* /

수신 : -NodeId<sub>j</sub>를 들어서 OnehopNode<sub>j</sub>에 저장  
-OnehopNode<sub>j</sub>를 들어서 j<i일때 NodeId<sub>j</sub>  
가 OnehopNode<sub>j</sub>안에 있으면 Con[i]<sub>j</sub>=Con  
[j]<sub>i</sub>=1 / \* Con[1..N] : connectivity  
vector \* /

프레임 2 :

전송 : -OwnPHead<sub>i</sub>없으면 \* PNodeStatus<sub>i</sub><-  
"PHead", \*PNodeStatus<sub>i</sub>전송  
PHead<sub>i</sub>전송, -Con[ ]<sub>i</sub> List 전송, -One-

hopNode<sub>i</sub> 전송

수신 : -PNodeStatus<sub>i</sub>를 받고 "PHead"일때  
OwnPHead가 없으면 OwnPHead<sub>j</sub><-i  
PHead<sub>i</sub>를 받아서 HeadsOnehopAway<sub>j</sub>  
에 저장

Con[ ]<sub>i</sub>를 받아 자신의 Con[ ]<sub>i</sub>를 수정  
OnehopNode<sub>i</sub>를 받아

\* i<j일때 NodeId<sub>j</sub>가 OnehopNode<sub>i</sub>안에  
있으면 Con[i]<sub>j</sub>=Con[j]<sub>i</sub>=1

\* OnehopNode<sub>i</sub> 중 OnehopNode<sub>j</sub>안에  
없는 것은 TwohopNode<sub>j</sub>에 저장

주지역 제어노드를 결정하기 위한 알고리즘으로 클러스터 형성 알고리즘 이외에 HEADNUMBER 라는 것이 있다<sup>6)</sup>. 그러나 HEADNUMBER 프로시쥬어가 어떤 상황하에서는 불필요하게 많은 주지역 제어노드를 만들 수 있다. 프로시쥬어 DELETEHEADS는 중복된 클러스터를 없애고 과다하게 많은 주지역 제어 노드를 제거한다<sup>6)</sup>. 여기서의 클러스터 형성 알고리즘은 HEADNUMBER를 사용할 때의 문제점인 과다한 제어 노드의 생성을 방지할 수 있다.

### 3.2중복된 클러스터 연결 알고리즘-LOCA(linking overlapped cluster algorithm)

중복된 클러스터 연결 알고리즘은 클러스터 형성 알고리즘에서 형성된 클러스터들간에 통신을 위한 게이트웨이를 결정하는데 있어 클러스터들중 중복된 것들을 연결하는 게이트웨이를 결정하는 알고리즘이다. 자신이 게이트웨이인가 판단하고 게이트웨이인 경우 어떤 주지역제어 노드 쌍을 연결하는 게이트웨이인가를 알아야 한다. 또한 게이트웨이 범위안의 노드들을 알아야 하며, 특히 각 주지역제어 노드와 보조지역제어 노드는 자신의 범위안에 어떤 게이트웨이가 있는가 알아야 한다. 중복된 클러스터 연결 알고리즘은 프레임 3 동안에 수행되며 프레임 3이 시작하기 전에 게이트웨이를 결정한 후 프레임 3 동안 주위의 한 홉 노드들에게 알린다. 이때 클러스터 형성 알고리즘에서 얻는 HeadsOnehopAway와

Con[] 정보를 이용해 게이트웨이를 결정한다. 게이트웨이 결정 역시 노드 id를 기준으로 한다. 게이트웨이 후보들중 노드 id가 제일 작은 것을 선택한다. 게이트웨이 후보들은 클러스터 1의 헤드 노드(H1)와 클러스터 2의 헤드 노드(H2)에 대한 연결성 벡터, 즉 Con[]<sub>H1</sub>과 Con[]<sub>H2</sub>를 가지고 있다. 그래서 게이트웨이 후보들은 Con[]<sub>H1</sub>과 Con[]<sub>H2</sub>에서 동일한 id를 찾아냄으로서 그것이 게이트웨이 후보임을 알 수 있다. 이때 게이트웨이 후보들중 제일 작은 id노드가 자신일 경우 자신이 헤드 쌍(H1, H2)의 게이트웨이가 됨을 결정한다. 그리고 프레임 3 동안에 한 홉 노드들에게 알린다. 알고리즘을 간단히 묘사 하면 아래와 같다.

알고리즘 LOCA / \* 가정 : 전송하는 슬롯=i,  
수신하는 슬롯=j  
\*/

게이트웨이 시험 :

- \* 노드자신의 노드 상태 PNodeStatus<sub>i</sub>는 “PHead” 가 아니어야 함
- \* HeadsOnehopAway<sub>i</sub>안의 모든 헤드 쌍(H1, H2)을 구함
- \* 각 candidategateway(H1, H2) < Con[]<sub>H1</sub>, Con[]<sub>H2</sub>의 중복된 id
- \* gateway1(H1, H2)가 자신의 노드 id와 동일하면 PNodeStatus<sub>i</sub> < - “Gateway”

프레임 3 :

- 전송 : PNodeStatus<sub>i</sub>가 “Gateway”이면 PNodeStatus<sub>i</sub>, gateway1(H1, H2) 전송
- 수신 : PnodeStatus<sub>i</sub>를 받고 “Gateway”이면 gateway1(H1, H2) 받아 게이트웨이 1 리스트에 저장

만약 우리가 주 지역 제어 노드를 설정할 때 HEADNUMBER와 DELETEHEADS를 이용했다면 중복된 클러스터를 연결할 때 게이트웨이 후보로 가장 유력한 것으로써 클러스터 헤드를 사용할 수 있다. 이리하여 과다한 링크의 생성을 방지할 수 있다. 그렇지만 여기서는 클러스터

형성 알고리즘을 사용해서 주 지역 제어 노드를 설정하므로 한 클러스터안에는 반드시 하나의 주지역 제어 노드만 있다. 따라서 위의 경우는 발생하지 않게 된다.

### 3.3 중복되지 않은 클러스터 연결 알고리즘- LNCA (linking nonoverlapped cluster algorithm)

분리된 넓은 클러스터들을 가능한 한 연결하기 위해 넓은 클러스터의 바깥부분에 있는 노드들을 게이트웨이로 선택해서 연결하고자 하는 알고리즘이 중복되지 않은 클러스터 연결 알고리즘이다. 중복되지 않은 클러스터 연결 알고리즘은 프레임 3 동안에 HeadsOnehoAway 정보와 Con[]를 주고 받아 HeadsTwohopAway 정보와 새 홉 노드들의 연결정보를 이용해 중복 되지 않은 클러스터들과 후보 게이트웨이쌍을 모호하지 않게 설정할 수 있다. 그리고 프레임 4 동안에 주위노드들에게 알린다.

게이트웨이 선택기준은 역시 노드 id이다. 2개 노드 id의 합이 가장 작으면서 서로가 무선 범위안에 있는 노드 쌍이 게이트웨이 쌍이 된다. 그리고 노드 id쌍의 합이 같은 경우에 가장 작은 id를 갖은 쌍을 선택한다.

일단 노드 자신이 중복되지 않은 클러스터에서 게이트웨이인가를 시험하기 위해 프레임 3을 통해 HeadsTwohopAway 그리고 2 홉 떨어진 노드의 연결성 벡터를 알 수 있다. 먼저 시험하는 노드는 주지역 제어 노드가 아니어야 한다. 그리고 클러스터의 바깥쪽 노드가 되기 위해 자신의 무선 범위안에 주지역 제어 노드가 반드시 1개(즉 자신의 주지역 제어 노드)만 있어야 한다. 그 이유는 한 클러스터 내부의 노드들은 대부분 한 홉안에 2개 이상의 주지역제어 노드를 포함하고 있기 때문이다. 알고리즘을 간단하게 묘사 하면 아래와 같다.

알고리즘 LNCA / \* 가정 : 전송하는 슬롯=i  
수신하는 슬롯=j  
\*/

프레임 3 :

전송 : -HeadsOnehopAway<sub>i</sub> 전송, Con[] (2 hop node id 정보) 리스트 전송

수신 : -HeadsOnehopAway<sub>i</sub>를 받아서 HeadsOnehopAway<sub>j</sub> 안에 없는 id를 복사해서 HeadsTwohopAway<sub>j</sub> 안에 저장  
-Con[] list를 받아서 노드 j안의 Con[] 리스트에 없는 벡터를 복사해서 연결

게이트웨이 시험 :

\* 노드 자신의 상태 PNodeStatus<sub>i</sub>는 "PHead"가 아니어야 함

\* HeadsOnehopAway<sub>i</sub>안에 자신의 주지역제어 노드를 제외한 헤드 id가 없어야 함

\* 겹쳐지지 않은 클러스터들의 헤드 쌍(H1, H2) 결정

-H1은 노드 i의 주지역제어 노드  
-loop h2={HeadsTwohopAway<sub>i</sub>}  
Con[]<sub>H1</sub>과 Con[]<sub>H2</sub>가 중복된 id를 가지지 않았을때 H2 < -h2  
end loop

\* 헤드 쌍(H1, H2)에 대한 후보 게이트웨이 쌍(gA, gB) 결정

loop g1={노드 i의 Con[]<sub>H1</sub>의 각 id}  
loop g2={노드 i의 Con[]<sub>H2</sub>의 각 id}  
만약 Con[]<sub>H1</sub>안에 g2가 있고 Con[]<sub>H2</sub>안에 g1가 있으면 후보 게이트웨이 쌍(gA, gB) 집합에 포함  
end loop  
end loop

\* 게이트웨이 쌍 결정

-헤드 쌍(H1, H2)에 대한 후보 게이트웨이 쌍(gA, gB) 집합안에서 gA와 gB의 합이 가장 작은 쌍들을 고른다.  
-쌍들중 노드 id가 가장 작은 것을 포함하는 게이트웨이 쌍(gA, gB) 선택.  
-gA 또는 gB가 i일때 PNodeStatus<sub>i</sub> < "Gateway"

프레임 4 :

전송 : -PNodeStatus<sub>i</sub>가 "Gateway"일때  
-헤드 쌍(H1, H2)와 게이트웨이 쌍(gA,

gB) 전송, PNodeStatus<sub>i</sub> 전송

수신 : -PNodeStatus<sub>i</sub>를 받고 "Gateway"인 경우  
-헤드 쌍(H1, H2)과 게이트웨이(gA, gB)를 받아 헤드 쌍 리스트에(H1, H2) 저장하고 게이트웨이 쌍(gA, gB)을 링크로 연결해 저장

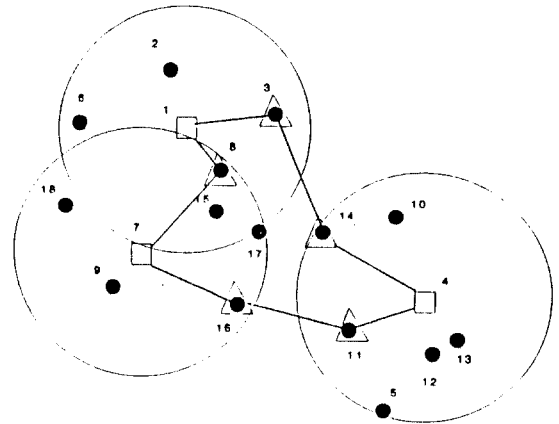


그림 4. 중복되지 않은 클러스터의 연결  
Fig. 4. Linking of nonoverapped clusters.

그림 4는 게이트웨이 쌍에 의해 연결된 큰 클러스터 들을 보여주고 있다.

비록 클러스터간의 통신량을 줄일 목적으로 근간 네트워크를 가능한 한 최소 신장트리에 가깝도록 하는 것이 바람직하지만 오류 허용(fault-tolerance)을 위해서 loop path나 과잉 링크를 갖는 것이 바람직할 수도 있다. 또한 필요한 클러스터 헤드의 최소 숫자를 선택하는 것은 사실 NP-complete 문제이다<sup>7)</sup>.

### 3.4 링크 활성화 알고리즘-LAA(link activation algorithm)

링크 활성화 알고리즘은 전송기간중 이웃 노드의 방송과 충돌없는 자신의 전송계획을 작성하는 알고리즘이며 프레임 2동안에 분산수행된다. 링크 활성화 알고리즘은 가능한 많은 링크를 활성화시킬 수 있도록 각 노드가 전송계획을 작성하도록 한다.

먼저 각 노드는 전송계획을 작성하기 위해 2 가지 자료구조를 갖는다. 즉 자신의 전송계획 (OwnSch<sub>i</sub>)과 이웃노드들의 전송계획(Schlist [i])들이다. 각 노드는 프레임 2를 통하여 한번씩 방송기회를 갖게된다. 그래서 전송충돌이 없는 자신의 전송계획 OwnSch을 Schlist을 근거로 작성해 이웃노드에게 전송한다. 또한 SchList [i] 역시 전송 한다. 이때 무선 범위안의 다른 노드들은 이 전송계획을 받아서 SchList[i]에 순서대로 저장한다. 즉 가장 최초에 작성된 전송 계획이 리스트끝에 유지될 수 있도록 한다.

각 노드가 수행하는 링크 확장 알고리즘을 간단히 묘사하면 아래와 같다.

알고리즘 LAA / \* 가정 : 전송하는 슬롯=i ;  
수신하는 슬롯=j  
\* /

전송 : SchList[i]들간에 충돌을 검사  
만약 충돌이 존재하면 SchList[i]에서  
연산 리스트의 끝쪽의 충돌을 일으키는  
스케줄을 수정

SchList[i]를 근거로 대칭적으로 OwnSch  
을 작성한다.

SchList[i]와 OwnSch<sub>i</sub>를 전송

수신 : SchList[i]를 받아서 SchList[i]를 수정한  
다.

-OwnSch<sub>j</sub>와 SchList[i]안의 SchList[j]  
j와 다음 경우 SchList[j]를 OwnSch<sub>j</sub>로  
복사

위 모듈에 의해 작성된 전송계획을 그림 5에서  
볼 수 있다.

전송 계획을 작성하는 다른 알고리즘으로 분산  
진전 알고리즘(Distributed Evolutionary Algo-  
rithm)을 사용하면 코드 분할 다중 액세스  
(Code Division Multiple Access)에서 발생하는  
충돌을 고려할 수 있고 링크에서의 통신량 변화  
에 적절히 대처할 수 있다<sup>9,10</sup>. 또 이 알고리즘을  
사용하면 링크 통신량에서의 변화성을 직접적으  
로 고려할 수 있고 또한 어떤 네트워크의 위상

<전송 계획 작성을 위한 정보>

node	OnehopNode							
1	3	6	7					
2	4	6						
3	1	5	7	8				
4	2	6						
5	3	7	8	9				
6	1	2	4					
7	1	3	5	8	9			
8	3	5	7	9				
9	5	7	8					

<작성된 전송계획>

node	slot: 1	2	3	4	5	6	7	8
1		B	3	6	7			
2		B	4	-	-	6	-	-
3		B	1	5	8	7	-	-
4		B	2	-	6			
5		B	7	3	9	8	-	-
6		B	-	1	4	2	-	-
7		B	5	8	1	3	9	-
8		B	9	7	3	5	-	-
9		B	-	5	-	7	-	-

그림 5. 전송 계획 작성의 예  
Fig. 5. Example for the transmission scheme.

(Topology)에 대해서도 한 노드에서 작성된  
계획이 충돌을 발생시키지 않는다. 반면에 링크  
확장 알고리즘은 간단하고 빠르게 갑작스런 한  
노드의 고장에 대해 튼튼하다.

### 3.5 보조지역 제어 노드 결정 알고리즘-SHDA (secondary determination algorithm)

보조지역 제어 노드 결정 알고리즘은 모든  
노드들의 유일한 보조지역 제어 노드를 모호하지  
않게 결정하는 알고리즘이며 프레임 4동안 수행  
된다. 알고리즘은 클러스터형성 알고리즘과 거의  
동일하다. 차이점은 그림 6에서처럼 제어 프레  
임 1의 슬롯 4의 번호배치가 다른 프레임 들과  
반대로 내림차순인 점이다.





그림 6. 제어 프레임 4의 슬롯 번호배김  
Fig. 6. Slot numbering of control frame 4.

보조지역 제어 노드는 클러스터내의 노드 id가 가장 큰 노드로 결정한다. 알고리즘을 간단히 기술하면 아래와 같다.

알고리즘 SHDA

- 전송 : OwnShead<sub>i</sub>=nothing이면  
 SNodeStat<sub>i</sub>="SHead"  
 OwnShead<sub>i</sub>=i  
 OwnShead<sub>i</sub> 전송
- 수신 : OwnShead<sub>i</sub> 받아서 OwnShead<sub>i</sub>가 i일때  
 만약 OwnShead<sub>j</sub>가 nothing이면 OwnS-head<sub>j</sub> < -i  
 something이면 SHeadOnehopAway < -i

IV. 시뮬레이션 결과

여기서는 위에서 제시한 알고리즘들을 이용해 시뮬레이션한 결과를 이동 무선 네트워크의 이동성, 생존성, 그리고 전송 지연 감소의 측면에서 살펴본다. 본 논문에서 설계한 제어구조 형성 알고리즘의 가장 큰 목적중에 하나인 네트워크의 이동성을 만족하는 성질을 그림 7과 그림 8에서 볼 수 있다.

그림 7은 13개의 노드로 구성되어 있고 재구성 기간을 통해 제어 구조가 형성된 상태에서 근간 네트워크를 이용해 전송계획에 따라 통신중이다. 그러나 이때 노드 1이 오른쪽 아래로 점차적으로 이동했을때 네트워크 연결성이 바뀌게 되고 전송계획간에 충돌이 일어난다. 그러므로 충돌이 일어난 노드들은 전송을 중단하게 된다. 그래서 그림 8처럼 다른 주기의 재구성 기간에 새로운 근간 네트워크와 전송계획을 작성해서 전송하게 된다. 이 경우에는 HEADNUMBER

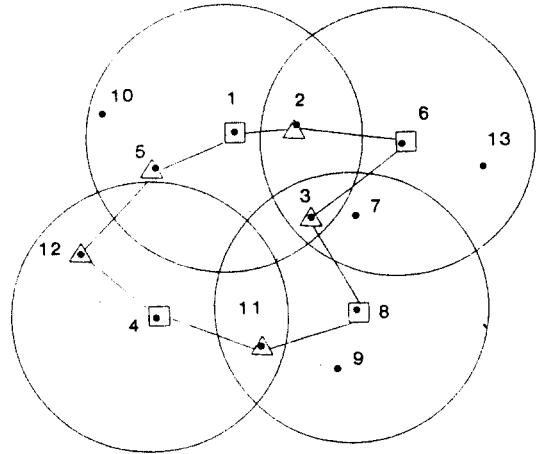


그림 7. 노드 1의 움직임 전  
Fig. 7. Before the movement of node 1.

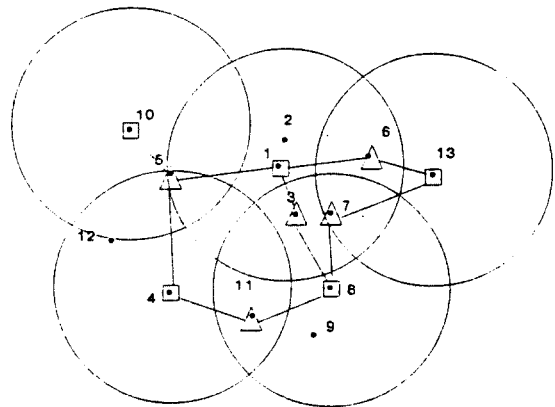


그림 8. 노드 1의 움직임 후  
Fig. 8. After the movement of node 1.

프로시저어나 분산 진전 알고리즘과 같은 다른 알고리즘들을 사용해도 다음 주기까지는 충돌로 인하여 통신이 두절된다.

이동 무선 네트워크는 또한 어떤 노드의 파괴로 인한 네트워크의 마비를 가져와서는 안된다. 이러한 생존성을 아래 그림 9, 그림 10에서 볼 수 있다.

그림 9에서 최악의 경우 즉, 주지역제어 노드인 4가 파괴되었을 때 해당 클러스터는 통신을 못하게 된다. 그러나 다음 주기에서 새로운 연결

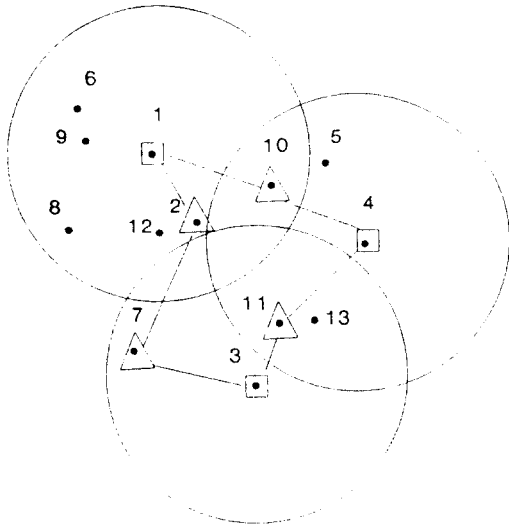


그림 9. 노드 4의 사라짐 전  
Fig. 9. Before the disappearance of node 4.

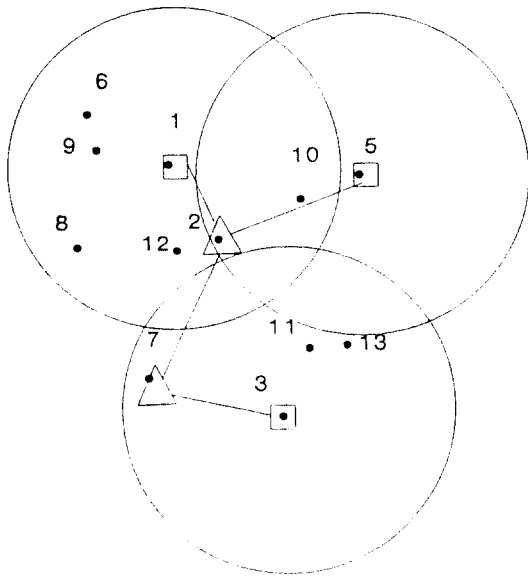


그림 10. 노드 4의 사라짐 후  
Fig. 10. After the disappearance of node 4.

정보를 얻어서 새로운 근간 네트워크와 전송계획을 작성해서 통신하게 되는 것을 그림 10에서 볼 수 있다. 이 경우에도 HEADNUMBER 프로시쥬어나 분산 집전 알고리즘과 같은 다른 알고

리즘을 사용해도 다음 주기까지는 통신을 못하게 된다.

클러스터 헤드가 파괴 되었을때 해당 클러스터를 다음 주기의 재구성기간 전까지는 통신을 못하게 된다. 그러므로 이때 즉 전송기간에 헤드가 파괴되어도 다음 재구성 기간 전까지의 자연을 감소 시키기 위해 항상 노드들은 보조지역 세이 노드를 유지하고 있다.

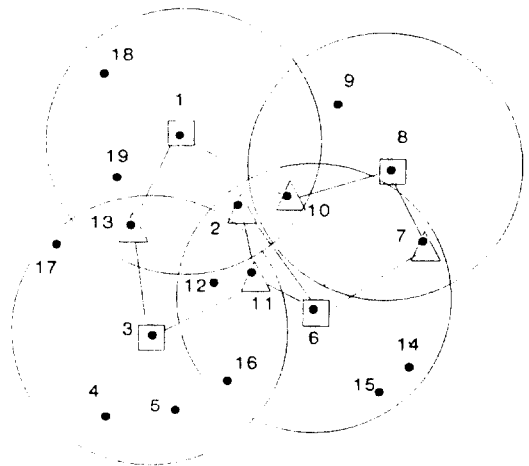


그림 11. 클러스터 헤드 1의 사라짐 전  
Fig. 11. Before the disappearance of cluster head 1.

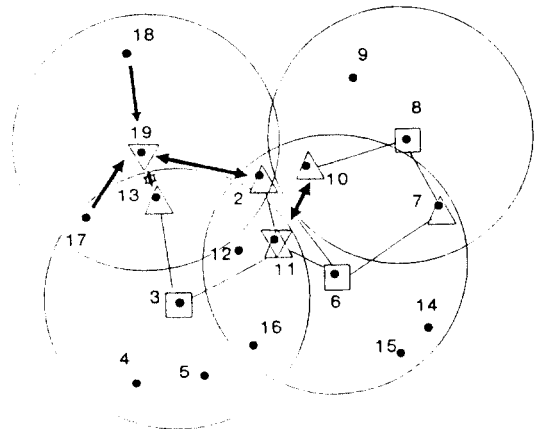


그림 12. 클러스터 헤드 1의 사라짐 후  
Fig. 12. After the disappearance of cluster head 1.

그림 11의 클러스터 헤드 1이 파괴되었을 때, 클러스터안의 노드들(2, 10, 13, 17, 18)이 자신들의 보조지역제어 노드({2, 13, 17, 18}→19, {10}→11)의 제어를 받아 남은 전송 기간동안 전송하는 링크를 그림 12에서 보여주고 있다. 만약 이 경우에 HEADNUMBER 프로시쥬어나 분산진전 알고리즘을 사용했다면 다음 주기까지는 총돌로 인하여 통신이 두절된다. 그러나 여기서 제시된 알고리즘을 사용하면 주지역 제어 노드가 파괴되어도 보조지역 제어 노드를 유지하고 있으면 통신이 가능하게 된다.

## V. 결 론

본 논문에서는 전투지역 환경과 요구사항에 관해 조사했다. 전투지역 환경에 적합한 네트워크가 되기위해 안전성, 이동성, 생존성, 적응성, 독자성, 잡음저항성 그리고 최소전송지연성을 만족 시켜야 한다. 이와 같은 조건을 만족 시키기 위해 공통채널을 사용하는 시분할 다중 액세스 채널 구조를 이용해서 분산 클러스터 제어 구조를 설계했다. 그리고 각 노드가 하나의 이동 무선 네트워크의 제어 구조 및 자신의 전송 계획을 구성하기 위한 5개의 분산 알고리즘을 설계 및 시뮬레이션을 통해 검증했다. 이 알고리즘들은 모든 노드들에서 동일하게 수행되며 다른 노드의 제어를 받지 않으며 독자적으로 수행된다. 단지 지역 정보만을 이용해서 동일한 알고리즘은 각 노드에서 분산수행이 된다. 그리고 설계된 네트워크가 동작할 때 고려되어야 할 사항들을 언급했다. 시뮬레이션은 워크스테이션 Solbourne 컴퓨터에서 UNIX 환경에서 C 언어를 이용했으며 X window를 이용해 수행했다. 시뮬레이터의 입력 매개변수로는 노드의 수, 통신반경, 임의의

위치를 생성하기 위한 seed number 등이다.

## 참 고 문 헌

1. M.J. Post, A.S. Kershenbaum and P.E. Sarchik, "A Distributed Evolutionary Algorithm for Reorganizing Network Communication", MILCOM, 1985.
2. Dennis J.Baker, Anthony Ephremides, "A Distributed Algorithm for Organizing Mobile Radio Telecommunication Networks", Proc. 2nd Int. Conf. Distributed Comput. Syst., Paris, France, Apr.8-10 1981.
3. Rober E. Kahn, Steven A.Gronemeyer, Jerry Burchfiel, Ronald C. Kunzelman, "Advances in Packet Radio Technology", Proceedings of the IEEE, Vol. 66, No.11, Nov. 1978.
4. Dennis J.Baker, Anthony Ephremides, Julia a.Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control", IEEE Journal on selec. areas in comm., Vol.SAC-2, No.1, Jan. 1984.
5. A.Hauptschein, M.Kajor, "Recognition and Self-Organization of Nodes into DTDMA Networks", IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-17, No.4 July 1981.
6. Dnnis J.Baker, Anthony Ephremides, "The Architectural Organization of a Mobile Radio Networks via a Distributed Algorithm", IEEE Trans. on comm., Vol.COM-29, No.11, Nov. 1981.
7. C.Li, "Clustering in packet radio networks", IEEE 1985 Int.Conf.on Com., Chicago, Ill., section 10.5.1, 1985.
8. Arne Nilsson, Wushow Chou, Charles J.Graff, "A Packet Radio Communication System Architecture in a Mixed Traffic and Dyanmic Environment", Proc. Comput. Networking Symp. IEEE CH1586-7/80, 1 980, pp 51-66.
9. T.G.Robertazzi, P.E.Sarachik, "Self-Organizing Communication Networks", IEEE Comm. Magazine, Jan. 1986-Vol.24, No.4.



李京鉉 (Kyung Hyun LEE) 正會員  
1967년 1월 13일생  
1989년 : 홍익대학교 전자계산학과 졸업  
(공학사)  
1991년 : 연세대 대학원 전자학과 졸업  
(이학석사)  
1991년 ~ 현재 : 삼성전자 정보통신 중앙  
연구소 부가연구실 재직



宋周錫 (Joo Seok SONG) 正會員  
1953년 3월2일생  
1976년 : 서울대학교 전기공학과 졸업  
(공학사)  
1979년 : 한국과학기술원 전기 및 전자공  
학과 졸업(공학석사)  
1988년 : U.C.Berkeley 전자학과 졸업  
(공학박사)  
1979년 ~ 1982년 : 한국전자통신연구소 근무  
1983년 ~ 1988년 : Electronic Research Lab 근무  
1988년 ~ 1989년 : Naval Postgraduate School 조교수  
1989년 ~ 현재 : 연세대학교 전자학과 조교수