JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 18, No. 12 : 2891~2898 Dec. 2014

이중화 IMA 시스템의 입력 데이터 동기화 방안

박홍열[1] · 김기일[2*]

Input Data Synchronization Scheme Based on Redundancy for IMA System

Hong-youl Park[1] · Ki-Il Kim[2*]

[1]Avionics S/W Team, Avionics System Department, Korea Aerospace Industries, Sacheon 664-710, Korea
[2*]Department of Informatics, Engineering Research Institute, Gyeongsang National University, Jinju 660-701, Korea

요 약

항공기용 Integrated Modular Avionics (IMA) 시스템의 경우 모듈 단위 이중화를 통하여 결함 감내 기능을 구현할 수 있다. 하지만, 이중화 구성 시 반드시 요구되는 소프트웨어 동기화는 하드웨어의 비동기적인 특성으로 인해 실제 구현 시 높은 복잡도를 야기한다. 이러한 문제점을 해결하기 위하여 현재 IMA시스템에서의 PALS(Physically Asynchronous Logically Synchronous) 디자인 패턴이 제안되었으나 실제 시스템 적용시 각 시스템의 특성에 따른 변화가 불가피하다. 본 연구는 PALS 디자인 패턴을 참조하여 Primary /Secondary 이중화를 이용하는 IMA 시스템에서의 입력데이터 동기화 설계방안을 연구하였다. 제안된 방식은 Rate Monotonic Scheduling (RMS) 방식을 고려하여 프레임 윈도우에 동기된 기법을 제안하고 있으며 시스템에 알맞은 동기화 시간을 분석하고 제안한다. 마지막으로 실험 및 분석을 통하여 제안된 방법의 타당성을 검증하였다.

ABSTRACT

It is feasible to develop a fault tolerant system through module level redundancy on the Integrated Modular Avionics (IMA). However, its great implementation complexity is one of important challenges when asynchronous hardware environment is naturally assumed. To solve this problem, Physically Asynchronous Logically Synchronous (PALS) on IMA has been proposed. But, it has adaptation problem by not addressing specific architecture for IMA system. In the paper, we propose how to synchronize the input data on the IMA system under primary/secondary redundancy architecture by referring to existing PALS. In the proposed scheme, we introduce window frame by considering rate monotonic scheduling and analyze the adequate the synchronization time. Finally, we verify the feasibility of the proposed design pattern through the systematic experiments.

키워드 : 동기화, IMA, PALS, 비율단조스케줄링, Primary/Secondary

Key word : Synchronization, IMA, PALS, RMS, Primary/Secondary

접수일자 : 2014. 05. 22 심사완료일자 : 2014. 11. 27 게재확정일자 : 2014. 12. 08

* **Corresponding Author** Ki-Il Kim (E-mail:kikim@gnu.ac.kr, Tel:+82-55-772-1373)
Department of Informatics, Engineering Research Institue, Gyeongsang National University, Jinju, 660-701, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2014.18.12.2891 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

IMA 아키텍처는 나날이 증가하고 있는 항공전자 요구기능을 효율적으로 구현하고 항공기의 운용과 유지보수비용을 획기적으로 줄일 수 있는 최신의 항공전자 시스템 아키텍처이다[1, 2]. 고성능의 프로세서와 신뢰성있는 데이터 네트워크, 시간과 공간의 독립적인 파티션을 지원하는 운영체제의 개발로 다양한 항공전자 기능을 통합할 수 있으며, 주요 기능의 경우 이중화, 분산배치 등의 시스템 설계로 하드웨어 결함에 대비하고 신뢰성을 향상시킨다.

대부분의 IMA시스템에서 각 노드상의 프로세서는 고유의 클럭에 의해 구동된다. 각각의 프로세서가 동일한 주기로 동작하지만 고유의 오프셋, 드리프트, 지터에 의해 서로 비동기적으로 동작하며 이러한 하드웨어 구조를 Globally Asynchronous/Locally Synchronous (GALS) 라 한다.

이러한 GLAS 기반의 시스템에서의 결함 감내를 위해 하드웨어적 접근과 더불어 소프트웨어적 접근도 필요하다. 즉, 시스템이 이중으로 구성될 때 각 소프트웨어 컴포넌트간 상태 동기화가 중요하며 비동기적 환경에서의 동기화는 구현과 검증에 있어 많은 어려움이 있다. 특히 race 상황과 데드락 상황을 고려한 구현시 보다 세심한 주의가 필요하다[3, 4].

이러한 비동기적 하드웨어상에서 소프트웨어 동기화를 쉽게 제공할 수 있는 디자인 패턴으로 Physically Asynchronous Logically Synchronous (PALS)가 제안되었다. PALS는 비동기적 환경에서 몇 가지 제약조건을 충족할 경우 적용될 수 있으며 복잡도를 현저히 줄일 수 있는 방법으로 검증되었다[5]. 하지만, 이러한 PALS 패턴의 경우 일반적인 IMA 구조에서의 동작 원리만을 정의하였기 때문에 이를 실제 적용하기 위해서는 각 시스템의 구성 및 응용프로그램의 특징을 반영하는 것이 바람직하다.

본 논문은 IMA환경에 PALS 적용한 연구를 참조하여 IMA를 위한 이중화 구성의 형태가 Primary/Secondary 구조인 경우 두 시스템의 입력 데이터 동기화가를 위한 방안을 제안하고자 한다. 제안 시스템의 경우 스케줄링 방식으로 RMS를 사용하고 있으므로 이를 반영한 시스템 구성 방안에 대한 분석 및 검증이 필요하다. 이를 위하여 기존으 PALS 접근 방법을 기반으로 제안 방법에 대한 동기화 기법을 구성하고 실험을 통하여 제안 방법의 타당성을 검정하고 그 결과를 제시하고자 한다.

본 논문은 다음과 같이 구성된다. 연구의 목적 및 배경을 설명한 서론에 이어 II장에서는 IMA구조상에서 동기화를 위하여 제안된 PALS동기화 기법에 대해 알아본다. III장에서는 동기화를 위한 시스템 구조 및 이를 위한 제안된 방안을 설명한다. IV장에서는 실험결과를 제시한다. 마지막으로 V장에서는 결론을 맺는다.

Ⅱ. 관련 연구

비동기적 하드웨어 환경에서의 동기화는 race와 데드락 상황을 고려하여 그 구현과 검증에 있어 많은 어려움이 발생하고 있다. 이러한 설계의 복잡도를 해결하기 위해 제안된 것이 바로 PALS디자인 패턴이다. 또한, 이를 확장하여 향후 항공전자 시스템인 IMA에 적용하기 위한 방안도 제안되었다. 본 절에서는 제안 메커니즘이 기반하고 있는 해당 기술을 간략히 살펴본다.

2.1. PALS

GALS 하드웨어 환경에서 PALS 디자인 패턴을 적용 하여 동기화를 구현 할 수 있다. PALS는 내고장성이 강한 실시간 네트워크를 이용하여 구현된다. 또한, 메시지의 전송을 정확히 예측할 수는 없으나 한계구간 이내에 보장되는 환경에 적용이 가능하다. 그림 1은 PALS의 기본원리를 보여준다.

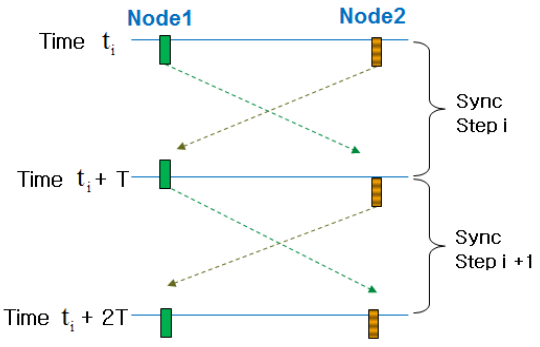

그림 1. PALS 기본원리
Fig. 1 PALS OverView

그림 1에서 볼 수 있듯이 각 노드의 동기화 로직은 T 구간마다 반복적으로 수행되며, T는 각 노드에 따라 결정된다. 기본 전제는 임의의 i 스탭에서 생성된 메시지는 목적지 노드의 i+1 스텝에서 동기화를 위하여 사용된다.

PALS는 두 노드의 동기화 시 개입될 수 있는 변이(클럭, 계산시간, 메시지 전송시간)값을 특정 범위(Bounded Range)로 한정 할 수 있을 때 적용이 가능하며 아래와 같이 제한된 클럭오차와 제한된 계산시간, 제한된 메시지 전송시간을 만족하도록 요구하고 있다.

- 제한된 클럭오차 : 모든 노드 j에서의 클럭 c_j 는 절대 시간 t 와 ϵ이내의 오차를 가져야 하며 이는 다음과 같이 표현된다. $|c_j - t| < \epsilon$
- 제한된 계산시간 : 노드의 다음상태를 위한 계산시간은 α이고 α는 특정구간 이내로 크기가 결정되어야 하므로 값을 범위는 $\alpha_{min} \le \alpha \le \alpha_{max}$이다.
- 제한된 메시지 전송시간 : 메시지 목적지에 μ시간 이내에 도달해야 하며 μ는 특정구간 이내로 크기가 결정되므로 이 조건은 다음의 수식으로 $\mu_{min} \le \mu \le \mu_{max}$ 정의된다.

스탭 i에서 생성된 메시지가 목적지 노드의 i+1에서 동기화되기 위해 송신노드의 최소 처리시간 H와 최소 동기화 주기 T에 대해 아래의 제약조건이 충족되어야 한다.

1) 스탭 i에서 생성된 메시지는 목적지의 스탭 i에서 사용될 수 없다. 수신 노드의 스탭 i는 늦어도 $t_i+\epsilon$에 수행되므로, $t_i+\epsilon \le t_i+H-\epsilon+\mu_{min}$이 만족되어야 한다. 이는 다음 (식 1)로 표현된다.

$$H \ge \max(2\epsilon-\mu_{min}, 0) \quad \text{(식 1)}$$

2) 스탭i에서 생성된 메시지는 수신측의 스탭 i+1에 사용될수 있도록 도착하여야 한다. 수신측의 i+1 연산은 빨라도 $t_i+T-\epsilon$에 수행되고 송신측 메시지는 늦어도 $t_i+H+\epsilon+\mu_{max}$에 도착해야 하므로 $t_i+T-\epsilon \ge t_i+H+\epsilon+\mu_{max}$가 성립되고 메시지 전송전 계산이 완료되어야 하므로 다음 (식 2)가 만족되어야 한다.

$$T \ge \mu_{max}+2\epsilon+\max(\alpha_{max}, 2\epsilon-\mu_{min}) \quad \text{(식 2)}$$

2.2. IMA상의 PALS적용

IMA시스템은 ARINC-653 운영체제와 같은 분할된 실시간 운영체제를 사용하여 많은 비행운용프로그램을 통합하며 시스템에서 각 파티션 스케줄링은 윈도우 프레임이 고정된 형태로 수행된다. 따라서, 파티션 스케줄링의 특성을 고려하여 Active/Standby와 같은 동기화 시스템의 상호간의 상태결정을 위해 일반적으로 PALS를 적용할 수 있다.

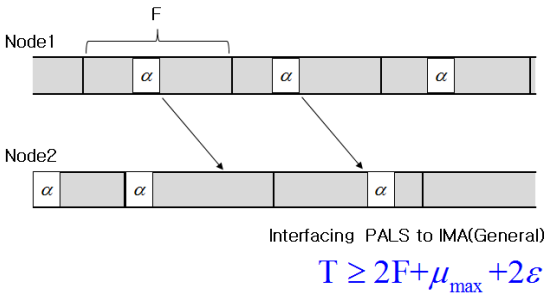

그림 2. IMA상의 PALS적용
Fig. 2 Interfacing Pals to IMA

그림 2와 같이 윈도우 프레임이 스케줄링 될 때 동기화 연산 α가 최악의 경우 두 개의 윈도우 프레임에 한번 수행 될 수 있으며, 이를 고려하여 PALS의 T제약 조건을 $T \ge \mu_{max}+2\epsilon+\max(2F, 2\epsilon-\mu_{min})$로 조정 할 수 있고, 일반적으로 2F가 $2\epsilon-\mu_{min}$큰 값이므로 T 값은 아래와 같이 정리된다.

$$T \ge 2F+\mu_{max}+2\epsilon \quad \text{(식 3)}$$

동기화 기능을 PALS 스탭이 아닌 IMA 윈도우 프레임 기준으로 수행됨에 따라 수신측의 스탭이 증가하지 않은 상황에 동기화 메시지가 도착할 수 있기 때문에 이러한 해결을 위해 송신측은 T시간 이상일 경우 첫 프레임에 동기화 로직을 수행하고 수신측은 버퍼를 운영하여 스탭 i에 도착한 동기화 메시지를 버퍼링하여 i+1 스탭에서 사용가능하도록 구성하여야한다.

Ⅲ. 제안 메커니즘

앞에서 살펴본 봐와 같이 PALS 시스템을 IMA에 적용하는 경우 Active/Standby와 같은 동기화 시스템의 상호간의 상태결정에 적용할 수 있다. 하지만, IMA 시스템에서의 동기화 방식은 여러 가지 형태 및 구조를 가지게 되므로 제안되는 시스템에 따른 새로운 메커니즘의 개발이 요구된다.

본 논문에서는 IMA 시스템에서의 입력 데이터 동기화 방안을 기존의 PALS 패턴을 참조하여 제안하고자 한다. 기존 시스템과는 달리 제안 방법은 IMA 시스템 상에서 Primary/Secondary 이중화 개념을 사용하는 동시에 운영체제의 스케줄링 방식을 고려한다. 구체적으로 제안 방법은 IMA환경에 적용된 PALS를 참조하여 RMS 환경에서의 Primary/Secondary 이중화 구성이 필요한 IMA시스템에 적용가능하고 응용프로그램의 복잡도를 감소시킬 수 있는 입력데이터 동기화 디자인 방안을 제안하고자 한다. 제안 시스템 구성 방법 및 응용 프로그램의 특징은 다음과 같다.

3.1. Primary/Secondary 형태의 이중화 시스템 구성

Primary/Secondary 형태의 이중화를 도입하는 경우 입력데이터의 동기화가 매우 중요하다. 이를 위하여 기존에 제안된 응용 프로그램 수준에서의 Hand Shaking을 통한 입력 이벤트의 동기화는 응용프로그램의 복잡도를 증가시키는 단점이 있다. 따라서, 이를 위한 새로운 메커니즘의 개발이 요구되고 있다. 그림 3은 본 연구에서 사용하고 있는 Primary/Secondary 형태의 이중화 시스템의 구조와 동작 방안을 간략하게 보여준다.

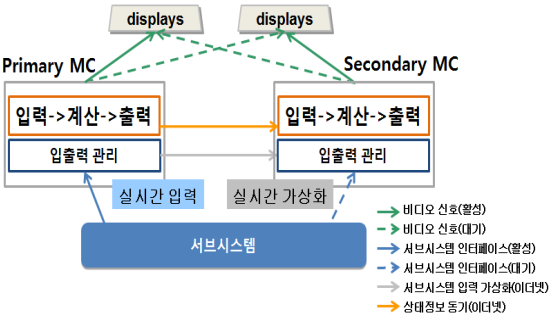

그림 3. Primary/Secondary Redundancy Concept
Fig. 3 Primary/Secondary 이중화 개념

그림 3의 시스템 구조의 경우 동일한 시간에 동일한 입력과 처리를 수행하고 제어권을 가진 결과만을 서브시스템으로 출력하여 Primary와 Secondary의 내부 상태를 동일하게 유지시켜 Backup 상황에 대처할 수 있다.

이 경우 1553 MUX BUS의 운영과 키버튼 처리를 위한 Serial 입력데이터의 운영 등은 인터페이스의 특성으로 제어권을 가진 Primary에서만 통신을 수행 할 수 있고 이런 데이터는 Primary측의 입력데이터를 실시간으로 Secondary로 동기화시켜 운영 할 수 있다. 하지만, 키버튼 처리를 위한 Serial 입력의 경우 Primary와 Secondary가 동시에 같은 입력으로 인식하지 못하는 문제가 발생 할 경우 현재 화면의 페이지 구성과 같은 시스템 내부의 상태 값이 불일치 하는 상황이 발생 할 수 있다. 이러한 문제를 해결 하기위해 어플리케이션 수준에서 입력값 확인을 위한 설계가 추가된다. 만약 어플리케이션이 전달받는 Primary와 Secondary의 입력데이터가 동기화 되어 있을 경우 어플리케이션 수준의 추가 상태확인은 불필요하므로 시스템의 복잡도는 감소 할 수 있다.

이러한 분석에 의하여 본 연구에서는 Primary/Secondary 이중화 구조를 가지며 RMS가 적용된 IMA 시스템에 적용 가능한 디자인 패턴을 제시하고 그 유용성을 확인해 보고자 한다.

3.2. IMA 시스템의 스케줄링 알고리즘 적용

그림 4는 ARINC-653 운영체제상의 RMS를 적용 할 경우 스케줄링 계층 구조를 보여준다.

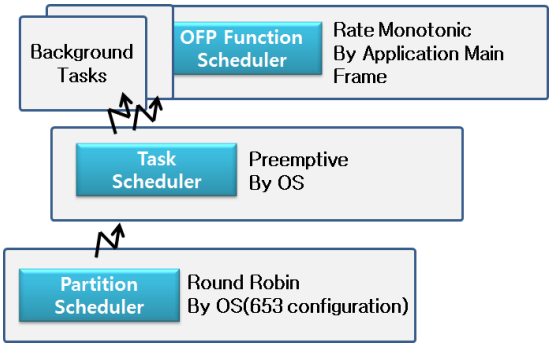

그림 4. OFP 스케줄링 계층
Fig. 4 OFP Scheduling Hierarchy

해당 구조에서 ARINC-653 파티션스케줄링에 의해 고정적으로 구성된 윈도우 major frame에 따라 스케줄이되며 프로세싱을 할당받은 파티션 윈도우 내부에서는 선점형 태스크 스케줄링이 제공된다. 선택된 태스크 중 비행운영프로그램의 주요기능을 수행하는 태스크는 내부에 RMS가 구성되어 있고, I/O, 그래픽 등 백그라운드 형태로 수행되어야 할 태스크들과 멀티태스킹 방식으로 운영된다.

3.3. 제안메커니즘

본 절에서는 IMA 시스템에서 앞서 설명한 이중화 구조 및 스케줄링 특성을 고려한 입력 데이터 동기화 방안을 제안한다. 제안 메커니즘의 핵심은 IMA환경에서 파티션 윈도우가 구성될 경우 해당 파티션이 수행되고 다음 수행시간을 할당 받을 때까지 유휴시간을 가지는 파티션 스케줄링의 단속적 특성을 활용하는데 있다. 그림 5는 파티션 스케줄링을 위한 윈도우 구성을 위한 예를 보여준다.

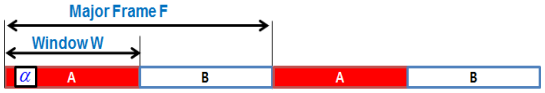

그림 5. 파티션 윈도우 구성
Fig. 5 Partition Window Configuration

이러한 환경에서 동기화를 위한 위한 방안의 핵심은 동기화를 위한 구간 T의 값을 구하는 것이다. 그림 5에서 윈도우 A가 수행된 후 다시 윈도우 A가 수행되기 위해 일정한 시간을 보장받는다. 이 경우 윈도우 A가 생성한 동기화 메시지를 전송 한 후 다음 파티션 윈도우로 스케줄링이 되기까지의 시간이 충분 할 경우 즉, (식 3)에서 감안했던 $\mu_{\max}+2\epsilon$의 시간보다 F-W의 시간이 충분 할 경우 아래와 같이 동기화 시간 T를 (식 4)와 같이 간략화 할 수 있다.

$$T \geq 2F \qquad \text{(식 4)}$$

그림 6은 기존연구와 F-W의 효과를 감안하고 있는 본 연구의 동기화 시간을 비교하여 보여준다. 기존의 방법에 비하여 F-W구간이 고려되면서 $\alpha_{\max}$에 대해 2F를 가정한 기존 연구의 값이 제안 시스템에는 적합하지 않으며 제안된 방법만으로도 충분히 동기화가 가능함을 보여주고 있다.

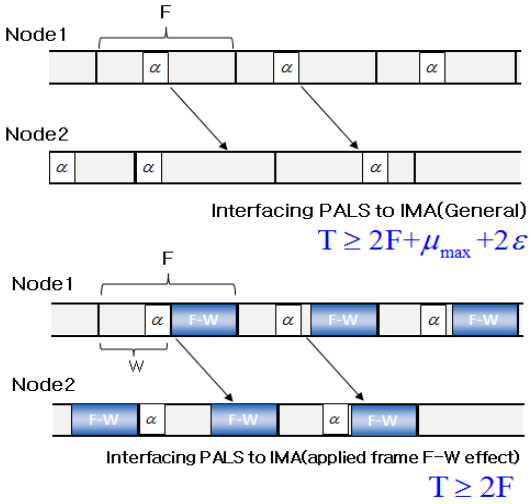

그림 6. Primary/Secondary를 위한 PALS IMA적용
Fig. 6 PALS to IMA for Primary/Secondary

제안된 방법을 적용하는 경우 새롭게 제시된 T에 대해 RMS적용을 위해 동기화 시간 T=2F로 고정하여 운영가능하며, 이는 50Hz 기본주기를 가지는 시스템의 경우 제안된 PALS 패턴은 25Hz 이하주기의 입력데이터를 동기화 할 수 있음을 의미한다. 즉, 기존의 연구에 대해 RMS를 적용 할 경우 F가 $\mu_{\max}+2\epsilon$보다 큰 동시에 비율단조의 특성으로 T=4F로 적용 할 수 있기 때문에 이는 결국 12.5Hz 이하 주기의 입력데이터에 대해 동기화를 수행 할 수 있음을 알 수 있다. 항공전자 시스템의 특성상 가장 높은 기본주기(50Hz)를 가지는 데이터는 실시간 계산데이터가 대부분이며, 시스템 내부의 운영상태에 영향을 줄 수 있는 키버튼 입력과 같은 조종사 입력데이터는 Low Rate로 설계됨을 감안할 경우 Low Rate의 전 영역을 처리 할 수 있는 제안된 설계방식의 유용성을 확인 할 수 있다.

3.4. 제안 메커니즘을 통한 입력 데이터 동기화 방안

제안된 메커니즘을 적용하여 입력 데이터를 동기화하기 위한 설계 시 그림 7과 같이 Primary와 Secondary 시스템에 각각 동기화 레이어를 구성하고 양쪽의 응용프로그램을 위한 동일한 입력데이터를 전송하게 된다.

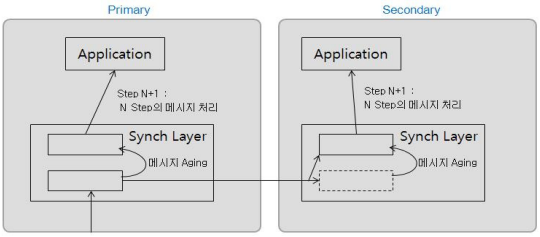

그림 7. 입력데이터 동기화 구조
Fig. 7 Input data synchronization architecture

Primary에서의 실제 메시지 입력에 대해 동기화 레이어는 입력을 처리하고 Secondary로 동일 메시지를 전송한다. 이때의 스케줄링 스탭은 i에 해당되며 Secondary로의 메시지 전송 시 태그에 i를 표시하여 전송한다. Primary는 스탭 i에 수신된 메시지를 다음 스탭에 사용하도록 한다. Secondary는 스케줄링 스탭 i+1에서 수신받는 동기화 메시지의 경우 Primary의 i스탭 혹은 i+1스탭의 메시지를 기대 할 수 있으므로 tag를 확인하여 i+1스탭의 메시지 일 경우 버퍼에 저장된 i스탭의 메시지를 어플리케이션에게 전달하고 다음 스탭을 위해 버퍼링한다. 만약 수신된 메시지가 i스탭일 경우 이 메시지를 직접 상위 어플리케이션에게 전달한다. 결국 Primary와 Secondary의 어플리케이션은 i+1 스케줄링 스탭에 i스탭의 입력메시지를 동일하게 처리 할 수 있다.

Ⅳ. 실험결과

제안한 동기화 기법의 검증을 위하여 IMA의 파티션 윈도우 스케쥴링 알고리즘을 모사하였고 실시간 운영체제 VxWorks가 탑재된 임무컴퓨터 두 대를 활용하여 Primary/Secondary 이중화를 구성하였다. 파티션 스케줄링을 모사하기 위해 동일 우선순위의 태스크 A와 B를 운영하였고, 라운드로빈 방식으로 스케줄링하였다.

그림 8은 실험에 사용된 임무컴퓨터의 구성과 모의된 스케줄링 윈도우의 개념을 보여준다. 실험을 위하여 20msec마다의 하드웨어 클럭을 이용하여 major frame을 형성하였고 태스크 A와 B 내부는 별도의 RMS를 구성한 상태에서 25Hz로 스케줄링 되는 기능을 활용하여 동기화 여부를 확인하였다.

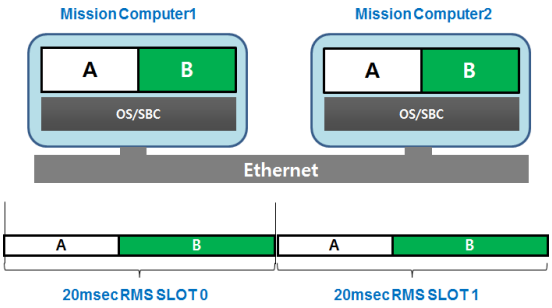

그림 8. 동기화 환경
Fig. 8 Synchronization Environment

2F(40msec)마다 증가하는 태크를 이용하여 상대와 동기화 여부를 확인하였으며, 기준 노드에서 tag값을 -1로 전송하여 PALS동기화를 시작시키고 버퍼와 실시간 수신데이터를 확인하여 태그값을 비교하여 (Received_tag+1 = Current_tag)의 관계가 유지되는지 확인하였다. 그림 9는 동기화 여부를 확인하기 위한 태그값의 변경 시나리오를 보여준다.

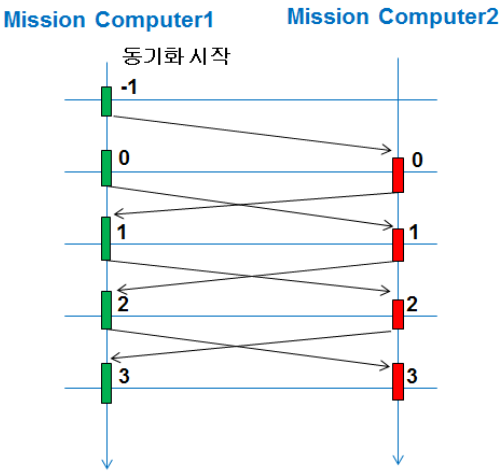

그림 9. 동기화 검증
Fig. 9 Synchronization Check

그림 9는 태스크 A와 B의 윈도우 사이즈를 변경하면서 태스크 A의 동기화 여부를 확인 결과를 보여주며 ARINC-653의 윈도 사이즈 변경과 동일한 효과를 위해 태스크에 부하를 걸어 실제 수행시간에 변화를 주었다. 그림 10은 태스크 실행시간의 변경을 WindRiver System Viewer로 확인한 결과이다.

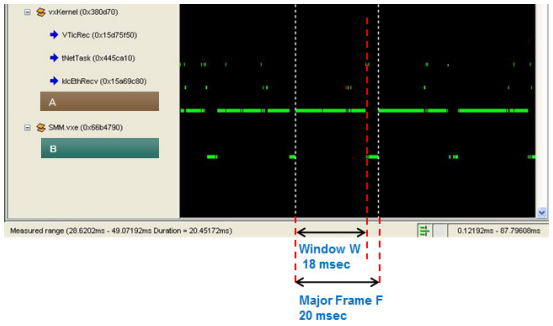

그림 10. 파티션 윈도우 크기 측정
Fig. 10 Window Size Measurement

앞에서 설명한 실험모델에 따라 태크스 A의 윈도우 크기를 증가 시키고 나머지 윈도우의 크기를 상대적으로 줄여가며 동기화 여부를 확인한 결과 태스크 A의 윈도우 크기가 18msec미만인 경우 동기화가 성공함을 확인하였다. 각 윈도우 크기에 따른 실험 결과는 표 1에 나타나 있다. 이 실험에서는 윈도우가 18인 경우 동기화가 보장되지 않는 것으로 확인되었으며 이는 메시지 교환을 위한 시간이 약 2msec 초과임을 알 수 있다. 반면, 윈도우 크기가 18미만인 경우에는 이러한 메시지 교환시간을 모두 포함하더라도 정상적으로 동기화가 보장됨을 알 수 있다.

표 1. 윈도우 크기에 따른 동기화 상태
Table. 1 Synchronization State According to Window

W(msec)	F-W(msec)	A Sync result
18	2	FAIL
17	3	OK
16	4	OK
15	5	OK
14	6	OK

이러한 연구와 더불어 윈도우 크기에 따른 동기화 방식을 기존의 PALS IMA와 비교하고자 한다. 이를 위하여 윈도우 크기가 17부터 20까지 0.5씩 증가시킨 경우를 비교하였다. 20인 경우는 기존의 PALS IMA의 경우를 나타내며 각 경우에 대하서는 다른 시드번호를 이용하여 100번의 다른 실험 환경을 구축하였다.

그림 11에서 알 수 있듯이 윈도우 크기가 17인 경우 제안된 방법은 동기화를 완벽하게 보장하게 되며 17.5인 경우 5%의 동기화에 실패한다. 더욱이 18이상일 경우 데이터 전송에 요구되는 시간 및 동기화 함수의 호출시간에 따라 성공률이 계속해서 높아지게 된다.

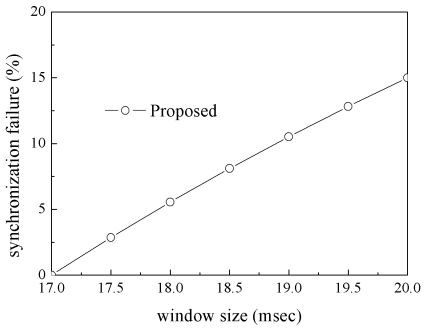

그림 11. 윈도우 크기에 따른 동기화 실패율
Fig. 11 Synchronization failure according to window size

또한, 윈도우 크기가 커질수록 동기화 실패율을 계속해서 높아지게 된다. 특히, 윈도우 크기가 20일 경우는 기존의 PALS IMA의 방법과 동일한 방법으로 동작하며 동기화 실패율이 약 15%정도 차이를 보이게 된다. 이는 기존의 PALS IMA방식을 사용하는 경우 완벽한 동기화가 불가능함을 의미한다. 따라서, 특정 프레임 크기가 정해진 경우 동기화를 위하여 데이터 전송 시간 및 동기화 호출 모듈의 수행시간에 맞추어 적당한 윈도우 크기를 고려하여 확정할 수 있으며 기존의 방식에 비하여 보다 적은 동기화 시간이 요구됨을 알 수 있다. 이는 그림 12와 같이 프레임 크기가 주어진 경우 완벽한 동기화를 보장하기 위하여 필요한 동기화윈도우크기를 보여준다.

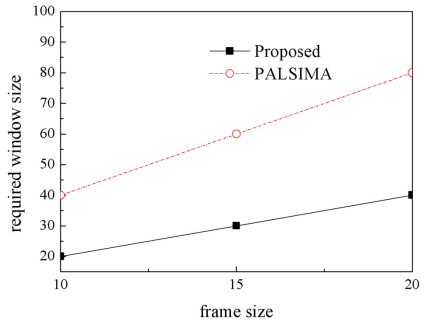

그림 12. 프레임 크기에 요구되는 윈도우 크기
Fig. 12 Required window size according to frame size

보장을 위한 동기화 윈도우의 경우 기존의 PALS IMA에 비하여 제안된 방식은 1/2로 줄어들었음을 알 수 있다.

Ⅴ. 결론 및 향후연구

본 논문에서는 RMS가 적용된 IMA환경에 적용 가능한 입력 데이터 동기화 설계기법을 제시하고 실험을 통해 제안된 방법의 타당성을 검증하였다. 검증 결과 IMA 프레임과 연동된 RMS를 통해 비행운영프로그램의 주요기능을 스케줄링 할 경우 제시된 'IMA 프레임 동기식의 PALS 패턴'을 참조하여 보다 효율적인 동기화가 가능하였다. 특히 이러한 패턴은 대량의 동기화 데이터를 운영하는Primary/Secondary 이중화 구조에 유용할 것으로 예상되며 기존 연구와 달리 모든 Low Rate 스케줄링 데이터를 동기화 할 수 있는 장점이 있다. 본 연구와 관련하여 실제 시스템에서의 해당 동기화 메커니즘을 구현 후 검증할 예정이며 다양한 IMA 구조에도 적용할 예정이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신산업진흥원의 고용계약형SW석사과정 (관리번호: NIPA-2014-HB301-14-1014)의 연구결과임

REFERENCES

[1] ARINC specification 651-1: Design Guidance for Integrated Modular Avionics, Aeronautical Radio, Inc., Annapolis, MD, November 1997.

[2] Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations, DO-297, RTCA Inc: Washington, DC, November 8, 2005.

[3] Lynch, N., *Distributed Algorithms*, Morgan Kaufmann Publishers, Inc: San Francisco, CA, 1996.

[4] Tel, G., *Introduction to Distributed Algorithms* Second Edition, Cambridge University Press: Cambridge, UK, 2000.

[5] Sha, L., A. Abdullah, M. Sun, j. Meseguer, P. Olveczky, PALS: Physically Asynchronous Logically Synchronous Systems, www.ideals.uiuc.edu/handle/2142/11897, May, 2009.

박홍열(Hong-Youl Park)

2006.02: 창원대학교 학사
2006.01 ~ 현재: 한국항공우주산업 주식회사 항전 S/W팀 연구원
※관심분야 : IMA, Fault Tolerance, Scheduling

김기일(Ki-Il Kim)

2002.02: 충남대학교 이학석사
2005.02: 충남대학교 이학박사
2006.03 ~ 현재: 경상대학교 정보과학과 부교수
※관심분야 : 센서네트워크, 이동네트워크, 항공기소프트웨어