

미래인터넷을 위한 자율 네트워킹 및 관리 기술 연구동향

유재형, 홍원기
포항공과대학교

요약

자율 네트워킹(autonomic networking)은 인간의 개입을 최소화하면서 환경의 변화에 따라 네트워크 스스로 자율적인 관리(self-management) 및 제어를 하도록 하는 혁신적인 개념이다. 본 고에서는 대규모이면서 이질적이고 동적인 특성을 갖는 미래인터넷의 관리에 필수적인 기술로 떠오르고 있는 자율 네트워킹의 개념과 연구동향 및 이슈에 대해 살펴본다.

I. 서론

2020년을 목표로 개발되고 있는 미래인터넷 구축을 위해 인터넷 설계시에 고려하지 못한 보안성, 이동성, 품질 및 관리의 문제를 해결하는 연구가 진행되고 있다. 이와 동시에 미래인터넷의 주요 응용 분야로 IoT(Internet of Things)[1,2], MANET(Mobile Ad-hoc Network)[3], WPAN(Wireless Personal Area Network) [4,5], DTN(Delay Tolerant Network)[6-8], SDN(Software Defined Network)[9]과 같은 네트워킹 기술들의 개발도 병행되고 있다. 이러한 다양하고 이질적인 네트워크로 구성되는 미래인터넷은 관리의 측면에서 다음과 같은 문제들을 야기할 것으로 예상된다.

- 확장성: 수천~수만 개의 노드들로 구성되는 단위 네트워크들이 상호 연동하고 백본 네트워크에 연결하여 서비스를 제공하는 미래인터넷은 전세계에 걸쳐 수백억 ~ 수조 개의 노드로 구성되어 현재와 같은 집중관리 방식으로는 운용이 불가능해진다.
- 관리의 복잡도: 미래인터넷은 이질적이고 다양한 노드와 단말들로 인해 관리의 복잡도가 크게 높아진다. 즉, 인간 관리자가 숙지해야 할 지식의 양이 방대해지고 이에 따라 네트워크의 구성 오류가 발생할 가능성도 높아진다.
- 동적 대응: 네트워크 구조는 다양한 유무선 네트워크 간의 연동 및 단말 이동, 에이전트와 콘텐츠 이동, 실시간 대역폭 변

경 (SLA, QoS 변동) 등으로 예측이 어려운 수준으로 동적으로 변화하여 현재와 같은 인간 관리자 중심의 관리체계로는 실시간 관리가 불가해진다.

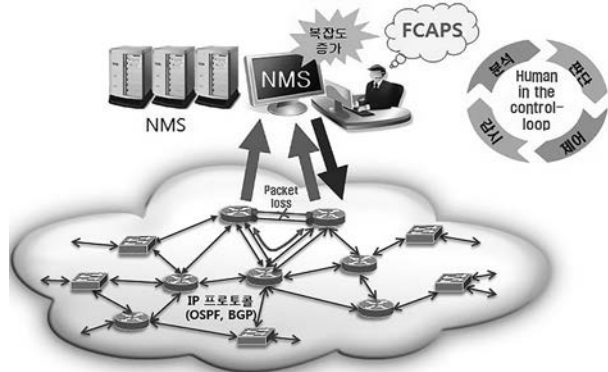


그림 1. 인간 중심의 집중식 관리 체계

〈그림 1〉은 집중관리 방식의 문제를 개념적으로 보인 것이다. 네트워크에서 어떤 이벤트(장애)가 발생하면 관리 시스템이 그 이벤트를 수집하고 분석하여 인간 관리자가 신속한 조치를 하는데 필요한 정보로 변환해준다. 하지만 현재의 관리 방식은 사람이 정확한 조치방법을 결정하기 않는 한 그 이벤트는 미해결 상태로 남는다는 근본적인 문제를 가지고 있다. 즉, 사람이 제어 루프에서 한 부분을 차지하고 있으며 때로는 실수를 할 수

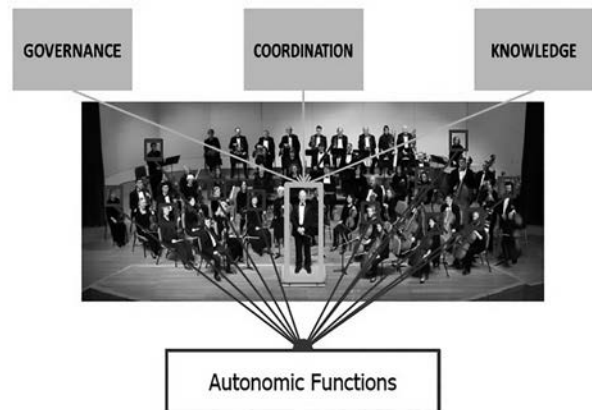


그림 2. 자율 네트워킹과 관리의 개념

도 있다는 문제를 갖고 있다(Human in the control loop).

자율 네트워킹 및 관리는 자동제어의 피드백 루프에서 진일보한 개념으로서 최소한의 중앙통제로 전체가 조화롭게 동작하도록 하는 것을 의미한다. 예를 들어 <그림 2>에 보인 오케스트라는 자율성과 중앙통제의 원리를 잘 표현하는 것으로서 연주의 목적을 잘 알고 있는 한 명의 지휘자가 전체를 통제(지휘)하여 숙련된 악기 연주자들이 전체적으로 잘 조화된 화음을 만들도록 하는 것과 같은 개념이다. 여기서 지휘자는 직접 악기(네트워킹)를 연주(제어)하지 않는다. 즉, 악보에 따라 악기를 연주하는 제어 루프에서 벗어나 있고 상위레벨에서 오케스트라가 이루어야 할 목표를 제시하는 역할을 한다.

자율 네트워킹 및 관리는 개별 네트워크 노드(또는 S/W 시스템)에 Self-★ (Self-configuration, healing, organization, optimization, protection 등) 능력을 부여하여 최소한의 인간 개입으로 전체 네트워크가 자율적으로 동작하게 하는 기술이다. 자율 네트워킹에서는 최상위 관리자가 결정한 정책을 만족하는 범위내에서 각 네트워크 노드가 인접한 다른 노드들과 연동하여 스스로 상태를 제어하므로 대규모 네트워크에서 확장성 문제를 해소할 수 있고 학습된 지식을 이용해 훨씬 복잡한 제어를 수행하도록 할 수도 있다.

이 다양한 분야의 연구를 소개하기 위해 II 장에서는 우선 자율 컴퓨팅의 역사와 개념 및 현재까지의 주요 연구결과를 설명하고 III 장에서는 미국과 유럽에서 자율 네트워킹 프로젝트로 진행된 주요 연구들에 대해 소개한다. IV 장에서는 자율 네트워킹 기술의 적용 분야와 연구 이슈를 살펴보고 V장에서 결론을 맺는다. 그리고 본 고에서는 자율 컴퓨팅과 자율 네트워킹, 자율관리라는 용어를 같은 의미로 혼용하여 사용하기로 한다.

II. 배경지식

1. 자율 컴퓨팅의 역사

최초의 자율관리 프로젝트는 1997년 DARPA에서 군사 목적으로 시작되었다[10]. 이 프로젝트는 SAS(Situational Awareness System)로 불리며 전장에서 병사들을 위한 개인용 통신 단말을 개발하는 것을 목표로 했다. 병사들이 주변 상황에 대한 정보를 입력하면 이 정보는 자율적으로 다른 병사의 단말에 전달된다. 정보는 음성뿐 아니라 지상에 설치된 센서 및 무인 비행기로부터 수집된 데이터를 포함한다. 이러한 개인 단말은 멀티홉 애드혹 라우팅(Multi-hop ad-hoc routing)[3]으로 통신하며 자율관리를 위해 적응형 라우팅을 사용하였다.

특히, 이 프로젝트는 약 1만개의 단말들 간에서 지연을 200ms 이하로 유지하며, 전파방해를 피하기 위해 20~2500 MHz의 광역 주파수를 사용하고 전송 대역폭은 단말 간 거리에 따라 10bps~4Mbps로 가변적으로 한다는 구체적인 목표를 가지고 추진되었다.

2001년에 IBM은 자율 컴퓨팅이라는 개념을 제안했다[12]. 이 개념은 복잡한 컴퓨팅 시스템을 인체에 비교한 것이다. 인체는 복잡계이지만 자율 신경계가 대부분의 신체 기능을 조절하므로 인간의 의식은 신체의 모든 기능을 제어하는 데에서 자유로울 수가 있다. IBM은 관리자의 부담을 줄이고자 복잡한 컴퓨팅 시스템도 자율적인 기능을 가져야 한다고 제안하였다.

IBM은 자율관리에 대해 4가지 기능을 정의했는데 이들은 자율관리 즉, 자기-구성/최적화/복구/보호(Self-configuration, self-optimization, self-healing 및 self-protection)이다. 이러한 기능들에 대해서는 2절에서 자세히 설명하기로 한다.

DARPA는 2004년에 SSP(Self-Regenerative Systems Programme)[24]라는 프로젝트에 착수하였다. 이것의 목표는 컴퓨터가 손상을 입어도 중요한 기능은 항상 수행할 수 있는 군사용 컴퓨팅 시스템을 개발하는 것이었다. 이 프로젝트에는 4개의 핵심요소가 있는데, 첫째로 S/W는 유사한 기능을 수행하는 다수의 버전을 상이한 기술로 생성하여 오류와 공격에 대한 내성을 갖도록 하였다. 둘째로 무의미한 블록들을 메모리 스택에 삽입하여 공격자가 취약점을 이용하기 어렵게 하고 특정 분기주소(branch address)를 찾기 어렵게 하였다. 셋째로 인가를 받은 클라이언트의 악의적인 갱신 시도를 막기 위해 신뢰 평가를 수행한다. 넷째로 군사 시스템의 운용자(내부자)가 시스템을 공격할 경우에 이를 차단하는 기술도 개발하고 있다.

미국의 NASA는 1998년과 1999년에 태양계의 혜성과 화성 탐사를 위해 각각 발사된 DS(Deep Space) 1호 및 2호 그리고 1996년에 발사된 화성 탐사선 패스파인더(Pathfinder)에 자율 제어 기술을 적용했다[25]. 그 이유는 탐사선과 지구상의 컨트롤 타워 간에 아주 긴 전파지연이 있기 때문이다. 즉, 탐사선과 교신하여 컨트롤타워에서 새 명령을 내리는 데 수 분~수 십분이 소요되므로 고가의 탐사선은 예외적인 상황에 대응하기 위해 자율적인 행동 능력을 가져야 했다.

이후에 수행된 주요 프로젝트에 대해서는 III장에서 서술한다.

2. 자율관리 속성

IBM이 제시한 자율관리의 주요 속성은 다음과 같다[11]. 현재의 자율 컴퓨팅 연구에서는 하나 이상의 self-★ 기능을 수행하는 시스템을 자율적이라고 말하고 있다.

• 자기 구성(Self-configuration)

자율 컴퓨팅 시스템은 상위레벨의 목표에 따라 자기구성을 한다. 상위레벨의 목표는 무엇을 원하는가를 알려주는 것이며 그것을 어떻게 실행하는가를 결정해 주지는 않는다. 이것의 예는 사용자의 요구에 따라 S/W를 다운로드 받아 설치하고 구성하는 것이다.

• 자기 최적화(Self-optimization)

자율 컴퓨팅 시스템은 자원의 사용을 최적화해야 한다. 이것은 서비스의 성능이나 품질을 개선하기 위해 어떤 상황이 발생하기 전에(proactively) 시스템의 변경을 시도하는 것이다.

• 자기 복구(Self-healing)

자율 컴퓨팅 시스템은 문제를 검출하고 진단할 수 있어야 한다. 검출된 문제는 메모리칩의 비트 에러와 같은 낮은 레벨의 H/W 장애일 수도 있고, 디렉토리 서비스의 잘못된 엔트리 입력과 같은 상위레벨의 S/W 문제일 수도 있다. 자율 컴퓨팅 시스템은 가능한 한 문제해결을 시도해야 한다. 장애 발생시에 예비시설로 교체하거나 S/W 업데이트를 다운받아 설치하는 것이 그 예로서 장애 또는 장애의 징조에 대해 반응적인(reactive) 무정지(fault-tolerant) 시스템의 구축이 있다.

• 자기 보호(Self-protection)

자율 컴퓨팅 시스템은 악의적인 공격으로부터 자신을 보호할 수 있어야 한다. 즉, 중요한 파일을 삭제하는 등의 사용자 부주의로 인한 S/W 변경도 막을 수 있어야 한다. 이 시스템은 보안, 개인정보 보호, 데이터 보호 등을 하기 위해 자율적으로 자신을 튜닝할 수 있어야 한다. 보안은 자기 보호의 중요한 요소로 H/W와 S/W 보안을 모두 고려해야 한다. 시스템은 보안 침해를 예측하고 사전에 이를 막을 수 있어야 한다.

3. MAPE-K 자율 루프(Autonomic Loop)

MAPE-K(Monitor, Analyse, Plan, Execute and Knowledge)는 자율 컴퓨팅 실현을 위해 IBM이 제안한 자율 루프(Autonomic Loop) 참조 모델이다. <그림 3>에 보인 MAPE-K 모델은 자율 시스템의 구조적인 측면을 나타낸 것으로 자율 컴퓨팅 및 네트워킹 연구 분야에서 많이 인용되고 있다.

MAPE-K의 관리대상 요소는 S/W 또는 H/W 자원이며 이들은 하나의 자율 관리기와 결합하여 자율적인 동작을 한다. 센서는 관리 대상 요소로부터 필요한 정보를 수집하며 실행기(effector)는 관리대상 요소에 대한 어떤 변경을 실행한다. 이러한 변경은 라우터에 특정 프로토콜을 동작/정지시키는 것이거나 링크에 할당된 메트릭 값 등의 파라미터 변경 등이다.

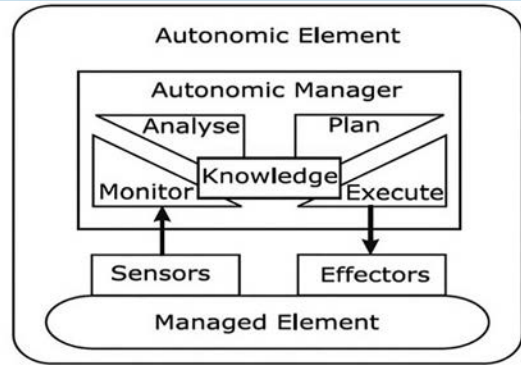


그림 3. IBM의 MAPE-K 자율 루프 참조모델

3.1 자율 관리기(Autonomic Manager)

자율 관리기는 센서로부터 수집한 정보를 내부에 저장하고 있는 지식(Knowledge)과 인간 관리자가 입력한 상위레벨의 정책 목표에 따라 분석하고 그 상태를 판단하며 필요에 따라 실행기에게 관리대상 요소에 대한 변경 명령을 내리는 역할을 수행한다. 관리기가 저장하고 있는 내부지식은 관리대상 요소에 대한 구조적 모델과 상황에 따른 변경이력 등이다.

인간 관리자가 입력해주는 목표는 ECA (Event-Condition-Action) 정책이나 효용함수(Utility Function) 정책 등으로 표현된다[12]. ECA 정책은 어떤 이벤트가 발생했을 때 주어진 조건을 만족하는 범위내에서 액션을 실행하도록 하는 것이다. 예를 들어 “웹 서버의 응답시간이 2초를 초과할 경우, 가용한 서버 자원이 남아 있으면, 가상머신(VM)의 수를 증가시켜라”하는 형태를 취한다. ECA 정책은 직관적으로 사용하기 쉬우나 다수의 정책이 실행될 때 정책 간의 충돌을 검출하기 어렵다는 문제를 가지고 있다.

효용함수는 다수의 파라미터를 입력으로 하고 각 상태가 바람직한 정도를 출력으로 한다. 앞의 예를 살펴 보면, 효용함수는 웹과 애플리케이션 서버의 응답 시간을 입력 파라미터로 하고 웹과 애플리케이션 서버들의 응답 시간의 여러 조합에 대한 효용성들을 출력한다. 이 방법을 사용하면 가용한 자원들이 충분하지 않은 상태에서도 자원을 웹과 애플리케이션 서버들에 대해 가장 바람직하게 분배하는 값을 찾을 수 있다. 하지만 효용함수는 그 함수에 의한 의사결정에 영향을 주는 모든 것들이 정량화되어야 하므로 문제 정의가 어렵다는 단점이 있다.

대규모 네트워크에서는 다수의 자율 관리기들이 분산되고 상호연동한다. 또한 전체를 최적화하기 위해 복잡한 최적해를 계산하게 되면 급격한 환경변화에는 대응이 불가능해진다. 따라서 빠르게 차선책(sub-optimal solution)을 계산하는 방법이 필요하다. 이를 위해 자율관리기에 창발적 설계(engineered emergence)를 추가하는 연구도 이루어지고 있다. 창발적 설계

라는 용어의 뜻은 원하는 결과를 전체적인 관점에서 달성하게 하는 것으로 자연계의 현상을 모방한 것이다[13]. 이것은 하위 레벨에 있는 컴포넌트들의 개별 동작에 대한 정확한 정보를 다루지 않고 상위레벨의 목표 달성에 초점을 맞추는 것이다. 이 방법에서는 단순한 규칙들이 국소적으로 가용한 정보를 사용해 동작한다. 이것은 엄격한 프로토콜과 순서 및 동기화에 초점을 두고 있는 전통적인 분산 애플리케이션의 설계방법과는 다르다. 하지만 자연계의 시스템들은 근본적으로 비결정론적이며 전체적으로 안정적이고 강건한 대규모 시스템을 구성하는 경우가 많다. 이러한 시스템들 중에서 자율 네트워킹 연구에 많이 인용되는 것은 곤충의 군집모델과 인체의 체온조절이나 면역체계, 평판모델, 게임이론과 같은 것들이다.

3.2 MAPE-K 루프의 구현 사례

IBM은 2004년에 MAKE-K 참조모델을 이용해 ACT (Autonomic Computing Toolkit)를 개발하였고 현재 온라인으로 다운로드 및 설치 사용이 가능하다[14]. ACT는 완전한 자율관리기는 아니며 이를 구현하기 위한 기반 플랫폼과 샘플 프로그램을 제공한다. ACT는 Java로 개발되었고 다른 S/W 애플리케이션과는 XML 메시지로 통신을 하며 센싱 기능으로는 관리대상 애플리케이션의 로그를 분석하는 방법을 사용한다.

IBM이 MAPE-K를 이용해 개발한 다른 툴로는 ABLE(Agent Learning and Building Environment)이 있다[15]. 이것은 기계학습과 추론 기능을 사용한 지능 에이전트 구축 툴킷이다. 이것을 이용해 구축된 자율관리기는 하나의 에이전트나 에이전트의 집합으로 구현되고 서로 다른 자율 기능들이 각각의 에이전트에 분리 설치되거나 하나로 일체화되어 설치된다. 이 툴킷은 Java로 구현되었으며 웹 서버나 데이터베이스와 같은 S/W 애플리케이션의 관리에 사용될 수 있다.

Ⅲ. 자율 네트워킹 연구동향

본 장에서는 MAPE-K 모델을 기반으로 하는 자율 네트워킹 연구동향을 유럽과 미국에서 수행된 대표적인 프로젝트들을 중심으로 살펴본다.

1. Autol 프로젝트

Autol(Autonomic Internet)[16]는 EU FP7 프로젝트의 하나로서 2008년~2010년까지 수행되었으며, 광역에 걸치는 가상 네트워크 자원과 서비스를 생성하고 이를 자율관리하는 기술

개발을 목표로 수행되었다.

Autol은 <그림 4>와 같이 IMO(Information Management Overlay) 컨트롤러와 노드로 구성된다. IMO 컨트롤러는 효용 함수를 사용하는 정책기반의 집중관리 시스템으로서 관리 애플리케이션의 요구사항에 따라 최적의 오버레이 네트워크를 설정하고 그 네트워크를 MAPE-K와 최적화 알고리즘을 결합하여 자율적으로 관리하는 기능을 수행한다. IMO 노드는 네트워크에 동적으로 배치되는 가상의 노드로서 가상자원의 정보를 수집, 처리하여 컨트롤러와 관리 애플리케이션에게 전달하고 컨트롤러로부터 명령을 받아 실행하는 기능을 한다.

IMO 노드는 IAP(Information Aggregator Point) 및 ICP(Information Collection Point)로 구성된다. ICP는 가상화된 네트워크 자원의 정보를 수집하여 가장 인접한 IAP에게 보내고 IAP가 데이터를 처리하여 관리 애플리케이션에게 전달한다.

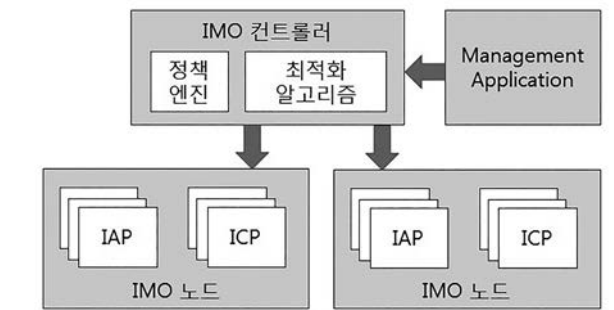


그림 4. Autol 구조

AutoI 프레임워크는 프랑스의 10개 지역을 연결하는 Grid5000 테스트베드에서 시험되었다. 시험은 분산 배치된 리눅스 서버들 위에 110개의 가상 라우터를 구성하여 가상 네트워크를 생성하고, 라우터를 다른 서버로 이동시키면서 그 라우터에 연결된 고객의 서비스도 함께 이전되도록 하는 시험을 성공적으로 수행하였다[16].

AutoI의 장점은 네트워크 가상화를 통해 이질적인 네트워크 자원을 유연하게 관리할 수 있다는 것이다. 하지만 집중화된 IMO 컨트롤러를 사용하므로 확장성 이슈가 남아 있고 또한 전체적인 보안 이슈가 미해결 과제로 남아 있다.

2. ANA 프로젝트

ANA(Autonomic Network Architecture)[17]는 EU FP6 프로젝트로 2006년~2010년까지 5년간 수행되었다. ANA는 현재의 IP 네트워크의 근본적인 문제를 해결하기 위한 것이다. 즉, IP 기반 네트워크에서는 프로토콜의 변경/수정이 불가능하며 모든 디바이스는 IP 주소를 필요로 하고 반드시 표준화된 라우팅 프로토콜이 설치되어야 하며 모든 구성은 사람이 직접 수

행해야 한다는 문제를 가지고 있다. ANA에서는 이러한 문제를 해결하는 방법으로서 네트워킹에 필요한 기능(프로토콜 및 감시기능 등)을 필요에 따라 동적으로 재구성할 수 있는 프레임워크를 개발하였다.

ANA에서는 네트워킹을 관리하기 쉬운 작은 단위인 구획(compartment)으로 분리하여 운용하며 특정 주소와 명명체계(naming)는 한 구획 내만에서 유효하도록 하였다. 이는 무선 센서 네트워크에서는 센서가 주소를 사용하지 않는 경우도 있기 때문이다. 주소와 명명체계는 미래에 정의되는 새로운 체계를 사용할 수도 있다. ANA에서 한 개의 노드 구획은 다음과 같은 것들을 수용한다.

- 기능블록(Functional Block): 기능 블록은 데이터를 처리하는 코드와 상태정보로 구성되며 프로토콜 및 알고리즘이 기능블록에 해당한다. 기능블록은 IDP를 통해 액세스 가능하다.
- IDP(Information Dispatch Point): 기능 블록과 IC에 입력 포트를 규정해준다.
- IC(Inforamtion Channel): 기능블록 간의 데이터 전달경로를 지정한다.

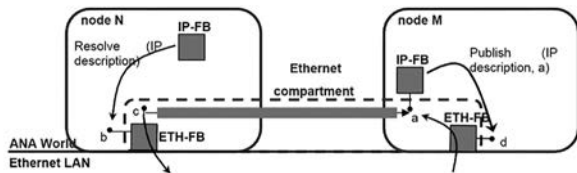


그림 5. ANA에서의 통신기능 구성의 예

〈그림 5〉는 ANA에서 노드 N과 M간에 기능블록과 IDP 및 IC를 이용하여 이더넷을 통해 IP 통신을 구현하는 예를 보인 것이다. 여기서 사각형은 IP와 Ethernet 기능을 처리하는 기능블록을 나타내고 흑색점은 데이터가 입력되는 IDP, 회색선과 화살표는 IC를 나타내며 실선과 점선은 구획을 나타낸다. 물론 이러한 기능 구성은 각 API를 이용한 프로그램으로 실행된다.

ANA에서의 자율관리는 측정 기능블록(MFB: Measuring FB)들과 피드백 루프에 의해 구현된다[17]. 측정 기능블록은 현재 사용되고 있는 다양한 측정 툴에 해당된다. 이러한 측정 MFB들을 통신 기능을 제공하는 다른 기능블록들과 결합하여 노드 설계에 사용하면 다양한 측정과 제어가 가능해진다.

ANA의 프레임워크는 리눅스로 개발되었으며 심비안, 아이폰, 맥 등의 단말에 적용하여 시험하였고 애드혹 네트워크에 대해서도 그 기능성과 확장성, 신뢰도를 검증하였다.

3. Cognitive 네트워크 구조

Cognitive 네트워크는 Cognitive Radio의 개념을 확장한 것

으로서 현재 네트워크의 상태를 인지하고 계획, 학습을 하면서 정해진 목표(end-to-end goal)에 따라 실행할 수 있는 네트워크이다[18]. 즉, Cognitive 네트워크는 자기 최적화와 자기 구성 및 프로토콜 계층 간 정보교환 설계를 강조한 자율 네트워킹의 특별한 변형이라고 할 수 있다. 〈그림 6〉처럼 Cognitive 네트워크는 인지정보를 상호 교환하는 Cognitive 노드의 집합으로 구성되며 각 노드는 3개의 기능 요소로 구성된다.

- Cognitive 평면: 데이터 분석과 의사결정 프로세스를 담당하며 네트워크를 최적의 운용상태로 유도한다. 또한 프로토콜 스택으로부터 감시정보를 수집하고 이를 바탕으로 구성 명령을 발행하여 프로토콜을 제어한다.
- 계층간 조절 및 신호 평면(Cross-Layer Coordination and Signaling Plane): 이 평면은 프로토콜 스택의 계층들과 Cognitive 평면 간에서 계층 별 정보 감시와 구성을 위한 중재 기능을 수행한다.
- NCE(Network Cognitive Engine): 이것은 네트워크 전체에 걸친 관리를 위해 여러 노드들에 있는 Cognitive 평면들과 연동한다. 이 프로세스는 각 노드에서 인지한 정보를 수집하고 분석하여 전체적인 지식을 구축하고 목표를 설정하며 이 결과를 각 노드에게 알려준다.

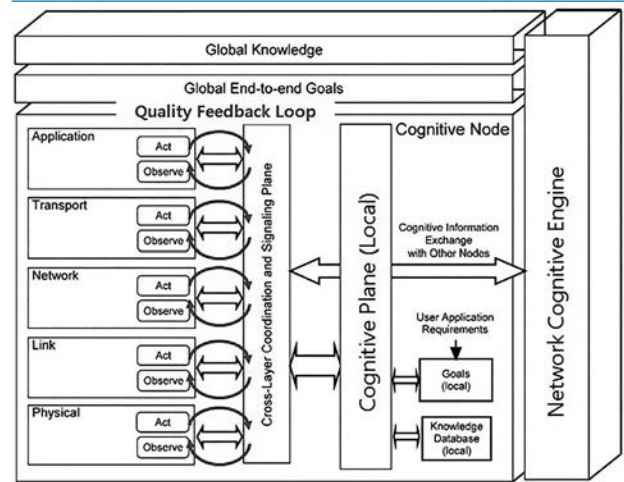


그림 6. Cognitive 네트워크 프레임워크

기능검증 용도로 CogNet은 NS-2 시뮬레이터에 구현되어 있다. CogNet은 TCP 윈도우 사이즈를 동적으로 재구성하는 시험에 사용되었고, 기존의 고정된 TCP 윈도우 사이즈 조절 메커니즘에 비해 우수한 전송성능을 보인 것으로 보고되어 있다[19].

4. SOCRATES 프로젝트

SOCRATES(Self-Optimization and self-ConfigurATIOn

in wireLess networkS)[20]는 EU FP7 프로젝트로 LTE 네트워크에서 SON(Self-Organization Nwtwork)을 실현하기 위해 2008년부터 2010년까지 3년간 수행되었다.

LTE 네트워크에서 기지국(eNodeB)들은 상호 X2 인터페이스로 연결되어 이웃 노드간에서 핸드오버 정보를 교환하므로 SOCRATES는 SON 구현을 위해 이 X2 인터페이스를 활용하였다.

구현된 SON 기능은 기지국이나 네트워크 관리 시스템에 설치되어 <그림 7>과 같은 제어 루프에 따라 수행된다. 여기서 정책 기능(Policy Function)은 관리자가 지정한 목표에 따라 SON 기능이 어떻게 대응할지를 결정해주며 정렬기능(Alignment Function)은 SON 기능을 실행하고 충돌이 검출되면 이를 해결하여 처리한다. 자기인지 기능(Autognostics Function)은 SON 기능과 감시 기능(Guard Function)이 보내는 명령에 따라 네트워크에서 적절한 성능, 고장 및 구성 데이터를 수집하여 SON 기능에 전달한다.

개발된 SOCRATES의 기능들은 이동단말의 핸드오버 최적화, 인접 셀 간의 커버리지 조정 및 부하분산 최적화 등의 시뮬레이션을 통해 검증하였으며 현재는 3GPP 표준에 반영하기 위한 노력을 하고 있다.

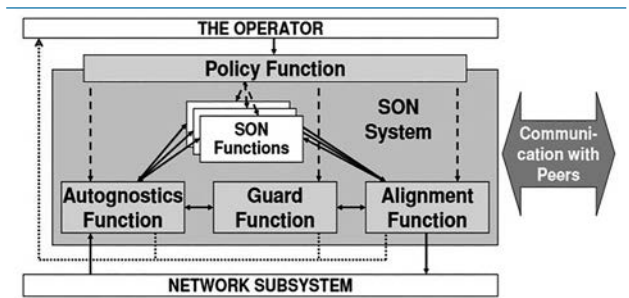


그림 7. SON 기능 프레임워크

5. BIONETS 프로젝트

BIONETS(BIO-inspired NExt generation networkS)[21]는 EU FP7 프로젝트로 2006년 ~2010년까지 약 4년 간 수행되었다.

BIONETS의 연구진들은 사회 전반에서 사용될 미래인터넷을 단순히 개별 노드들이 연결된 링크의 집합으로 보지 않고 대규모 시스템의 신경계로 보아야 한다는 생각으로 자연 및 사회현상을 수학적으로 모델링하여 자율 네트워킹에 필요한 핵심 알고리즘들을 개발하였다.

BIONETS는 <표 1>과 같이 통신 시스템을 3개의 레벨로 구분하고 다양한 분야의 원리와 이론들을 자율 네트워킹 기술 개발

에 활용하고 있다. 또한, 그 연구결과를 무선 메시 네트워크의 자율관리에 적용하였고, TCP의 전송 효율 문제를 개선하기 위해 다양한 TCP 프로토콜들을 상황에 따라 선택적으로 사용하게 하는 방법을 개발하여 시연하기도 했다.

표 1. BIONETS의 연구분야

구분	패러다임
사용자	사회과학, 미시경제학, 평판모델, 게임이론, 공유의 경제, 네트워크 게임 등
서비스	진화 알고리즘, 인공 배아 발생, 오토마타 이론, 카테고리 이론, 분산협업, 보안 등
프로토콜	기계학습, 인공 화학, 전염병 확산이론, 진화 게임, 트래픽 엔지니어링, 스케일 프리 네트워크, 브랜칭 프로세스 등

BIONETS는 자연과 사회현상을 모델링하여 네트워킹에 활용하는 이론적 토대를 만든 것으로 높이 평가 받는다. 특히 생물의 행동을 이용한 알고리즘들은 국소 정보만을 사용하여 전체를 최적해에 근사하게 동작시킬 수 있으므로, 미래인터넷의 중요 연구과제인 확장성 문제를 근본적으로 해결할 수 있다는 장점이 있는 반면 전체에 걸친 수렴시간이 빠르지 못하다는 근본적인 문제를 남기고 있다.

하지만 BIONETS에서 다룬 연구주제는 더 많은 생물과 물리현상 및 사회현상을 관찰할 경우 기존의 연구방법과 완전히 다른 패러다임을 만들어 낼 가능성이 매우 큰 분야임은 틀림없다.

6. ETSI GANA

ETSI의 AFI(Autonomic network engineering for the self-managing Future Internet) 그룹은 앞에서 설명한 자율 네트워킹 관련 연구 결과들을 종합하여 2009년부터 GANA(Generic Autonomic Network Architecture)[22,23]의 표준화를 진행 중이다. GANA는 미래 네트워크의 자율관리를 위한 시나리오, 유즈케이스 및 요구사항을 문서(GS AFI 001)[22]로 작성했고 참조모델이 문서(GS AFI 002)[23]화 되어 있다.

또한, AFI는 GANA의 효율성을 입증하기 위해 우선적으로 참조모델을 오픈플로우 기반 SDN[9]에 적용하는 연구를 지원하고 있다. ONF에서 정의한 SDN에는 자율적인 제어와 관리가 고려되어 있지 않고 오픈플로우 컨트롤러 S/W의 모듈화도 되어 있지 않아, SDN에서 사용하는 집중제어 방식이 미래 인터넷의 관리를 더욱 복잡하게 만들 수 있다고 판단하고 있다. 이를 위해 GANA에서 정의한 자율 루프를 오픈플로우 컨트롤러와 연동시키는 API를 개발하거나 오픈플로우 컨트롤러 내부에 S/W 모듈로 탑재하도록 하는 연구를 추진하고 있다.

IV. 적용분야 및 연구 이슈

IV장에서 설명한 연구사례들처럼 미래인터넷의 자율 네트워킹 및 관리에 관한 많은 연구가 진행되었지만 SOCRATES의 연구결과를 제외하고는 당분간 통신사업자의 유무선 네트워크에는 크게 적용되지 않을 것이다. 유무선 인터넷은 대규모 통신사업자인 경우에도 수십만개 수준의 네트워크 노드로 구성되어 있다. 인터넷이 잘 구축된 대부분의 국가들에서는 트래픽이 증가해도 노드의 수를 늘리지 않고 대용량 노드로 대체하고 있기 때문에 통신사업자가 관리하는 노드 수는 더 이상 크게 증가하지는 않을 것이다. 또한 집중화된 운용관리 시스템들의 개발에 수천억~수조원을 투자한 통신사업자들은 이들을 쉽게 포기하지도 않을 것이다. 결국 미래인터넷의 자율 네트워킹 기술은 무선 센서 네트워크와 IoT, SDN 등 인간에 의한 집중관리가 어렵거나 프로그래밍이 용이한 네트워크에 우선 적용될 것으로 전망된다.

무선 환경에서 배터리로 동작하는 대부분의 센서 노드들은 노드 비용(가격과 전력소비량)의 대부분을 무선통신 기기가 차지하고 센서 자체와 CPU 및 메모리의 비용은 상대적으로 작다. 따라서 하나의 노드에 여러 종류의 센서들을 설치하는 것이 경제적이며 강건성을 높이는 방법이다. 하지만 주어진 과제를 수행하기 위해 어느 센서를 언제 구동시킬 것인지는 그 순간의 상황과 사용자의 의도에 따라 달라진다. 여기에서 센서 노드의 자율적 판단 능력이 필요하다. 또한 노드가 저가의 핸드폰 정도의 처리능력을 갖는다면 MAPE-K와 같은 자율관리 기능을 탑재할 수도 있고, 소형의 배터리로 동작하는 노드라면 메모리 크기가 작고 처리 성능도 낮으므로 경량화된 자율 네트워킹 기능을 구현하여 탑재하여야 한다.

무선 애드혹 네트워크에서는 한 노드의 데이터를 다른 노드로 전달할 때에 소요되는 시간과 전력을 최소화하는 연구가 많이 진행되었다. 이 연구들 중, 멀티홉 연결방식은 전체 네트워크 상태를 몰라도 인접한 노드를 이용해 경로 연결을 할 수 있지만, 모든 노드들은 패킷 중계를 위해 항상 대기(listening) 상태에 있어야 하므로 전력 소모 문제를 야기한다. 직접연결 방식은 노드가 무선기지역에 도달할 수 있는 거리 내에 있어야 한다는 제약이 있다. 따라서 노드들의 클러스터가 대표 노드를 선출하고 대표 노드들이 멀티홉으로 전달하도록 하는 하이브리드 방식의 연구도 수행되고 있다. 이 방식에 자율 네트워킹을 추가하면 노드들의 가용도에 따라 대표 노드를 선출하거나 배터리 소모 비용을 네트워크 노드들 전체에 고르게 분배하면서 대표 노드를 선출하는 것도 가능하다(예를 들어 무선 기지역에 가깝게 위치한 노드들이 중계 역할을 수행할 확률이 크므로 배터리 소

모량이 더 커진다)[10].

상태와 정책 간에서 요동이 일어나는 상태의 급변은 자율 네트워킹에서 항상 문제가 되는 이슈이다. 이것은 감도를 조절하여 시스템을 제어하는 다른 기술(예: 제어 이론)에서 해결책을 찾을 수도 있다. 그러나 제어이론 등은 지속적으로 변하는 환경 하에서 입력 데이터가 명확히 정의되지 않으면 시스템에 적용하기 어렵다는 문제를 가지고 있다.

하나의 네트워크에서 발생하는 모든 이벤트를 효과적으로 처리할 수 자율관리 기능을 개발하기는 매우 어렵고 특히 소형화된 노드에 복잡한 기능을 탑재시키는 것도 비현실적이다. 이러한 경우를 대비해 자율관리기 내에 다양한 자율관리 기능을 구현해 놓은 후 동적으로 업로드(또는 다운로드)시키는 방법으로 선택 적용할 수 있는 구조 연구도 필요하다.

효과적인 자율 네트워킹을 실현하기 위해서는 자율관리기가 주변 환경변화를 인지하고 판단하여, 선택 가능한 기능들 중에서 가장 근사한 해를 제공하는 기능을 적용토록하고 그 결과를 측정하여 변화된 상태에 적합한 다른 기능을 실행하도록 하거나 파라미터를 변경하는 등으로 몇 차례의 반복 실행을 통해 문제를 해결하도록 하는 연구도 필요하다.

현재의 인터넷보다 수십배 이상 큰 대규모 시스템에서는 전체를 항상 최적상태로 유지하는 것은 매우 어려운 문제이다. 변동성이 큰 어떤 환경에서 대규모 시스템에 대한 최적해를 구하고 적용하는 것은 일정한 시간이 소요되므로 변동이 빈번하면 오히려 시스템을 불안정하게 만드는 요인이된다. 따라서, 이러한 규모의 네트워크에서는 서비스 수준을 만족하는 범위내에서 기본적인 강건성을 유지하는 차선의(sub-optimal) 해를 실시간으로 구하도록 하는 연구도 필요하다. 또한, 제어를 한 뒤에 안정상태로 수렴하는 시간을 중요한 관리측도로 삼아야 하나 현재의 자율 네트워킹 분야에서는 많은 연구가 이루어지고 있지 않다.

V. 결론

자율 네트워킹 패러다임은 대규모의 이질적인 요소로 구성되고 이동성을 기본으로 하는 미래 인터넷을 효과적으로 관리하는 데에 필수적인 원칙으로 떠오르고 있다. 이를 실현하기 위해 선진국에서는 2000년대 초반부터 국가주도로 광범위한 분야에서 연구를 진행하고 있고 그 가능성과 가치가 차차 입증되고 있다. 하지만 국내에서는 아직까지 체계적인 연구가 이루어진 바 없고 이제서야 새로운 연구주제로 관심을 끌기 시작한 수준이다.

자율 네트워킹은 실용화의 관점에서는 아직 미성숙한 주제이

다. 하지만 생물, 물리, 경제 및 사회과학 등 더 많은 분야의 기술들이 네트워크 기술과 융합되어 그 효과가 검증될 때에 현실적이고 실용성 있는 기술로 떠오를 것이며, 가까운 미래에 이 기술은 일반적인 네트워크 기술에 통합되어 통신시스템 설계 프로세스에 자연스럽게 포함될 것이다. 20여년 전 별개의 연구 분야로 다루어졌던 분산 처리 기술이 이제는 컴퓨팅 기술의 일부가 된 것처럼 가까운 미래에는 자율 네트워킹과 관리 기술이 네트워크 기술의 당연한 일부가 될 것이다.

미래인터넷의 자율 네트워킹과 관리 기술의 개발을 위해서는 IV 장에서 설명한 외국의 사례들처럼 다양한 분야의 연구가 필요하다. 이것은 단 몇 개의 팀들로 구성된 국내 연구진만으로 단기에 해결할 수 있는 것은 결코 아니다. 미국과 유럽처럼 체계적인 국제규모의 계획을 수립하고 산학연이 공동으로 참여하여 연구개발을 추진하는 방안을 검토해야 할 것이다.

참고 문헌

- [1] ESTI, "Machine to Machine Communications," Available on "<http://www.etsi.org/technologies-clusters/technologies/m2m>".
- [2] Jayavardhana Gubbi et al., "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems* 29, 2013
- [3] Mehran Abolhasan et al., "A review of routing protocols for mobile ad hoc networks", <http://www.elsevier.com/locate/adhoc>
- [4] Tim Winter et al., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Internet Engineering Task Force (IETF), 2010.
- [5] N. Kushalnagar et al., "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," RFC 4919, IETF, 2007.
- [6] K. Scott et al., "Bundle Protocol Specification," IETF RFC 5050, November 2007.
- [7] M. Ramadas et al., "Licklider Transmission Protocol - Specification," IETF RFC 5326, September 2008.
- [8] V. Cerf et al., "Delay-Tolerant Networking Architecture," IETF RFC 4838, April 2007.
- [9] 유재형, "SDN/OpenFlow 기술 동향 및 전망", KNOM Review Vol. 15, No. 2, Dec. 2012
- [10] Markus C. Huebscher et al., "A survey on autonomic computing—degrees, models and applications", *ACM Computing Surveys (CSUR) Volume 40 Issue 3, August 2008 Article No. 7*
- [11] Zeinab Movahedi et al., "A survey on Autonomic Network Architectures and Evaluation Criteria", *IEEE Communications Surveys and Tutorials, Vol. 14, No.2, 2012*
- [12] Kephart et al, "An artificial intelligence perspective on autonomic computing policies", *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004*
- [13] Anthony et al., "Emergent graph colouring", *First Annual International Workshop at the 3rd International Conference on Autonomic Computing (ICAC), 2006*
- [14] IBM Autonomic Computing Toolkit, <http://www.ibm.com/developerworks/autonomic/r3/probdet1.html>
- [15] IBM Agent Learning and Building Environment, <https://publib.boulder.ibm.com/infocenter/series/v5r3/index.jsp?topic=%2Frzahx%2Frzahxabletoolkit.htm>
- [16] "Autonomic Internet (AutoI) FP7 project." [Online]. Available: <http://ist-autoi.eu/>
- [17] ANA, <http://www.ana-project.org/web/>
- [18] Dzmitry Kliazovich et al., "CogProt: A Framework for Cognitive Configuration and Optimization of Communication Protocols", *MobiLight 2010, LNICST 45, 2010*
- [19] D. Kliazovich et al., "Cogprot: A framework for cognitive configuration and optimization of communication protocols," *2nd International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT), 2010.*
- [20] "SOCRATES project", <http://www.fp7-socrates.org/>
- [21] "BIONETS project", <http://www.bionets.eu/>
- [22] "ETSI GS AFI 001, V1.1.1", June 2011, http://www.etsi.org/deliver/etsi_gs/AFI/001_099/001/01.01.01_60/gs_afi001v010101p.pdf
- [23] "ETSI GS AFI 002 V1.1.1", April 2013, http://www.etsi.org/deliver/etsi_gs/

AFI/001_099/002/01.01.01_60/gs_afi002v010101p.pdf

- [24] Howard Shrobe et al., "Self-Adaptive Systems for Information Survivability: PMOP and AWD RAT", MIT-CSAIL-TR-2007-023, April 10, 2007
- [25] Roy Sterritt et al., "Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions", Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based System, 2005)

약 력



유재형

1983년 연세대학교 전자공학과 학사
 1985년 연세대학교 전자공학과 석사
 1999년 연세대학교 컴퓨터공학과 박사
 1986년~1994년 KT 연구개발본부 전임연구원
 1995년~2002년 KT 운용시스템연구소 망관리연구실장
 2003년~2012년 KT 네트워크연구소 인터넷연구담당 상무보
 2013년~현재 포항공과대학교 컴퓨터공학과 연구부교수
 관심분야: Autonomic Networking, SDN(OpenFlow/LISP), Network Management



홍원기

1983년 Univ. of Western Ontario, BSc in Computer Science
 1985년 Univ. of Western Ontario, MS in Computer Science
 1986년~1991년 Univ. of Waterloo, PhD in Computer Science
 1991년~1992년 Univ. of Waterloo, Post-Doc Fellow
 1992년~1995년 Univ. of Western Ontario, 연구교수
 1995년~현재 포항공과대학교, 컴퓨터공학과 교수
 2007년~2011년 포항공과대학교, 정보통신대학원장
 2007년~2010년 포항공과대학교, 정보통신연구소 연구소장
 2008년~2010년 포항공과대학교, 컴퓨터공학과 주임교수
 2008년~2012년 포항공과대학교, 정보전자융합공학부장
 2012년~2014년 KT 종합기술원, 원장
 관심분야: Network Management, Future Internet (SDN/SDI/LISP), ICT Convergence