

# 생체모방 자율제어 네트워크 환경 구현 연구 동향 : Programming Protocol-Independent Packet Processors 기술 개요

진정하, 김기천  
건국대학교

## 요약

자연계에서 존재하는 다양한 생명체는 자신들의 생존과 종의 번성을 위해 효율적인 행동 규칙을 만들어 진화해 왔다. 이러한 생명체의 다양한 생존원리로부터 착안을 하여 자연계가 아닌 다른 환경에서 적용이 가능하도록 알고리즘을 만들어 적용시키는 것을 생체모방 알고리즘이라 한다. 자연계의 환경자체가 불확실한 변화가 다양하게 포함되고 있으며, 제한된 자원 환경을 어떻게 효율적으로 활용하는가의 문제가 걸려 있음으로 인하여 이러한 생체모방 알고리즘은 적용환경의 변환에 빠른 적응력을 제공할 수 있고, 자원 제약형 환경에서 안정적으로 확장성과 적응성을 제공할 수 있어서 상호 운용성 측면에서 많은 이득을 줄 수 있다. 이와 같은 생체모방 알고리즘을 네트워크의 관점에서 적용시켜 보면, 전자의 경우에는 자율적인 네트워크 구성을 용이하게 제공할 수 있음을 나타내고, 후자의 경우에는 IoT 환경과 같은 자원 제약형 환경에서의 상호 운용성을 제공할 수 있다. 이렇듯이 생체모방 알고리즘을 네트워크에 접목시켜 연구하는 것은 최근의 네트워크 분야의 연구 이슈와 상호 보완적으로 작용하여 시너지 효과를 제공할 수 있다. 자연계의 군집 현상 및 동기화 현상을 네트워크 환경에서 적용하여 사용할 수 있는 생체모방 알고리즘 기술은 다양하게 존재하고 있으며 이를 활용하는 연구를 통해 SDN(Software Defined Networking)에서의 자율제어 네트워크 구성에 접목하거나 IoT 환경과 같은 자원 제약형 환경에서의 보다 효율적인 상호 연결성을 제공하는 방향으로 발전할 수 있다. 이러한 생체모방 자율제어 네트워크 환경 구현을 위해 기존의 OpenFlow 환경과 새로이 부각되는 P4: Programming Protocol-Independent Packet Processors 기술에 대해서 정리하여 향후 생체모방 자율제어 네트워크 구현 방안을 제시하고자 한다.

## I. 서론

생체모방 알고리즘(Biologically Inspired 또는 Bio-Inspired

Algorithm)은 자연계에 존재하는 다양한 생명체의 행동 규칙을 연구하여 모델링한 알고리즘이다. 이러한 생체모방 알고리즘은 자연계를 구성하고 있는 다양한 생명체가 인위적인 제어 없이 단순하면서도 적은 효율적인 행동 규칙을 통하여 자연계에서 살아남을 수 있으며, 생명체가 속한 생태계를 유지하고, 환경을 관리하며, 상호간의 동기화를 통해 생존과 종의 번식이라는 대전제의 기능을 수행하고 있다. 자연계에서 도출된 생체모방 알고리즘은 우리가 흔히 접할 수 있는 형태의 중앙집중식 알고리즘이 아닌 수평적인 구조로 자율적인 알고리즘 기능을 갖게 된다. 이는 자연계의 생명체는 각각의 개체가 정해져 있는 행동 규칙을 수행함으로써 자신이 속해있는 생태계에서 일관된 형태를 나타낼 수 있는 하향식 형태를 갖는 것이 아닌 상향식의 형태의 분산처리 알고리즘이다. 자연계에서 도출된 생체모방 알고리즘은 공학뿐만 아니라 인문, 사회학 분야에서도 많은 연구가 진행되고 있으며, 특히 컴퓨터 공학에서는 생체모방 알고리즘을 네트워크에 적용시켜 최적화 및 동기화를 수행하는 형태로 접근하고 있다.

현재까지 연구되어 온 생체모방 알고리즘의 예에는 생체 시스템을 모델링한 유전자(Genetic) 알고리즘, 생명체가 최적의 길을 찾기 위해 페르몬 분비를 사용하는 과정을 모델링한 Ant Colony Optimization(ACO) 알고리즘, 생명체 발광 주기의 동기화 과정을 묘사한 반딧불이(Firefly) 알고리즘, 생명체의 먹이 찾는 행동을 모델링한 벌(Bee) 알고리즘 등과 같은 고전적인 방식으로부터, 보다 고차원적이고 복잡한 형태의 날아다니는 새 무리, 야생에서의 동물 무리, 물고기 무리, 그리고 박테리아 무리 등과 같은 생물 군집에서 쉽게 관측이 가능한 무리 집중에 의한 질서화된 운동, 즉 군집 이론(Flocking Theory)에서 자기 질서화(Self-Ordered) 현상을 모델링에 대한 연구가 진행되고 있다. 생물 군집 현상은 자연계에서 무질서하게 존재하는 개체들이 외부의 인위적인 작용 없이 스스로가 질서화된 상태를 형성해가는 현상을 의미하고 있다. 이러한 생물 군집 현상은 다음과 같은 세가지 법칙인 각 개체간의 서로 일정한 간격을 유지해야 한다는 분리성(Separation), 각각의 개체는 주변의 다른 개체들 이동 방향의 평균적인 수치로 자신의 이동 방향

을 제어하는 의미하는 정렬성(Alignment), 주변의 다른 개체들과 일정한 거리를 유지함으로써 자신의 위치를 결정하는 결함성(Cohesion)을 중요시 하고 있다[1].

이러한 군집 이론은 1986년 Reynolds가 컴퓨터 시뮬레이션을 수행한 이후 다양한 분야에서 이를 활용한 연구 결과들이 도출되고 있으며, 각 개체들이 군집 이론의 세 가지 법칙들을 분산 방식으로 처리하고, 각 개체들의 초기 데이터가 특정 범위에 있었다고 가정한다면 일정 시간이 경과한 후에 전체 개체들의 속도와 위치등의 데이터는 각 개체들의 초기 데이터의 평균적인 수치로 수렴함을 수학적으로 증명하는 것과 같은 연구를 통해 군집 이론에 관한 연구가 매우 활발히 진행되고 있으며 각 개체에서 정보의 분산 처리가 요구되는 분야에서 최적의 솔루션을 구하는데 이론적인 바탕을 제공하고 있다[2]-[6].

본 기고문에서는 지금까지 연구가 진행된 Ant Colony Optimization 알고리즘, Bee 알고리즘, Firefly 알고리즘과 최근 활발히 연구가 진행되고 있는 Flocking 알고리즘과 같은 생체모방 알고리즘을 정성적으로 분석하여 네트워크의 자율제어를 구성하는데 있어서 주요 이슈사항에 대한 해결책이 될 수 있음을 자세히 살펴본다. 아울러 생체모방 알고리즘의 자율제어 네트워크 구현을 위한 기술 연구 동향을 조사하여 효율적이고 상호 운용적인 적용형 네트워킹 기술을 위한 연구 동향을 살펴본다.

## II. 생체모방 알고리즘

통신 및 네트워킹 기술 분야에서 생체모방 알고리즘의 필요성에 대하여 자연계의 생체 시스템과 통신망의 유사한 환경의 예를 들어 설명하고 기존의 통신망 및 네트워킹 기술에서 이슈사항을 해결하기 위해 사용되는 고전적인 Ant Colony Optimization 알고리즘, Bee 알고리즘, Firefly 알고리즘부터 새로이 연구가 진행중인 Flocking 알고리즘, Huddling Penguins 알고리즘에 대하여 자세히 기술한다.

### 1. Ant Colony Optimization 알고리즘

Ant Colony Optimization 알고리즘은 개미 집단의 습성을 관측하여 개미집으로부터 멀리 떨어져있는 먹이를 시력이 없는 개미 집단이 가장 빠른 경로를 찾아서 운반하는 행위를 모방하는 방식이다.

개미는 집을 나와서 먹이를 찾아 다닐 때 자신이 지나온 길에 페로몬(Pheromone)을 분비하게 되는데 이는 시력이 없는 개미가 먹이를 발견하면 자신이 지나온 길에 남아 있는 페로몬을 따

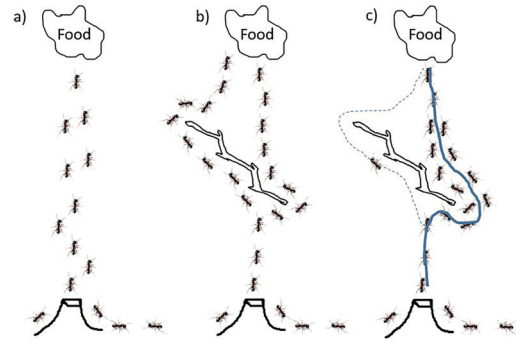


그림 1. Ant Colony Optimization 알고리즘 개요

라 개미집으로 먹이를 운반할 수 있게 한다. 이러한 방식으로 개미무리가 다양한 경로를 이용하여 먹이를 운반하게 되면 일정 시간이 지나서 개미집과 먹이와의 거리가 가장 짧은 경로에 다수의 페로몬이 쌓이게 되어 개미는 페로몬이 가장 많이 쌓여 있는 경로를 최단 경로로 인식하고 선택하여 먹이를 운반하게 된다. 다음의 그림은 Ant Colony Optimization 알고리즘으로 개미가 먹이를 최단경로를 찾아서 이동시키는 현상을 도식화시킨것이다.

이러한 Ant Colony Optimization 알고리즘은 네트워크에서의 스케줄링 문제, 이동체 경로 탐색 등과 같은 다양한 네트워크 최적화 문제에 적용되어 왔다[7][8].

### 2. Bee 알고리즘

꿀벌이 꽃으로부터 꿀을 채집하는 자연계의 현상에서 영감을 얻은 생체모방 알고리즘 기술을 Bee 알고리즘이라 한다.

Bee 알고리즘에서 벌의 역할을 주변에 꿀이 많은 꽃을 찾는 역할을 수행하는 정찰벌(Scouter), 꿀을 채집하는 채집벌(Forager), 벌집에서 대기하고 있는 대기벌(Onlooker)로 구분하고 있으며, 모든 벌은 세 가지의 역할을 번갈아 가며 수행하게 된다. 정찰벌은 꿀이 풍부한 꽃을 발견하게 되면 정찰벌에서 채집벌로 역할을 바꾸어 꿀을 채집하고 꿀을 벌집으로 가져온 채집벌은 벌집의 대기벌에게 자신이 채집한 꽃의 위치를 특유의 8자 춤(waggle dance)을 통해 전달하고, 채집했던 방향으로 날아가 계속해서 꿀을 채집한다. 대기벌은 벌집에 위치하고 있다가 채집벌이 알려준 위치로 가서 꽃으로부터 꿀을 채취하는 채집벌의 역할을 수행한다. 해당 지역의 꿀의 채집이 모두 끝나게 되면 채집벌은 벌집으로 복귀하여 정찰벌이나 대기벌의 역할중 하나를 선택하여 수행하게 된다.

이러한 Bee 알고리즘은 다양한 최적화 기법 연구에 적용되고 있으며, 통계적 품질관리(Statistical Quality Control), 작업 스케줄링, 로봇공학 등에서 주로 논의되고 있다[9][10].

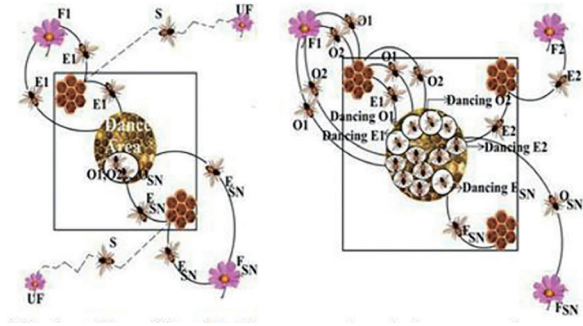


그림 2. Bee 알고리즘 개요

### 3. Firefly 알고리즘

Firefly 알고리즘은 반딧불이 반짝거릴 때 초기엔 각 개체의 고유한 진동수에 따라 반짝이게 되나 시간이 지남에 따라 각 개체 상호간의 작용으로 상대방의 반짝거림에 맞추어 최종적으로는 한 마리의 거대한 반딧불처럼 동시에 반짝거리는 현상에서 영감을 얻어 개발된 집단 동기화(Collective Synchronization) 알고리즘이다[11].

이러한 집단 동기화 현상은 쿠라모토 모델을 통하여 이해할 수 있는데, 쿠라모토 모델에서는 동일한 진동자들이 서로 약하게 연결되어 있으며 진동자들의 상호작용이 위상 차이의 사인 함수에 의존한다고 가정하여 반딧불이의 동조 현상을 모델링하고 있다[12]. 동기화가 이루어지지 않은 초기 상태의 각 개체들은 고유의 진동수를 갖고 진동하게 된다. 하지만, 각 개체들의 상호 작용으로 인하여 고유의 진동수에 변화가 발생함으로써 개체 간 결합 세기가 증대되어 각 개체의 고유 진동수를 넘어서는 수치를 얻게 되면, 각 개체 간의 동기화가 발생하여 위상 혹은 진동수가 같아지게 된다.

이와 같이 Firefly 알고리즘은 집단 동기화 현상을 효과적으로 모델링 하여 이를 네트워크에서의 동기화시 적용하고 있다.

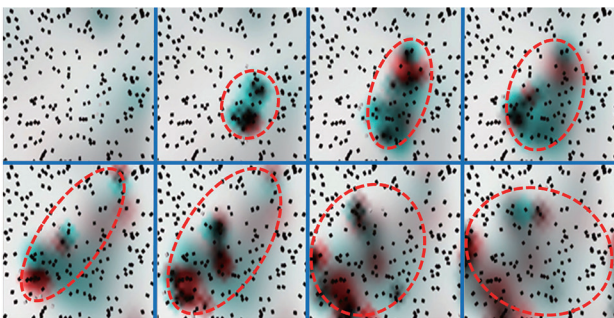


그림 3. Firefly 알고리즘 개요

### 4. Flocking 알고리즘

Flocking 알고리즘은 자연계에서 다양한 무리가 이동하는 현상에서 영감을 얻어 연구되고 있는 알고리즘이다. 철새 무리가 일정한 형태를 유지하며 집단을 형성하여 날아가는 모습이나 작은 물고기 무리가 자신들을 외부의 침입자로부터 보호하기 위해 집단을 형성하는 자기 질서화(Self-Ordered) 현상을 모델링하여 네트워크에서의 노드 배치, 스케줄링 관리, 자원 관리 등의 분야에서 다양하게 활용되고 있다[13][14].



그림 4. Bird Flocking 알고리즘

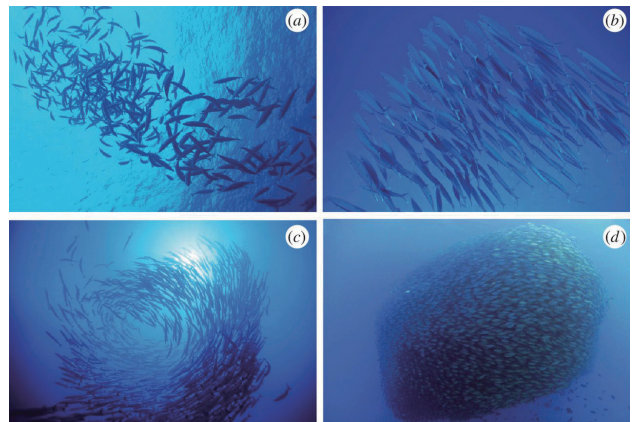


그림 5. Fish Flocking 알고리즘

Cucker-Smale 모델로 불리는 Flocking 모델은 F. Cucker와 S. Smale에 의하여 각 개체들이 간단한 균집 이론의 3가지 법칙인 분리성(Separation), 정렬성(Alignment), 결합성(Cohesion)을 통해 분산 처리 방식으로 수행하고 일정 시간의 경과한 후에 특정 수치로 수렴된다는 사실을 나타내고 있다.

Cucker-Smale 모델은 분산 처리가 요구되는 분야에서 최적화된 솔루션을 구하는데 있어서 이론적인 바탕을 제공하고 있다.



## 5. Huddling Penguins 알고리즘

Huddling Penguins 알고리즘은 추운 남극의 환경에서 살아남기 위한 황제펭귄이 집단을 이루어 서로간에 체온을 나눔으로써 일정 체온을 유지시키는 방식을 모방한 모델이다[15][16]. 황제펭귄은 남극의 차가운 바람으로부터 자신의 열 손실을 최소화시키기 위해 서로의 몸을 맞대어 체온을 유지하는데 무리의 중심과 경계에 위치한 펭귄의 이동으로 일정 체온을 유지하고 있다. 경계에 위치한 펭귄은 자연스럽게 바람의 방향으로 이동하게 되고, 중심에 위치한 펭귄들이 경계로 순환되는 현상을 모방하고 있다.

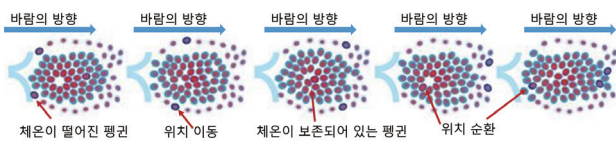


그림 6. Huddling Penguins 알고리즘 개요

Huddling Penguins 알고리즘은 평균적으로 동일하게 에너지소모를 유지할 수 있도록 모델링 되어 있다.

## Ⅲ. 자율제어 네트워크 환경

새로이 등장한 자율제어 네트워크 서비스에서는 어플리케이션들의 요구사항이 복잡해짐에 따라 이를 위한 새로운 네트워크 정책과 프로토콜을 필요로 하고 있다. 하지만, 이미 인터넷 환경은 수많은 사용자들을 바탕으로 광범위하게 영향을 미치고 있으므로 새로운 프로토콜을 도입시키거나 성능향상을 도모한다는 이유로 인터넷 자체의 물리적인 구조를 변화 시킨다는 것은 어려움이 따른다. 또한, 네트워크에서 이상 트래픽이 발생하거나 새로운 이벤트가 발생하는 경우 네트워크 관리자들에 이를 해결하기 위해 새로운 정책을 수립하여 네트워크에 반영할 수 있는 형태로 변환하여 적용하게 되는데, 여기서 비효율적인 처리요소가 발생하게 된다.

이러한 변화를 효율적으로 수용할 수 있는 미래 인터넷 기술 중 하나가 바로 SDN(Software Defined Network)이라고 할 수 있다. SDN은 패킷 포워딩 하는 부분과 네트워크 제어 기능을 분리시켜 직접 프로그래밍할 수 있는 구조이며 네트워크 추상화를 통해 설계와 운용과정을 보다 용이하게 만든 개방형 플랫폼이다[17]. 초기 SDN에서는 중앙집중식 환경으로 구성되어 적용되었으나, 다양한 환경에서 발생하는 문제점을 자율적으로

제어하기 위하여 다중 컨트롤러를 이용한 SDN 환경이 대두되고 있다[18].

네트워크에서의 주된 문제점에 대하여 앞장에서 언급한 생체모방 알고리즘의 도입의 필요성을 살펴본다.

### 1. 네트워킹 환경의 초 거대화

미래 인터넷의 규모는 구성 노드 수 측면에서 현재보다 자리수가 기하급수적으로 증가할 것으로 예상되고 있다[19]. 이동 애드혹 네트워크(Mobile Ad-Hoc Network: MANET)를 구성하는 경우 및 센서 네트워크(Sensor Network)를 구현하는 경우 미래 애플리케이션 서비스에 대해서 최대 수십에서 수백만 개의 노드 수를 고려해야 원활한 서비스가 가능하게 된다.

이렇듯 네트워크에 연결되는 노드 수가 증가하게 되면 네트워크에 매우 큰 부하를 야기시키게 되며, 발생하는 트래픽 부하는 네트워크에서 가지고 있는 용량을 초과할 것이고, 이는 전송 경로 상의 혼잡 및 충돌 발생에 의한 패킷 손실 등으로 이어져 통신의 신뢰성에 심각한 타격을 줄 것이다. 이를 적당한 노드 수의 선택으로 해결하려 한다면 시스템의 복잡도를 증가시키게 되어 신뢰성 보장을 위한 제어 정보의 주기가 네트워크의 규모와 함께 증가하게 된다. 아울러, 네트워크 규모가 증가함에 따라 연결 가능한 경로도 증가하게 되어 최적 경로 선정을 위한 탐색이 기하급수적으로 증가하게 되므로 경로 결정을 위한 라우팅 테이블, 테이블 갱신을 위한 제어 정보 교환에서의 오버헤드 등이 비례하여 증가하게 된다.

기존 네트워크 기술로는 MANET 및 센서 네트워크와 같은 대규모 네트워크 환경에서 다량의 노드 설치, 효과적인 통신 및 유지 관리가 불가능 함에 따라 네트워킹 알고리즘은 자율적으로 네트워크 크기 변화에 확장이 가능(Scalable)하여야 하고 적응적(Adaptive)인 동작을 수행할 수 있는 생체모방 알고리즘의 도입이 필요로 한다.

### 2. 자율네트워크 환경

기존의 네트워크 환경은 정적인 통신 채널로 구성된 통신 시스템이나, 향후 도래하게 될 네트워크의 구조는 노드의 이동성, 트래픽의 방대한 양, QoS를 보장하기 위한 대역폭 요구량, 채널상태, 네트워크 구조 등에 있어서 동적으로의 구성이 필요할 것이다. MANET에서는 노드의 이동이 발생하게 되는 경우 네트워크의 크기, 전송 거리, 채널 상태, 통신 링크 등이 계속하여 갱신이 발생하게 된다[20]. 또한, 센서 네트워크의 트래킹 시스템인 경우 노드의 이동에 따라 해당 지역의 센서에서 발생하는 트래픽 양은 급격히 증가하다가 감소하게 된다. 이러한 경우 기존의 정적인 네트워크 관리 방안은 발생하는 동적인 부하량

을 감당할 수 없으므로 인하여 네트워크 용량에 비효율성을 발생시키게 된다. 따라서, 자율네트워크를 구성하기 위해 생체모방 알고리즘의 구현을 통해, 특히 생체 시스템의 인공 면역 체계(Artificial Immune System)와 같이 환경 변화에 적응이 뛰어난 알고리즘이나 활성화-억제(Activator-Inhibitor) 시스템과 같이 동적인 환경에 적응력이 뛰어난 생체모방 알고리즘을 적용함으로써 효율적으로 변화를 감지하고 예측 되어진 시스템 패턴에 적응이 가능하다.

### 3. 자원제약적인 환경

네트워크 환경의 변화는 다수의 노드가 네트워크에 연결되는 형태로 급속하게 진행되고 있다. 이렇듯 네트워크 기술이 발전함에 따라서 서비스 양, 대역폭 등과 같은 서비스 질의 수준을 보장해주어야 함을 요구하고 있으나 물리적인 네트워크 상에서의 연결은 그 한계를 나타내고 있다. 이러한 자원제약적인 환경에서 효율적으로 자원을 활용하기 위한 고차원적인 자원 할당 관리 방식이 필요로 한다.

생체모방 알고리즘은 한정된 환경에서 효율적으로 자원을 사용할 수 밖에 없는 생태계를 기반으로 하는 알고리즘이기 때문에 발전하는 네트워크의 문제점인 자원제약적인 환경에서 보다 능동적인 자원 할당 방식을 제공할 수 있다. 예를 들어, Ant Colony Optimization 알고리즘의 경우 개미가 효율적인 방법으로 먹이와 개미집간의 최적 이동 경로를 산정하게 하는 방식으로 이는 네트워크 환경에서 비용 효율적인 방법으로 경로 산정에 적용시켜 사용이 가능하다.

### 4. 상호운용성

사물 인터넷(Internet of Things)은 개개의 사물을 네트워크에 연결시켜 대규모 무선 네트워크로 확대하는 것이 목표이나 사물 인터넷을 이루는 구성 노드가 다양한 이기종으로 구성 될 것이라는 문제점을 해결하는 방법은 요원한 현실이다. 이기종 간의 네트워크 연결을 위해서는 통신 단말의 능력과 이기종 간 통신 규약이 모델링 되어 효과적으로 관리되어야 할 것이다.

생체모방 알고리즘의 기반이 되는 자연계에서도 다양한 이종 간의 연결 및 관리방안이 관측되어지며 이를 크게 생체 시스템과 생명체의 개체 구성 부분으로 구분이 가능하다. 생체 시스템의 경우 다수의 유기체가 외부의 교란에도 안정적인 내부 상태를 유지할 수 있는데, 이는 신경체계, 내분비체계, 면역체계 등과 같은 이질적인 시스템들 간의 상호 협력을 기반으로 항상성(Homeostasis)을 유지함으로써 가능하게 한다. 생체 시스템을 기반으로 하는 생체 모방 알고리즘은 이기종 간의 연결 구조를 갖는 네트워킹 기술 개발에 적용이 가능하다. 반면에 곤충 집단

과 같은 생명체의 개체 구성은 외부의 환경 자극에 대하여 서로 다른 다양한 반응 능력을 갖고 동작하고 있다. 생명체의 개체 구성을 기반으로 하는 생체모방 알고리즘은 집단 지성을 통하여 임무 할당과 프로세스 선택을 최적화 시킬 수 있는 네트워킹 기술에 접목시킬 수 있다.

## IV. 생체모방 알고리즘 자율제어 네트워크 구현 기술 동향

SDN과 OpenFlow의 관계가 밀접하다고 알려져 있으나, 사실상 SDN은 훨씬 상위 개념으로 네트워크 구조 혹은 새로운 패러다임이고, OpenFlow는 SDN을 위한 하위 인터페이스 기술 중의 하나이다. SDN이 개념적인 아키텍처라고 한다면 OpenFlow는 통신을 하기 위한 프로토콜이라고 볼 수 있다. 생체모방 알고리즘을 자율제어 네트워크에 적용하기 위해 우선적으로 SDN이라는 상위의 큰 개념을 살펴보고 하위의 인터페이스인 OpenFlow 기술과, 새로이 연구되고 있는 P4: Programming Protocol-Independent Packet Processors 기술에 대해서 살펴본다.

### 1. SDN

SDN은 소프트웨어 프로그래밍으로 네트워크 경로설정 및 제어, 그리고 복잡한 운용관리를 유연하게 처리할 수 있는 네트워킹 기술이다. SDN에서는 표준화된 인터페이스를 제공함으로써 네트워크의 데이터 평면(data plane)과 제어 평면(control plane)을 분리하여 네트워크 운용자가 다양한 상황을 염두에 두고 데이터 평면에서 이루어지는 통신 기능을 제어 평면의 프로그래밍을 통하여 다양한 방식으로 제어할 수 있게 한다. 네트워크 제어기능을 스위치에서 분리하여 중앙화 함으로써 네트워크 자원의 가시성을 확보할 수 있으며, 이를 바탕으로 유연한 제어와 효과적인 네트워크 관리를 가능하게 하는 네트워크 제어·관리 기술이다. SDN에서는 다음과 같은 두가지 기본적인 원칙을 바탕으로 이루어지고 있다.

우선, SDN은 개방형 인터페이스와 소프트웨어를 통하여 하드웨어 기반의 스위치 및 라우터에서 처리하는 데이터 포워딩 기능이 반드시 제어가 됨을 의미하는 소프트웨어 정의 포워딩(Software Defined Forwarding)이 가능해야 한다.

그리고, SDN에서 진보된 네트워크 관리 툴의 개발이 추상화를 통해 가능 하도록 해야 하며 이런 추상화 도구들은 전체 네트워크의 상태를 모니터링 하면서 토폴로지 변화나 새로운 플로우 입력 등과 같은 이벤트에 따른 대처나 네트워크 요소를

제어하는 기능 등이 포함 가능한 글로벌 관리 추상화(Global Management Abstraction)를 목표로 하고 있다.

SDN의 개념 구조를 살펴 보면 L0에서 L3 스위칭 기능을 하는 데이터 전달장치가 위치하는 인프라 계층(Infrastructure Layer), 전체 네트워크 상태의 글로벌 뷰를 사용하여 네트워크 동작을 제어하는 네트워크 컨트롤러의 제어 계층(Control Plane), 그 상위에서 동작하는 응용이 위치하는 응용 계층(Application Layer)으로 구성되는 총 3개의 계층 구조로 표현된다. 각 계층 간의 원활한 연동을 위해서 사우스바운드 인터페이스(Southbound Interfaces)와 노스바운드 인터페이스(Northbound Interfaces)가 존재하게 된다[21].

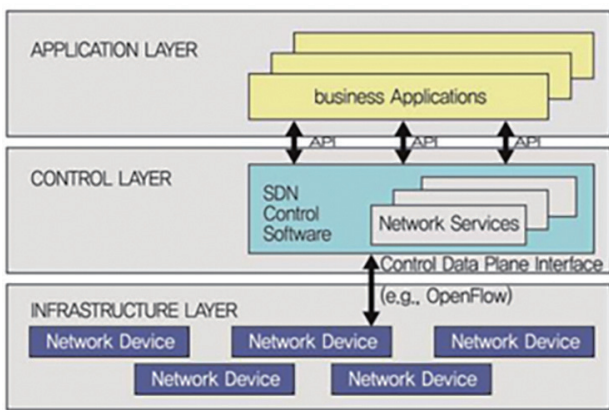


그림 7. SDN 개념 구조

네트워크를 관리하는 기능은 SDN 컨트롤러에 집중화 되어 있어서 네트워크를 포괄적으로 관리하고 있음으로 네트워크 설계와 운용을 단순한 형태의 네트워크로 구성이 가능해 진다. 따라서, 단순하게 추상화된 네트워크를 네트워크 관리자는 프로그램만으로 구성하여 다수의 장치에 분산되어 있는 구성정보를 수작업을 통해 관리할 필요 없이 보다 손쉽게 제어가 가능하다.

## 2. OpenFlow

SDN의 대표적인 표준 기술인 OpenFlow는 2009년 ONF(OpenFlow Network Foundation)에 의해 표준화된 이래 CISCO, HP 등 큰 규모의 네트워크 벤더들이 SDN 스위치를 꾸준히 출시하고 있으며, SDN 망을 제어하는 오픈 소스 기반의 Controller도 Ryu, FloodLight, OpenDayLight 등이 등장하고 있다.

SDN 으로 구현된 네트워크에서 노드들은 OpenFlow 스위치와 연결되고, 그 OpenFlow 스위치들은 OpenFlow Controller와 연결된다. 그래서 OpenFlow Controller가 직접 연결되어 있는 OpenFlow 스위치들과 간접적으로 연결된 노드들, 그리고

SDN 네트워크 자체의 현황을 중앙집중화(Centralization)된 형태로 모니터링이 가능하다. 중앙집중화된 모니터링을 토대로 OpenFlow Controller는 SDN 네트워크 내의 자원을 관리하면서 서비스를 제공할 수 있으며, 이런 서비스들은 응용 서비스 프로그램의 형태로 제공된다. 응용 프로그램은 이러한 서비스를 바탕으로 API를 이용해 Northbound로 Controller에 관여하여 SDN 네트워크에 제어할 수 있기 때문에 Programmable 한 환경이 가능하게 되며, 오픈소스로 된 Controller들은 이 점에 착안해 다양한 서비스와 프로토콜을 지원하고 있다.

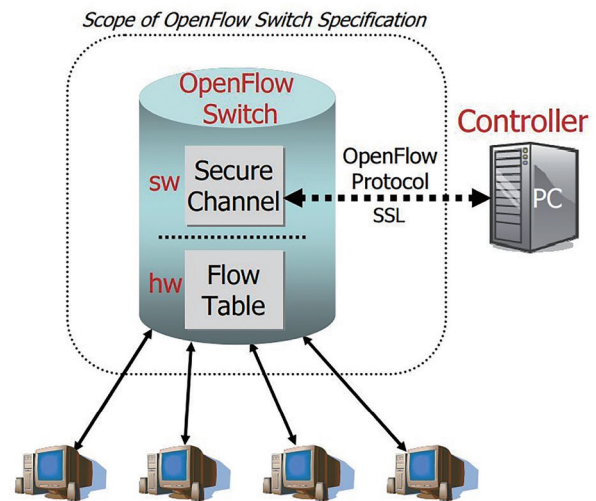


그림 8. OpenFlow 개념 구조

이러한 기술들을 이용한다면 SDN네트워크에서 새로운 프로토콜과 서비스를 설계하고 구현할 수 있으며, SDN 네트워크 내에 있는 노드들의 주소를 수집하여 관리가 가능하다. SDN 기반의 네트워크에서 기존 네트워크의 노드만이 아니라 OpenFlow Controller를 통해 이동성 노드에 대한 관리도 Home agent와 Foreign agent를 통하지 않고 가능하며 SDN 네트워크 내의 라우팅 메커니즘도 OpenFlow Controller를 통해 관리가 가능할 수 있게 된다.

## 3. P4: Programming Protocol-Independent Packet Processors

SDN은 기존 인터넷 라우터들에 데이터 평면만을 두고, 제어 평면을 상위의 컨트롤러에 둬으로써 분산되어 있던 라우팅 제어 및 관리를 하나의 컨트롤러에서 담당하고 있어서 네트워크 라우팅 경로 설정 및 모니터링이 가능해졌으나, 패킷을 처리하는 방식 그 자체는 프로그래밍하거나 변경할 수 없는 제약사항이 여전히 존재하고 있다. 기존에 알려진 프로토콜 스택을 처리하는 방식을 스위치상에서 프로그래밍하여 사용하기 때문에 이에 대



한 변경이 불가능함에 따라 이를 유연하게 하고자 최근 스위치에 패킷 프로세싱 방식을 소프트웨어로 프로그래밍하려는 시도가 진행되었고, 그 중 대표적인 방식으로 P4: Programming Protocol-Independent Packet Processors가 대두되고 있다.

프로그래밍 가능한 스위치에 패킷 프로세싱 방식을 프로그래밍하는 것으로, 여기서 P4는 그것을 가능하게 해주는 프로그래밍 언어에 해당한다. P4는 API와의 관계를 나타내고 있으며 포워딩 테이블을 스위치의 기능으로 고정시켜 상주시키도록 설계하고 있다.

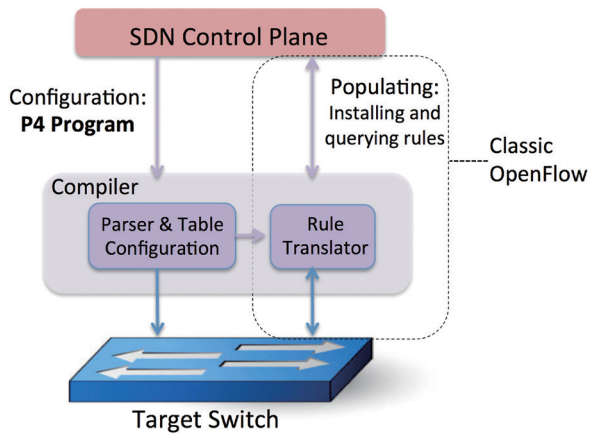


그림 9. P4: Programming Protocol-Independent Packet Processors 개념 구조

P4는 C와 Python 등의 범용 언어와는 대조적으로, P4는 네트워크의 데이터 전송에 최적화 된 구조로 특정된 언어로서 패킷 전송 데이터 플랜 프로그래밍을 가능하게 하도록 설계된 프로그래밍 언어이다.

P4는 컨트롤러가 패킷의 구문 및 처리를 즉각적으로 다시 정의 내릴 수 있도록 재구성성을 제공하여야 하며, 스위치는 특정된 패킷 포맷에 종속되지 않도록 프로토콜 독립적이어야 하고, 시스템에 종속되지 않도록 적용 시스템에 독립적인 목표를 갖고 있다.

P4는 프로토콜에 의존하지 않도록 설계되어 있어서 프로그램에 필요한 프로토콜 헤더 형식과 필드 이름을 명기하여 컴파일된 프로그램과 대상 장치에 의해 해석되어 순서대로 처리 된다.

P4 프로그래밍 패킷들은 범용 칩 CPU와 FPGA 시스템, 네트워크 프로세서 및 ASIC과 실행 시스템의 다양한 종류에 대해 컴파일이 가능하도록 구현되어 있으며, 시스템의 구현에 의존하지 않도록 설계되어 있다. P4에서는 각 시스템에서 구현시 실행 가능한 형식으로 구현되며 P4프로그래밍 패킷을 소스 코드로 변환할 수 있는 컴파일러와 포함되어 있어야 한다[22].

P4를 통한 패킷 전달 모델은 구성과 실장의 두가지 제어로서

이루어 진다. 동작 프로그램의 파서를 통해 일치하는 작업 단계의 순서를 설정하고 각 단계에서 처리되는 특정 헤더 필드를 지정하도록 구성한다. 구성정보에서는 프로토콜을 지원하는 방식과 스위치에서 어떻게 패킷을 처리하는지를 결정한다. 실장의 동작은 구성에서 특정지어진 일치하는 작업에서의 항목을 추가하거나 삭제한다. 실장은 어떤 상황에서도 패킷에 의해서 정책이 결정되어짐을 의미한다.

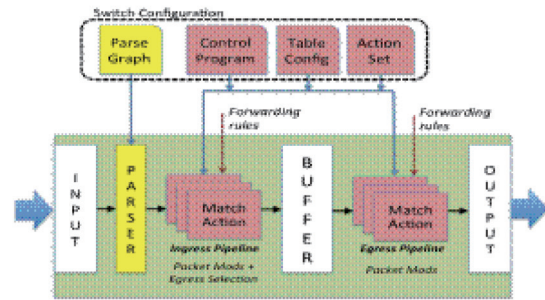


그림 10. P4 패킷 처리 전달 모델 개념

이렇듯, P4 기술을 사용하여 SDN 네트워크를 구성하게 되면 기존 SDN보다 유연하고 효율적인 네트워크 인프라 제어 및 관리가 가능하다는 장점과, 보다 진보된 상호 운용성을 제공할 수 있다. 더불어, 생체모방 알고리즘을 적용시켜 네트워크를 구현함으로써 적응형 자율제어 네트워크 구현에 많은 효용을 제공할 수 있다. P4 컨소시엄에서는 이러한 P4를 OpenFlow 2.0으로 정착시키고자 P4 workshop을 열고 관련 학계의 참여를 촉진하고 있으며 P4에 대해서 활발히 홍보하고 는 추세이다.

## V. 결론

본 고에서는 생체모방 알고리즘을 활용하여 현재 네트워크의 문제사항을 해결할 수 있는 방안에 대하여 살펴 보았다. 향후 네트워크의 발전이 자율제어 네트워크 환경으로 초거대화해 이기종간의 다양한 연결을 주제로 진행되고 있으며, 이를 원활하게 사용하기 위해서는 다양한 생체모방 알고리즘을 네트워크에 적용시키는 연구가 필요하다고 보여진다. 여기서, 생체모방 알고리즘을 적용하여 자율제어 네트워크 환경을 구현함에 있어서 기반 기술이 되는 SDN 환경을 살펴보고, SDN을 구현하기 위한 기술로서 OpenFlow 기술을 정리해 보았다. 하지만, OpenFlow 기술의 제약사항이 존재하여 이기종간의 상호 운용성 지원이나 적응형 자율제어 네트워크 구성에는 어려움이 따르게 된다. 따라서, 현재 새로이 연구되고 있는 P4: Program-

ming Protocol-Independent Packet Processors 기술을 사용하여, 프로그래머블한 패킷으로 OpenFlow 환경에서의 제약 사항을 해결할 수 있다. 생체모방 알고리즘을 접목한 자율제어 네트워크 환경에서 지능형 패킷을 활용하는 구현 기술과 관련된 국내의 연구는 미비한 실정이며, 상기 기술의 연구를 통해 자율 네트워크에서 프로그래밍이 가능한 패킷들이 처리됨으로써 이기종간의 시스템에서 상호 운용성을 확보 할 수 있고, 자원 제약적인 네트워크 환경에서 보다 효율적인 방식으로 네트워킹 동작이 가능해 진다.

## 참고 문헌

- [1] Craig W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model", ACM Computer Graphics, vol. 21, no. 4, pp. 25-34, 1987.
- [2] Hui Yu, Ji-Gui Jian, "Flocking motion control of mobile agents based on distance-dependent adjacency matrix", International Conference on Wavelet Analysis and Pattern Recognition, vol. 1, pp. 17-22, Aug. 2008.
- [3] Seung-Yeal Ha, Taeyoung Ha, Jong-Ho Kim, "Emergent Behavior of a Cucker-Smale Type Particle Model With Nonlinear Velocity Couplings", IEEE Transactions on Automatic Control, vol. 55, no. 7, pp. 1679-1683, July 2010.
- [4] Demetriou, M.A., "Distributed parameter methods for moving sensor networks in unison", American Control Conference, 2008, pp. 273-278, June 2008.
- [5] Schwager, M., Slotine, J.-J., Rus, D., "Consensus learning for distributed coverage control", IEEE International Conference on Robotics and Automation, pp. 1042-1048, May 2008.
- [6] Bin Xu, Kurdila, A.J., Stilwell, D.J., "Geometric ergodicity of the distributional consensus problem in vehicle network control", IEEE Conference on Decision and Control (CDC), pp. 7499-7506, Dec. 2010
- [7] Dorigo, M., Birattari, M., Stutzle, T., "Ant colony optimization", IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28-39, Nov. 2006.
- [8] M. Dorigo, C. Blum, "Ant colony optimization theory: A survey", Theoretical Computer Science, vol. 344, pp.243-278, Nov. 2005.
- [9] Karaboga, D., "An idea based on honey bee swarm for numerical optimization.", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [10] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M, "The Bees Algorithm," Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
- [11] G. Werner-Allen, G. Tewari, A. Patel, R. Nagpal, and M. Welsh, "Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects," In SenSys, 2005.
- [12] J.A. Acebron, L.L. Bonilla, C.J. Perez-Vicente, F. Ritort, R. Spigler, The Kuramoto model: "A simple paradigm for synchronization phenomena", Rev. Mod. Phys., vol. 77, pp. 137-185, 2005.
- [13] Olfati-Saber, R., "Flocking for multi-agent dynamic systems: algorithms and theory," IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 401-420, March 2006.
- [14] J. A. Carrillo, M. Fornasier, J. Rosado, and G. Toscani, "Asymptotic flocking dynamics for the kinetic Cucker-Smale model," SIAM Journal on Mathematical Analysis, 2010.
- [15] D.P. Zitterbart, B. Wienecke, J.P. Butler, B. Fabry, "Coordinated Movements Prevent Jamming in an Emperor Penguin Huddle," PLoS one 6(6), e20260, 2011.
- [16] A. Waters, F. Blanchette, A.D. Kim, "Modeling Huddling Penguins," PLoS one 7, e50277, 2012.
- [17] 윤현식, 강경순, 김학서, 박혜숙. "SDN에서 데이터 경로 설계에 대한 고려사항". 한국통신학회지 제30권 제3호, p.22-28. February 2013.
- [18] 백상현, 장인선, 서동은, 이종화. "미래 네트워크의 새로운 패러다임 SDN/NFV에 대하여". 한국통신학회지 제32권 제 7호, p.82-92. June 2015.
- [19] "OUTLOOK: Visions and research directions for the Wireless World," WWRP(World Wide Radio Forum), no.4, July, 2009.
- [20] I. Chlamtac, M. Conti, J.J. Liu, "Mobile ad hoc networking: imperatives and challenges," Elsevier Ad



Hoc Networks, vol. 1, no. 1, pp. 13-64, 2003.

- [21] ONF White Paper, "Software-Defined Networking: The New Norm for Networks," Apr. 2012.
- [22] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, P4: Programming protocol-independent packet processors. SIGCOMM Comput. Commun. Rev., 44(3):87-95, July 2014.

## 약 력



진 정 하

2002년 금오공과대학교 전자통신공학과 공학사  
2006년 건국대학교 공학석사  
2014년~현재 건국대학교 박사과정  
관심분야: 미래인터넷, SDN/NFV, IoT, sensor network



김 기 천

1988년 서울대학교 전산학 공학사  
1992년 미국 northwestern Univ. 공학박사  
1992년~1996년 한국통신기술(주) 선임연구원  
1996년~1998년 신세기 통신(주) 책임연구원  
1998년~현재 건국대학교 컴퓨터공학과 교수  
관심분야: 미래인터넷, SDN/NFV, IoT, sensor network