



네트워크 기반 대규모 임베디드 시스템의 상호협동을 위한 Smart Message 기법

김문자*, 조인준**, 조기환***

● 목 차 ●

- 1. 서론
- 2. 상호협동 컴퓨팅 모델
- 3. Smart message 기법
- 4. 관련연구와 비교
- 5. 결론

1. 서론

일반적으로 임베디드 시스템은 임의의 시스템에 내장(Embedded)된 형태로 구성되며, 일단의 특정 하드웨어 구성요소를 제어하기 위하여 사용되며, 제한되고 전문화된 기능을 수행하는 장치로 정의된다. 최근 인터넷을 기반으로 하는 응용들의 급격한 팽창에 의한 정보화 기반구조 흐름 중에 하나는 이러한 임베디드 시스템들이 네트워크를 수단으로 상호 연결되고 협력한다는 점이다. 예로 생산라인에 참여하는 수많은 로봇에 부착된 센서들은 서로 유기적으로 정보를 주고 받음으로써 전체 라인의 유기적 기능을 다할 것이다. 한편 하드웨어 기술 발전에 힘입어 가격이 하락하고 미래 정보화 사회 환경에서 그 효용성이 커짐에 따라서 전문화된 기능을 수행하는 임베디드 시스템의 수는 기존의 범용 기능을 수행하는 PC를 훨씬 능가하고 있다. 지리적 혹은 기능적으로 분산된 수천 혹은 수만 개의 임베디드 시스템들은 네트워크를 수단으로 상호협력 능력을 갖음으로써 더욱 강력하고 유용한 시스템으로 발전할 것이다[1].

템으로 발전할 것이다[1].

대규모 임베디드 시스템들의 연결 형태는 일단의 임베디드 시스템 집합이 네트워크 그룹을 형성하고 이러한 그룹들이 모여서 차 상위 네트워크 그룹을 구성하는 계층적 구조를 갖는다. 이때 차 상위 네트워크 그룹들은 지리적 위치, 관리 주체의 측면에서 그룹단위 노드 기능은 물론 각 네트워크 그룹 내에서 사용하는 상호연결 기술은 현실적으로 이질적인 경우가 대부분이다. 또한 단위 임베디드 시스템 측면에서 각 노드들은 현재 온도 혹은 주위에서 발생하는 사건을 감지하는 등의 특정 작업의 실행에 최적화된 하드웨어로 구성된다. 즉 임베디드 시스템은 적용 특성상 현장 적용이라는 본래의 특성으로 인하여 노드 자체 혹은 기능의 이동성을 가지며 주로 휴대형 전원을 사용함에 의해 전원의 제한성을 갖는다. 노드 이동성이 주어지지 않더라도 특정 기능만을 수행하는 소규모 형태의 임베디드 시스템 모두에게 전원 시설을 설비하는 것은 현실적이지 않을 수 있다[8].

위와 같은 네트워크 기반 대규모 임베디드 시스템 고유 특성들을 수용하기 위해서는 임베디드 시스템간 통신 방법론에 새로운 접근이 요구된다. 즉 기존의 정적 구조의 통신 방법을 대규모 임베디드

* 전주공업대학 영상멀티미디어과 초빙 전임강사
 ** 배재대학교 컴퓨터공학과 부교수
 *** 전북대학교 전자정보공학부 조교수

시스템 시스템에 그대로 적용할 수 없다. 가장 커다란 요인으로 노드의 이동으로 인한 위상 변화이다. 다음은 대규모 노드에서 발생할 수 있는 노드 고장에 따른 위상변화를 들 수 있다. 사람에 의해 조작되는 인터넷 노드들과는 다르게 네트워크로 연결된 임베디드 시스템은 일반적으로 사람의 보살핌이 적용되지 않거나 적용할 수 없는 형태가 보통이다. 즉 고장에 대해서 광범위하게 산재된 각 개별 노드들을 reset할 수 없을 것이다. 마지막으로 통신 단절에 따른 위상 변화를 들 수 있다. 노드의 이동성은 네트워크 접속의 의사적(spurious) 단절현상과 전원으로 제한성에 따른 의도적(voluntary) 단절현상을 수반한다. 이러한 네트워크 위상변화는 응용 측면에서는 노드의 가용성 문제로 귀결된다. 한편 TCP/IP와 같은 기존의 통신 방법론은 노드 단위로 유일한 이름을 제공하는 naming 기법을 제공하고 있다. 그러나 위와 같이 노드의 네트워크 위상이나 노드 가용성이 동적으로 변화하는 시스템 상황에서는 고정된 바인딩을 이용하는 기존의 naming을 그대로 사용할 수 없다.

네트워크 기반 대규모 임베디드 시스템 환경에서 네트워크 위상 및 노드 가용성에 변화가 발생하더라도 네트워크를 구성하는 임베디드 시스템 노드들 사이에 광역의 작업이 가능해야 한다. 즉 현장의 센서 임베디드 시스템 노드에게 데이터 전파(dissemination), 역으로 임베디드 시스템 노드들로부터 데이터 수집(collection), 그리고 필요에 따라서는 전달받은 데이터를 라우팅하고 나아가서는 지리적으로 영역이 구분된 상황에서 전체 시스템에서 일어나는 동작을 추적(trace)하는 등의 작업이 가능해야 한다[4][5]. 이를 위해서는 기존의 분산 컴퓨팅 모델과 다른 새로운 컴퓨팅 모델이 필요하다.

네트워크 기반 대규모 임베디드 시스템의 각 노드 혹은 노드 그룹은 지리적 위치, 동작, 전원 혹은 하드웨어 관점에서 독자적인 특성을 갖는다. 이러

한 독자적인 특성을 아우르는 응용을 구축하기 위해서는 먼저 각 노드는 programmable하고 나아가 사용자가 정의한 응용을 지원하여야 한다. 또한 네트워크 위상과 노드 가용성이 변화하는 상황에서 지능적인 데이터 전달 방법론이 필요하다. 더불어서 기존의 분산처리 컴퓨팅에서 적용하였던 노드 간 동기화가 매우 어려우며 심지어는 요청/응답 형태의 기본적인 통신 절차가 불가능하게 된다. 따라서 응용은 결과의 질(QoR : Quality of Result)를 만족하는 한 부분적인 실행을 수용할 수 있어야 한다. 한편 대규모 노드로 구성되고 네트워크 위상 및 노드 가용성이 변화하는 네트워크 환경에서는 유일한 식별자를 사용하는 것보다는 노드의 기능적 특성을 기반으로 노드 지정이 가능한 naming 기법이 요구된다. 본고에서는 미국 Rutgers 대학을 중심으로 제시된 Smart message를 소개한다[1][8][9]. Smart Message는 이미 잘 알려진 Active network 개념의 Active message[3], Smart packet[7] 그리고 Mobile agent[6]와 유사한 기초에서 출발하고 있다. 즉 이동 가능한 데이터와 코드로 구성되어 네트워크 내의 각 노드를 단계적으로(순차적으로) 이동하면서 필요한 경우에 방문한 노드에서 코드를 실행하고 데이터를 변화한다. 그러나 Smart Message는 스스로 라우팅 경로를 결정하며 방문한 노드에서 제공하는 최소한의 기능을 활용 고유의 접근 특성을 갖는다. 이러한 고유 특성은 네트워크 위상 및 노드 가용성이 수시로 변화하는 네트워크 기반 대규모 임베디드 시스템 노드들 사이에 상호협동 컴퓨팅 모델의 기초가 된다. 위에서 언급한 방법들 사이의 구체적인 공통점과 상이점은 4장에서 다루기로 한다.

2. 상호협동 컴퓨팅 모델

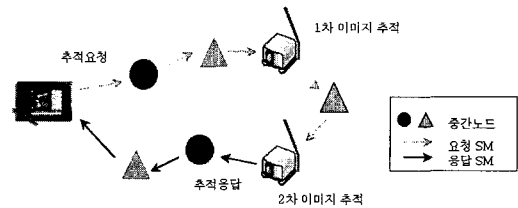
네트워크 기반 대규모 임베디드 시스템에서 가장 원천적인 응용은 현장에 구축된 다수의 임베디

드 시스템들이 감지하거나 생성한 데이터를 전체적으로 모으거나, 혹은 임의의 데이터를 일단의 임베디드 시스템 집합에 전파로 생각할 수 있다. 기존에 이러한 응용을 위한 임베디드 시스템을 구성하는 시점에 네트워크 위상이 결정된다. 그러나 네트워크 기반 대규모 임베디드 시스템 환경은 동적으로 임의의 중간 노드가 이동하거나 고장이 발생하므로 전체 분산 응용이 동작하지 못하는 사태를 겪게 된다. 나아가 현장 상황에 따라서 임베디드 시스템에 적용되는 응용 자체가 변화할 수 있다. 그 결과로 범용적 특성의 PC 네트워크가 아닌 임베디드 시스템에 의해 형성된 네트워크는 동적인 네트워크 변화와 새로운 응용의 적용이 상대적 어렵다. 따라서 사용자가 정의한 분산 응용을 적용적으로 수용하는 새로운 컴퓨팅 모델이 요구된다[1].

대규모 임베디드 시스템 네트워크에서 동적인 네트워크 위상 및 노드 가용성 변화를 수용하는 분산처리 시스템은 각 임베디드 시스템 노드 스스로 자신의 자원과 특성을 고려하여 적용된 응용이나 데이터 라우팅에 관련된 작업을 스스로 유도하는 형태를 가져야 한다. 즉 전체 분산 응용을 유효 적절하게 수행하기 위해서는 각 노드들이 상호간에 협동하는 컴퓨팅 모델을 가져야 한다. 따라서 동적인 분산 응용에 참여하는 노드의 집합은 각 노드들의 고유 특성에 의해 식별되며, 응용 형태로 제어되는 라우팅 수단에 의해 경로를 선택한다. 또한 응용은 주어진 QoR이 만족된다면 부분적인 실행을 최종결과로 수용할 수 있어야 한다. 이러한 상호협동 컴퓨팅 모델은 Smart Message(SM) 개념에 의해서 구현된다.

네트워크 환경에서 SM의 실행은 SM이 방문한 각 노드에서 설정된 특성에 따라서 다르게 된다. 즉 노드들은 전달된 SM 내의 실행 정보와 자신이 갖고 있는 특성이 일치한 경우에 임베디드 시스템은 해당 센서를 통하여 데이터를 읽어서 메시지에 포함시키거나, 혹은 전달된 데이터와 자신이 읽은

데이터를 이용하여 새로운 정보를 산출하는 등의 작업을 실행한다. 반면에 전달된 SM 정보와 특성이 일치하지 않는 경우에는 단지 메시지의 전달과정에 참여하는 단순한 네트워킹 노드로 역할을 수행한다. 따라서 특정 응용을 처리하기 위해서 상호협동에 참여하는 노드들은 제한된 기억장치와 실행 환경을 갖고 임의의 시간동안 SM을 지원한다. 반면에 응용의 실행에 직접 참여하지 않는 노드들은 SM 자신의 라우팅 기능에 의해서 SM의 전달에 참여함으로써 간접적으로 상호협동을 지원한다. 이러한 SM의 기능이 대규모 임베디드 시스템들에 대해서 능동적인 상호협동의 기반구조를 자연스럽게 제공한다.



(그림 1) SM에 의한 상호협동 컴퓨팅 모델

(그림 1)은 SM에 의한 네트워크 기반 대규모 임베디드 시스템의 상호협동 컴퓨팅 모델의 개념도를 나타내고 있다. 이러한 모델의 예로 고속도로, 공항 등의 지리적으로 광역에 산재한 임베디드 시스템 형태의 카메라 네트워크에서 물체의 궤도를 추적하는 응용을 가정할 수 있다. 임의의 특징을 갖는 물체를 추적하기 위해서 좌측의 정보단말이 네트워크에 SM을 보낸 경우를 살펴보자. 대규모 네트워크 환경에서 SM은 물체의 이미지를 획득할 수 있는 카메라 노드를 찾아서 이동하여야 한다. 이러한 과정에서 SM은 물체의 이동 방향, 속도, 그리고 지리적인 정보를 사용할 수 있을 것이다. SM이 물체의 이미지를 획득하는 카메라 노드로 전달되는 과정에는 SM을 라우팅 해주는 중간노드들이 필연적으로 참여하게 된다. 임의의 카메라 노드에

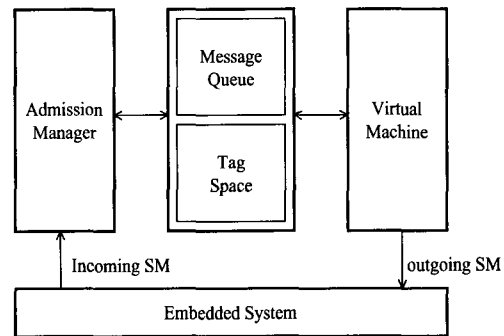
SM이 전달되면 그 노드는 물체의 이미지를 획득하고, 최초로 원했던 특성을 갖고 있는지(특성의 예로 원하는 물체가 사람인 경우에 전면 얼굴) 분석한다(그림에서 1차 이미지 추적). 만약에 원하는 특성을 갖는 이미지를 얻지 못한 경우, 혹은 이동하는 물체의 궤도를 추적하는 경우에는 다른 이미지를 얻거나 추적을 계속하기 위해서 다른 카메라 노드로 SM은 라우팅 된다. 이때 다음 카메라 노드의 선택은 물체의 이동특성을 기준으로 한다. 그림의 2차 이미지 추적에서 SM은 원하는 특성을 획득하였다고 판단되면 획득한 이미지와 함께 최초 추적을 요청한 노드에게 응답하는 SM을 전달한다. 이때 최초 요청한 노드로 라우팅 되는 과정은 지리정보를 기반으로 할 것이다.

3. Smart Message 기법

네트워크 기반 대규모 임베디드 시스템 환경에서 네트워크 위상 및 노드 가용성의 변화를 능동적으로 수용하기 위한 SM 시스템 구조는 네트워크를 구성하는 모든 노드를 지능화하기보다는 통신에 사용되는 메시지에 지능적 요소를 부여함으로써 상호협동에 참여하는 노드에 부과되는 요구사항을 최소화하는 접근방법을 사용한다. 따라서 새로운 응용이나 프로토콜의 구현을 위해서 네트워크 내의 모든 노드들을 수정해야 하는 기존의 통신 시스템 구조에 융통성을 부여하고 있다. 사실 특정한 단위 작업을 수행하는 수천, 수만 개의 네트워크 기반 임베디드 시스템을 모두 수정하는 일이란 경우에 따라서는 불가능 할 수 있다. 응용의 실행에 필요한 데이터나 코드를 통신 메시지에 포함시킴으로써 융통성 있는 시스템을 구성하려는 시도는 Active Network 개념으로 더 잘 알려져 있다. 그러나 SM은 확장성을 가장 본질적인 요소로 삼는다는 점과 부분적인 실행 결과를 최종 결과로 수용하는 점이 Active Network과 구별된다.

3.1 시스템 구조

SM은 방문한 노드에서 일단의 응용을 수행하기 위해서 사용되는 코드와 데이터를 포함한다. 또한 응용을 위한 상호협동 작업이 진행되면서 SM은 동적으로 구성된다. 이러한 과정은 기본적으로 태그 개념을 기반으로 한다[1]. 각 노드는 SM의 실행 과정에 일관성을 제공할 태그로 구성된 Tag Space라는 제한된 데이터 저장공간을 갖는다. 노드에서 SM의 실행은 Tag Space 내의 태그를 참조, 생성 혹은 삭제 포함한다. 태그는 데이터 교환 혹은 SM 사이의 동기화는 물론 naming과 라우팅 과정에도 사용된다. 또한 임의 노드에서 SM이 실행되는 동안에 네트워크 내의 다른 노드에게 전달한 새로운 SM의 발생을 포함한다. 이러한 새로 발생된 SM들은 다음 노드를 찾기 위해서 라우팅 정보를 수집하는 등의 작업을 수행한다. 따라서 네트워크 내에서 실행되는 SM들의 집합이 곧 응용인 셈이다.



(그림 2) 시스템 구조

(그림 2)는 네트워크 기반 대규모 임베디드 시스템 환경에서 SM에 의한 상호협동 작업을 위한 시스템 구조를 보인다. 각 노드들은 SM의 수신과 실행에 대해 독립적인 프로그래밍 환경을 갖는다. 이를 구체화하기 위해 각 노드에서 제공해야할 최소 요구사항은 Tag Space, Admission Manager 그리고 Virtual Machine이다. 각각의 구체적인 기능은 다음과 같다.

3.1.1 Tag Space

태그는 유일한 식별자, 접근제어 리스트, 유효기간 그리고 데이터로 구성된다. 식별자는 파일시스템에서 파일 이름처럼 태그의 이름으로 사용된다. 임의의 SM이 태그를 접근하는 권한은 접근제어 리스트에 포함된 정보에 따른다. 유효기간은 해당 노드의 Tag Space에 태그가 유지되는 기간을 나타낸다. Virtual Machine은 응용을 실행하는 과정에서 Tag Space를 관리하며 Admission Manager는 SM의 접근제어를 결정하기 위해서 Tag Space를 접근한다.

임의의 태그는 다수의 응용에 의해서 사용된다. 네트워크 내의 노드는 각 상태(예로 임의의 센서) 대해서 하나씩의 태그를 갖는다. 이때 태그를 읽는다는 것은 해당 센서가 감지한 값을 읽는 것과 같다. 또한 SM은 관련이 있는 노드를 선정하거나 라우팅정보를 유지하기 위해서 태그를 사용하기도 한다. 예로 네트워크 전반에 걸쳐서 노드 사이에 데이터를 전달하는 응용에서 사용하는 SM은 방문한 노드에 태그를 생성하여 그동안 발견한 라우팅 정보를 태그의 데이터 부분에 저장한다. 이러한 방법으로 전통적으로 관리자에 의해서 노드 단위로 유지 관리되어 왔던 라우팅 정보를 태그를 수단으로 관리함으로써 현재의 네트워크 위상에 적절한 라우팅 정보를 동적으로 관리할 수 있다.

3.1.2 Admission Manager

다수의 SM이 전달되는 상황에서 노드의 Virtual Machine이 임의의 작업을 실행하고 있는 동안에 새로운 SM이 도착하면 먼저 이를 저장해야 한다. 더불어 전원, 메모리, 네트워크 전송률 등 자원이 제한되는 임베디드 시스템 환경에서 어떤 노드의 현재 가용한 자원의 상태를 파악하여 새로운 SM의 수용여부를 판단하여야 한다. 따라서 Admission Manager는 새로 전달된 메시지를 받아서 메시지 큐에 저장하고, 노드 내 자원의 가용 여부 혹은 원하는 태그의 유무 여부에 따라서 접근을 제어한다.

(그림 3) SM 메시지 포맷에서 보이는 바와 같이 각 SM은 자신에게 필요한 자원을 명시적으로 포함하고 있다. 또한 Virtual Machine은 SM이 정의한 작업을 실행하기 위해서 SM에 명시된 자원이 충분한지 여부를 판단하여 불충분하다고 판단되면 작업을 중단하고 이를 제거한다.

3.1.3 Virtual Machine

잘 알려진 바와 같이 Virtual Machine은 서로 다른 하드웨어 플랫폼 사이에서 SM의 실행을 위한 하드웨어 추상화 계층이다. 따라서 Virtual Machine은 사용되는 메모리와 실행시간 관점에서 부담이 적어야 하고, 이동하는 응용 코드가 안전하게 실행되는 환경을 제공함은 물론 SM을 지원하기 위해서 필요한 프리미티브들이 제공되어야 한다. 현재 Java virtual machine이 이러한 조건을 만족하는 도구로 볼 수 있다.

3.2 Smart Message 포맷

SM은 동적인 메시지 전달체계를 위한 상태, 응용에서 요구하는 자원, 노드와 SM의 관련 여부를 처리하는 핸들러, 그리고 응용의 실행을 위한 코드와 데이터 부분으로 구성된다. (그림 3)은 SM 포맷 구조를 보인다.

헤더	자원 테이블	핸들러	상태	코드	데이터
----	--------	-----	----	----	-----

(그림 3) Smart Message 포맷

고정된 크기의 헤더 필드는 전자서명과 SM 구성 요소에 대한 정보를 포함한다. 전자서명은 SM을 식별하기 위해서, 노드에서는 전달된 SM의 태그에 대한 접근 제어를 위한 수단으로 사용된다. 자원 테이블 필드는 실행 시간, 접근하기를 원하거나 생성될 태그들, 요구되는 메모리 크기 그리고 네트워크 트래픽 등 SM의 실행을 위해서 요구되는 자원들을 포함한다. Admission Manager는 임의의 노드

에 SM이 도착하면 자원 테이블의 내용과 실제 노드에서 사용 가능한 자원을 비교하여 접근제어를 한다. 또한 SM은 노드에 도착하면 노드와 SM의 관련 여부를 결정하기 위해서 즉시 실행될 핸들러 필드를 포함한다. 만약 해당 노드가 SM의 관심에 해당한다면 작업을 실행할 준비를 하게 된다. 상태 필드는 SM이 전달된 뒤에 작업을 실행할 때 필요한 노드 상태를 포함한다. 코드 필드는 독립적인 프로그램으로서 데이터 혹은 필요에 따라서는 다른 프로그램과 같이 사용되기도 한다. 데이터 필드는 이동 데이터를 포함하며 주로 코드가 실행될 때 사용된다.

3.3 라우팅과 실행

기존의 네트워크에서 라우팅은 네트워크를 구성하는 노드 각각에 미리 정의된 동일한 라우팅 프로토콜을 가정하고 있다. 잘 알려진 라우팅 테이블 기반의 IP(Internet Protocol)가 대표적인 경우이다. 3.1절 시스템 구조에서 언급하였듯이 SM은 태그 특성을 이용하여 라우팅을 정의한다. 즉 SM은 네트워크 전반에 걸쳐 자신의 경로를 결정하는 책임을 갖게되며, 이는 전적으로 응용 수준에서 처리된다. 따라서 SM은 다양한 라우팅 알고리즘에 해당하는 라우팅 기능 중에서 선택적으로 사용이 가능한 장점을 갖는다. 임의의 노드에서 SM은 자신의 다음 경로를 결정하기 위해서 “경로발견” SM을 생성하여 주위에 전파할 수 있다. 이때 전파된 “경로발견” SM이 되돌아 올 때까지 원래의 SM은 라우팅 태그에서 블록되며, 되돌아온 결과를 기준으로 라우팅 태그를 수정함으로써 다음 라우팅이 진행된다. 태그의 정보는 지속성을 가지므로 한번 새로운 라우팅 정보를 획득하게 되면 유사한 경우에 SM에 의해서 반복해서 사용된다.

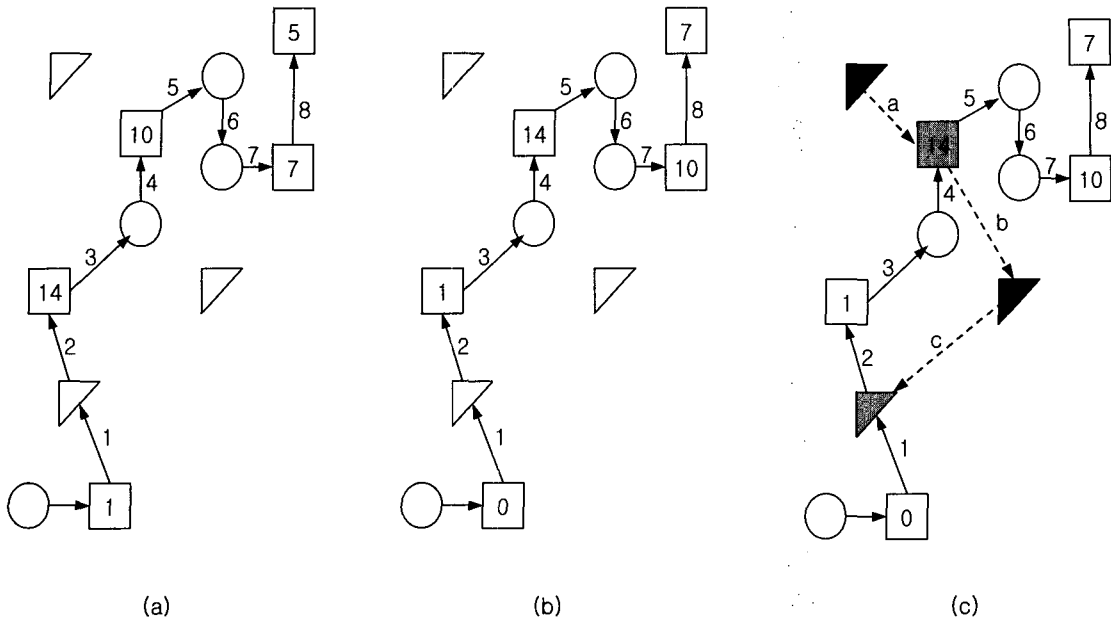
SM의 실행은 Virtual Machine에서 실행되는 태스크 형태를 취한다. 태스크가 실행되는 동안에 Virtual Machine은 SM 내의 데이터와 필요에 따라

서는 접근하고 있는 태그를 변화시킨다. 실행이 완료되면 원래 SM 혹은 내용이 변화된 SM이 다른 노드로 라우팅 된다. 변화된 새로운 SM은 자신이 원하는 자원을 충분히 지원하고 원하는 태그가 존재하는 노드에서 다시 실행된다. 실행중인 태스크는 다른 태스크에 의해서 선점되지 않으며 순차적으로 실행된다. 그러나 다른 SM에 의해서 태그가 변환되기를 기다리기 위하여 태스크 스스로 Virtual Machine을 양보할 수 있다. 한편 만약 실행시간이 SM의 자원 테이블에 정의된 실행시간을 초과하게 되면 해당 태스크는 강제로 종료된다. 반면에 핸들러는 SM의 실행과는 다르게 순차적이지 아니라 SM이 도착하는 즉시 실행되며 절대 선점되지 않는다. 이는 핸들러의 실행시간이 SM에 비해서 상대적으로 매우 짧으며 SM이 현재 노드에서 실행되어야 할지 다른 노드로 전달되어야 할지를 신속히 결정할 필요가 있기 때문이다.

3.4 Smart Message 예제

위에서 서술한 시스템 구조와 동작절차를 기반으로 SM이 실행되는 과정을 (그림 4)를 통하여 예제로 살펴보기로 한다. 그림에서 서로 다른 형태의 세 가지 노드가 원, 삼각, 사각으로 표시되어있다. 사각 노드는 (a) 그림의 밑에 위치한 원 노드에 의해 보내진 SM의 실행될 노드를 보인다. SM에 의해 구현된 응용은 다섯 개의 사각 노드를 방문하여 각 노드에서 확보중인 혹은 획득한 데이터(그림에서 사각 노드에 표시된 숫자)를 다음 사각 노드로 전파하는 것이다. 화살표 옆의 번호는 SM이 순차적으로 전달되는 순서를 표시한다. 이때 원 노드와 삼각 노드는 SM이 사각 노드를 방문하는 라우팅 과정에서 중간 노드로 역할을 수행한다.

각 사각 노드에서 확보한 데이터를 다른 사각 노드로 전파하기 위해서 SM은 네트워크 내에서 노드 사이를 이동할 때마다 이동 상태를 관리하고 최근에 획득하였던 데이터를 메시지에 포함하여 다음



(그림 4) SM의 실행 예제

노드로 라우팅 된다. (b)의 그림은 SM이 네트워크를 순회하고 난 뒤의 노드 상태를 보인다. 이러한 예는 각 노드에 부착된 센서에서 얻은 값을 기준으로 SM의 다음 경로가 결정되는 객체 추적 응용에 간단히 적용될 수 있다.

(c)의 그림에서 보이는 바와 같이 임의의 네트워크에는 다수의 응용이 동시에 실행되는 경우가 있다. 첫 번째 응용에 참여하는 노드는 사각 노드들이며, 두 번째 응용에 참여하는 노드는 삼각 노드들로써 두 번째 응용이 실행되는 경로상의 노드는 검게 표시되어져 있다. 여기서 두 응용 모두의 실행에 참여하는 노드는 회색으로 표시되어져 있다. 회색의 삼각 노드는 첫 번째 응용의 실행 경로상에서 중간 노드로 작용하지만 두 번째 응용에서는 SM이 직접 실행되는 노드이다.

4. 관련연구와 비교

SM은 programmable 관점에서는 Active network 기법인 Active message[3], Smart packet[7]과, 코드의

이동성 측면에서는 Mobile agent[6]와 접근방법을 공유하고 있다. Active message와 유사하게 네트워크 내의 임의의 노드에 코드를 포함한 메시지가 도착하면 코드는 데스크 형태로 실행된다. 그러나 Active message는 목적지에 대한 임의의 핸들러를 지정하는 방법인 반면에 SM은 메시지에 라우팅 기능을 실행할 코드를 포함한다. 또한 전자는 시스템 수준의 네트워크에서 빠른 통신을 지향하고 있는 반면에 후자는 네트워크 기반 대규모 임베디드 시스템 환경에서 다양한 응용의 원격 프로그래밍 능력 확보를 목표로 한다. SM은 제한된 전원의 절약과 동적인 네트워크 환경에서 QoR에 따른 부분적인 실행을 허락하는 점이 성능보다는 중요한 지표가 되기 때문이다.

Smart packet 구조는 코드 이동성 특성을 활용하여 네트워크를 관리하는 융통성 있는 수단을 제공한다. 이는 인터넷 특히 IP 헤더의 option 부분을 이용하여 구현된다. 따라서 라우팅은 인터넷의 라우팅 메커니즘에 따르며, Smart packet이 임의의 특정 목적지에 도착하면 실행된다. 이와는 달리 SM은

네트워크 내의 경로상에 위치하는 노드 모두에서 자신의 다음 경로를 능동적으로 선택한다. 또한 SM은 네트워크 내의 경로상에 노드에서 실행된 상태를 메시지 내에 포함하여 전달하는 고유 특성을 갖는다.

코드가 이동한다는 점에서 유사한 개념을 공유하는 Mobile agent는 노드와 노드 사이에 코드를 이동할 때 안전한 전송을 보장하는 네트워크 기반구조를 가정하고 있다. 반면에 SM은 네트워크 내에서 자신 스스로 라우팅 책임을 진다. 따라서 SM은 네트워크 내의 모든 노드들이 스스로 라우팅 가능한 구조를 제공하고 있음을 전제로 한다.

최근 이동통신의 범용화에 따라 새로운 네트워크 구조로써 Ad hoc 네트워크[2]에 대한 연구가 크게 진행되고 있다. 기존의 이동 네트워크는 라우팅을 임의로 하는 기반구조와 정보 생성 혹은 소비를 전제로 참여하는 호스트로 양분된 구조를 갖는다. Ad hoc 네트워크는 라우터 형태의 기반구조와 호스트가 결합된 구조로서 호스트는 라우터 역할을 포함한다. 따라서 기반구조가 약하거나 없을 경우, 심지어는 기반구조 사용료가 부담이 되는 경우에 독자적인 이동통신망을 구축하기 위한 유용한 수단으로 전망된다. 그러나 라우터인 호스트가 이동한다는 고유의 특성으로 인하여 기존에 사용되었던 정적인 라우팅 프로토콜을 그대로 사용할 수 없다. 또한 현재까지 많은 Ad hoc 라우팅 방법이 제안되어 왔지만 일반적으로 최적화된 단일 라우팅 프로토콜을 정의할 수도 없다. 즉 Ad hoc 네트워크는 현재 상태에 적절한 라우팅 프로토콜을 선택으로 적용하는 것이 타당하다. SM 기법은 Ad hoc 네트워크의 다양한 프로토콜을 선택적으로 사용할 수 있도록 해주는 통신 방법론으로 유용하게 활용될 수 있을 것으로 예상된다.

5. 결론

본고는 네트워크 기반 대규모 임베디드 시스템을 위한 상호협력 모델과 이를 효율적으로 지원하기 위한 Smart message 기법을 통신 방법론 관점에서 살펴보았다. 수천, 수만 개의 임베디드 시스템이 네트워크를 수단으로 연결된 환경은 노드들의 고장, 이동에 따른 의사적 통신단절, 그리고 제한된 전원에 따른 의도적 통신단절 등의 고유 특징적 요소를 갖는다. 이러한 환경에서 새로운 응용을 수용하기 위하여 모든 노드를 재설정하는 것은 거의 불가능할 것이다. 또한 다수의 노드들이 참여하는 분산 응용은 일부 노드 혹은 통신 고장으로 전체 응용이 실패라는 결과를 낳는다. 따라서 이러한 고유 특성들을 수용하기 위한 programmable 기법의 적용은 매우 유효 적절한 접근으로 보인다. Smart message는 네트워크 내의 임베디드 시스템 노드들은 분산 응용에 필요한 컴퓨팅 및 통신 자원을 제공함으로써 메시지를 통하여 전달되는 코드를 실행하여 융통성 있는 상호협력의 효과를 얻는다. Smart message 기법은 데이터의 수집, 배포, 내용기반 라우팅, 객체 추적 등의 응용에 다양한 융통성을 지원하는 통신 방법론으로 활용될 것으로 예상된다. 특히 Ad hoc 네트워크와 같이 다수의 라우팅 방법론에서 현재 상태에 최적의 방법론을 선택적으로 적용코자 할 경우에 획기적인 수단이 될 것으로 보인다.

참고문헌

- [1] C. Borcea, D. Iyer, P. Kang, A. Saxena, and L. Iftode, "Cooperative Computing for Distributed Embedded Systems," *Rutgers University Technical Report DCS-TR-464*, Nov. 2001
- [2] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of

Multi-Hop Wireless Ad Hoc Network Routing Protocols,” *Proc. of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 85-97, 1998

- [3] 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
- [4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next Century Challenges : Scalable Coordination in Sensor Networks,” *Proc. of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 263-270, 1999
- [5] C. Intanagonwiwat, R. Govindan, and A. Estrin, “Directed Diffusion : A Scalable and Robust Communication Paradigm for Sensors Networks,” *Proc. of the 6th annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000
- [6] D. Milojicic, W. LaForge, and D. Chauhan, “Mobile Objects and Agents,” *USENIX Conference on Object-oriented Technologies and Systems*, pp. 1-14, 1998
- [7] B. Schwartz, 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
- [8] P. Stanley-Marbell, C. Borcea, K. Nagaraja, and L. Iftode, “Smart Messages : A System Architecture for Large Networks of Embedded Systems,” *Rutgers University Technical Report DCS-TR-430*, Jan. 2001
- [9] P. Stanley-Marbell, and L. Iftode, “Scylla : A

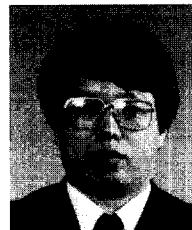
Smart Virtual Machine for Mobile Embedded Systems,” *Proc. of the Third IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 2000

저자약력



김문자

1984년 전남대학교 계산통계학과 졸업 (학사)
1986년 전남대학교 계산통계학과 졸업 (석사)
1985년-2000년 한국전자통신연구원 선임연구원
2000년-현재 전주공업대학 영상멀티미디어과 초빙
전임강사
관심분야: 데이터베이스, 이동컴퓨팅, 분산처리시스템
e-mail: mjkim@jtc.ac.kr



조인준

1982년 전남대학교 계산통계학과 졸업 (학사)
1985년 전남대학교 전자계산학과 대학원 졸업 (석사)
1999년 아주대학교 컴퓨터공학과 대학원 졸업 (박사)
1990년 정보처리 기술사 (전산 조직 응용)
1983년-1994년 한국전자통신연구원 선임연구원
1994년-현재 배재대학교 컴퓨터공학과 부교수
관심분야: 정보통신 보안, 컴퓨터 네트워크(이동컴퓨
팅), 전산조직응용
e-mail : injune@mail.paichai.ac.kr



조 기 환

1985년 전남대학교 계산통계학과 졸업 (학사)
1987년 서울대학교 계산통계학과 졸업 (석사)
1996년 영국 Newcastle 대학교 전산학과 졸업 (박사)
1987년-1997년 한국전자통신연구원 선임연구원
1997년-1999년 목포대학교 컴퓨터과학과 전임강사
1999년-현재 전북대학교 전자정보공학부 조교수
관심분야 : 이동컴퓨팅, 컴퓨터통신, 분산처리시스템
e-mail : ghcho@dcs.chonbuk.ac.kr