결함허용 모듈재구성 방식을 이용한 SDR 시스템

정용창° 윤성재 이병호
한양대학교 정보통신대학원
{sqrface° sjyoon bhrhee}@scann.hanyang.ac.kr

Fault-tolerant module reconfiguration for SDR system

Yong Chang Jung° Sung Jae Yoon Byung Ho Rhee
The Graduate School of information and communications, Hanyang University

요 약

본 논문은 다양한 환경변화에 적응하는 무선 통신 기술인 SDR시스템에서 소프트웨어 다운로드 과정을 마친 후 소프트웨어 모듈들을 사용자의 요구 또는 환경에 맞게 유연한 재구성을 할 수 있도록 설계하고 재구성 과정에서 발생할 수 있는 다양한 고장이나 결함을 탐지하여 그에 대한 안전한 복구 과정을 통해 전체 시스템의 신뢰성과 가용성을 향상시키는 결함허용 방식의 모듈재구성 메커니즘을 제안한다.

1. 서 론

SDR(소프트웨어 정의 무선통신:Software Defined Radio)기술은 차세대 이동통신으로서 초고속 데이터 전송이 가능하도록 하고 다양한 이동통신 무선접속 환경을 어떻게 하나로 통합하느냐 하는 것을 목적으로 꾸준히 연구되어오고 있으며 앞으로 다가올 무선 및 이동통신 환경에서 사용자의 요구에 의해서 부상하고 있는 기술이다. SDR기술은 다중 규격, 다중 대역, 다중 서비스 시대에 시스템의 통합 해결책을 제시해 줄 수 있는 차세대 기반 기술로 고려되어 3G 시스템의 업그레이드 및 4G 시스템의 접근으로 이해되고 있을 뿐만아니라 단순히 무선 통신 기술의 발전이 아닌 컴퓨터, 반도체, DSP 기술 등의 여러 가지 시스템 기술이 요구되는 기술의 융합 및 조화를 필요로 하는 기술이다. 개방 구조를 바탕으로 설계된 공통 하드웨어 시스템에 객체지향 응용소프트웨어의 다운로드로 다중모드, 다중 규격 변경이 가능하도록 구성된 단일 단말기는 모든 다양한 서비스 및 양질의 서비스 제공이 가능하여야한다.[1]

서비스 플랫폼의 교체 없이 사용자 요구에 따라 신속하게 서비스 변경이 가능하도록 하는 객체지향 구조 미들웨어 기술을 기반으로 시스템 모듈 변경이 가능한 모듈간 정보/제어 인터페이스 정의가 필요하다. 시스템 기능 모듈들은 독립적으로 동작할 수 있도록 각 모듈들을 라이브러리화하여 개방 인터페이스에 의해 연결하고 각 모듈에 특정 동작을 할당하기 위한 소프트웨어 동작으로 구성된다. 다운로드에 의해 현장에서 실시간 재구성이 가능한 유연한 통신 장치를 제공할 수 있는 SDR 기반 시스템은 이동 통신 기술의 디지털화 증대와 컴퓨터 통신의 무선화에 따라 이동통신은 점점 더 컴퓨터와 같이 하드웨어에 의존적이지 않고 소프트웨어에 의한 유연한 적용이 가능한 특징을 가지게 된다. 그러므로 주변 환경에 유연하게 적용되는 SDR 시스템을 위해 요구된 모듈들을 다운로드하고 효율적인 재구성 및 관리에 대한 연구가 필요하다.

이에 따라 본 논문에서는 SDR 시스템의 구조와 다운로드과정, 다운로드 된 모듈들을 재구성하는 과정 및 그 과정에서 발생할 수 있는 결함을 알아보고 결함허용 방식을 이용하여 결함발생시에도 고장 없이 복구하여 정상적으로 재구성과정을 수행할 수 있도록 하는 방법에 대해 제안하였다.

2. SDR 시스템

2.1 SDR 시스템의 구조

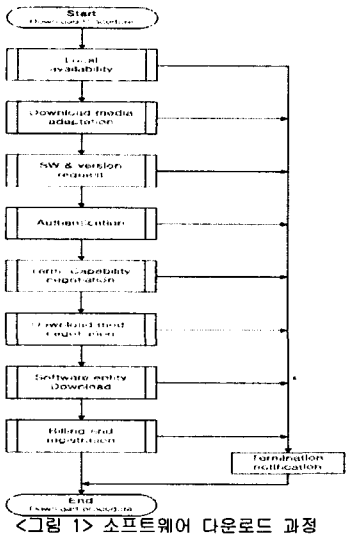
시스템 동작을 위해 필요한 소프트웨어를 응용 소프트웨어(software application)라고 정의하며 개방 인터페이스로 독립된 기능의 서브시스템들을 연결하는 구조로 되어있고 각 서브시스템은 시스템의 동작 상태에 의해 결정될 것이다. 응용소프트웨어는 Smart Card(SIM Card), OTA(Over-the-Air) 등의 다운로드를 이용하여 모듈간 인터페이스와 파라미터의 변경 및 추가가 가능할 것이다. 서로 다른 서비스 플랫폼 상에 응용소프트웨어 다운로드를 위한 미들웨어 기술의 적용은 필수적으로 요구된다.

SDR 시스템은 소프트웨어의 재구성성이 요구되므로 다양한 하드웨어를 추상화시키기 위해 RTOS(Real-Time Operating Systems, 실시간 운영체제)가 제공하고 응용이 재구성될 수 있도록 컴포넌트 기반 컴퓨팅을 지원하는 것은 응용소프트웨어와 RTOS 사이에 위치하는 중간 계층인 내장형 미들웨어가 제공

하게 된다.[2]

2.2 소프트웨어 다운로드(Software Download)

SDR 시스템은 대부분의 중요한 기능들이 소프트웨어로 구현되어 다양한 통신 방식이나 부가 서비스를 제공한다. 따라서 SDR 기술을 적용한 단말이나 기지국 장비의 기능을 변화하기 위해 새로운 프로그램들을 적용하는 소프트웨어 다운로드 기능은 SDR의 성공적인 전개를 위해 매우 필요하다.



<그림 1> 소프트웨어 다운로드 과정

이러한 소프트웨어 다운로드를 통해 새로운 사용자 Application 및 프로토콜 스택 뿐 아니라 물리 계층의 모뎀 기능 등의 다운로드가 가능하다.

소프트웨어 다운로드를 위해 사용자의 이동 단말과 기지국은 약간 다른 접근 방법을 취하게 된다. 일반적으로 기지국에서는 새로운 소프트웨어 버전으로 업그레이드 해야 하는 경우에만 소프트웨어 다운로드가 수행되며 이는 소프트웨어를 관리하는 서버로부터 유선 네트워크를 통해 용이하게 이루어 질 수 있다. 반면에 사용자의 단말에서는 이동 및 사용자 요구의 변화(예를 들어 새로운 Application Transaction의 요구, 무선 접속 변화 등)에 의해 잦은 소프트웨어의 변경이 필요하며, 이 경우 소프트웨어 다운로드는 앞에서 언급된 바와 같이 Smart Card, 유선 및 무선(OTA) 네트워크 등의 방법을 통해 수행될 수 있다.[3]

3. 결함허용 모듈 재구성을 이용한 SDR 시스템

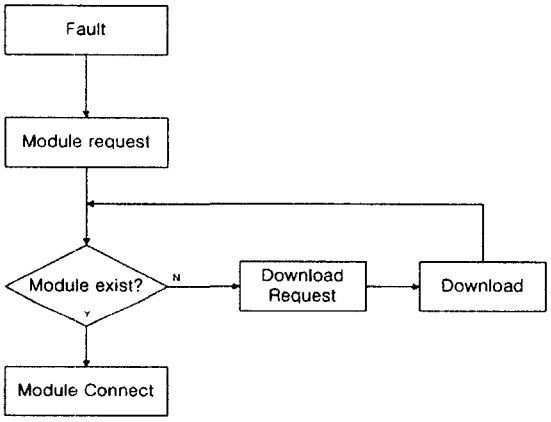
3.1 모듈 재구성

사용 환경 변화 또는 시스템의 요구에 따라서 모듈들을 다운로드하거나 사용 모듈의 재구성이 필요하게 되는데, 보유하고 있는 모듈로 재구성이 가능한지 여부를 판단하여 각각의 콤포넌트나 상위로부터 메시지로 요청된 모듈들은 모듈관리자에 의해 Load, Remove, Update 과정을 통해 재구성이 이루어지고 구성에 대한 응답을 하게 된다.[4]

3.2 재구성간 fault 발생시 문제점

다양한 무선환경 속에서는 고려되지 않은 원인으로 수많은 경우의 오류나 고장이 발생하게 하는 요소들이 있고 이를 모두 배제할 수 는 없다. 이와 같이 시스템에 치명적인 fault가 발생하게 되면 이를 감지해내서 가능한한 빨리 복구하여 정상적인 임무를 수행하느냐가 매우 중요한 문제이다.[5]

fault는 논리적 오류나 에러와 같은 소프트웨어적인 결함과 시스템 보드 고장이나 불안정한 상태와 같은 하드웨어적인 결함을 들 수 있는데 어느 경우든 <그림2>과 같이 fault가 발생하여 감지하게 되면 각 컴포넌트나 상위의 메세지에 따라 모듈들의 재요청이 발생하게 된다. 결함 발생 후에는 이미 모듈 관리자의 모듈을 신뢰할 수 없으므로 다운로드를 재 수행해야하는 경우도 예상된다.



<그림 2>결함 발생후 모듈 재연결 과정

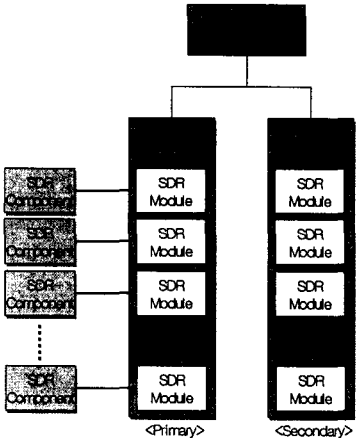
fault발생 후 복구되는 시간을 고려해 볼때, fault 발생으로 인해 요청하는 컴포넌트의 수를 C, 한개의 요청으로 한번의 재 다운로드를 거쳐 모듈을 연결하게 되는 최대 복구시간을 T 라고 가정했을때, 최대 총 복구시간은 CT가 된다. 그러나 이 최대 총 복구 시간은 빈번한 fault가 예상되는 경우, 즉 fault가 발생하고 난 후 다시 fault가 발생하게 되는 간격이 더 짧아지는 경우가 발생하게 된다면 결국 요청의 반복이 이루어지게 되고 다운로드 과정 또한 반복하게 되어 복구하는데 필요한 시간이 더욱 커지게 되며 복구는 더욱 어려워지게 된다.

3.2 결함허용 모듈재구성 방식

<표1> 모듈관리 정보테이블 구성 예

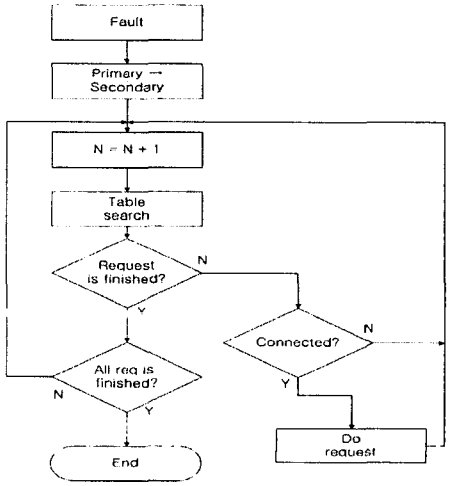
No.	component	version	connect	recent req	req finished
1	RF	1.0	yes	load	yes
2	Antenna	1.5	no	update	no
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

결함허용 모듈재구성 방식에서의 모듈관리자는 <표 1>과 같이 모듈관리 정보에 대한 테이블을 갖게 되고 다운로드나 요청 수행할 때마다 갱신하고 secondary 모듈 관리자로 복제하게 된다. 미들웨어에서 제공하는 FWC(Fault Watch Component, 결함 감시 컴포넌트)로부터 결함이 감지되면 각 컴포넌트가 사용중인 모듈과 진행중인 재구성 단계에 관한 정보를 가지고 있는 primary 모듈 관리자가 미리 복제된 secondary 모듈 관리자로 즉시 전환되어 바로 수행할 수 있게 된다. 이때 모듈 관리자가 관리하는 모듈들을 함께 복제를 하게 되어 같은 프로세스를 수행을 하고 있기 때문에 모듈저장소가 따로 필요하지 않게 되고 보관중인 모듈들에 대한 신뢰성도 높아지게 된다<그림 3>.



<그림 3> Warm Standby로 대기하는 secondary 모듈 관리자

FWC로부터 결함에 대한 감지를 통한 메시지를 받아 primary 모듈관리자에서 secondary 모듈관리자로 전환하는 과정<그림 4>를 살펴보면, fault 즉 시스템의 모듈 재구성 과정중에 발생할 수 있는 모든 경우의 결함이



<그림 4> Secondary manager로의 전환 과정

FWC에 의해 탐지되어 메시지가 전달되면 primary 모듈 관리자가 가지고 있던 정보들과 함께 수행하던 프로세스 전체를 secondary 모듈관리자로 넘기게 된다. 이때 모듈관리 정보 테이블을 검색하여 모든 컴포넌트의 각 모듈들이 요청수행이 완료 되었는지에 대한 여부를 가려낸다. 다음은 완료되지 않은 모듈에 대해서는 연결여부를 확인하고 연결되어있던 모듈에 대해서만 그 요청에 대해 수행해주게 된다. 이때 요청을 수행해주면서 바뀌게 되는 모듈정보 테이블의 내용은 그때마다 갱신해주어야 한다. 모듈정보 테이블 상의 모든 모듈들의 요청수행이 끝나면 전환 과정이 종료되면서 바로 전환된 secondary가 primary로 primary는 secondary로 그 임무가 바뀌게 되고 정상적인 프로세스는 계속 진행되게 된다. 만일 이 과정의 중간에서 fault가 발생하게 된다고 하더라도 그 상태에서 전환과정을 다시 수행하게 되면 기존의 요청에 대한 재 다운로드과정이 없기 때문에 복구시간에는 큰 차이가 없게 된다.

4.결론

본 논문은 SDR 시스템의 모듈재구성과정에서의 치명적 오류나 시스템 불안정으로 인한 결함이 발생했을때에 대한 문제점을 제시하고 이에 대한 시간적 손실을 고려하여 재구성을 수행하는데 끊김없이 가동될 수 있도록 모듈관리자를 복제하여 전환과정을 거치는 결함허용 모듈재구성방식을 이용한 SDR 시스템 설계를 제안하게 되었고 전체 시스템의 신뢰성과 가용성이 증가하게 되는데 그 의미를 지니고 있다.

향후 과제로는 결함허용 방식이 갖게 되는 문제점 중 하나인 가용량 증가에 따르는 비용문제, 메모리문제 등에 대한 보완이 이루어 지고 모듈재구성 과정을 포함하여 미들웨어, 다운로드 과정 또한 연계된 결함허용 방식이 적용되어야 하며 구성된 하드웨어 또한 결함허용 기술이 적용되어야 할 것이다. 앞으로 SDR시스템의 원활한 다운로드의 구현과 모듈재구성이 가능하게 됨과 동시에 최적의 하드웨어의 개발이 이루어지고 보다 안정적이고 신뢰성있는 시스템을 개발하는데 필요한 기초자료가 될 수 있을 것으로 보인다.

참고문헌

[1] Software Defined Radio (SDR) Forum, http://www.sdrforum.org.

[2] 홍성수, SDR을 위한 RTOS와 내장형 미들웨어의 설계, 한국통신학회지, 제19권, 11호, 2002.

[3] K Moessner, R Tafazolli, 3G mobile Communication Technologies, Conference publication No 471, IEE 2000

[4] Chia-ching Lin, Hung-Lin Chou, Chin-Lien Chiu, Min-Chiao Wang and Shiao-Li Tsao, "Design of a SDR Software Framework", SDR Forum document SDRF-02-I-0019-V0.00.

[5] Barry W.Johnson, Design and analysis of Fault-Tolerant Digital Systems, Univ. of Virginia, Charlottesville, 1989.