

主題

## 유비쿼터스 네트워킹을 위한 컴퓨팅 미들웨어 기술의 설계

한국전자통신연구원 임베디드S/W기술센터 김 정 기, 신 창 민, 유 석 대, 박 승 민

차 례

1. 서론
2. 유비쿼터스 네트워킹을 위한 미들웨어 기술 동향
3. 유비쿼터스 네트워킹을 위한 새로운 미들웨어 설계
4. 결론

### 요 약

최근에 정보 산업과 이동 통신 기술이 발전함에 따라 퍼스널 컴퓨터를 넘어 컴퓨터의 개념이 매우 빠르게 확장되고 있다. 이동 정보 단말 형태의 PDA, HPC(hand-held PC), 휴대폰 등이 개발되었고, 셋탑박스(set-top box), 지능형 디지털 TV, 인터넷 냉장고 등의 정보가전은 임베디드 운영체제(embedded OS)를 내장하여 컴퓨터로 발전하고 있으며, 제어 및 계측 기기들도 마이크로 프로세서를 내장하여 임베디드 시스템으로 발전하고 있다. 이렇게 새롭게 확장된 컴퓨터 시스템들은 인간 생활의 편리성과 다양성을 위해 상호 연결되어 하나의 네트워크를 형성하고 있으며, 상호간에 정보를 공유하고 협력하여 언제 어디서나 통신 및 컴퓨팅이 가능한 유비쿼터스 네트워킹(Ubiquitous Networking)으로 발전하고 있다. 본 논문에서는 이러한 유비쿼터스 네트워킹

에서 필요한 컴퓨팅 미들웨어(Middleware) 기술의 최근 동향을 살펴보고 본 연구에서 제안하는 새로운 미들웨어 기술을 설계한다. 이러한 미들웨어 기술은 편재(遍在)되어 있는 컴퓨팅 장치를 상호 이용하기 위해 컴퓨터의 환경을 탐지하여 저장하고 재이용하는 상황 인식 기술, 편재된 장치들이 네트워크에 연결되면서 정보를 전달하는 Ad-hoc 네트워킹과 동적 라우팅 기술, 그리고 정보 전달을 위해 제어 코드와 데이터를 패킷(packet) 형태로 전달하는 협력 메시지(Cooperative Message) 기술 등이 포함된다.

### 1. 서론

일반 가정의 가전제품 속에 마이크로 프로세서 등이 내장되면서 가전제품이 더욱 지능화되었고, 이를 이용하여 인터넷 연결과 정보처리 등의 컴퓨팅을 위한 정보가전의 개념으로 발전되었

다. 인터넷 냉장고, 셋탑박스(settop box), 인터넷 TV 등의 정보가전뿐만 아니라, 무선 통신 기술과 정보 산업의 발달로 PDA, 휴대폰, 전자책 등을 이용하여 어디서나 통신과 컴퓨팅이 가능하게 되었다. 이러한 정보가전 및 이동 통신 장치들은 통신과 컴퓨팅을 수행하기 위해 내부에 운영체제 등을 내장하여 임베디드 시스템으로 형성되고 있으며, 기존의 제어 및 계측 기기들에 마이크로 프로세서와 운영체제 등이 내장되면서 임베디드 시스템으로 발전되고 있다. 이렇게 휴대용 기기나 정보 가전제품 등이 하나의 네트워크에 연결되어 언제 어디서나 통신 및 컴퓨팅이 가능하고 컴퓨팅 시스템이 협력함으로써 새로운 서비스가 가능한 개념을 유비쿼터스 네트워크(Ubiquitous Networking)이라 부르고 있다.

최근에는 우리 주변에 있는 모든 물체에 컴퓨터를 내장하여 서로 네트워크로 연결되고 상호간에 협조와 타협을 해 가면서 인간의 삶에 보이지 않게 컴퓨팅을 제공한다는 유비쿼터스 컴퓨팅(Ubiquitous Computing)의 개념이 대두되고 있다[1, 2]. 유비쿼터스 컴퓨팅은 전자제품뿐만 아니라 자동차, 안경, 그림, 벽, 심지어 병이나 약, 쓰레기에도 마이크로 프로세서를 내장하여 통신 및 협업을 통해 인간에게 보이지 않게 컴퓨팅을 제공하는 개념으로 사용된다. 근래에는 유비쿼터스 네트워킹이라는 개념에서 유비쿼터스 컴퓨팅으로 발전되면서 개념이 혼용되어 사용되는 경향이 있지만, 본 논문에서는 유비쿼터스 네트워킹 개념에서 제공되는 미들웨어 기술에 대해 언급하고자 한다.

유비쿼터스 네트워킹을 위해 컴퓨팅 시스템에서 이를 운영할 임베디드 운영체제는 필수적인 요소이며, 그 상위에 올라가는 모듈의 상당 부분이 네트워크 지원을 위한 미들웨어 부분이 된다. 본 논문에서는 이러한 네트워크 지원을 위한 미들웨어로 상황인식 처리 기술과 센서 네트워크를

위한 동적 라우팅 및 Ad-hoc 네트워킹 기술, 그리고 마지막으로 코드 이동성을 지원하는 스마트 메시지 기술에 대해 최근 동향을 언급하고 본 연구에서 개발하고자 하는 미들웨어 시스템의 구조와 특징을 설계하고자 한다.

유비쿼터스 네트워킹을 위해 우선적으로 필요한 것은 사용자가 처한 환경을 컴퓨터가 인식하는 것으로부터 시작된다. 그러므로써 사용자에게 대한 통신이나 응용을 지원할 수 있다. 이러한 사용자가 처한 환경에서 사용자의 현재 위치, 행동 및 작업, 감정 상태 등을 객체(Object)라고 나타낼 수 있으며, 사용자나 사용자의 객체에 대한 정보 값과 그 정보들의 변화를 상황(Context)이라고 표현할 수 있다. 이렇게 사용자의 환경으로부터 상황정보를 얻어내는 과정을 상황인식(Context-Awareness)이라 한다. 객체들이 발견되면 객체에서 얻은 상황정보는 현 시스템에서 사용될 수도 있고 네트워크를 통해서 일반적으로 서버에 등록되거나 다른 곳에서 이용될 수 있다. 서버에 저장된 상황정보는 필요에 따라 유비쿼터스 네트워크에 연결된 다른 단말에서 이용되거나 실행된다.

이러한 상황인식이 이용될 수 있는 응용은 이동 단말을 가지고 다니면서 컴퓨팅을 수행할 때 일어날 수 있다. 단말에 대한 상황을 인식하고 수행을 위해 상황정보가 서버나 다른 단말로 이동할 때 통신 방식은 Ad-hoc 네트워크 형식을 취하며, 정보 이동을 위해 동적 라우팅(Dynamic Routing) 개념이 필요하다. 또한 상황정보들을 라우팅을 통해 전달하기 위해 스마트 메시지(Smart Message)와 같은 코드 및 데이터 이동 방식이 필요하다.

## 2. 유비쿼터스 네트워킹을 위한 미들웨어 기술 동향

본 장에서는 앞에서 언급한 유비쿼터스 네트워킹을 위해 최근에 연구되고 있는 컴퓨팅 미들웨어 기술을 살펴본다.

### 2.1 상황인식 기술

일반적인 컴퓨터 시스템 실행은 사용자에게 의한 분명한 입력 값에 의해, 컴퓨터의 내부 처리를 끝내고 입력 값에 상응하는 출력 값을 만들어 낸다. 일반적으로 입력 값이 같으면, 같은 결과 값을 만들어 낸다. 그러나 상황인식의 응용에서는 사용자 입력에 대해 사용자의 상태, 물리적인 환경, 컴퓨팅 자원의 상태, 기존 정보를 통한 분석 등의 상황정보를 통해 사용자의 상황에 맞는 결과 값을 제공한다[3]. 이러한 상황에 대한 정의는 응용마다 다양하며 사용자 ID, 위치, 시간, 온도, 심리적 요소 등이 상황 정보에 해당한다.

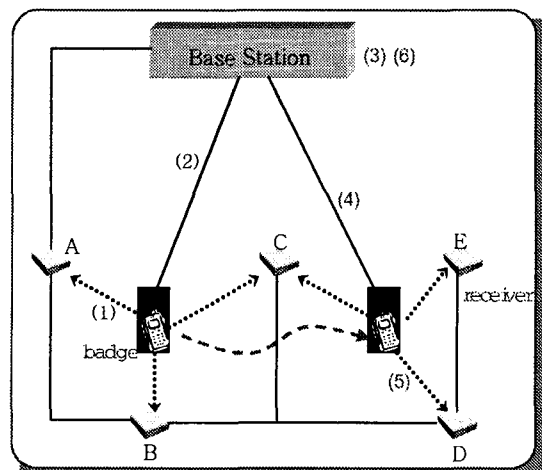
이렇게 함으로써 응용이 사용되는 상황에 따른 변화에 적응적(adaptive)으로 서비스를 제공할 수 있으며, 사용자 환경에 대한 탐지(detection)와 센싱(sensing)을 통해 컴퓨터 자원 및 서비스를 최대한 활용할 수 있다. 이러한 상황인식의 최초 시도는 Olivetti 사의 액티브 배지(Active Badge)라고 볼 수 있다[4]. 이러한 상황인식 처리를 위해 필수적인 요소는 상황 탐지(context detection) 기능, 서버에 저장하기 위한 상황 표현 및 데이터베이스 기술, 관련성 있는 상황을 선택하고 검색하는 기술, 상황에 기반하여 서비스를 자동으로 실행하는 기술 등이 있다.

관련 연구로 Yau와 Karim[5]는 상황인식 미들웨어의 구조와 상황을 정의하는 언어(Context Definition Language)를 제안하고 있다. 미들웨어는 하부에서 객체를 불러들이는 ORB(Object Request Broker), S/W와 H/W를 연결하는 인터페이스, AOC(Adaptive Object Container)로 구성되어 있으며, AOC는 상황을 인식하고, 객체를 동작하기 위해 함수를 부르고, 객체의 데이터를

저장하는 역할을 수행한다.

Couder와 Kermarrec[6]은 상황인식을 처리하는 일반적인 구조와 상황 객체를 표현하는 모델(Contextual Object Model)을 제안하고 있다. 구조는 클라이언트-서버(Client-Server) 형식으로 구성되어 있으며, 클라이언트는 시스템과 네트워크를 모니터링하여 상황을 탐지(Detection)하고 상위 계층으로 통지(Notification)해 주는 계층과, 상황정보를 관리하고 처리를 위해 선택하고 서버에 저장을 처리하는 적응적 계층(Adaptive Layer), 그리고 응용으로 구성되어 있다. 서버에서는 정보 서버 역할을 수행하여 상황 정보를 저장하고 관리하며 클라이언트 요청에 응답한다.

Harter 등[7]은 건물 내부에서 이동 단말의 위치를 검출하기 위해 (그림 1)과 같은 공간 정보를 이용한 시스템 구조를 제안하고 있다. 메인 컴퓨터(Base Station)는 수신기(receiver) A, B, C를 통해 이동 단말(또는 배지)의 메시지를 받고 위치를 파악하여 어떤 처리를 수행할 수 있다. 단말이 이동하면, 수신기 C, D, E를 통해 단말의 메시지를 받고 원래 단말이 이동하여 새로운 위



(그림 1) 건물 내에서 이동 단말의 위치 탐지

치로 옮긴 것을 파악할 수 있으며 새로운 처리를 할 수 있다. 이 때 단순히 단말의 위치만을 확인할 때는 단순히 공간 인덱싱 시스템(Spatial Indexing Proxy) 만을 이용하지만 상황 정보를 저장하고 불러와 처리 할 때는 서버의 데이터베이스까지 접근하여 처리하게 된다.

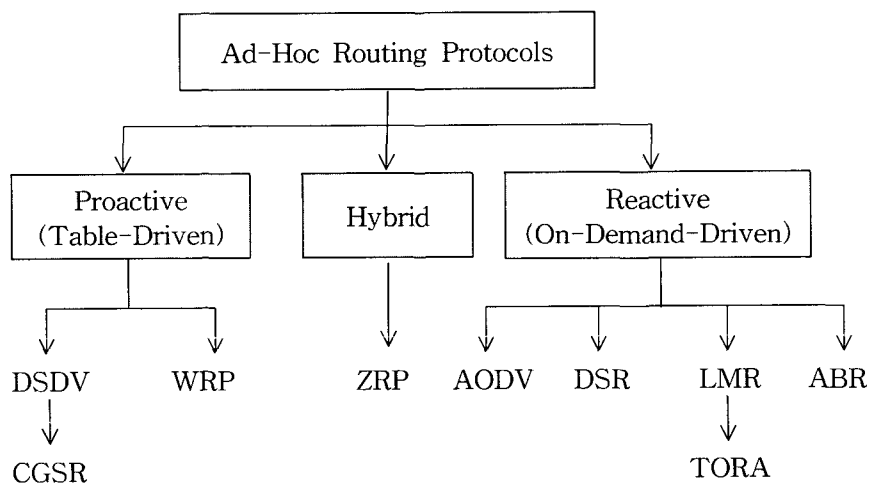
이렇게 단말이 이동하여 새로운 위치에 있는 단말에 데이터를 문제없이 전달해 주는 과정을 마이크로 모빌리티(Micro Mobility)라 한다. 모빌리티 개념은 IETF의 모바일 IP(Mobile Internet Protocol, MIP) 워킹그룹이 인터넷 상에서의 이동성을 제공한 것에 기반을 두고 있다[8]. 즉, 이동 단말에게 유일한 주소를 가지게 하고 이동 단말이 이동했을 경우에도 패킷이 효율적으로 전달될 수 있도록 하는 호스트 모빌리티를 제안하고 있다. 모바일 IP에서의 모빌리티는 에이전트 발견(agent discovery), 등록(registration), 터널링(tunneling)의 3단계로 수행된다. 그러나 이와 같은 MIP에서 지원하는 호스트 모빌리티는 너무 느리고 무거워 로컬 모빌리티에는 적합하지 않다.

그래서, IETF에서의 Seamoby (Seamless mobility) 워킹 그룹은 로컬 시그널에 의한 이동

성 운영을 통한 로컬 모빌리티를 제안하고 있다 [9]. Seamoby의 작업 영역은 상황 전달(Context Transfer), 마이크로 모빌리티, 페이징 등을 포함한다.

## 2.2 Ad-hoc 네트워크와 동적 라우팅 기술

유비쿼터스 네트워크에서 컴퓨팅 노드간 통신은 근거리 무선통신 기술을 주로 이용하는데, 각 노드들의 이동성 때문에 고정된 네트워크가 형성되지 않는다. 이런 노드들의 위치 변화에 의해 자율적으로 구성되는 임시적인 네트워크 형태를 이동 Ad-Hoc 네트워크라 한다. 즉, 이동 Ad-Hoc 네트워크는 다수의 노드가 필요한 인프라스트럭처(infrastructure)를 구성하는 것이 경제적으로 불리하거나 물리적으로 어려운 환경에서 인터넷과 같은 대형 통신망의 서비스뿐만 아니라 각 노드간 통신의 지원에 중점을 둔 네트워크이다. 현재 이동 Ad-Hoc 네트워크는 IETF (Internet Engineering Task Force) MANET (Mobile Ad-hoc NETwork) 워킹 그룹에서 라우팅 기법에 관한 표준화가 이루어지고 있다[10]. 이동 Ad-Hoc Network의 존재 형태는 Stand-alone 형태로 구성이 되어 기존의 인터넷에서 제공하는



(그림 2) Ad-Hoc Network에서 라우팅 프로토콜 분류

콘텐츠를 이용할 수 없는 형태로 존재할 수 있다. 그리고 게이트웨이(gateway) 노드를 이용하여 기간 망과 연결할 수 있다[11].

이동 Ad-Hoc Network에서의 라우팅 프로토콜은 (그림 2)에서와 같이 Proactive 방식과 Reactive 방식, 그리고 두 방식을 혼합한 Hybrid 방식으로 크게 나누어진다. Proactive 방식은 네트워크 위상(topology) 변경이 있을 때마다 각 노드들이 자신의 라우팅 정보를 갱신하도록 하여 항상 최신의 라우팅 정보를 유지한다. 따라서 네트워크 위상의 변경이 자주 있을 경우에는 제어 메시지의 오버헤드가 크다는 단점이 있다. Reactive 라우팅 방식은 트래픽이 발생하는 시점에서 루트를 탐색하므로 Proactive 방식이 가지는 제어메시지의 오버헤드를 줄일 수 있다. 그러나 초기 경로 탐색을 위한시간이 더 필요하다[12, 13].

이동 Ad-Hoc 네트워크의 특성은 각 이동 단말 기기들이 호스트의 역할 뿐만 아니라 라우터의 역할도 수행한다는 것이다. 또한 중앙 통제식이 아닌 동적인 네트워크 형태를 가지며 이동 노드들 간에 싱글홉(single-hop) 라우팅과 멀티홉(multi-hop) 라우팅이 지원되어야 한다. 그리고 이동노드는 제한된 용량의 배터리를 갖는 경우가 많으므로 절전을 위한 알고리즘이 필요하며, 무선 통신이므로 보안과 인증 문제가 존재한다.

이러한 이동 Ad-Hoc 네트워크의 특성은 센서 네트워크에도 적용이 된다. 그러나 센서 네트워크(Sensor Network)와 이동 Ad-Hoc 네트워크의 차이점은 다음과 같다. 우선 센서 네트워크에서의 센서 노드의 수는 이동 Ad-Hoc 네트워크에서의 노드 수보다 훨씬 많으며, 센서 노드들은 매우 밀접하게 배열이 되고 통신의 실패율이 높다. 또한 센서 네트워크의 위상(topology)은 매우 자주 바뀌며, 이동 Ad-Hoc 네트워크가 point-to-point 통신에 기반하는 데

반하여 센서 노드들은 방송(broadcasting) 형식의 통신을 주로 사용한다. 그리고 센서 노드들은 전력과 메모리, 컴퓨팅 능력의 제한을 가지고 있다. 따라서 위와 같은 차이점을 고려하여 센서 노드들에 적용되는 이동 Ad-hoc 라우팅 알고리즘이 개발되어야 한다.

센서 네트워크에 적절히 적용될 수 있는 새로운 네트워크 기술로써 액티브 네트워크(Active Network)가 있다[14]. 기존의 인터넷 개념에서는 네트워크에 대한 새로운 기술 및 서비스의 적용이 어렵고 사용자의 요구에 능동적으로 대응하기도 어려웠다. 따라서 사용자의 요구에 실시간으로 대응하고 새로운 기술과 서비스를 즉각적으로 적용할 수 있는 네트워크로써 액티브 네트워크가 등장하였다. 액티브 네트워크에서는 패킷에 실행 가능한 어떤 코드를 같이 전달하여 라우터에서 실행함으로써 네트워크 관리자의 작업 없이 다양한 기능을 수행하도록 한 것이다. 이와 같이 함으로써 네트워크는 새로운 응용을 쉽게 적용할 수 있는 유연성과 확장성을 가지게 된다.

액티브 네트워크의 응용 예로서는 Web caching, 지능형 라우팅(Intelligent routing), Active storage nets, Multimedia stream filters 등이 있다. 지능형 라우팅은 성능요소, 지역적인 라우팅 정책, 대역폭, 트래픽 통계량 등을 인터넷 라우팅 및 전송 프로토콜로 제공한다. 이렇게 함으로써 네트워크 장비의 기능을 실시간으로 변경하여 예측 불가능한 전송 트래픽의 변화에 능동적으로 대응이 가능하다. 이와 같은 액티브 네트워크의 지능형 라우팅 기능은 위상이 자주 바뀌는 센서 네트워크의 라우팅 엔진으로써 적합하다.

버클리 대학의 Smart Dust는 대표적인 센서 네트워크 프로젝트로써 작은 크기의 실리콘 모트(silicon mote)라는 입방체 안에 완전히 자율적인

센싱과 통신 플랫폼 능력을 갖춘 컴퓨팅 시스템이다[15]. 운영체제는 버클리 대학의 TinyOS를 사용하고 있다.

### 2.3 스마트 메시지 기술

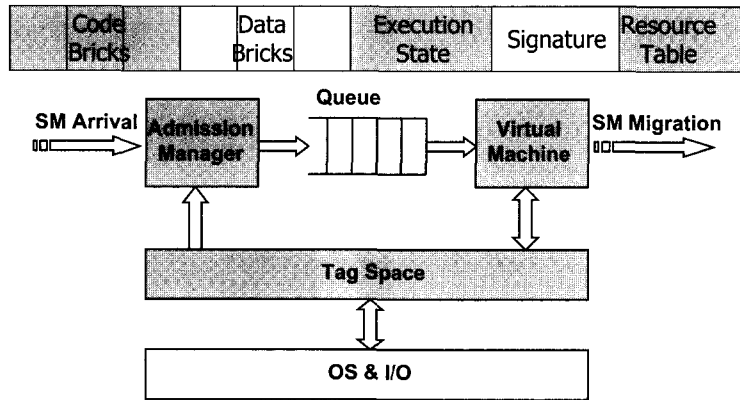
기존의 분산처리 환경은 컴퓨팅 노드들이 동일한 기능을 수행하고 안정된 네트워크에서 효율적으로 분산 처리하여 정확한 결과를 얻기 위한 구조였다. 그러나 유비쿼터스 네트워크에서는 일반적으로 대규모의 ad-hoc 네트워크와 이기종 노드들이 하나의 네트워크를 형성하여, 특정 응용을 실행하는 환경을 제공한다. 또한 각 노드들은 원격지에 설치되어 사용자의 관리를 벗어난 상태에서 작동되고 전원 공급, 사용 메모리, 프로세싱 능력, 네트워크 대역폭 등에 있어 매우 제한적 자원을 사용함으로써 매우 불안정한 네트워크를 구성하게 된다. 이런 형태의 구성을 네트워크 임베디드 시스템(Networked Embedded System, NES)이라고도 한다[16].

이런 환경에서 노드들의 협업을 위해 메시지를 전달하는 방식 중의 하나로 스마트 메시지(Smart Message)가 제안되었다[17]. 스마트 메시지는 센서 네트워크에서 데이터를 전파하고 모으는 간단한 응용으로부터 교통 상황에 적응하기 위해서 자동차들이 서로 협력하는 응용, 지능적 카메라를 이용하여 물체를 추적하는 등의 복잡한 협동 응용까지 포함하고 있다. 스마트 메시지는 액티브 네트워크 분야에서 제안된 액티브 메시지(Active Message)[18], 스마트 패킷(Smart Packet)[19], 이동 에이전트(Mobile Agent)[20]와 유사한 기능을 수행한다. 즉, 이동 가능한 데이터와 코드로 구성되어 네트워크 내의 각 노드를 단계적으로 또는 순차적으로 이동하면서 필요한 경우에는 방문한 노드에서 코드를 실행하고 데이터를 변환한다. 그러나, 스마트 메시지는 스스로 라우팅 경로를 결정하여

방문한 노드에서 제공하는 최소한의 기능을 활용하는 고유의 접근 특성을 갖는다. 이러한 고유 특성은 네트워크 위상 및 노드 가용성이 수시로 변화하는 대규모의 NES 노드들 사이에 상호협동 컴퓨팅 모델을 제공한다.

스마트 메시지는 (그림 3)과 같이 실행코드들(Code Bricks)과 데이터 섹션들(Data Brick)로 이루어진 이동 가능한 실행 유닛이고 경량의 실행 상태가 가능하다. 실행코드는 사용자 정의 응용 프로그램과 라우팅 코드가 포함되어 있고, 데이터 섹션에는 스마트 메시지가 이동하는 동안 실행 상태와 같은 이동 데이터가 포함된다. 실행상태(Execution State)는 부분적인 코드 수행 후에 이동이 가능하도록 하기 위해서 프로세스의 상태를 보관한다. 보안적인 고려를 위해서 서명(Signature)를 두어 공용 메모리에 대한 접근을 제어하고 있으며, 해당 메시지가 각 노드로 이동하여 사용할 자원들에 대한 정보를 자원 테이블(Resource Table)에 기록하고 있다.

노드를 구성하는 내부 처리 구조는 네 가지 모듈로 되어있다. 스마트 메시지와 별개인 운영체제와 I/O에 관한 부분을 제외하고 나머지 구성 요소들은 스마트 메시지의 전달과 실행, 인증을 위해서 필요하다. 수용관리기(Admission Manager)는 메모리, CPU 사이클, 대역폭 또는 전원 제한 등과 같은 제약으로 출입 허용에 대한 관리를 수행하여 스마트 메시지를 수신하고 저장하는 역할을 수행한다. 가상 머신(Virtual Machine)은 수신된 스마트 메시지에 의해 생성된 작업의 실행을 위한 하드웨어 추상화 계층처럼 동작한다. 실행을 위한 라이브러리가 포함되어 있다. 태그 공간(Tag Space)은 들어오는 스마트 메시지의 실행을 위한 이름 기반 메모리를 제공하고 호스트의 OS와 I/O 시스템에 대하여 단일한 인터페이스를 제공하고 있다.



(그림 3) 스마트 메시지와 노드의 구조

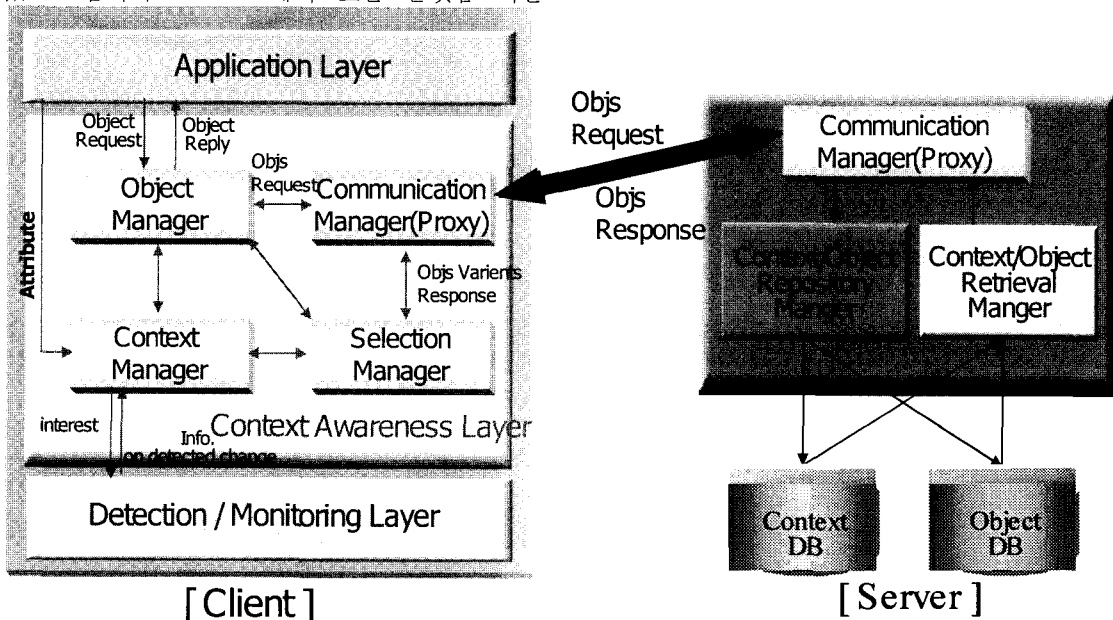
### 3. 유비쿼터스 네트워킹을 위한 새로운 미들웨어 설계

본 장에서는 앞에서 언급한 유비쿼터스 네트워킹 연구 동향을 바탕으로 한국전자통신연구원(ETRI)의 임베디드S/W기술센터에서 연구하고 있는 "임베디드 소프트웨어 표준 플랫폼 기술

개발" 중에서 "편재형 컴퓨팅 미들웨어 기술 개발"의 일환으로 제안하고 있는 유비쿼터스 네트워킹 미들웨어 구조를 설계한다.

#### 3.1 새로운 상황인식 구조의 설계

본 연구에서는 Couder와 Kermarrec[6]의 상황인식 처리 구조를 기반으로 새로운 구조를 제안



(그림 4) 새로운 상황인식 처리 시스템 구조

한다. 기존의 연구에서는 클라이언트 측의 미들웨어 구성에 연구의 초점이 맞추어져 있기 때문에 상황 정보 서버의 구조가 명확하지 않아, 상황 정보를 저장하는 방법과 검색 모듈이 구성되어 있지 않다. 또한 클라이언트와 서버간의 통신 및 프락시 모듈의 기능이 없어 네트워크 장애에 따른 재전송 및 캐싱(caching) 기능이 부족한 단점이 있다.

이러한 문제점을 보완하기 위해 본 연구에서는 (그림 4)와 같은 새로운 구조를 제안한다. DML (Detection/Monitoring Layer)은 사용자나 컴퓨팅 환경에 대해 위치, 컴퓨팅 능력, 가용한 대역폭 등 응용 프로그램과 상호작용에 관련된 이벤트와 자원 등을 감지하여 상위 계층에 통보한다.

CAL(Context-Awareness Layer)은 감지된 환경 정보에 대해 상황 정보를 인식하고 처리하는 역할을 수행한다. 그 중에서 상황관리기(Context Manager, CM)는 사용자의 취향, 기기의 성능, 현재 위치 등을 포함하고 있는 주어진 환경에 대한 상황(context) 명세를 관리한다. 그리고, 상황 정보는 상황 속성(contextual attributes) 형식의 데이터베이스로 저장된다. 상황 속성은 상황의 특별한 상태를 나타내는 쌍 (이름, 값)으로 된 정보이다. 하위 계층의 DML에서 현재 환경에 대한 상황 변화가 감지 되었을 때 그에 대한 정보를 넘겨 받는다. 객체 관리기(Object Manager, OM)는 응용 프로그램에서 사용되고 있는 상황 객체에 포함되어 있는 모든 정보를 위한 데이터 구조를 관리한다. 응용 프로그램이 객체를 참조했을 때, 정보서버(Information Server)로부터 정보를 검색하고 적당한 상황 정보를 답변하는 역할을 수행한다. 검색된 객체를 위해 데이터 구조의 새로운 엔트리를 생성하는 역할도 수행한다. 선택관리기(Selection Manager, SM)는 응용 쪽에서 요구가 들어오면, 객체 리스트의 속성

(attributes) 값과 CM의 현재 상황 명세를 사용하여 가장 유사한 상황 정보를 찾아 낸다. 사용자의 선호도(preference)를 참조하여 필터링하고 적합한 결과를 OM에 보낸다. 통신관리기(Communication Manager, Proxy)는 서버와의 통신을 담당하며 네트워크 오류에 대비하여 캐싱 기능을 수행한다.

서버 쪽의 구조를 살펴보면, 상황저장관리기(Context Repository Manager)는 상황에 대한 명세 및 각 명세에 따른 객체 내용을 데이터베이스에 저장하는 역할을 수행한다. 상황검색관리기(Context Retrieval Manager)는 데이터베이스에 저장된 상황의 명세 및 객체 내용을 검색하는 역할을 수행한다. 서버 쪽의 통신관리기(Communication Manager, Proxy)는 클라이언트와의 통신을 담당하며 네트워크 오류에 대비하여 캐싱 기능을 수행한다.

이러한 상황 인식 시스템을 이용하여 사용자의 이동 객체에 대한 발견은 센서(sensor or receiver)에 의해 수행되며, 객체 등록은 클라이언트에서 상황을 인식하여 서버 부분에 데이터베이스 형태로 저장하고, 객체의 실행은 클라이언트의 응용에서 상황 정보를 요구하면 서버의 정보를 검색한 다음 상황 정보를 전달하여 수행되는 구조로 되어 있다.

그리고 이동 단말에 대한 마이크로 모빌리티와 빠른 핸드오프(hand-off)를 지원하는 이동성 엔진을 설계하고 있다.

### 3.2 Ad-hoc 네트워크와 동적 라우팅 설계

본 연구에서는 액티브 네트워크 기술을 이용하여 센서 네트워크에서 이동 Ad-hoc 기능이 지원되는 라우팅 엔진을 개발하고 있다. 또한 센서 네트워크에 적당한 라우팅 알고리즘을 찾기 위해 라우팅 알고리즘을 평가할 라우팅 시뮬레이션



(Simulation) 도구를 개발할 예정이다. 이러한 도구를 통해 센서 네트워크에 가장 효율적인 라우팅 알고리즘을 선택하거나 새로운 알고리즘을 설계할 예정이다.

### 3.3 새로운 협력 메시지(Cooperative Message) 설계

기존의 스마트 메시지는 유비쿼터스 네트워크에서 코드 이동에 대한 기본 개념을 설계하고 있지만 헤더 부분에 대한 정의가 미비하여 코드 처리에 대한 판단을 위해 메시지 전체를 해석해 봐야 한다는 문제점이 있다. 또한, 스마트 메시지가 수송관리를 거친 뒤에도 태그 공간에서 수행하기에 부적합하면 다시 수송관리로 보내져 검증을 다시 수행함으로써 자원을 낭비하는 문제점이 있다.

그러한 문제점을 해결하기 위해 본 연구에서는 (그림 5)와 같은 새로운 메시지 형식인 협력 메시지(Cooperative Message)와 처리 시스템을 제안하고 처리 구조를 설계한다. 메시지의 형식은 크게 메시지 헤더와 자원 요구 테이블, 실행코드 부, 데이터 부, 그리고 부분적인 코드 이동에 따른 상태 보존을 위해서 상태(state) 정보를 두고 있다. 우선 메시지 헤더에는 메시지의 ID와 인증 및 권한 부여, 메시지 전체 검증을 위한 서명(Signature) 등이 포함되어 있으며, 메시지의 유효성이 보장되는 생존기간(Lifetime) 값을 포함하고 있다. 또한 메시지 처리의 효율성을 위하여 각 메시지에 대한 우선순위(Priority)값의 필드도 포함된다.

메시지 속성, 인증 및 확인을 위한 내용을 헤더에 포함시킴으로써 각 NES에서 메시지 처리를 용이하게 하도록 한다. 메시지를 수신한 각 노드에서는 메시지 검증을 위하여 메시지 전체에 대한 참조 없이 헤더를 보고 해당 메시지의 타입을 확인 가능하며, 타입에 맞는 검증단계를 수행할 수 있다. 메시지 자체에 대한 검증이 완료되

면, 시스템이 허용해 줄 수 있는 자원과 메시지가 요구하는 자원의 차이가 날 수 있기 때문에 곧바로 사용 자원에 대한 검증 과정을 수행한다. 이런 수행은 사용 가능한 자원의 보유여부와 함께 협력 메시지가 사용하고자 하는 특정 자원이 NES에 포함되어 있는지를 확인하는 과정이다.

메시지 헤더 값을 통하여 각 NES 노드들은 해당 메시지에 대한 수용 여부를 결정하게 된다. 다음으로 자원 요구 테이블은 각 메시지가 다음에 사용할 자원들에 대한 사용 내역을 기록하고 있다. CPU 사용시간, 공용 메모리 요구량, 사용하고자 하는 자원에 대한 기록이 포함되어 있다. 코드 부와 데이터 부는 실행에 필요한 코드 블록과 데이터 블록들이 포함되어 있다. 코드 부의 마지막에는 예외 처리 코드(Exception Code)를 포함한다. 예외 처리 코드는 해당 메시지가 현재 NES 노드에서 실행될 권한이 없거나 실행될 필요성이 없을 때 처리를 위한 부분으로서 일반적으로 현재 노드를 건너뛰기 위한 코드와 강제 종료를 위한 코드가 포함되어 있다.

협력 메시지가 노드에 도착하면 Entry Module에서는 메시지를 확인하는 과정을 수행한다. 메시지 헤더에 포함된 ID와 서명(Signature)를 확인하고 현재 메시지의 생존기간이 유효한지 검사한다. 그 다음 자원(Resource) 요구량과 자신이 가지고 있는 자원의 정도를 비교한다. 메시지를 확인하고 비교하는 동안 요구 조건을 만족하지 않는 경우에는 코드 부의 예외처리 과정을 수행하도록 한다. 각 메시지는 자신의 우선순위(Priority)가 존재하기 때문에 큐(Queue)에 적재된 메시지는 우선 순위에 의해서 순서가 결정된다. 메시지 처리의 기아 현상을 제거하기 위하여 큐에 대기한 시간이 길어질수록 우선순위를 상향시켜주는 정책을 혼합하여 사용한다.

가상 머신(Virtual Machine)은 메시지에 포함되어있는 코드를 수행하는 공간이다. 현재의

NES에서 실행될 코드 부분은 이미 결정되어 있기 때문에 코드 실행에 있어 필요한 공용 메모리 공간을 이때 확보하게 된다. 공용 메모리 공간으로서 Content Memory를 제공하고 있는데 가상 머신에 제공될 때는 이름을 기반으로 하는 일정한 형식을 취하여 제공된다.

#### 4. 결 론

현재까지의 컴퓨팅 환경은 사용자가 컴퓨터 모니터 앞에 앉아 가상의 세계를 경험하도록 하는 것이었다. 그러나 컴퓨팅 기술의 빠른 향상으로 인하여 과거의 컴퓨팅 환경은 유비쿼터스 네트워킹 환경으로 변해가고 있다. 여러 종류의 많은 컴퓨터들이 현실 세계의 다양한 전자 제품들에 이식되어 상호 연결되고 언제, 어디서나 컴퓨팅을 이용할 수 있고 컴퓨터가 협력을 통해 새로운 응용이 가능한 인간과 컴퓨터 간의 최적화된 컴퓨팅 환경이 되고 있다. 또한 이를 넘어서 인간 주위의 모든 사물에 마이크로 프로세서가 내장되어 컴퓨팅이 가능하게 되는 유비쿼터스 컴퓨팅 사회로 발전되고 있다.

본 연구에서는 유비쿼터스 네트워킹 기술에서 근간이 되는 미들웨어 기술의 최근 동향을 파악하여 현재 문제점이 되고 있는 부분을 기술하고, 문제점 극복을 위한 새로운 미들웨어 설계 내용을 제안했다. 이러한 설계 내용을 바탕으로 성능 평가를 통해 새로운 미들웨어 기술이 구현될 것이다. 또한 상황인식 기술의 활용도를 높이기 위해 자동 인식의 활용 영역을 확대하고 시스템에 포함시킬 예정이다. 무선 네트워크 환경을 제공하기 위해 HomeRF, Bluetooth, 무선 LAN 등 다양한 통신 방식이 존재하게 되는데, 응용에 따라 효과적으로 사용할 수 있는 무선 방식을 분류하고 본 연구에 반영해 나갈 예정이다.

#### 참고문헌

- [1] 사카무라 겐(저), 최운식(역), 유비쿼터스 컴퓨팅 혁명, (주)동방미디어 출판사, 2002.
- [2] 리처드 헌터(저), 윤정로(역), 최장욱(역), 유비쿼터스: 공유와 감시의 두 얼굴, (주)북21 출판사, 2002.
- [3] A. K. Dey, "Context-Aware Computing: The CyberDesk Project," Proc. of the AAAI 1998 Spring Symposium on Intelligent Environments (AAAI Technical Report SS-98-02), pp. 51-54, Mar. 1998.
- [4] Roy Want, Andy Hopper, Veronica Falcao, Jonathan Gibbons, "The Active Badge Location System," ACM Transactions on Information Systems, Vol. 10, No. 1, pp. 91-102, 1992.
- [5] Stephen S. Yau, Fariaz Karim, "Context-Sensitive Middleware for Real-Time Software in Ubiquitous Computing Environments," Fourth International Symposium on Object-Oriented Real-Time Distributed Computing, pp. 163-170, 2001.
- [6] P. Couder, A.M. Kermarrec, "Improving Level of Service for Mobile Users Using Context-Awareness," 18th IEEE Symposium on Reliable Distributed Systems, pp. 24-33, 1999.
- [7] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster, "The Anatomy of a Context-aware application," Wireless Networks Vol. 8, Issue 2/3, pp. 187-197, 2002.
- [8] <http://www.ietf.org/html.charters/>

- mobileip-charter.html.
- [9] <http://www.ietf.org/html.charters/seamoby-charter.html>.
- [10] M.S. Corson and J.P. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," IETF RFC 2501, Jan. 1999.
- [11] 권혜연, 신재욱, 이병복, 최지혁, 남상우, 임선배, "이동 Ad Hoc 네트워크 기술동향," 전자통신동향분석, 제18권 제2호, 2003.
- [12] C. K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall PTR, 2002.
- [13] 김상하, "이동 Ad Hoc 네트워크 라우팅 프로토콜 기술," 정기간행물 전파지, <http://www.kora.or.kr/kora/radar/200209/sub8.html>, 2002.
- [14] 이종화, "Active Networking 기술 소개 및 연구 현황," <http://new.itfind.or.kr/KIC/etlars/i-ndustry/jugidong/873/87304.html>, 1998.
- [15] <http://webs.cs.berkeley.edu/tos/index.html>.
- [16] C. Borcea, D. Iyer, P. Kang, A. Saxena, and L. Iftode, "Cooperative Computing for Distributed Embedded Systems," Proc. of 22nd International Conference on Distributed Computing Systems, pp. 227-236, 2002.
- [17] P. Stanley-Marbell, C. Borcea, K. Nagaraja, and L. Iftode, "Smart Messages: A System Architecture for Large Networks of Embedded Systems," Proc. of the Eighth Workshop on Hot Topics in Operating Systems, pp. 181-189, 2001.
- [18] T. von Eicken, D. Culler, S. Goldstein, and K. Schauer, "Active Messages: A Mechanism for Integrated Communication and Computation," Proc. of the 19th Annual International Symposium on Computer Architecture, pp. 256-266, May, 1992.
- [19] B. Schwarts, A. W. Jackson, W. T. Strayer, W. Zhou, R. D. Rockwell, and C. Partridge, "Smart Packets for Active Networks," ACM Transactions on Computer Systems, pp. 397-413, Feb. 2000.
- [20] D. Milojicic, W. LaForge, and D. Chauhan, "Mobile Objects and Agents," USENIX Conference on Object-oriented Technologies and Systems, pp. 1-14, 1998.



김정기 (金汀基)  
 e-mail : jkk@etri.re.kr  
 1992년 전북대학교 컴퓨터 공학과(공학사)  
 1994년 전북대학교 컴퓨터 공학과(공학석사)  
 1999년 전북대학교 컴퓨터 공학과(공학박사)  
 1996년-1998년 시스템공학

연구소 연구원  
 1998년-현재 한국전자통신연구원 임베디드S/W기술센터 선임연구원

<주관심분야> 임베디드S/W, 유비쿼터스 컴퓨팅, 병렬 정보검색



신창민(辛昌珉)

e-mail : cmshin@etri.re.kr

1996년 고려대학교 컴퓨터학과(이학사)

1998년 고려대학교 컴퓨터학과(이학석사)

1998년-2001년 (주)대우통신 제품연구소

2001년-현재 한국전자통신연구원 임베디드S/W기술센터 연구원

<주관심분야> 편재형 컴퓨팅, 무선통신, 모바일 컴퓨팅, 운영체제, 3D 그래픽스



박승민 (朴勝敏)

e-mail : minpark@etri.re.kr

1981년 울산대학교 전자공학과(공학사)

1983년 홍익대학교 전자공학과(공학석사)

1998년-현재 충남대학교 전자공학과 박사과정

1983년-1984년 (주)LG전자

1984년-현재 한국전자통신연구원 임베디드S/W기술센터 S/W개발도구연구팀장, 책임연구원

<주관심분야> 임베디드S/W, 유비쿼터스 컴퓨팅, RTOS, 이동 통신망



유석대 (劉錫大)

e-mail : sdyu@dcs.chonbuk.ac.kr

1999년 전북대학교 컴퓨터학과(이학사)

2001년 전북대학교 전산통계학과(이학석사)

2002년-현재 전북대학교 컴퓨터통계정보학과 박사과정

2003년-현재 한국전자통신연구원 임베디드S/W기술센터 위촉연구원

<주관심분야> Ad-hoc 네트워크, 스마트 메시지, 이동 네트워크 보안