리플렉티브 메모리 시스템을 이용한 효과적인 네트워크 설계

論 文
54D-6-8

Effective Network Design Using Reflective Memory System

李 聖 雨†
(Sung-Woo Lee)

Abstract - As the increasing integrity of VLSI, the BIST(Built-In Self Test) is used as an effective method to test chips. Generally the pseudo-random test pattern generation is used for BIST. But it requires too many test patterns when there exist random This paper proposes and presents a new efficient network architecture for Reflective Memory System (RMS). A time to copy shared-data among nodes effects critically on the entire performance of the RMS. In this paper, the recent researches about the RMS are investigated and compared. The device named Topology Conversion Switch(TCS) is introduced to realize the proposed network architecture. One of the RMS based industrial control networks, Ethernet based Real-time Control Network (ERCnet), is adopted to evaluate the performance of the preposed network architecture for RMS.

Key Words : Reflective Memory System, ERCnet, Topology Conversion Switch, Ethernet

1. 서 론

리플렉티브 메모리 시스템(Reflective Memory System)은 다중 컴퓨터 환경에서 분산된 노드들이 메시지를 통해서 데이터를 공유하는 과정에서 생기는 다양한 문제를 해결하는 효과적인 방법 중 하나이다[1]. 리플렉티브 메모리의 핵심 개념은 원격의 노드들 사이에서 공유 데이터가 자동으로 복사되어 서로의 메모리에 업데이트 되는 것이다. 만약, 공유를 할 데이터가 노드의 메모리에 업데이트 되면, 그 데이터는 정해진 일정 시간 안에 자동으로 다른 모든 노드들에게 전달된다. 이 과정에서 공유 데이터가 전체 노드에게 자동 복사 되는 시간이 전체 성능을 결정하는 중요한 요소 중 하나이다. 즉, 리플렉티브 메모리 시스템에 적용되는 내부 네트워크의 데이터 전송 속도가 데이터 자동 복사 시간에 결정적인 영향을 미친다.

지금까지 리플렉티브 메모리 시스템에 관한 다양한 연구가 이루어졌다[2][3][4]. 특히, Network Shared Memory (NSM), Shared Common RAM Network(SCRAMNET+), VME Microsystems International Corporation(VMIC)'s VMIC RM은 상업화를 목적으로 구현되어 있는 시스템들이다. 이들은 모두 링형 토폴로지를 기반으로 하여 네트워크가 구성된 공통점을 가지고 있다. 기존 링형 네트워크에서는 데이터가 노드를 지나갈 때 마다 버퍼에 복사된 후 재전송되어 시간지연이 발생하게 된다. 이 시간지연을 줄이기 위해서 NSM은 새로운 링형 네트워크 구조인 단 방향 슬롯 링형 구조(Unidirectional slotted ring architecture)를 제안하고 있으나 구조상 처리할 수 있는 데이터양이 적어진다는 단점이 있다[2]. SCRAMNET+ 와 VMIC 는 기존 링형 네트워크에서 사용하는 방식을 그대로 사용하고 있어 처리 할 데이터가 많아지고 노드의 수가 증가하면 할수록 전체 성능이 저하된다[3][4].

본 논문에서는 리플렉티브 메모리 시스템의 장점을 최대한 살리고 기존 연구들의 단점을 보완하는 새로운 링형 네트워크 구조를 제안한다. 제안되는 네트워크 구조에서는 노드들 사이에서 데이터가 전달될 때, 전 노드로부터 전송되어 온 데이터가 내부 버퍼에 복사된 후 재 전송되는 과정 없이 바로 다음노드로 전달되는 구조로 되었다. 이로써 노드 내부에서 발생하는 데이터 복사와 재 전송의 지연시간을 없앨 수 있다. 또한 본 논문에서는 이 설계 방법을 구현 하기위한 첫 단계로 토폴로지 변환 장치(Topology Conversion Switch)를 소개하고 전체 성능을 검증하기 위해 리플렉티브 메모리 방식의 산업용 통신망으로 개발된 Ethernet based Real-time Control Network (ERCnet)에 적용하여 기존 구조와 성능을 비교한다.

본 논문은 다음과 같이 구성된다. 우선, 2장에서 리플렉티브 메모리 시스템에 대해 간략히 소개하고, 3장에서 기존의 연구된 결과들을 비교 검토한다. 4장에서 본 논문에서 제안하는 구조의 설계 방법과 구현에 관련된 사항을 설명한 후, 5장에서 제안되는 구조를 위해 개발된 TCS에 관해 설명하고, 성능 측정을 통해 검증한다. 6장에서는 ERCnet의 응용사례를 설명하고, 마지막으로 7장에서 결론을 맺는다.

† 교신저자, 正 會 員 : 電力硏究員 發電硏究室 先任硏究員
E-mail : swlee@kepri.re.kr
接受日字 : 2005年 1月 21日
最終完了 : 2005年 4月 14日

2. 리플렉티브 메모리 시스템

리플렉티브 메모리 시스템은 하드웨어적 분산형 공유 메모리(Distributed Shared Memory) 시스템의 한 종류로 다중 노드가 하나의 메모리를 통해 데이터를 공유하는 공유 메모리 시스템과 메시지를 통해 공유 데이터를 전달하는 메시지 전송 시스템의 장점을 모두 가지고 있다. 분산형 공유 메모리 시스템은 소프트웨어적 분산형 공유 메모리 시스템과 하드웨어적 분산형 공유 메모리 시스템으로 나누어지진다. 소프트웨어적 분산형 공유 메모리 시스템은 각 노드별로 지역 메모리를 관리하고, 전체적인 가상 메모리 주소 공간에 의거하며 메시지 교환을 통해 일관성을 유지하는 구조로 공유 메모리 시스템의 장점과 메시지 전송 시스템의 장점을 통합한 구조이다. 반면, 하드웨어적 분산형 공유 메모리 시스템은 각각의 메모리가 노드에 분산되어 있지만 통신망 전체는 하드웨어적으로 동일한 주소 공간에 의거하여 일관성을 유지하며 프로그래머와 사용자에게 투명성을 제공한다.

리플렉티브 메모리 시스템의 경우 한 노드에서 다른 노드의 새로운 데이터를 요구할 때, 통신망에서 공유된 데이터는 모든 노드의 리플렉티브 메모리영역에 존재하기 때문에 요구 노드는 별도의 데이터 요구 없이 필요한 데이터를 실시간으로 사용할 수 있다. 즉, 모든 노드들은 각각의 지역 메모리를 가지고 있지만 결과적으로 하나의 공유메모리에 연결된 형태로 통신을 하게 된다[5]. 리플렉티브 메모리 시스템은 물리적으로 듀얼 포트 메모리를 사용하며 이 메모리를 논리적으로 전역 공유 어드레스에 맵핑하여 사용한다. 이 시스템에서 데이터의 갱신은 여러 다른 종류의 토폴로지를 통해 이루어질 수 있으며 갱신 데이터의 크기 또한 가변적으로 정할 수 있다.

3. 관련된 기존 연구

지금까지 리플렉티브 메모리 시스템에 관한 연구들 중에서 링형 네트워크를 내부 데이터 공유에 사용하는 연구는 Network Shared Memory (NSM), Shared Common RAM network (SCRAMNET+), VME Microsystems International corporation (VMIC)'s VMIC RM이 대표적이다. 그림 1에서 보는 것은 기존 링형 네트워크의 동작이다. 기존 링형 네트워크에서는 노드들 사이에서 데이터가 전달 될 때, 어떤 노드가 전 노드로부터 받은 데이터를 자신의 수신 버퍼에 저장하고 이를 송신 버퍼로 복사 한 후 다음 노드로 다시 전송하는 방식으로 동작된다. 그러므로 데이터가 노드를 지나갈 때 마다 버퍼에 복사되어 재 전송되는 시간지연이 발생하게 된다.

3.1 NSM

Network Shared Memory(NSM)는 그림 1과 같은 구조를 내부 네트워크로 사용하는 리플렉티브 메모리 시스템에서 각 노드의 시간 지연을 줄이기 위해서 그림 2의 왼쪽 그림에서 보는 것 같은 새로운 링형 네트워크 구조인 단 방향 슬롯 링형 구조 (A uni-directional slotted ring architecture)를 제안하고 있다. 그러나 그림 2의 오른쪽 그림에서 보듯

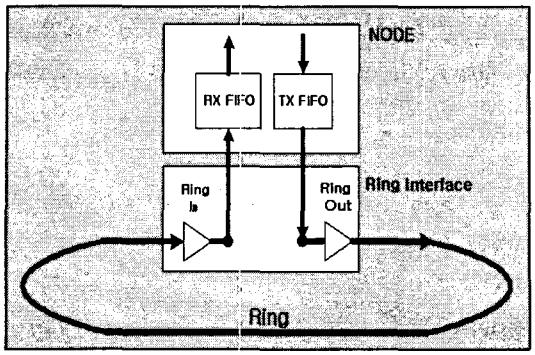

그림 1. 기존 링형 네트워크
Fig 1. The established ring type network

이 NSM 구조는 결국 노드 외부에 존재는 슬릿(FIFO)들의 데이터 용량이 한정되어 있어서 노드가 처리할 수 있는 데이터양이 적어진다는 단점이 있다[2].

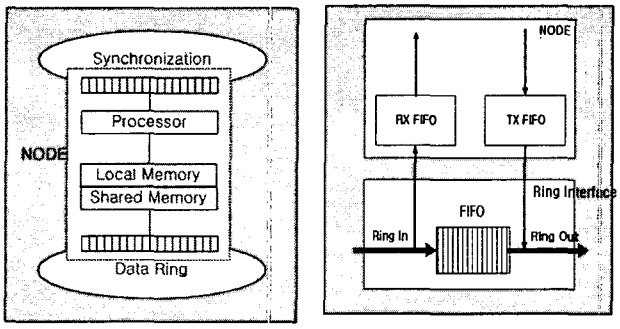

그림 2. NSM 의 구조
Fig 2. A structure of NSM

3.2 SCRAMNET+,VMIC

SCRAMNET+ 와 VMIC는 그림 3에서 수신과 송신 사이에 Transceiver FIFO가 존재하여 그림 1과 같은 기존 링형 네트워크에서 사용하는 방식을 그대로 사용하고 있다. 즉, 데이터가 전달될 때 노드 내부의 송·수신 버퍼에 복사되어 재 전송 된다. 그러므로 처리할 데이터가 많아지고 노드의 수가 증가하면 할수록 각 노드 내부의 데이터 복사 지연으로 인해 전체 성능을 저하시키는 결과가 발생한다[3][4].

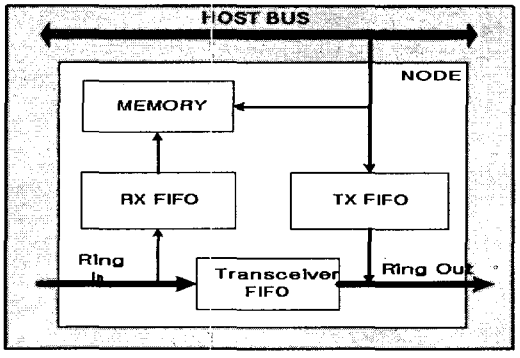

그림 3. SCRAMNET+, VMIC의 구조
Fig 3. A structure of SCRAMNET+, VMIC

4. 새로운 브로드캐스팅 방식의 링형 및 TCS

4. 1 새로운 브로드캐스팅 방식의 링형 네트워크

위에서 검토한 기존 리플렉티브 메모리 시스템의 문제를 해결하기 위해 본 논문에서는 새로운 브로드캐스팅 기반의 링형 네트워크를 제시한다. 그림 4에서 각 노드가 접속되는 링 인터페이스에 스위치를 추가해서 링형 토폴로지를 버스형 토폴로지로 변환 할 수 있다. 어떤 노드의 리플렉티브 메모리에 새로운 데이터가 업데이트 되었다면 이 데이터는 그 노드의 송신 버퍼에 기록되고 동시에 내부 스위치에 의해 송신 상태로 스위칭 한다. 송신 모드의 노드는 데이터를 전체 노드에게 브로드캐스팅 한 후 다시 수신 모드로 스위칭 하게 된다. 이때 브로드캐스팅된 데이터는 각 버스의 버퍼에 복사되어 재 전송되는 과정이 없기 때문에 버스형 네트워크를 통해 전달된 것 같은 효과를 가지게 된다. 데이터를 송신한 노드를 제외한 모든 수신 모드 노드들은 단순히 자신을 지나가는 데이터를 읽어서 자신의 리플렉티브 메모리에 업데이트 한다. 다시 말해, 노드에서 걸리는 지연을 최소화함으로써 리플렉티브 메모리 시스템이 많은 양의 데이터를 고속으로 공유할 수 있게 한다.

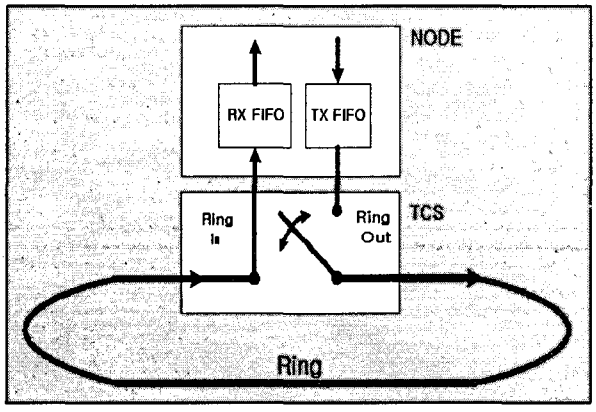

그림 4.새로운 브로드케스팅 기반의 링형 네트워크
Fig 4. A ring type network of the based new broadcasting

4.2 TCS

TCS란 링형의 네트워크를 브로드케스팅 버스형 네트워크로 변환하여 사용하기위해 제안된 일종의 링 인터페이스이다. 이 장치는 브로드케스팅된 데이터를 수신하여 노드에서 사용하는 중앙처리장치(CPU)에 넘겨줌과 동시에 다음 노드로 전달해 줌으로써 브로드캐스팅할 때 각 노드를 지나갈 때 마다 생기는 시간 지연을 줄여주는 기능을 한다. TCS의 블록 다이어그램은 그림 5와 같이 컨트롤러와 아날로그 스위치로 구성되며 컨트롤 신호는 수신 상태인지 송신 상태인지에 따라 주어지며 수신 상태일때는 수신장치로 들어온 데이터를 아날로그 상태에서 바로 송신 장치로 전달함과 동시에 노드의 수신 버퍼로 전달한다. 송신 상태일 때는 수신장치와 송신장치의 연결을 끊고 전송하고자하는 데이터를 송신 장치를 통해 전송하게 된다.

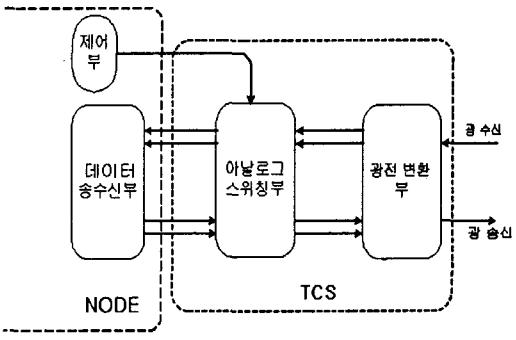

그림 5. TCS 의 블록다이어그램
Fig 5. The block diagram of TCS

5. ERCNet 성능 해석 및 측정

5.1 성능해석

이 섹션에서는 ERCNet의 데이터 업데이트 주기를 수식으로 표현하고, 오실로스크프로 측정한 실제값을 그림 6에 나타내었다. 이를 위한 파라미터들은 다음과 같다.

◇ 노드 개수 : N
◇ i번째 노드의 짧은 주기 데이터 양 : P_i (Byte)
◇ i번째 노드의 긴 주기 데이터 양 : Q_i (Byte)
◇ 한 프레임의 데이터크기 : $F_d = 1024\,Byte$
◇ Cell 크기 : $M_c = 128\,Byte$
◇ 헤더크기(Mheader) : 16(byte)
◇ 한 프레임에 들어가는 cell의 수: $N_{cell} = F_d / M_c = 8Byte$
◇ i 번째 노드에서 i+1 번째 노드 사이 거리: L_i (m)
◇ 1 bit time : T_b
◇ 광전변환시간: T_{oe} (μs)
◇ 전광변환시간: T_{eo} (μs)
◇ Interframe gap : $T_g = 96\,T_b$
◇ 전체 노드에서의 업데이트주기: T_{up}
◇ 노드의 토큰전달시간: T_{H_i} (μs)
◇ 노드의 토큰인식지연시간: T_{dh} (μs)
◇ 노드의 데이터 프레임 생성 시간 : T_{D_w} (μs)
◇ 노드의 토큰프레임 생성시간: T_{T_w} (μs)
◇ 노드의 토큰프레임전송지연시간: T_{D_i} (μs)

이때 ERCNet 에서 어떤 i번째 노드가 전송할 짧은 주기 셀의 개수와 긴 주기 셀의 개수를 각각 N_s, N_l 이라고 표시하면,

$$N_s = \lceil P_i / M_c \rceil$$
$$N_l = \lceil Q_i / 5 \times M_c \rceil$$

라고 나타내어질 수 있다. 이를 통해 짧은 주기 프레임의 개수 N_{F_s}와 긴 주기 프레임의 개수 N_{F_l}는 다음과 같이

$$N_{F_s} = \lceil N_s / N_{cell} \rceil$$
$$N_{F_l} = \lceil N_l / N_{cell} \rceil$$

표시할 수 있다.

그리고 각 노드 사이의 전송 선로에 의한 전송지연(Propagation delay)을 전체 네트워크에서 나타내는 값인 T_{prop} 는

$$T_{prop} = \sum_{i=0}^{N} 5L_i$$

와 같이 나타내어진다. 이때 짧은 주기 한 프레임을 보내는 데 걸리는 시간 T_{fs} 은

$$T_{fs} = (N_{Fs} - 1)\{(F_d + M_{header} + M_{CRC})T_b + T_g\}$$

이고 긴 주기 한 프레임을 보내는 데 걸리는 시간 T_{fl}은

$$T_{fl} = (N_{Fl} - 1)\{(F_d + M_{header} + M_{CRC})T_b + T_g\}$$

이 된다. 하지만 마지막 프레임의 경우 N_{cell} 만큼의 크기로 보내지 않을 수도 있으므로 짧은 주기 마지막 프레임을 보낼 때의 시간은

$$T_{rs} = \{(N_S \bmod N_{cell}) \times M_c + M_{header} + M_{CRC}\}T_b$$

가 되고 긴 주기 마지막 프레임을 보낼 때의 시간은

$$T_{rl} = \{(N_l \bmod N_{cell}) \times M_c + M_{header} + M_{CRC}\}T_b$$

여기서 한 노드가 자신의 통신 프레임을 전송하는데 걸린 시간 $T_{D_{cell}}$은

$$T_{D_{cell}} = T_{f_s} + T_{r_s} + T_{f_l} + T_{r_l} + T_{oe} + T_{eo} + T_{prop}$$

이 되고 전체 네트워크에서의 업데이트 주기 T_{up}은

$$T_{up} = \sum_{i=1}^{N} T_{H_i} = \sum_{i=1}^{N} \{T_{Ath} + (N_{F_s} + N_{F_l})T_{D_{cr}} + T_{D_{cell}} + T_{t_{cr}} + T_{D_l}\}$$

로 표시할 수 있다. 이 식을 이용하여 계산된 결과를 통해 정해진 데이터양과 4개의 노드를 이용하여 위의 식을 테스트해 본 측정결과가 그림 6에 나타나 있다. 그림 6은 토큰 홀드 시간 측정결과를 스캔한 것이다. User LED가 On일 때는 Low 신호이고, Off일 때는 High 신호이다. LED 신호가 High일 때 메니저 노드가 토큰을 받고부터 데이터를 전송하고 난후 토큰을 다음 노드로 전송하기까지(토큰 홀드 시간(T_{H_i} : 400μs))의 시간을 나타낸 측정값이다. 성능 해석을 통해 나온 값을 표1로 나타내어 보면 다음과 같다.

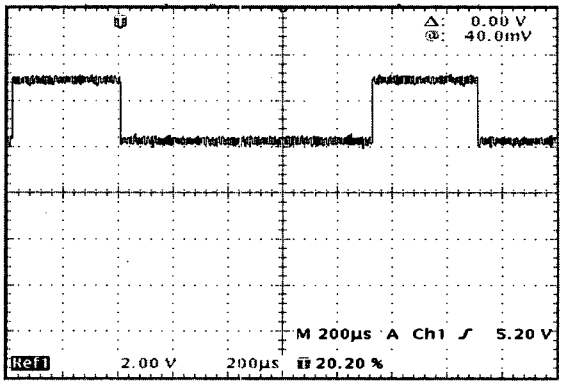

그림 6. 토큰 홀드 시간
Fig 6. The token hold time

표 1. ERCNet의 실험 데이터
Table 1. The experiment data of ERCnet

N	4	N_{F_l}	0.25
P_i	1920	T_{Ath}	10μs
Q_i	1152	$T_{D_{cr}}$	20.2μs
N_s	15	$T_{T_{cr}}$	1.1μs
N_l	2	T_{D_l}	34μs
N_{F_s}	2	T_{H_i}	400μs

이러한 실험치를 위의 식에 대입해보면,

T_{f_s} = (1024+16+32)*8bit*10ns/bit= 85760ns= 85.76μs
T_{r_s} = (7*128+16+32)*8bit*10ns/bit= 75520ns= 75.52μs
T_{f_l} = (1024+16+32)*8bit*10ns/bit= 85760ns= 85.76μs
T_{r_l} = (1*128+16+32)*8bit*10ns/bit= 14080ns= 14.08μs
$T_{D_{cell}}$ = 85.76us+75.52us+85.76us+14.08us+0.2us+0.15us
= 261.47μs
T_{H_i} = 10us + (2+0.25)(20.2us) + 261.47us + 1.1us
+ 34us = 352.02μs

실험을 통해서 나온 T_{H_i} 값은 400μs 이고, 수식을 통해 나온 T_{H_i} 값은 352μs 로서 거의 비슷함을 알 수 있다.

5.2 성능 측정

ERCnet는 물리 계층으로 100Mbps의 패스트 이더넷을 사용하기 때문에 공유되는 데이터 역시 이더넷 프레임에 실려서 전송된다. 이더넷 규격에 의해 프레임의 최대 길이는 헤

더를 포함해서 1500byte 이고, 최소 길이는 데이터가 없더라도 64byte를 보내야한다. 즉, 공유할 데이터가 1500byte를 넘게 되면 데이터를 프레임의 길이에 맞게 잘라서 여러번 전송하게 된다. 또한 64byte 보다 작은 데이터를 공유하게 될 때는 최소 프레임의 길이를 고려하여 전송한다.

본 논문에서는 ERCnet이 TCS 없이 기존의 링형으로 네트워크를 구성하여 동작할 때와 TCS를 사용하여 새로운 브로드캐스팅 기반의 링형으로 구성 할 때의 동작 성능을 실험을 통해 비교한다. 네 개의 노드를 연결하고 처음 노드에서 보낸 공유 데이터가 전체 망을 순환해서 돌아올 때까지의 시간을 측정하여 비교한다. 이 시간을 데이터 순환 시간(Data Rotation Time)이라 한다.

5.2.1 성능 측정 결과 고찰

그림 7에서 비교 해놓은 것처럼, TCS를 사용하여 구성한 네트워크 망에서는 1500 바이트의 데이터를 공유하는데 147 μs의 시간 지연이 걸리고, TCS 없이 보편적인 링형으로 구성한 경우에는 같은 데이터를 공유하는데 608μs의 시간 지연이 걸린다. 즉, 약 560μs이상의 시간 지연 차이가 남을 알 수 있다. 또한 공유할 데이터가 작은 경우와 공유할 데이터가 커질 때의 시간 지연을 비교해 보았을 때 데이터 크기별로 늘어나는 시간 지연은 그 차이가 비슷하다. 이것으로 데이터가 전송될 때 걸리는 시간 지연은 대부분 노드 내부에서 데이터를 복사하고 재 전송하는데 걸리는 차이임을 알 수 있다. 이 측정 결과로 본 논문에서 제안하는 네트워크 구조가 다수의 노드가 접속된 리플렉티브 메모리 시스템에서 데이터 공유 시간 지연을 줄여 전체적인 성능을 향상 시킬 수 있음을 알 수 있다.

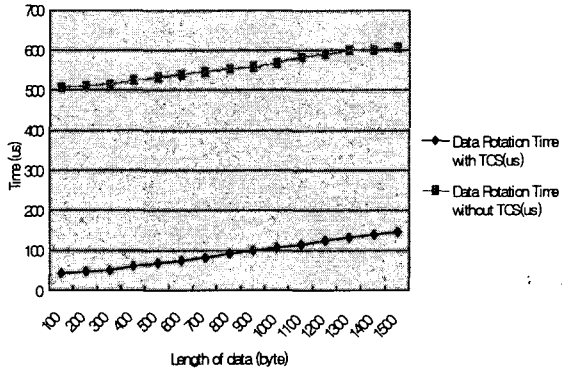

그림 7. 데이터 공유 시간 지연
Fig 7. A data hold time delay

6. 응용 사례 ERCnet

ERCNet은 리플렉티브 메모리 방식의 통신망으로서 산업환경의 분산 제어 시스템으로 사용할 수 있도록 개발되었으며, 원자력 발전소용 분산 제어 시스템으로 사용될 예정이다. ERCNet은 링형을 기본 토폴로지로 사용하며 물리계층으로 광 매체를 사용한다. 매체 제어 방식(Medium Access Control)으로 토큰 패싱 방식을 사용하여 결정적인 동작성과 빠른 속도, 큰 처리 용량을 가진다. 리플렉티브 메모리를 구현하기위해 각 노드들은 자기만의 데이터 영역을 가짐과 동시에 다른 모든 노드에 대한 메모리 영역을 가지게 된다. ERCNet의 노드들은 자신의 데이터를 다른 모든 노드에게 브로드캐스팅하여 전체 노드가 전체 노드에 대한 데이터를 가지고 있도록 한다. 그림 8은 ERCnet의 전체 구성을 나타

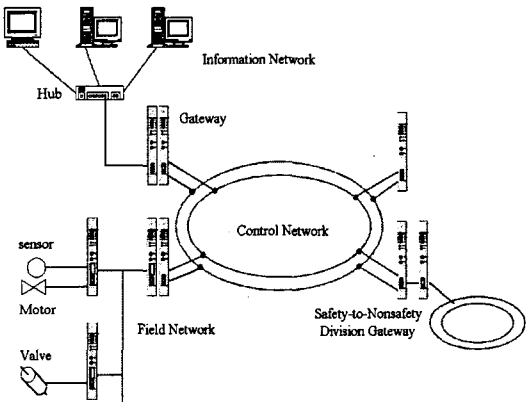

그림 8. ERCnet의 구성도
Fig 8. A structure of ERCnet

낸 것으로 필드망(Field Network)에서 여러 콘트롤러들을 제어하고 그들 사이의 정보를 공유 및 관찰하기 위해서 제어 통신망(Control Network)에서 데이터를 리플렉티브 메모리 방식으로 공유해준다. 정보 통신망(Information Network)은 제어 통신망을 통해 공유되는 데이터를 관찰하고 기기 제어를 위한 명령을 보낸다.

본 논문에서 제시하는 네트워크 설계 방법은 그림 8에서 보인 ERCnet 시스템의 구조 중 제어 통신망에 적용되었다. 즉, 각 노드들을 링으로 연결 할 때 링과 노드사이를 TCS로 연결함으로써 브로드캐스팅 링형을 구성한 것이다.

7. 결 론

본 논문에서는 리플렉티브 메모리 시스템의 성능에 관한 중요한 요소를 설명하고 기존 연구들을 비교 검토하였다. 또한 기존 리플렉티브 메모리 시스템의 성능을 개선하기 위해 고속 링 형 네트워크 구조를 제안하였다. 이 구조는 리플렉티브 메모리에 업데이트된 데이터가 기존 링형에서 전송될 때, 각 노드에서 걸리는 지연을 최소화함으로써 리플렉티브 메모리 시스템이 더 많은 양의 데이터를 고속으로 공유할 수 있게 한다. 제안된 구조를 구현하는 첫 단계로 TCS를 소개하고 실제 리플렉티브 방식의 산업용 통신망으로 개발된 ERCnet에 적용하여 성능을 비교함으로써 전체 성능 향상을 검증하였다. 주어진 실험 환경에서 1500바이트의 데이터를 공유하는데 약 560μs 이상의 시간 지연이 적어지는 것을 확인 하였다. 이 결과로 제안된 구조가 리플렉티브 메모리 시스템에서 처리해야할 노드가 많아지는 경우에 시스템의 전체적인 성능을 향상시킬 수 있음을 검증하였다.

참 고 문 헌

[1] M. Jovanovic, V. Milutinovic, "An Overview of Relfective Memory Systems", IEEE Concurrency, vol.

7, pp. 56-64, 1999.
[2] Ramanjuan R.S., Bonney J.C., Thurber K.J., ''Network shared memory: a new approach for clustering workstations for parallel processing'', Proceedings of the Fourth IEEE International Symposium, pp. 48-56, 1995.
[3] Systran Corporation, ''SCRAMNet+ Overview'', www.systran.com
[4] VME Microsystems Corp., ''VMIC's Reflective Memory Network'', www.vmic.com
[5] Chia Shen, Ichiro Mizunuma, "RT-CRM: Real-Time Channel-Based Reflective Memory", IEEE Transaction on Computers, vol 49, no 11, pp 1202-1214, November 2000.

저 자 소 개

이 성 우 (李 聖 雨)

1960년 9월 29일생. 1987년 대전산업대 전기공학과 졸업, 1989년 건국대학교 대학원 전기공학과 졸업(석사), 1999년 동 대학원 박사. 현재 한국전력공사 전력연구원 발전연구실 선임연구원

Tel : 042-865-5396
Fax : 042-865-5304
E-mail : swlee@kepri.re.kr