

스페이스와이어 링크의 시각 동기 성능 개선

Improvement of Time Synchronization over SpaceWire Link

류 상 문*
(Sang-Moon Ryu)

Abstract: This paper deals with the time synchronization problem over SpaceWire links. SpaceWire is a standard for high-speed links and networks between spacecraft components, which was invented for better, cheaper, faster on-board data handling in spacecraft. The standard defines Time-Code for time distribution over SpaceWire network. When a Time-Code is transmitted, transmission delay and jitter is unavoidable. In this paper, a mechanism to remove Time-Code transmission delay and jitter over SpaceWire links is proposed and implemented with FPGA for validation. The proposed mechanism achieves high resolution clock synchronization over SpaceWire links, complies with the standard and can be easily adopted over SpaceWire network.

Keywords: clock synchronization, time synchronization, SpaceWire, link interface

I. 서론

우주 비행체(spacecraft)는 자세 제어, 위치 제어, 전력 관리, 명령 처리, 데이터 처리 등의 주요 기능을 담당하는 다양한 서브 시스템(sub-system)으로 구성되며 개개 서브 시스템은 개별 동작이 가능한 다수의 유닛(unit)들로 구성된다. 시스템을 구성하는 유닛들은 내부 네트워크나 버스의 형태로 연결되며 대부분 자체의 클럭(clock) 신호를 이용해서 각자 시간 정보를 유지하고 이를 기준으로 동작한다.

요구 사항에 따라 전체 시스템 또는 일부 유닛은 시각 동기화되어 동작하여야 하는데, 상이한 부품의 특성과 온도 환경 등에 의해 시간이 흐를수록 시각 동기화 상태를 벗어나게 되며 그 차이는 지속적으로 증가하게 된다. 따라서 주기적인 시각 동기화가 필수적이며 유닛 간 시각차는 요구 사항의 허용 범위 이내로 유지되어야 한다.

이러한 시각 동기화는 분산형 시스템에서는 필수적이며 다양한 형태의 시스템에 대해 관련 연구[1-11]가 국내외에서 진행되었으며, 네트워크를 기반으로 하는 계층 제어 장치에서의 정밀 시각 동기화를 위한 IEEE 1588 표준[12]도 제정되었다.

본 논문은 우주 비행체의 온보드 데이터 처리(on-board data handling)를 위해 만들어진 스페이스와이어(SpaceWire)[13]로 연결된 유닛 사이의 시각 동기화에 대한 성능 개선 방법을 제안한다.

스페이스와이어는 우주 비행체의 온보드 데이터 처리를 위해 IEEE 1355 표준[14]을 바탕으로 1999년에 고안되었으며 2003년에 유럽의 ESA (European Space Agency) 표준으로 채택되었다. 그 후 ESA, NASA (National Aeronautics and Space Administration) 그리고 JAXA (Japan Aerospace Exploration Agency)의 우주개발 프로그램에 적용되어 왔으며 2008년에 그 내용이 개정되었다.

스페이스와이어는 점 대 점 방식의 양방향 통신을 전이중

으로 지원하여 주며, 10m 정도의 거리에서 2M~400Mbps의 직렬 통신을 가능하게 해준다. 그리고 물리 계층에서는 LVDS (Low Voltage Differential Signaling)를 적용하여 송신기의 소비 전력이 상당히 적으며, 고속 전송이 가능하다. 또한 네트워크 구성이 용이하며 시간 정보 분배를 위한 time-code 전송 기능이 있다.

스페이스와이어 네트워크의 구성을 위해서는 스페이스와이어 링크 인터페이스(link interface)와 라우터가 필요하다. 링크 인터페이스는 우주 비행체 내부의 유닛(unit) 사이의 통신을 담당하는 것으로 스페이스와이어 표준에 정의되어 있는 신호 레벨(signal level), 문자 레벨(character level), 교환 레벨(exchange level) 그리고 패킷 레벨(packet level)을 담당하고, 라우터는 스페이스와이어 표준에 정의되어 있는 네트워크 레벨(network level)을 담당한다.

스페이스와이어 네트워크에 접속되어 있는 클럭 마스터(clock master)는 time-code를 전송하여 시간 정보를 분배할 수 있는데 time-code의 전송에는 전송 지터(jitter)가 발생하게 되어 정밀한 시각 동기화가 어렵다.

본 논문은 스페이스와이어 표준을 어기지 않고 이러한 time-code 전송 지터의 영향을 없앨 수 있는 방법을 제안하고 하드웨어 구현과 실험을 통하여 이를 검증한다. II 장에서 스페이스와이어 링크 인터페이스를 통하여 연결된 유닛 사이의 시각 동기화와 동기화 오차 문제에 대해서 기술하고 이를 해결하기 위해 타 연구자에 의해 제안된 방법과 그 문제점에 대해 언급한다. 본 논문이 제안하는 방법은 III 장과 IV 장에서 구현 및 실험 결과와 함께 소개되며 V 장에서 결론을 맺는다.

II. 스페이스와이어 링크의 시각 동기화

스페이스와이어 표준의 문자 레벨(character level)에는 그림 1과 같이 정보 전달을 위한 8-bit 데이터 문자(data character), 링크 제어를 위한 4-bit 제어 문자(control character) 그리고 8-bit NULL과 14-bit Time-code를 포함하는 제어 코드(control codes)가 정의되어 있다.

정보는 데이터 문자와 패킷의 종료를 알려주는 EOP로 구성되는 패킷 단위로 전달되며, 흐름 제어를 위하여 8개의 문자가

* 책임저자(Corresponding Author)

논문접수: 2009. 8. 1., 수정: 2009. 9. 23., 채택확정: 2009. 10. 7.

류상문: 군산대학교 제어로봇시스템공학과(smryu@kunsan.ac.kr)

※ 본 연구는 교육과학기술부의 과학기술위성 3호 개발 사업의 지원을 받아 수행되었음.

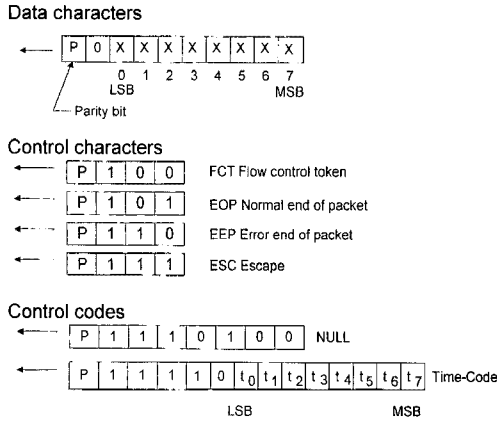


그림 1. 스페이스와이어 제어 문자와 데이터 문자.
Fig. 1. SpaceWire control characters and control codes.

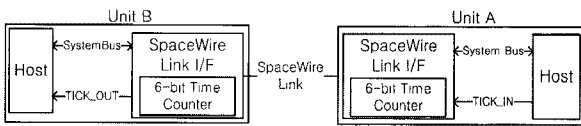


그림 2. 스페이스와이어 링크 및 링크 인터페이스.
Fig. 2. SpaceWire link and link interfaces.

전송될 때마다 FCT가 전송된다. 그리고 NULL은 패킷 전송이 없을 때 링크의 연결 상태를 유지하기 위해서 전송된다.

Time-code는 시간 정보를 전달하기 위해 전송되며, $\langle t_5 - t_0 \rangle$ 에 시간 정보가 포함되며 t_6 과 t_7 은 제어 플래그(control flag)로 사용된다. 제어 플래그는 예시로 마련되어 있으며 현재는 그 용도가 정해져 있지 않다.

스페이스와이어 표준에 따르면 그림 2처럼 시스템을 구성하는 유닛 A와 B가 스페이스와이어 링크로 연결되어 통신하면서 유닛 A가 시간 정보를 전달하는 마스터로, 유닛 B가 이를 수신하는 슬레이브로 동작하는 경우, 유닛 A의 호스트는 주기적으로 TICK_IN 신호를 발생하고, TICK_IN을 수신한 스페이스와이어 링크 인터페이스는 내부의 6-bit time counter를 1만큼 증가시키고 이 값을 $\langle t_5 - t_0 \rangle$ 에 담은 time-code를 전송해야 한다.

그리고 time-code를 수신한 유닛 B의 스페이스와이어 링크 인터페이스는 수신된 time-code의 $\langle t_5 - t_0 \rangle$ 값을 자신의 6-bit time counter에 저장하여 유닛 A과 동일한 시간을 유지하여야 한다.

스페이스와이어 링크 인터페이스는 호스트로부터 시간 전송 요청(TICK_IN)을 받으면 전송 중인 문자나 NULL이 전송된 후 time-code를 전송하게 된다. 따라서 time-code의 전송 요청과 실제 전송 시작 사이에 그림 3의 Δ 로 표현되는 시간 지연이 발생할 수 있다.

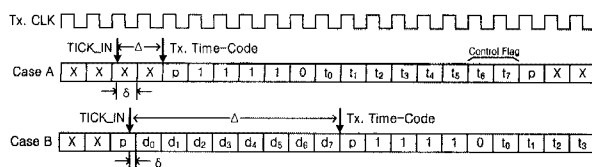


그림 3. 스페이스와이어 링크의 time-code 전송.
Fig. 3. Transmission of a time-code over SpaceWire link.

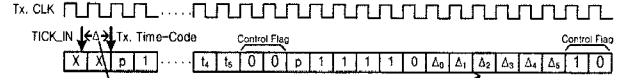


그림 4. Pinsard와 Cara가 제안한 time-code 전송.
Fig. 4. Transmission of a time-code proposed by Pinsard and Cara.

만일 문자나 NULL이 전송 완료되는 시점에 time-code 전송이 요청되면 시간 지연이 매우 작겠지만, 10-bit 데이터 문자가 전송되기 시작하는 시점에 전송이 요청되면 데이터 문자의 전송에 필요한 전송 클럭(Tx. CLK)의 10 주기에 해당하는 지연이 발생하게 된다. 기본 전송 속도인 10[Mbps]의 경우에는 최대 전송 지연이 1[μ sec]이며, 최저 속도인 2[Mbps]의 경우에는 최대 전송 지연이 5[μ sec]가 된다.

더욱이 이러한 time-code 전송 지연은 일정한 값으로 유지되는 것이 아니라 0에서부터 최대값까지 무작위적으로 변하게 되어 정밀한 시간 동기화가 필요한 시스템의 경우 그 성능에 영향을 주게 된다.

Pinsard와 Cara는 [1]에서 이러한 time-code 전송 지연에 의한 시간 동기 오차를 줄이기 위해서 그림 4로 표현되는 다음과 같은 방법을 제안하였다.

Time-code 전송 요청과 전송 시작 사이의 지연 Δ 를 전송 클럭의 수로 환산하고 $\langle \Delta_5 - \Delta_0 \rangle$ 로 변환하여 이를 추가로 전송한다. 전송 형태는 time-code와 유사하게 하고 시간 정보를 담고 있는 time-code와 구별하기 위해 t_6 과 t_7 로 이루어지는 제어 플래그를 이용하는데, 그림 4는 time-code가 시간 정보를 담고 있는 경우에는 "00"이, 전송 지연 정보를 담고 있는 경우에는 "10"이 적용된 경우를 보여준다.

수신측에서는 시간 정보에 이어 수신되는 전송 지연 정보를 이용하여 정밀한 시간 동기화를 꾀할 수 있도록 하자는 것이다. 이 방법의 가장 큰 문제는 일반적인 스페이스와이어 링크 인터페이스는 전송 지연 정보를 담고 있는 변형된 time-code를 인식하지 못하기 때문에 전송된 시간 지연 정보를 표준에서 정한 시간 정보로 오인할 수 있다는 것이다. 또한 실제 전송 지연 시간 Δ 를 전송 클럭의 수로 표현하므로 그림 3에서 δ 로 표현된 [0-전송 클럭 주기]의 값에 해당하는 시간 동기화 오차는 제거할 수 없다.

본 논문에서 제안하는 방식은 추가적인 time-code 전송 없이 전송 지연과 지터를 완전히 없앨 수 있으며, 제안된 방식이 구현되지 않은 일반적인 스페이스와이어 링크 인터페이스와 함께 사용하는데 아무런 문제가 없다.

III. Time-code 전송 지연 및 지터 제거

1. RTC를 이용한 time-code 전송

대부분의 컴퓨터 시스템은 시간 유지를 위하여 RTC (Real-Time Clock)를 사용한다. 본 논문이 제안하는 방식은 스페이스와이어 링크 인터페이스 내부에 전송 클럭에 의해 구동되는 RTC와 이를 이용해서 time-code를 전송하는 Tx. time-code controller를 구현하는 것이다. Tx. time-code controller는 그림 5과 같이 카운터 ST와 R로 이루어지는 RTC와 카운터 R의 값이 설정값 F와 일치할 때 TICK_IN 신호를 발생하여 time-code 전송을 요청하는 회로로 구성되어 있다. 여기서 카운터 ST와 R의 크기를 결정하는 m과 n은 시스템의 요구 사항에

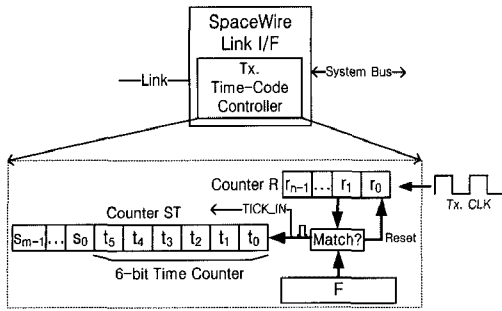


그림 5. Tx. time-code controller의 블록도.
Fig. 5. Block diagram of Tx. time-code controller.

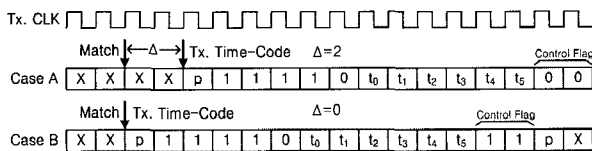


그림 6. 제안된 방식에 의한 time-code 전송.
Fig. 6. Transmission of a time-code by proposed mechanism.

따라 적절히 선택하면 된다.

그림 5에서 RTC의 $\langle r_{n-1} - r_0 \rangle$ 로 구성되는 카운터 R은 전송 클럭 1주기마다 1씩 증가하다가 설정값 F에 도달하면, 즉 match가 발생하면 reset되고 이때 $\langle s_{m-1} - s_0, t_5 - t_0 \rangle$ 로 구성되는 카운터의 ST의 값이 1 증가하며 $\langle t_5 - t_0 \rangle$ 의 값이 time-code로 전송된다. 그러므로 RTC의 $\langle t_5 - t_0 \rangle$ 가 그림 2의 6-bit Time counter로 사용되고 호스트의 개입 없이 스페이스와이어 링크 인터페이스 내부에서 주기적으로 TICK_IN 신호를 발생하는 것으로 생각할 수 있다.

따라서 스페이스와이어 링크 인터페이스의 전송 클럭의 주기를 P_c 라 하고 time-code 전송 주기를 P_t 라 한다면 이들은 $P_t = F \cdot P_c$ 의 관계를 갖는다.

그림 6은 제안한 방식으로 동작하는 경우의 time-code 전송 상황을 보여준다. Case A는 문자나 NULL이 전송되는 도중에 match가 발생해서 전송이 완료된 후에 time-code가 전송되는 상황을 보여 주고, Case B는 문자나 NULL이 전송 완료되는 시점에서 match가 발생되어 time-code가 즉시 전송되는 상황을 보여 준다.

Δ 의 크기는 항상 전송 클럭 주기의 정수배가 되므로 편의를 위하여 앞으로 Δ 의 값을 그에 상응하는 전송 클럭의 수로 표기한다. 그러면 그림 6의 Case B에서는 Δ 가 0이 된다. 만일 데이터 문자 전송을 위해 첫 번째 비트가 송신된 시점에서 match가 발생되면 데이터 문자의 전송이 완료되기 위해 9개의 전송 클럭이 필요하므로 Δ 는 9가 된다. 따라서 Δ 는 [0-9] 범위의 값을 갖게 된다. 이론적으로는 8-bit NULL만이 전송되는 경우에는 Δ 가 1/8의 확률로 [0-7]의 값 중 하나가 되고, 10-bit 데이터 문자만이 전송되는 경우에는 Δ 가 1/10의 확률로 [0-9]의 값 중 하나가 된다.

그림 6의 Case B처럼 전송 지연 Δ 가 0인, 즉 전송 지연이 없는 time-code를 ‘무지연(zero-delay) time-code’라고 부르고 이들의 제어 플래그를 Δ 가 0이 아닌 경우의 time-code의 제어 플래그와 값을 달리하여 구별할 수 있다. 그림 6에서는 무지연 time-code의 제어 플래그 값을 ‘11’로 설정한 경우를 보여

준다. 수신측에서는 무지연 time-code가 수신된 경우에만 이를 이용하여 시각 동기화를 수행하면 정밀한 시각 동기화를 이룰 수 있다. 그리고 TICK_IN 신호가 전송 클럭에 동기화되어 발생하므로 그림 3의 δ 로 표현된 전송 지연에 의한 시각 동기화 오차를 제거할 수 있다.

앞서 언급했듯이 8-bit NULL만이 전송되면 무지연 time-code의 전송 비율은 12.5%(1/8)가 된다. 그리고 10-bit 데이터 문자만 전송되면 무지연 time-code의 전송 비율은 10%(1/10)가 된다. 따라서 제안된 방식을 사용하면 전송되는 time-code 중 무지연 time-code가 발행하는 비율은 10%와 12.5% 사이가 된다.

제안된 방법은 스페이스와이어 표준에 어긋나지 않으므로 제안된 방법이 구현되지 않은 일반적인 스페이스와이어 링크 인터페이스와 접속하여 사용하여도 문제가 발생하지 않는다.

2. 구현 및 시각 동기화 실험

상용의 스페이스와이어 링크 인터페이스 부품을 이용해서는 제안된 방법을 실험 검증할 수가 없다. 따라서 제안된 방법을 실험하기 위한 스페이스와이어 링크 인터페이스 회로를 VHDL언어로 개발하여 Xilinx사의 FPGA XC3S700AN에 구현하였다. 개발된 스페이스와이어 링크 인터페이스는 표준에 정의되어 있는 문자 계층, 교환 계층 그리고 패킷 계층을 준수하며 그 블록도는 그림 7과 같다.

Tx. time-code controller는 1절에 기술된 방법에 따라 time-code를 transmitter에게 전달하고, transmitter는 이를 상대방 스페이스와이어 링크 인터페이스에 전송한다. Receiver로 수신된 time-code는 Rx. time-code register에 저장되고, 이때 수신된 time-code의 제어 플래그 값에 따라 host 측에 interrupt가 요청되어 host가 이를 이용한 시각 동기화를 이룰 수 있도록 하였다.

시각 동기화 실험은 그림 7의 스페이스와이어 링크 인터페이스 2개를 time-code 송신측과 수신측으로 구별하여 각각 개별 FPGA에 구현하여 수행하였다. 송신측에서 time-code를 전송하고, 수신측에서는 무지연 time-code가 수신된 경우에만 time-code에 실려있는 시간 정보를 이용하여 내부 RTC를 설정하였다. 내부 RTC를 설정할 때에는 수신된 시간 정보를

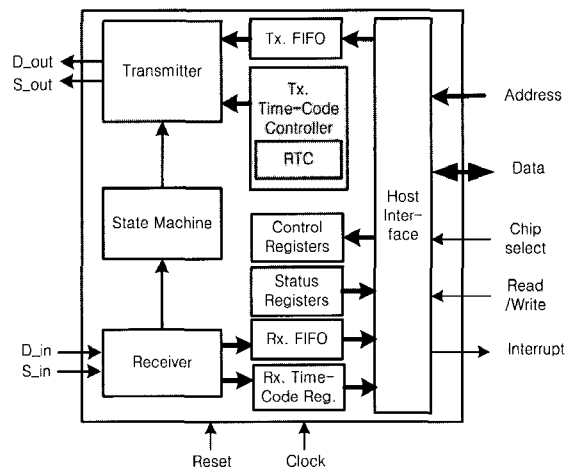


그림 7. 제안된 스페이스와이어 링크 인터페이스 블록도.
Fig. 7. Block diagram of the proposed SpaceWire link interface.

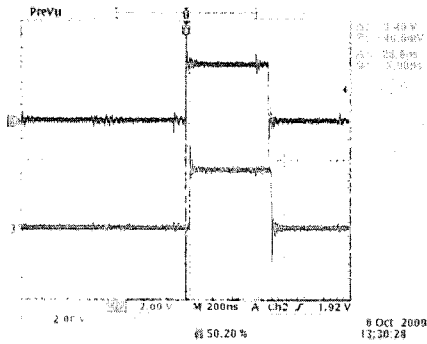


그림 8. 무지연 time-code를 이용한 시각 동기화 결과.
Fig. 8. Result of time synchronization using zero-delay time-code.

그대로 적용하는 것이 아니라 14-bit time-code가 전송되고 수신측에 인식되는데 소요되는 시간을 고려하여 정밀한 시각 동기화가 이루어지도록 하였다.

본 논문에서 제안된 정밀 시각 동기화 방식의 효과는 전송속도와의 관계가 없으므로 임의로 최저 속도인 2[Mbps]로 설정하였으며, FPGA의 동작 클럭 주파수는 10[MHz]로 설정하였다. 두 개의 FPGA가 각각 독립적인 10[MHz] 클럭으로 동작하고 Reset되는 순간이 완벽히 일치하지 않으므로 아무리 정밀한 동기화를 수행하더라도 시각 동기화 오차는 0[sec]에서 동작 클럭의 1주기에 해당하는 100[nsec] 사이의 범위로 발생하게 된다.

동기화 성능 시험을 위해서 그림 5의 카운터 ST의 최하위 비트값이 변할 때 펄스를 출력하도록 하고 이 펄스의 상승 모서리를 측정하여 그림 8에 나타내었다. 위쪽 펄스는 송신측에서 그리고 아래쪽 펄스는 수신측에서 관측된 것이다. 두 펄스의 상승 모서리 간격은 약 28[nsec]로 관측되었으므로 시각 동기화 오차는 약 28[nsec]가 되며 상당히 정밀한 시각 동기화가 가능함을 알 수 있다.

3. 무지연 time-code 전송 비율 실험

실험은 400개의 데이터 문자로 이루어진 패킷을 특정 빈도로 전송하는 도중에 time-code를 초당 10회의 빈도로 50,000회 전송하였을 때 관측된 Δ 의 분포를 측정하였다.

그림 9는 패킷이 10[패킷/초]의 빈도로 전송될 때 관측된 Δ 의 분포를 보여준다. 패킷의 전송 빈도가 매우 낮아 8-bit NULL이 주로 전송되기 때문에 12.5%에 매우 근사한 비율로 무지연 time-code가 전송되었고 Δ 가 8이나 9인 경우는 매우 희박함을 확인할 수 있다.

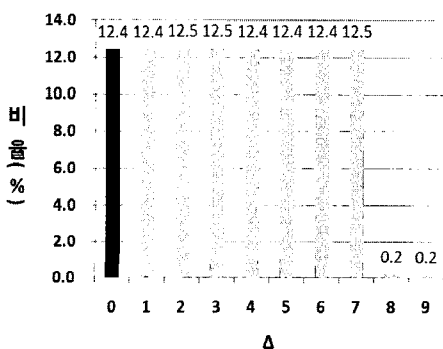


그림 9. 10[패킷/초]의 빈도로 전송하는 경우 Δ 의 분포.
Fig. 9. Distribution of Δ with 10[packet/sec.] packet transmission.

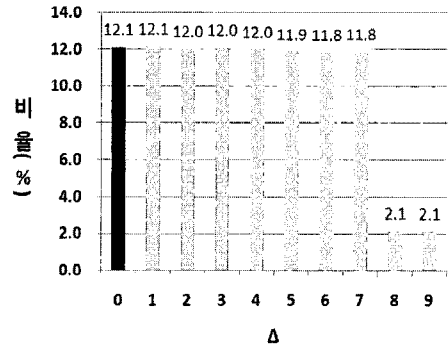


그림 10. 100[패킷/초]의 빈도로 전송하는 경우 Δ 의 분포.
Fig. 10. Distribution of Δ with 100[packet/sec.] packet transmission.

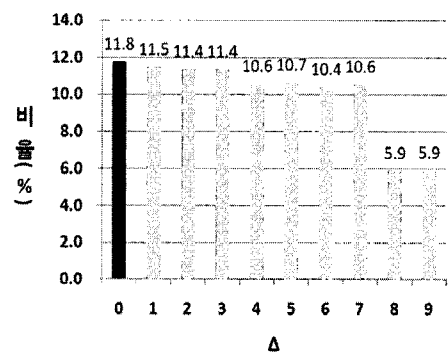


그림 11. 300[패킷/초]의 빈도로 전송하는 경우 Δ 의 분포.
Fig. 11. Distribution of Δ with 300[packet/sec.] packet transmission.

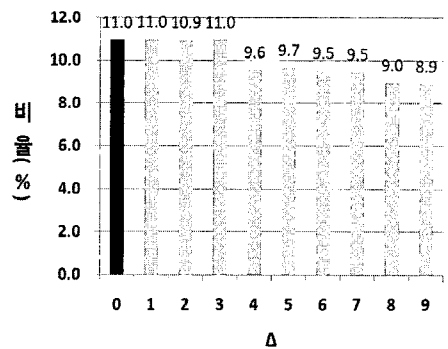


그림 12. 500[패킷/초]의 빈도로 전송하는 경우 Δ 의 분포.
Fig. 12. Distribution of Δ with 500[packet/sec.] packet transmission.

그림 10, 11, 12는 패킷을 각각 100[패킷/초], 300[패킷/초], 500[패킷/초]의 빈도로 전송하는 경우에 관측된 Δ 의 분포를 보여주는데, 패킷 전송 비율의 증가에 따라 무지연 time-code의 전송 비율이 감소하는 것을 볼 수 있다. 그리고 이에 반해 Δ 가 8이나 9인 time-code의 전송 빈도는 증가했음을 확인할 수 있다. 이것은 패킷의 전송 빈도 증가에 따라 8-bit NULL의 전송 빈도는 줄어들고 10-bit 데이터 문자의 전송 빈도가 증가했기 때문이다.

IV. 무지연 time-code 전송 비율 개선

정밀 시각 동기화에 사용할 수 있는 무지연 time-code의 전송률은 시각 동기화 관련 성능에 영향을 준다. 무지연 time-

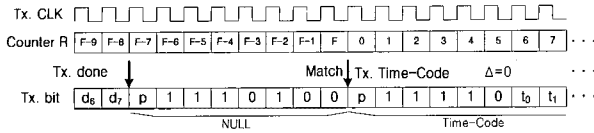


그림 13. 무지연 time-code 전송을 위한 NULL 삽입.
 Fig. 13. Insertion of a NULL for a zero-delay time-code.

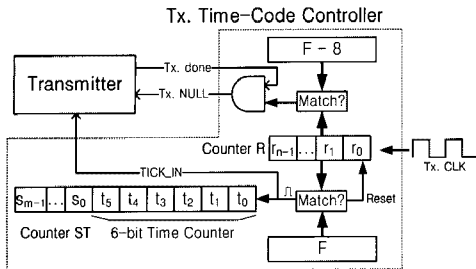


그림 14. 개선된 Tx. time-code controller가 포함된 스페이스와 이어 링크 인터페이스.
 Fig. 14. SpaceWire link interface with improved Tx. time-code controller.

code의 전송 비율이 클수록 전송되는 두 유닛 간의 정밀 시각 동기 상태를 지속적으로 유지할 수 있으며, 시각 동기화 오차를 일정 범위 이내로 유지하는데 필요한 총 time-code 전송량이 줄어들게 된다.

본 장에서는 III 장의 1절에 제안된 방법을 바탕으로 무지연 time-code 전송 비율을 증가시킬 수 있는 방법을 소개하고 그 효과를 실험을 통하여 보여준다.

1. 무지연 time-code 전송 비율 개선 방법

정보의 흐름에 영향을 주지 않는 NULL을 적당한 시기에 추가로 전송하여 무지연 time-code의 전송될 수 있는 기회를 만들면 무지연 time-code의 전송 비율을 증가시킬 수 있다.

그림 13처럼 문자의 전송이 완료되는 시점(Tx. done)에서 카운터 R의 값이 (F-8)이면 다음에 전송해야 할 문자를 전송하는 것이 아니라 8-bit NULL을 전송하여 NULL의 전송이 완료되는 시점과 카운터 R의 값이 F가 되는 시점이 일치하도록 만든다. 그러면 무지연 time-code가 전송되고 이어서 전송이 잠시 미루어졌던 문자가 전송되게 된다.

2. 구현 및 실험

그림 14는 제안된 방법을 구현하기 위한 Tx. time-code controller의 블록도이다. 문자의 전송이 완료(Tx. Done)되고 카운터 R의 값이 (F-8)과 같으면 다음에 전송해야 할 문자를 전송하는 것이 아니라 8-bit NULL을 전송(Tx. NULL)하여 NULL의 전송이 완료되는 시점과 카운터 R의 값이 F가 되는 시점이 일치하도록 구현하였다.

III 장의 3절과 동일한 실험 환경과 방법으로 Δ의 분포를 관측하여 제안된 방식의 효과를 파악하였으며, 실험 결과를 보여주는 그림 15, 16, 17, 18은 각각 그림 9, 10, 11, 12의 실험 결과에 대응한다.

패킷의 전송 빈도가 매우 낮은 경우에는 제안된 무지연 time-code 전송 비율 개선 방법의 효과가 없는 것을 알 수 있다. 이것은 10-bit 데이터 문자의 전송은 매우 적고 8-bit NULL의 전송이 지배적이기 때문이다. 그러나 패킷의 전송 빈도가 높아갈수록, 즉 10-bit 데이터 문자의 전송 비율이 높

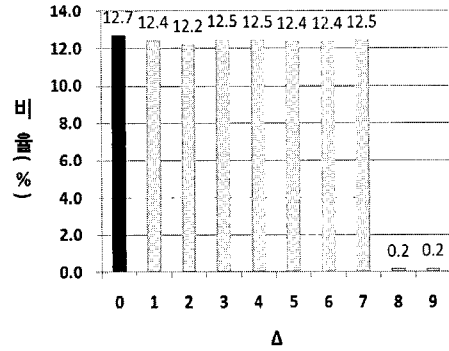


그림 15. 10[패킷/초]의 빈도로 전송하는 경우 Δ의 분포.
 Fig. 15. Distribution of Δ with 10[packet/sec.] packet transmission.

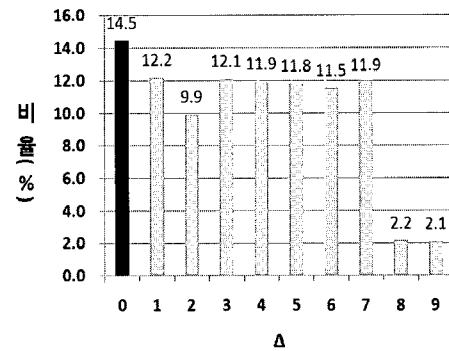


그림 16. 100[패킷/초]의 빈도로 전송하는 경우 Δ의 분포.
 Fig. 16. Distribution of Δ with 100[packet/sec.] packet transmission.

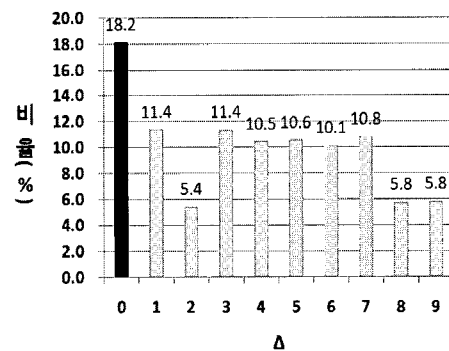


그림 17. 300[패킷/초]의 빈도로 전송하는 경우 Δ의 분포.
 Fig. 17. Distribution of Δ with 300[packet/sec.] packet transmission.

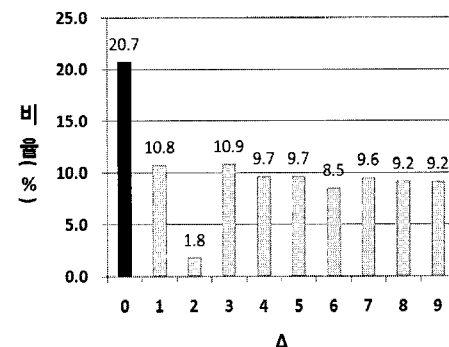


그림 18. 500[패킷/초]의 빈도로 전송하는 경우 Δ의 분포.
 Fig. 18. Distribution of Δ with 500[packet/sec.] packet transmission.

아갈수록 그 효과가 나타나 무지연 time-code 전송 비율이 최대 20%까지 되는 것을 알 수 있다.

V. 결론

스페이스와이어는 고신뢰, 저전력, 고속 데이터 처리를 위해 고안되었다. 구현 비용과 관련 부품의 크기가 작기 때문에 반도체 기술의 지속적인 발달과 위성의 크기와 무게에 따른 발사 비용 부담 때문에 위성의 소형화가 요구되는 현실에 적합한 해결책이라 할 수 있다. 그리고 향후 고신뢰 통신이 요구되는 우주 개발 분야 이외의 분야에도 응용될 것으로 기대되고 있다.

스페이스와이어 표준은 스페이스와이어 네트워크에 접속된 유닛들 사이와 시간 정보 공유를 위해 time-code 분배에 관한 내용을 규정하고 있으며, 본 논문은 스페이스와이어 링크로 연결된 두 유닛 사이의 시각 동기화 오차 문제를 지적하고 이를 극복할 수 있는 정밀 시각 동기화 방법을 제안하였다. 그리고 표준에 부합하는 스페이스와이어 링크 인터페이스에 직접 구현하고 실험하여 그 효과를 보였다.

제안된 방법은 스페이스와이어 표준에 내재되어 있는 time-code 전송에 따른 전송 지연과 그 지터를 완전히 제거할 수 있으며, 제안된 방법이 구현되어 있지 않은 일반 스페이스와이어 링크 인터페이스와 접속이 가능하다.

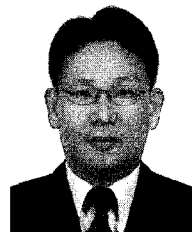
그리고 스페이스와이어 네트워크 구성에 필수적인 스페이스와이어 라우터에도 그대로 적용할 수 있으며, 이를 통하여 네트워크 레벨의 정밀 시각동기화도 이를 수 있을 것으로 기대한다.

참고문헌

- [1] F. Pinsard and C. Cara, "High resolution time synchronization over spacewire links," *Proc. of 2008 IEEE Aerospace Conference*, pp. 1-8, Mar. 2008.
- [2] 조현태 외, "TDOA 기반의 실시간 위치 측정 시스템을 위한 정밀 무선 시각 동기 시스템," *한국통신학회논문지*, 제34권 제1호, pp. 86-97, 2009.
- [3] 조현태 외, "ZigBee를 이용한 무선 네트워크 환경에서의 정밀 시각 동기 기법 설계 및 구현," *한국통신학회논문지*, 제33권 제5호, pp. 561-570, 2008.
- [4] 조정현, "LAN에서 시간에 민감한 트래픽을 위한 시각 동기화 기법," *한국통신학회논문지*, 제33권 제6호, pp. 234-241, 2008.

- [5] 서영수, "CAN 시간동기를 이용한 복수 전동기 동기제어," *제어·로봇·시스템학회 논문지*, 제14권 제7호, pp. 624-628, 2008.
- [6] 박수환 외, "분산 컴포넌트 시스템에서 동기화된 시간의 정밀도를 높일 수 있는 기법," 2008 한국컴퓨터종합학술대회논문집, 제35권 제1호, pp. 500-503, 2008.
- [7] 황소영 외, "인터넷 기반 분산 환경에서 시각 동기를 위한 임베디드 시스템," *정보과학회논문지, 컴퓨팅의 실제*, 제11권 제3호, pp. 216-223, 2005.
- [8] A. Hanzlik and A. Ademaj, "A composable algorithm for clock synchronization in multi-cluster real-time systems," *Proc. of 2006 International Workshop on Intelligent Solutions in Embedded Systems*, pp. 1-12, Jun. 2006.
- [9] K. Sun, P. Ning, and C. Wang, "Secure and resilient clock synchronization in wireless sensor networks," *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 2, pp. 395-408, Feb. 2006.
- [10] K. Arvind, "Probabilistic clock synchronization in distributed systems," *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 5, pp. 474-487, May 1994.
- [11] D. L. Mills, "Precision synchronization of computer network clocks," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 2, pp. 28-43, Apr. 1994.
- [12] IEEE Standard 1588, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems."
- [13] ESA Standard, "ECSS-E-ST-50-12C : SpaceWire-Links, nodes, routers and networks," 2008.
- [14] IEEE Standard 1355, "IEEE Standard for Heterogeneous Interconnect (HIC)."

류 상 문



1992년 금오공과대학교 전자공학과 졸업. 1995년 한국과학기술원 전기및전자공학과 석사. 2006년 동 대학원 전자전산학과 박사. 1995년~2000년 LG전자(주). 2000년~2004년 한국과학기술원. 2006년~현재 군산대학교 제어로봇시스템공학과 조교수. 관심분야는 임베디드 제어 시스템, 실시간 제어 시스템, 결합허용 임베디드 시스템, 스페이스와이어 네트워크.