JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 4 : 724~730 Apr. 2017

# 타임코드 확장을 통한 스페이스와이어 네트워크의 시각 동기화 성능 개선

류상문*

# Improvement of Time Synchronization of SpaceWire Network through Time-Code Extension

**Sang-Moon Ryu***

Department of Information and Control Engineering, Kunsan National University, Gunsan 54150, Korea

## 요 약

항공우주분야 시스템의 네트워크 구현을 위해 고안된 스페이스와이어에는 네트워크의 시각 동기화를 위한 타임코드가 정의되어있다. 스페이스와이어 네트워크에서 타임코드가 링크를 통과할 때마다 14[bit-period]의 전송 지연과 최대값이 10[bit-period]인 전송 지터가 발생하며 이것은 시각 동기화 오차의 주요 원인이다. 본 논문은 스페이스와이어 표준에 정의되어 있는 타임코드를 확장하여 시각 동기화 성능을 개선하는 방법을 제안한다. 타임 마스터와 시각 동기화를 수행하는 노드들은 확장된 타임코드들을 이용하여 시각 정보가 전송되는 과정에서 발생한 전송 지연과 지터를 파악하고 이를 이용하여 시각 동기화 보정을 수행할 수 있다. 제안된 방법의 효과는 OMNeT++ 기반의 스페이스와이어 네트워크 시뮬레이션 환경을 이용하여 분석되었으며 그 결과 수 [bit-period] 이내의 오차로 시각 동기화가 가능하다는 것이 확인되었다. 제안된 방법은 소규모 스페이스와이어 네트워크 시스템에 적합하며 이전 연구 결과들에 비해 구현에 따른 비용 대비 매우 효과적인 성능 향상을 얻을 수 있다.

## ABSTRACT

SpaceWire invented for spacecrafts has Time-Code defined for time synchronization over SpaceWire network. A Time-Code suffers transmission delay of 14[bit-period] and jitter up to 10[bit-period] whenever it passes through a SpaceWire link, which is the primary cause of time synchronization error. This work presents a simple method to improve the time synchronization which uses two extended Time-Codes. Nodes on a SpaceWire network can find how much delay and jitter a received Time-Code has suffered while it passes through the network, and they can correct time synchronization error with this information. The proposed method was validated in a simulation environment developed based on OMNeT++. The simulation result showed that time synchronization error less than a few bit-periods can be achieved. The proposed method is cost effective and suitable for small-scale SpaceWire network systems.

**키워드** : 스페이스와이어, 시각동기화, 타임코드 확장, OMNeT++

Key word : SpaceWire, time synchronization, Time-Code extension, OMNeT++

**Received** 21 November 2016, **Revised** 05 December 2016, **Accepted** 27 January 2017

* **Corresponding Author** Sang-Moon Ryu(E-mail:smryu@kunsan.ac.kr, Tel:+82-63-469-4688)
Department of Information and Control Engineering, Kunsan National University, Gunsan 54150, Korea

Open Access https://doi.org/10.6109/jkiice.2017.21.4.724 print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/ by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

## Ⅰ. 서 론

항공우주 분야의 네트워크 기반 임베디드 시스템 구현을 위해 ESA(European Space Agency)에 의해 표준화된 스페이스와이어(SpaceWire)[1, 2] 는 네트워크 구성이 용이하고 네트워크의 시각 동기화를 위한 타임코드(Time-Code) 전송 기능이 포함되어 있다. 스페이스와이어 네트워크에 접속되어 있는 타임 마스터(time master)는 그림 1과 같이 패리티(P)를 포함해 14비트로 정의되어있는 타임코드를 약속된 주기에 따라 주기적으로 전송하여 네트워크에 접속된 모든 노드(node)들에게 시각 정보를 분배할 수 있다. 타임코드에서 $< T_7 - T_6 >$는 제어 플래그로 모두 '0'이어야 하며 이외의 값에 대해서는 그 용도가 아직 정해져있지 않다. $< T_5 - T_0 >$는 타임 마스터가 자체적으로 유지하고 네트워크에 배포하는 6비트 클럭 카운트(clock count)이며 타임 코드를 수신한 노드들은 이 클럭카운트를 이용하여 시각 동기화를 수행한다.

| P | 1 | 1 | 1 | 1 | 0 | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

LSB MSB

**Fig. 1** Time-Code of SpaceWire

타임코드를 이용한 시각 동기화는 스페이스와이어 표준[1]의 8.12 System time distribution에 정의되어 있는데 간략히 정리하면 다음과 같다. 클럭 마스터는 타임코드 배포 주기 마다 클럭 카운트를 1씩 증가(만일 6비트가 모두 1인 63인 경우에는 0이 됨)하여 네트워크에 배포한다. 타임코드를 수신하는 노드는 유효한 타임코드에 포함된 클럭 카운트만을 이용하여 시각 동기화를 수행한다. 타임코드의 유효성은 타임코드를 수신한 노드에 의해 판단되는데, 직전에 수신된 클럭 카운트보다 큰 클럭 카운트를 포함하는 타임코드를 유효한 타임코드로 그렇지 않은 타임코드를 무효한 타임코드로 판단한다.

타임코드의 전송 과정에서 전송 지연 및 지터(jitter)가 발생하며 이것은 타임 마스터와 노드들 사이의 시각 동기화 오차의 주요 원인이 된다[3, 4]. 전송 지연은 타임코드가 타임 마스터나 라우터의 송신부에서 출력되는 과정에서 발생하는 출력 지연으로 타임코드의 길이에 대응하는 14bit-period이다. 전송 지터 역시 타임코드의 출력 지연으로 타임코드가 타임 마스터나 라우터의 송신부에서 출력되어야하는 순간에 이미 다른 문자(4비트, 8비트 또는 10 비트로 구성)가 출력되고 있는 중이라면 이 문자의 출력이 완료될 때까지 타임코드의 출력이 지연되기 때문에 발생하며 그 최대 크기는 10bit-period이다[4].

본 논문에서는 위와 같이 스페이스와이어 네트워크에서 타임코드를 이용한 시각 동기화를 수행할 때 발생하는 동기화 오차를 타임코드의 확장을 통하여 감소하는 새로운 방안을 제안한다. 이를 위해 2장에서 관련된 이전 연구에 대해 소개하고, 3장에서는 타임코드의 확장 및 이를 이용한 시각 동기화 오차 개선 방법에 대해 소개한다. 4장에서는 제안된 방안에 대해 시뮬레이션을 통해 확인된 효과를 소개하고 끝으로 5장에서 결론을 맺는다.

## Ⅱ. 관련 연구

스페이스와이어 네트워크의 시각 동기화 오차를 줄이기 위해 이전 연구 [3]에서는 표준에 정의된 타임코드의 제어 플래그를 변형한 확장된 타임코드를 정의하고 서브타임카운터(sub time-counter)를 추가로 구현하여 사용하는 방법을 제안하였다. 타임코드가 전송될 때마다 발생하는 전송 지터에 관한 정보를 확장된 타임코드에 실어서 추가로 전송하고, 이 전송 지터 정보를 수신한 노드에서는 이를 자신의 서브타임카운터의 상태에 반영하여 전송 지터의 영향을 줄이는 방법이다.

이 방법은 타임 마스터에 인접한 노드에서는 1bit-period 미만의 오차로 정밀 시각 동기화를 수행할 수 있으나 시각 동기화 1회 수행마다 타임코드가 2개 전송되어야하며 시각 동기화 회로의 동작이 스페이스와이어 표준에 부합하지 않는 큰 단점이 있다. 스페이스와이어 표준에서는 시각 동기화가 주요 전기 신호의 상승 모서리에 수행되도록 되어 있는데 이 방법은 하강 모서리에서 수행되도록 되어 있어 호환성이 매우 부족하다. 그리고 타임코드가 링크(link)를 통과할 때마다 발생하는 전송 지터를 모두 고려할 수 없으며 전송지연에 따른 시각 동기화 오차는 제거할 수 없기 때문에 노드가 타임 마스터로부터 멀어질수록 성취할 수 있는 시각 동기화 오차는 커지게 된다.

이전 연구 [5]에서는 타임코드의 추가 전송 없이 서브타임카운터만을 이용하여 1bit-period 미만의 오차로 정밀 시각 동기화를 수행할 수 있는 방법이 제안되었다. 이 방법은 전송 지터가 최소로 발생하는 타임코드를 판별하여 이를 이용하는 방법으로 타임 마스터에 인접한 노드에서만 정밀 시각 동기화를 수행할 수 있다는 한계와 타임코드 활용 효율이 낮다는 단점이 있다.

다른 연구 [6]은 [7]에서 제안된 브로드캐스팅(broadcasting)을 이용하여 스페이스와이어 네트워크의 시각 동기화 오차를 줄이는 방식을 제안하였다. 이 방법은 범용 네트워크에서의 시각 동기화 방법과 유사하며 스페이스와이어 표준에 부합하지 않으며 시각 동기화 수행을 위한 추가의 소프트웨어 구현 및 구동에 따른 부담이 큰 단점이다.

Gaisler사는 정밀 시각 동기화를 위해 “SpaceWire Time Distribution Protocol”을 제안하고 이를 회로로 구현한 SPWTDP라는 IP(intellectual property)를 개발하였다[8, 9]. 이 방법은 소프트웨어 구현과 구동에 따른 부담이 없지만 시각 동기화를 수행하는 노드의 시각 동기화 정보가 저장되어 있는 메모리 영역이 접근될 수 있도록 RMAP(Remote Memory Access Protocol)[10]이 함께 적용되어야한다. 따라서 이 방법을 적용하기 위해서는 필요한 회로들을 직접 개발하거나 이미 개발된 IP들을 유료로 구입하여야 하는 부담이 발생한다.

이전 연구 [11]은 [5]에서 적용한 서브타임카운터의 사용은 배제하고 확장된 타임코드를 적용한 방법을 제안하였다. 제안된 방법은 타임 마스터에 인접한 노드에서의 시각 동기화 성능은 [5]에 비해 다소 떨어지지만 관련 회로 구현을 단순화할 수 있으며 타임 마스터로부터 라우터 하나를 거쳐서 연결된 노드에서의 시각 동기화 성능까지 개선할 수 있다. 이 방법은 구현에 따른 부담 대비 시각 동기화 성능은 많이 개선되지만 라우터가 1개만 사용되는 매우 작은 규모의 스페이스와이어 네트워크에만 적용할 수 있다는 한계가 있다.

## Ⅲ. 타임코드 확장을 통한 시각 동기화

본 논문에서는 스페이스와이어 표준[1]의 8.12 System time distribution에 정의된 내용을 바탕으로 하여 타임코드 확장만을 이용한 시각 동기화 방법을 제안한다. 제안된 방법은 [11]의 방법을 보완한 것으로 확장된 타임코드들을 정의하고 이들을 이용해 타임코드의 전송 지연 및 지터 발생 정보를 추가로 전송하여 시각 동기화 오차를 획기적으로 줄일 수 있다. 이 방법은 [3, 5, 6, 8]의 경우보다 시각 동기화 성능은 떨어지지만 서브타임카운터나 복잡한 프로토콜을 구현하기 위한 추가의 하드웨어나 소프트웨어가 필요 없기 때문에 구현에 따른 부담이 매우 적으며 스페이스와이어 표준의 사양을 크게 벗어나지 않고 구현할 수 있다. 제안된 방법은 타임 마스터와 정밀 시각 동기화를 수행해야하는 노드가 연결된 경로에 존재하는 링크의 통신 속도가 동일한 경우에만 적용할 수 있으며 네트워크의 규모도 일정 수준 이하로 제한된다. 스페이스와이어가 범용 네트워크가 아닌 네트워크 기반 임베디드 시스템을 위한 용도로 제안된 표준이기 때문에 이러한 제한을 수용하여 시스템을 구성하는 것은 어렵지 않다고 판단된다.

### 3.1. 타임코드의 확장

본 연구에서는 아직 용도가 정해지지 않는 타임코드의 제어 플래그 $< T_7 - T_6 >$를 이용하여 표 1과 같이 타임코드의 기능을 세분화하고 이를 이용한다. 표에서 NTC(Normal Time-Code)는 $< T_7 - T_6 >$가 모두 ‘0’인 기존의 타임코드를 의미한다. LCT(Link Count Time-Code)는 NTC가 전달되는 경로에 존재하는 링크의 수인 링크 카운트(link count)를 전달하는 타임코드로 $< T_5 - T_0 >$는 링크 카운트로 사용된다. CJT (Cumulative Jitter count Time-Code)는 NTC가 전달되는 과정에 발생한 전송 지터의 합인 누적 지터 카운트(cumulative jitter count)를 전달하는 타임코드로 $< T_5 - T_0 >$는 누적 지터 카운트로 사용된다.

**Table. 1** Extension of Time-Code

| Extended Time-Code | $< T_7 - T_6 >$ | $< T_5 - T_0 >$ |
|---|---|---|
| NTC | 00 | clock count |
| LCT | 01 | link count |
| CJT | 10 | cumulative jitter count |
| reserved | 11 | N/A |

### 3.2. LCT

NTC가 타임 마스터나 라우터로부터 출력될 때마다 즉 링크를 한 개 통과할 때마다 NTC의 길이에 해당하

는 14bit-period의 전송 지연이 발생한다. 따라서 NTC를 수신하는 노드는 수신된 NTC가 전송 과정에서 몇 개의 링크를 경유했는지 알면 전송 지연으로 유발되는 시각 동기화 오차를 보정할 수 있다. LCT는 이러한 정보를 포함하며 다음의 규칙에 의해 스페이스와이어 네트워크에 전파된다.

a. 타임 마스터는 NTC를 출력한 직후에 LCT를 출력할 수 있다. NTC를 출력할 때마다 LCT를 출력할 필요는 없으며 LCT의 출력 빈도는 시스템의 요구 사항에 따라 정해진다.
b. 타임 마스터가 LCT를 출력할 때 링크 카운트는 1로 설정한다.
c. 라우터는 유효한 NTC가 수신된 직후에 수신되는 LCT만을 처리하며, 무효한 NTC가 수신된 직후에 수신되는 LCT는 무시한다.
d. LCT를 수신한 라우터는 수신된 LCT의 링크 카운트가 63보다 작으면 이를 1 증가하고 그렇지 않으면 그 값을 유지하여 LCT가 수신된 포트를 제외한 모든 포트에 출력한다.

LCT를 수신한 노드는 수신된 링크 카운트로부터 타임 마스터로부터의 거리(링크 수)를 파악할 수 있다. 규칙 a에 의해 타임 마스터는 NTC를 출력한 직후에만 LCT를 출력하므로 라우터 역시 NTC를 출력한 직후에 LCT를 출력하게 된다. 규칙 c는 LCT가 스페이스와이어 네트워크 내에서 무한 반복적으로 전파되는 것을 방지하기 위한 것이다. 규칙 d에 의해 타임 마스터와 각 노드들 사이의 최대 링크 수가 63 이하인 규모의 스페이스와이어 네트워크에만 LCT를 적용할 수 있다.

### 3.3. CJT

NTC가 타임 마스터나 라우터로부터 출력될 때마다 즉 링크를 한 개 통과할 때마다 최대값이 10bit-period인 전송 지터가 발생한다. 만일 NTC를 수신한 노드가 NTC가 전송되는 과정에서 얼마의 전송 지터가 발생했는지를 알면 전송 지터의 누적으로 유발되는 시각 동기화 오차를 보정할 수 있다. CJT는 이러한 정보를 포함하며 다음의 규칙에 의해 스페이스와이어 네트워크에 전파된다.

a. 타임 마스터는 NTC를 출력한 직후에 CJT를 출력한다. CJT에 앞서 출력된 NTC의 출력 과정에서 발생한 전송 지터의 양(단위는 bit-period)을 CJT의 누적 지터 카운트로 설정한다.
b. 라우터는 유효한 NTC가 수신된 직후에 수신되는 CJT만을 처리하며, 무효한 NTC가 수신된 직후에 수신되는 CJT는 무시한다.
c. CJT를 수신한 라우터는 수신된 CJT의 누적 지터 카운트를 변경하여 CJT가 수신된 포트를 제외한 모든 포트에 출력한다. 이때 각 포트에서 출력되는 CJT의 누적 지터 카운트는 수신된 CJT의 누적 지터 카운트에 해당 포트에서 NTC가 출력되면서 발생한 전송 지터의 양(단위는 bit-period)을 더한 값으로 설정되며 그 최대값은 63으로 제한된다.

NTC와 CJT를 연달아 수신한 노드는 수신된 누적 지터 카운트로부터 NTC 전송 과정에 발생한 전송 지터의 누적 합을 파악할 수 있다. 규칙 a에 의해 타임 마스터는 NTC를 출력한 직후에만 CJT를 출력하므로 라우터 역시 NTC를 출력한 직후에 CJT를 출력하게 된다. 규칙 b는 CJT가 스페이스와이어 네트워크 내에서 무한 반복적으로 전파되는 것을 방지하기 위한 것이다. 규칙 c에 의해 타임 마스터와 각 노드들 사이의 최대 링크 수가 6($6 = \lfloor 63/10 \rfloor$) 이하인 규모의 스페이스와이어 네트워크에만 CJT를 적용할 수 있다.

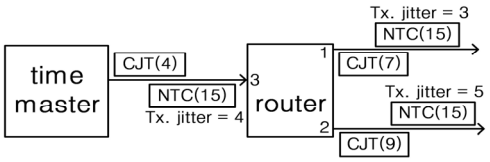


**Fig. 2** Example of transmission of CJTs

그림 2는 타임 마스터에서 출력된 CJT의 누적 지터 카운트가 변경되는 과정의 예를 보여준다. 그림에서는 타임 마스터가 클럭 카운트가 15인 NTC를 출력했고 이 과정에서 전송 지터가 4bit-period 발생했다. 그래서 타임 마스터는 누적 지터 카운트가 4인 CJT를 출력했다. 라우터의 3번 포트에 수신된 NTC는 포트 1번과 2번에서 출력되고 이 과정에서 전송 지터가 각각 3bit-period과 5bit-period 발생했다. 따라서 라우터는 포트 1번에는 누적 지터 카운트가 7(=4+3)인 CJT를, 포트 2번에는 누적 지터 카운트가 9(=4+5)인 CJT를 출력했다. 이 과정은 NTC와 CJT가 라우터를 경유할 때마다 수행된다.

### 3.4. LCT와 CJT를 이용한 시각 동기화 오차 개선

타임 마스터와 시각 동기화를 수행하는 노드들은 유효한 NTC가 수신된 직후에 수신되는 LCT나 CJT에 포함된 링크 카운트나 누적 지터 카운트만을 이용하여 시각 동기화 오차를 보정한다. 수신된 LCT의 링크 카운트를 $l$ 그리고 수신된 LJT의 누적 지터 카운트를 $j$라고 하면 수신된 NTC의 전송 과정에서 $(14\times l)$bit-period에 해당하는 전송 지연과 j bit-period에 해당하는 누적 지터가 발생했다는 것이다. 따라서 정해진 통신 속도에 의해 타임코드가 전송되는 비트-주기를 $p$초라고 하면 전송과정에서 발생한 총 시간 지연은 식 (1)과 같게 된다.

$$p(14\times l+j) \tag{1}$$

시각 동기화를 수행하는 노드는 이미 LCT를 수신하여 타임 마스터로부터의 거리가 파악된 상태에서 유효한 NTC와 LJT를 수신할 때 마다 NTC에 포함된 클럭 카운트와 식 (1)의 총 시간 지연을 적용하여 시각 동기화를 수행하면 시각 동기화 오차를 개선할 수 있다.

### 3.5. LCT와 CJT 적용에 따른 스페이스와이어 네트워크 규모 제한에 대한 검토

3.2절에 언급된 바와 같이 LCT는 타임 마스터와 각 노드들 사이의 최대 링크 수가 63 이하인 규모의 스페이스와이어 네트워크에만 적용할 수 있으며, 3.3절에 언급된 바와 같이 CJT는 타임 마스터와 각 노드들 사이의 최대 링크 수가 6 이하인 규모의 스페이스와이어 네트워크에만 적용할 수 있다. 따라서 LCT와 CJT를 모두 적용하려면 스페이스와이어 네트워크의 규모는 타임 마스터와 각 노드들 사이의 최대 링크 수가 6 이하로 제한된다.

그림 3은 타임 마스터와 각 노드들 사이의 최대 링크 수가 6 이하인 경우에 해당하는 스페이스와이어 네트워크의 예를 보여준다. 그림에서 R은 라우터를 N은 노드를 의미하며 라우터에 접속될 수 있는 대부분의 노드들은 생략되었다. 현재 상용으로 공급되는 스페이스와이어 라우터는 최대 32포트를 지원한다. 따라서 그림 3과 같이 구성된 네트워크에서 모든 라우터가 32포트를 보유하고 있다면 31개 노드를 접속할 수 있는 라우터는 9개, 30개 노드를 접속할 수 있는 라우터는 12개 그리고 29개와 28개 노드를 접속할 수 있는 라우터는 각각 1개와 3개이다. 따라서 이 경우에 접속 가능한 총 노드의 수는 타임 마스터를 포함하여 752개이다. LCT와 CJT를 적용하여 타임 마스터와 각 노드들 사이의 최대 링크 수가 6 이하로 제한되어도 상당한 규모의 네트워크 기반 임베디드 시스템을 구현할 수 있다.

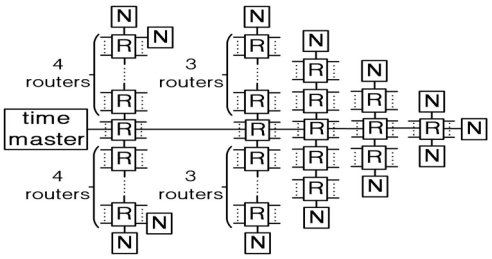


**Fig. 3** Example of SpaceWire network with maximum 6 links between the time master and every node

## Ⅳ. 시뮬레이션 및 결과

이전 연구 [4]에서 개발된 OMNeT++ 기반 스페이스와이어 네트워크 시뮬레이션 환경에서 그림 4와 같이 3개의 라우터와 8개의 노드로 구성된 스페이스와이어 네트워크 시스템을 모사하도록 구성하고 NTC와 함께 LCT와 CJT를 이용한 시각 동기화 방법의 효과를 NTC만을 이용하는 기존의 시각 동기화 방법과 비교하였다. 네트워크의 통신 속도는 기본 속도인 10[Mbps]로, 네트워크 사용률은 50% 정도로 설정하였으며, 20만회의 시각 동기화 수행에 따른 시각 동기화 오차에 대한 통계를 분석하였다.

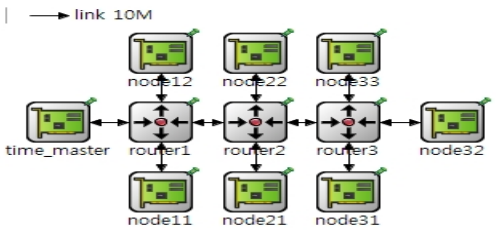


**Fig. 4** Assumed SpaceWire network system

표 2는 기존의 방법과 제안된 방법에 따른 시각 동기화 오차의 평균과 표준 편차를 보여준다. 기존 방법을 적용하면 타임마스터로부터 거리(링크 수)가 멀어질수록 오차가 크게 증가하는 것을 알 수 있다. 그러나 제안

된 방법을 적용하면 오차가 획기적으로 감소하게 되는데, 통신 속도 10Mbps에 대응하는 비트-주기는 100nsec이므로 오차의 평균이 1~2 bit-period 정도로 매우 작다는 것을 알 수 있다. 그리고 타임마스터로부터 거리(링크 수)가 멀어질수록 오차가 증가하는 현상은 기존의 방법과 동일하지만 증가 폭이 링크 당 1bit-period로 매우 작다는 것을 알 수 있다.

**Table. 2** Statistics of time synchronization error [nsec]

| | Conventional | | Proposed | |
|---|---|---|---|---|
| | mean | std. dev. | mean | std. dev. |
| node 11 | 3723 | 374 | 95 | 41 |
| node 12 | 3776 | 391 | 95 | 41 |
| node 21 | 5393 | 442 | 133 | 41 |
| node 22 | 5497 | 461 | 133 | 41 |
| node 31 | 7128 | 469 | 168 | 42 |
| node 32 | 7140 | 497 | 168 | 42 |
| node 33 | 7127 | 500 | 168 | 42 |

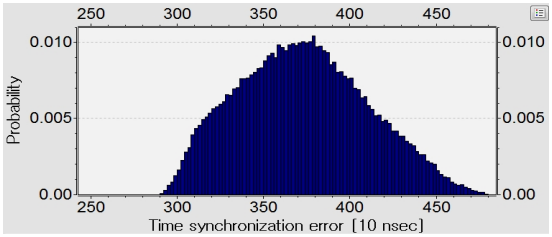


**Fig. 5** Probability graph of time synchronization error at node 11 with the conventional way

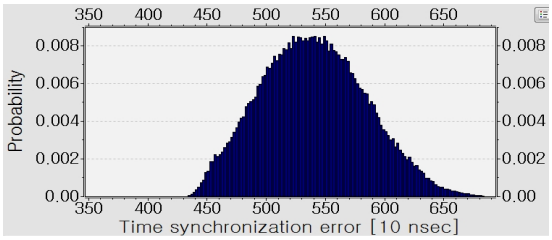


**Fig. 6** Probability graph of time synchronization error at node 21 with the conventional way

그림 5-7은 노드 11, 21, 31에서 관찰된 기존의 방법에 따른 시각 동기화 오차의 분포를, 그림 8-10은 노드 11, 21, 31에서 관찰된 제안된 방법에 따른 시각 동기화 오차의 분포를 보여준다. 노드 12의 분포는 그림 5와 8의 노드 11과 유사하게, 노드 22의 분포는 그림 6과 9의 노드 21과 유사하게, 그리고 노드 32와 33의 분포는 그림 7과 10의 노드 31과 유사하게 나타났다. 제안된 방법의 적용에 따른 오차의 발생 범위는 모두 2~3 bit-period 이내로 적용 이전 대비 크게 향상되는 것을 알 수 있다.

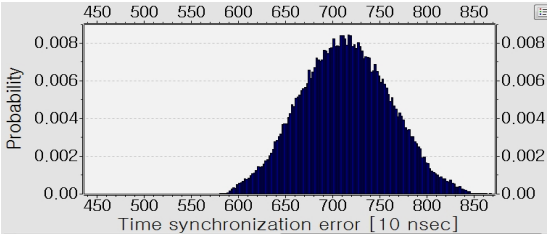


**Fig. 7** Probability graph of time synchronization error at node 31 with the conventional way

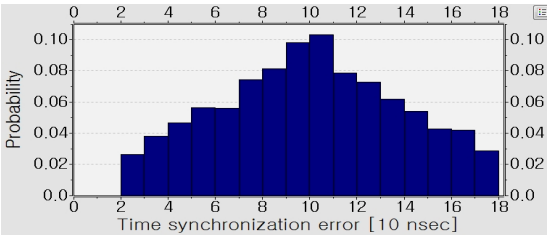


**Fig. 8** Probability graph of time synchronization error at node 11 with the proposed way

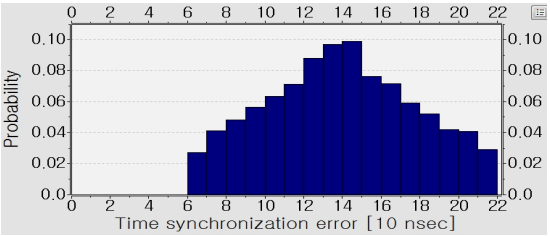


**Fig. 9** Probability graph of time synchronization error at node 21 with the proposed way

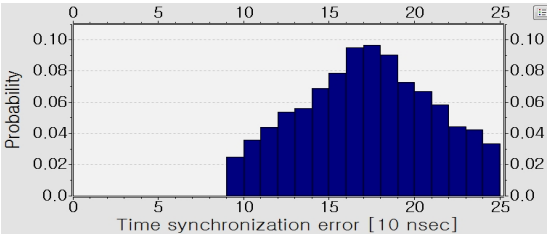


**Fig. 10** Probability graph of time synchronization error at node 31 with the proposed way

## V. 결 론

본 논문에서는 스페이스와이어 네트워크에서 시각 동기화를 위해 배포되는 타임코드의 전송 중에 발생하는 전송 지연과 전송 지터의 영향을 감소시키는 방법을 제안하였다. 스페이스와이어 표준에 정의되어 있는 타임코드를 세분화하여 타임 마스터와 수신 노드 사이의 링크 수 정보를 전달하는 LCT와 타임 마스터에서 출력된 타임 코드가 노드에 도착할 때까지 발생한 전송 지터의 누적 합 정보를 전달하는 CJT를 추가로 정의하고 이들을 이용하여 시각 동기화 오차를 보정하는 방법을 제안하였다. OMNeT++ 기반의 스페이스와이어 네트워크 시뮬레이션 환경에서 수행된 시뮬레이션 결과를 통하여 제안된 방법이 시각 동기화 오차를 현저히 줄일 수 있음을 확인하였다. 제안된 방법을 적용하기 위해서는 타임 마스터와 정밀 시각 동기화를 수행하는 노드까지의 경로에 존재하는 모든 링크들에서의 통신 속도가 동일해야한다는 제한이 있다. 타임 마스터와의 정밀 시각 동기화가 필요하지 않는 노드들에는 이러한 제한이 적용되지 않는다. 제안된 방법은 스페이스와이어 링크 인터페이스와 라우터에 포함되는 전송 회로를 조금만 수정하면 쉽게 구현할 수 있고 기존의 방법 대비 복잡한 하드웨어나 소프트웨어 부담이 없다는 장점이 있기 때문에 수 bit-period 이내의 시각 동기화 오차를 허용할 수 있는 경우에는 투자 대비 매우 큰 효과를 얻을 수 있을 것으로 판단된다.

## REFERENCES


[1] ESA Std. ECSS-E-ST-50-12C, *SpaceWire-Links, nodes, routers and networks*, ESA, 2008.

[2] ESA SpaceWire web site [Internet]. Available: http://spacewire.esa.int/.

[3] F. Pinsard and C. Cara, “High resolution time synchronization over spacewire links,” in *Proceedings of 2008 IEEE Aerospace Conference*, pp. 1-8, 2008.

[4] S.-M. Ryu, “Analysis of transmission delay of timecode over SpaceWire network using OMNeT++,” *Journal of Korea Institute of Information and Communication Engineering*, vol. 19, no. 9, pp. 2022-2028, Sep. 2015.

[5] S.-M. Ryu, “Improvement of Time Synchronization over SpaceWire Link,” *Journal of Control, Robotics and Systems*, vol. 15, no. 11, pp. 1144-1149, Nov. 2009.

[6] B. V. Leeuwen, J. Eldridge and J. Leemaster, “SpaceWire Network Simulation of System Time Precision,” in *Proceedings of International Spacewire Conference 2011*, pp. 113-120, 2011.

[7] A. Roberts, S. G. Dykes, R. Klar and C. C. Mangels, “A Link-Layer Broadcast Service for SpaceWire Networks,” in *Proceedings of 2007 IEEE Aerospace Conference*, pp. 1-10, 2007.

[8] A. Hashmi, H. Berry, O. Temam, and M. Lipasti, “Spacewire time distribution protocol implementation and results,” in *Proceedings of 2014 International SpaceWire Conference*, pp. 19-25, Oct. 2014.

[9] “SpaceWire-Time Distribution Protocol VHDL IP Core User’s Manual,” Aeroflex Gaisler, Technical Report SPWCUC-REP-0005, 2014.

[10] ESA Std. ECSS-E-ST-50-52C, *SpaceWire-Remote memory access protocol*, ESA, 2010.

[11] S.-M. Ryu, “Time Synchronization over SpaceWire Network using Hop Count Information,” in *Proceedings of 2016 fall conference of Korea Institute of Information and Communication Eng.*, pp. 715-718, 2016.

**류상문(Sang-Moon Ryu)**

금오공과대학교 전자공학 학사
KAIST 전기및전자공학 석사
KAIST 전기및전자공학 박사
LG전자 주임연구원
KAIST 선임연구원 및 연구교수
군산대학교 IT정보제어공학부 교수
※관심분야 : 결함허용 임베디드 시스템, FPGA 기반 임베디드 시스템