

차량용 지능형 네트워크에서의 동기식 이더넷중심 오류 주입 알고리즘 구현[☆]

Synchronize Ethernet-based Fault Injection Algorithm Implementation for Intelligent Automotive Network

장 은 지¹ 김 인 영² 이 응 재^{1*}
Eunji Jang Inyoung Kim Woongjae Lee

요 약

본 논문은 차량용 지능형 네트워크에서 각광 받게 될 이더넷중심의 프로토콜과 데이터 전송 시 임의의 오류에 내성이 있는 알고리즘을 제안하였다. 동시에 시뮬레이션과 실험을 통해 구현과 검증은 시도하여 시스템의 유용성을 증명하였다. 기존의 차량 통신 서비스 시스템에 적용시키기 위한 알고리즘에 대하여 문헌 조사하고 최종적으로 새로운 알고리즘을 구현하였다. 실제 차량용 실시간 데이터의 경우 임의의 값이 불규칙적으로 발생하기 때문에 임의의 데이터를 생성하여 실험을 진행하고 payload의 설정을 표준 형식에 맞추었다. 구현된 알고리즘의 성능을 확인하고자 기존의 알고리즘에서 단일 데이터에서부터 혼합형 (Hybrid-type) 데이터까지 확인하였을 때의 문제점을 확인하고 제안하는 알고리즘의 실효성을 확인하였다.

☞ 주제어 : 동기식 이더넷, 차량용 네트워크, 오류 주입 알고리즘, 지능형 시스템

ABSTRACT

In this paper, we propose the protocol of Ethernet that will receive a popular interesting in the automotive intelligent network, it also attempts to implementation and verification through simulation and experiments to propose a fault tolerance algorithm when the data transfer on it. It has proven the usefulness of the system in order to apply toward an existing automotive communication system. In the case of actual real-time data for automotive industry, we generated a randomly-generated data which is the set of payload into a standard format to complete the experiment. Among the implemented existing algorithms performance, we confirmed the effectiveness of all range from a single data to mixed (Hybrid-type) data, to verify the proposed algorithm.

☞ keyword : Synchronized ethernet, Automotive intelligent Network, Fault tolerant algorithm, Intelligent system

1. 서 론

스마트카로 비견되는 차량 통신 시스템에서 대용량 데이터 전송에 대한 요구사항이 계속하여 증가함에 따라 차량 내·외에 운영 가능한 고 성능, 대 용량의 통신망을 필요로 하게 되었다 [1]. 최근의 전기차와 스마트 차량에는 자동차의 브레이크나 자세 제어 등 안전성을 높여주는 기능 위주의 ECU에서 차선유지 및 이탈경고, 주차 보조 등 다양한 편의성을 제공하는 ECU가 개발되어 탑재

되고 있는 실정이다 [2]. 원격차량진단, 긴급 구난등과 같은 차량 통신 서비스에서 요구되는 데이터 크기 또한 증가하고 있으며 안전, 편의를 동시에 만족하는 인포테인먼트 (Infotainment)기능을 중심으로 발전하고 있다. 그 결과 해외 기업에서는 우선적으로 차량용 1Gbps 대역에서 동기식 이더넷 기반의 네트워크 적용을 추진하고 있다 [2]. 해외 자동차 업계에서는 차량용 AutoSAR, GINI등과 같은 소프트웨어 플랫폼과 동기식 이더넷 환경 구축을 위한 공동 프로젝트를 시작하여 차량용 전자 기기를 위한 개방형 시스템과 인터페이스의 표준화를 목표로 연구를 진행하고 있다.

차량 데이터의 전송 시 주로 데이터를 다루는 것은 영상기반 장애물 인식 시스템, 자동주차시스템, 차량용 블랙박스 등 다양한 응용분야에서 대용량의 정보 처리 시스템이 요구되고 있어 2020년 전후로 차량용 네트워크

¹ Dept. of Digital Media, Seoul Women's University

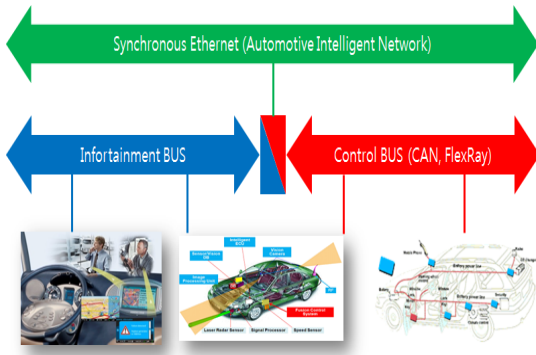
² SH C&C

* Corresponding author (wjlee@swu.ac.kr)

[Received 31 March 2016, Reviewed 2 April 2016, Accepted 1 June 2016]

☆ This work was supported by a research grant of Seoul Women's University in 2016.

백본 망의 표준화를 준비하는 입장에서 산업용 이더넷 국제표준 RAPIenet (Real-time Automation Protocols for Industrial Ethernet) 승인을 비롯하여 차량용 이더넷을 응용한 안전 시스템의 개발 및 양산 기술 적용이 진행되고 있다 [3].



(그림 1) 동기식 이더넷중심 차량 지능 네트워크
(Figure 1) Synchronous Ethernet for Automotive Intelligent Network

동기식 이더넷 통식은 1Gbps급 대용량 데이터 처리를 최초로 시도하는 프로토콜을 적용하기 위하여 지금까지 시도되지 않았던 각종 데이터의 흐름을 효율적으로 활용하고자 하며, 기존의 프로토콜들이 가지는 약점을 보완하기로 하였다. 이러한 연구는 현재 상용화되어있는 고급차량의 OEM사들의 제품(BCM89810) 중 한 가지를 선정하여 [4]의 페이로드(Payload) 카드 포맷에서 2개의 스위치를 이용하여 총 4개의 포트를 제어하는 방식을 도입하여 진행 한다. 기존의 문헌 [5]에서 제시하였던 단순 네트워크카드 (NIC; Network Interface Card)의 듀얼 포트 (Dual-port) 의 경우 하드 리얼리티 시스템에서 중복성을 부여하는 측면에서 4개 이상의 포트를 이용한 NIC를 제어하는 알고리즘보다 안정하여야 하며 따라서 이번 연구에서는 동기식 이더넷중심의 차량용 페이로드를 4개의 포트를 이용하여 데이터를 전송하는 방법을 제시하고 그것의 실효성을 검증하는 것을 목표로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 조사하고 3장에서는 제안하는 알고리즘을 기술한다. 4장에는 Fault tolerance 시스템을 통한 실험 방법을 기술하고 5장에서는 실험의 결과를 정리하였다. 끝으로 그 내용을 바탕으로 6장에서 결론을 맺는다.

2. 관련연구

1960년대 초반부터 엔지니어들은 마이크로 전자 기기의 설계에서 Fault Injection의 결합이 중요한 것을 인식했다. 그러나 이때 그들은 시뮬레이션 타이밍 설정이 늦었고 당시의 시뮬레이션은 모델의 완성 직전에 수행되었다 [6].

초기의 임베디드 시스템을 비롯하여 현재도 역시 간단한 임베디드 시스템에서는 시뮬레이션을 통한 검증이 필요 없는 간단한 시스템들이 많았다. 그러나 임베디드 시스템의 사용이 증가와 응용분야를 차량용으로 진출함에 따라 고정에서 불량품에 따른 오류비용(Cost of Failure)가 증가하기 시작했고 이에 따라 소프트웨어 및 시뮬레이션 보드의 제작을 통해 먼저 완성될 하드웨어의 성능 및 문제를 테스트 하는 시스템이 절실하게 필요하게 되었다 [7].

그 결과 나타난 Fault Injection 방법은 총 3가지의 범주로 분류된다. 시뮬레이션 기반의 방법, 소프트웨어로 구현된 방법과 하드웨어 기반의 기술적 방법으로 크게 나누어진다. 설계의 초기 단계에서는 시뮬레이션 기반의 방법을 이용하며 시험모델을 이용하거나 제품의 제작 직전의 단계에서는 하드웨어 기반의 기술적 방법을 주로 이용한다 [8].

설계자들은 대개 상위레벨에서 Petri net [9], Queuing model 등을 이용해 DPE (Designers usually Perform dependability parameter Estimation)를 진행하게 되지만 설계자들은 원하는 결과를 얻기 위해서 점진적으로 하위레벨 모델의 세부사항에 다가야 한다 [13]. 이로 인한 오류 주입 알고리즘에 의한 소프트웨어를 통해 검증된 시스템들이야말로 검증절차에 합격한 신뢰성이 있다고 할 수 있다 [14].

이미 해외 선진기업에서는 많은 이더넷 프로토콜 및 인터넷 계층을 연결하는 각종 PHY 디바이스들이 출시되고 있으나 차량용 동기식 이더넷중심으로 소프트웨어 개발은 아직 상당부분 연구가 필요한 상황이다. 특히 소프트웨어의 임베디드기능 부재는 물론 하드웨어 제품도 역시 미약하여 두 분야 모두 개발이 필요하다 [10].

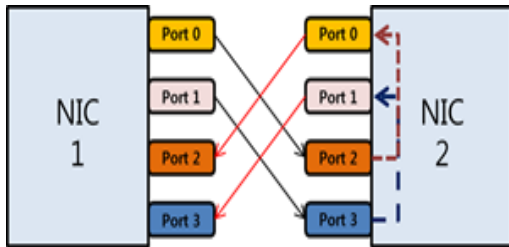
대용량 데이터 전송이 가지는 장점이 점차 증가하고 자율주행의 기능이 각광을 받고 있는 시점에서, 그러나 패킷 전송은 원격지와 정확한 동기화를 이루어야 하는 어플리케이션을 다룰 경우 수용의 한계를 가지고 있다. 이에 IEEE에서는 시간 프로토콜의 표준화를 추진, IEEE

1588이 이를 근간으로 하고 있다 [10][11].

현재 IEEE에서 표준화가 진행 되고 있는 동기식 이더넷 기술은 이더넷 기술 활용 측면에서 관심을 받고 있는 새로운 기술이다. 기존의 비동기식 이더넷이 제공할 수 없는 QoS (Quality of Service)기능을 대신하여, 사이클 개념을 사용 한다 [12].

3. 제안된 알고리즘

차량용 데이터의 종류는 각종 센서를 통하여 카메라 영상 또는 비트단위의 운전자의 조정으로 유닛별로 전송 또는 제어되어야 한다. 차량용 페이로드(Payload) 카드로부터 포맷을 정해지고 어떠한 포맷일지라도 잘못 될 경우 치명적인 문제를 가지게 됨으로 오류를 발생하지 않는 무결성의 원칙을 가지고 있다. 우리는 페이로드 (Payload) 카드 포맷에서 2개의 스위치를 이용하게 되지만 그림 1에서는 생략하고 여 총 4개의 포트를 동시에 전송하고 제어하는 방식을 도입하여 실험하고자 한다.



(그림 2) 제안된 이더넷 통신 구조
(Figure 2) Proposed Ethernet-based Communication. Structure

제안된 이더넷 통신 구조에서는 실험을 위하여 제한적으로 통신 구조를 가정하였으며 오류 주입을 위한 일대일대응에 의한 전단사함수 (Bijective Function)의 개념으로 실험하였다. 초기 데이터의 개수와 NIC의 개수를 정의하도록 되어있으며 그 이유는 실험에 사용할 데이터의 개수와 NIC의 개수를 앞으로도 확장을 시도할 수 있도록 유동적으로 조정할 수 있도록 하였다. 차량용 통신 네트워크상에 제어가 필요한 기기의 개수가 앞으로 증가 될 것을 예상하고 또한 사용자마다 유동적일 수 있기 때문에 이와 같이 설정하였다.

실제 시뮬레이션을 통한 오류 주입 실험에서는 각각의 NIC에 페이로드의 숫자를 100, 5,000, 10,000, 16,384,

65,536 (Packet) 데이터용량을 증가하면서 처리하였다. 이 중에서 5,000, 10,000 (Packet)의 경우 임의의 수치를 정해 실험한 것이고, 16,384, 65,536 (Packet) 경우 추후 실험을 진행할 이미지의 픽셀 개수와 동일한 개수의 데이터를 처리한 결과를 비교하기 위한 것으로 각각 128 x 128의 이미지와 256 x 256의 이미지 픽셀 수를 의미한다.

시뮬레이션의 초기화 함수는 각각의 포트에 1~4,095 (Packet) 값 중에서 한 값으로 임의의 IP를 생성 [15][16]하도록 하고 오류가 주입되는 시점을 생성한다. 만일 하나의 파일의 데이터가 5,000 (Packet) 일 때 이 중에서 하나의 데이터에 임의의 NIC에서 자동으로 문제가 발생하도록 설정하였다.

설정된 근거는 보편타당성을 위하여 임의의 NIC 중 0 번째의 NIC를 피하여 주입되며 초기 로딩 데이터는 문제가 없다는 가정 하에 역시 첫 번째 데이터에는 문제가 없도록 설정 했다.

각각의 NIC는 그림 2과 같이 상호 연결이 되도록 전단사함수의 개념에서 기본적으로 설계 되었다. 포트 0, 1은 전송을 담당하고 포트 2, 3은 수신을 담당하는 포트로 구현하여 이를 충족하고 필요시에는 양방향 통신을 가능하도록 하였다. 이때 만약 데이터를 수신하게 되면 NIC 1은 포트 2, 3으로 수신한 데이터를 0, 1에 복사를 하고, 만약 포트 0과 1의 데이터가 다른 경우 NIC 1의 포트 0과 1의 CRC 코드와 NIC 2의 포트 0, 1의 코드를 비교하여 이전의 NIC 포트 번호와 동일한 것이 정상 데이터라고 판별하여 그 값으로 모든 값을 변환하는 시뮬레이션의 기본 원칙을 따르게 된다.

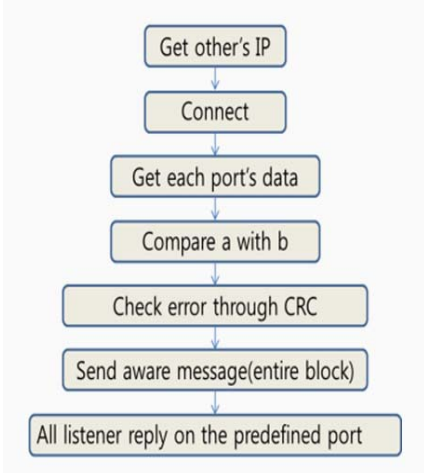
그림 2과 같이 NIC를 설계하게 되면 현재의 매쉬 (Mesh) 구조를 링(Ring) 구조로 변환 시 마지막의 NIC 정보를 다시 처음의 NIC에 전송하면 되게 된다.

차량용 통신 구조가 CAN, FlexRay, MOST로 발전하는 가운데 새로운 동기식 이더넷통신구조가 얼마나 혁신적인, 효율적인 구조로 시장에 자리 잡게 되는 것은 아직 미지수이나 이러한 실험을 반드시 거쳐야 하는 것이다.

4. Fault Tolerance 시스템

순환중복검사 (Cyclic Redundancy Check)를 일반적으로 네트워크 등을 통하여 데이터를 전송 시 전송된 데이터에 오류가 있는지를 확인하기 위한 체크 값을 결정하는 방식이지만 두 번에 걸친 비교와 함께 오류를 덧붙여 전송하며, 이진법에 의한 진행을 다수의 검사를 진행할

수 있다 [17] [18]. 검사방식의 신빙성을 위하여 CRC 테이블을 만들어 변수 화시키고, 데이터가 들어오면 다항식을 위하여 테이블 탐색을 통해 CRC를 결정하게 된다.



(그림 3) 기존 알고리즘의 Flow Chart
(Figure 3) Existing Algorithm's Flow Chart

기존의 알고리즘 [4]의 경우 듀얼포트 NIC를 이용하여 1개의 포트만을 사용하였다 [4]. 그러나 새롭게 적용되는 제안된 알고리즘과 임베디드된 하드웨어는 적어도 페이로드를 4개 이상의 포트를 이용하여 차량내부 혹은 외부, 내외부의 통신 환경이 구축이 필요하다 [19].

(표 1) 제안된 알고리즘의 의사 코드
(Table 1) Proposed Pseudo-code

```

Algorithm
[DATA_NUM = number of packet, NIC_NUM = number of NIC]
[init NIC]
make random port IP, range:1-4095
init fault injecting timing.
(random iteration value, choose random NIC)
iter =0;
for(iter<DATA_NUM)
  set each iteration's data
  for (current_NIC_NUM = 1) to NIC_NUM
    connect next NIC
    get other NIC's data
  if (iter == random iteration value
    
```

```

&& current_NIC_NUM == choose random NIC
&& first)
inject fault
endif

get crc16 result, using ccitt
compare port data(port0, port1)
if (compare result !=0)
  fault data repair
endif

current NIC := current NIC+1
endifor
iter := iter+1
endifor
    
```

표 1은 오류 주입 검증을 위한 제안된 알고리즘의 기본 골격을 표현하고 있다. 이때 구현한 NIC 페이로드의 순서도는 상단의 그림 2와 같이 설명할 수 있다.

(표 2) 데이터의 전송 방법
(Table 2) Data Transmission Methodology

```

Procedure get data
[PORT_NUM = number of PORT]
[set port data]
for(j=2) to PORT_NUM
  transport NIC[current_NIC].port_data[j] to
  NIC[current_NIC].port_data[j-2]
  j:= j+1
endifor
    
```

표 2는 포트간 데이터의 전송을 표현하고자 하는 의사 코드이며 이를 표 1에서 언급한 제안된 알고리즘의 기본 골격에 따라 깊은 연관성을 가지며 설명되어 질 수 있다.

5. 실험 및 검증

5.1 실험 조건

제안하는 알고리즘이 실제 구현 가능성과 유용성을 확인하기 위하여 다음과 같이 실험을 진행하였다. 우선 데이터의 전송 시 동작여부를 확인하기 위하여 페이로드에 컨트롤 데이터(제어데이터; **Control data**)와 어시스턴

스 데이터(보조데이터; Assistant data)를 구분하였고 정수형 데이터와 문자형 데이터를 생성하여 사용하였다.

이때 실제 사용되는 데이터의 사이즈는 상황에 따라 그 사이즈가 달라질 수 있으므로 정수형의 경우에 0부터 255까지의 코드가 있다고 가정하여 컨트롤 데이터를 받아 처리를 한다고 가정, 프로그램의 수행 시 시간에 따라 데이터의 내용이 달라 질 수 있도록 데이터를 생성하였다.

(표 3) 정수(Int)형 데이터 구분
(Table 3) Int-type Data

Number of Packet	100	5,000	10,000	16,384
epoch	3,000	3,000	3,000	3,000
Data size (byte)	-	-	-	74,908

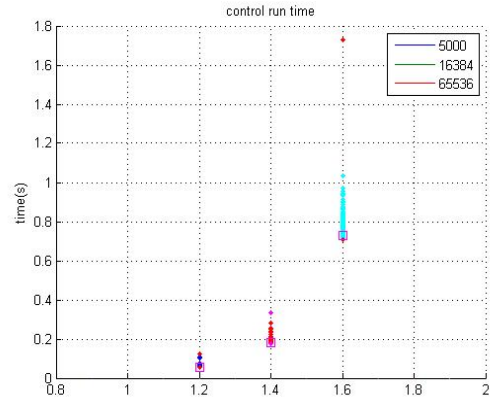
(표 4) 문자(Char)형 데이터의 구분
(Table 4) Char-type Data

Number of Packet	100	343	5,000	10,000	16,384
epoch	3,000	3,000	3,000	3,000	3,000
Data size (byte)	-	-	-	-	81,689

위에서 보는바와 같이 표 3 와 표 4 에서는 정수(Int)형 데이터와 문자(Char)형 데이터에 관하여 실험을 보여주고 있다. 동시에 메시지 데이터의 처리를 가정하여 메시지의 길이와 그 내용을 시간에 따라 달라 있도록 srand() 함수를 이용하여 임의의 데이터를 생성 하고, 생성된 데이터가 정상적인 문자 데이터를 값을 나타낼 수 있도록 아스키 코드의 값을 지정하였다. 데이터의 생성 및 처리 확인 시 가독성을 높이기 위하여 소문자 데이터만 생성이 되도록 수치를 조정했다.

5.2 실험 결과

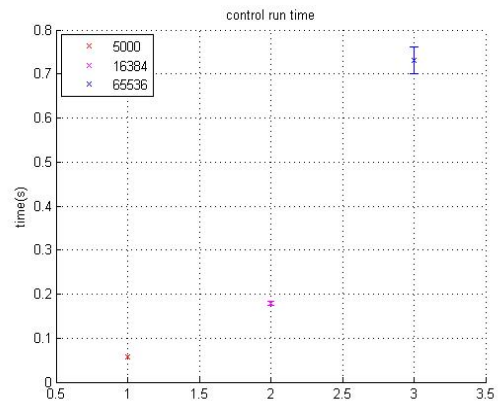
대용량 데이터를 처리하는 방식을 확인하고자 실험의 목적이 있기 때문에 실험을 가장 많은 용량의 결과로 보아야 하지만 실시간 처리상황에서 오류 주입에 의한 Fault Tolerance 실험에 근거하여 진행하였다.



(그림 4) 컨트롤 데이터 처리시간
(Figure 4) Control Data's Processing Time

그림 4은 비트 형태의 데이터를 처리한다는 가정 하에 숫자 데이터를 생성하여 오류를 복구하고 데이터를 전송하는데 걸리는 시간을 측정된 결과이다. 각각의 점들은 한 데이터 세트를 모두 처리하는데 걸린 시간을 나타낸 것으로 오류는 임의의 순간 임의의 NIC에서 한 차례 발생하게 한 것이다.

총 3,000번의 N값을 가지는 실험 (N=3,000)을 진행했으며 그림에 따르면 대부분의 실험은 유사한 값을 보였으나 평균값과 많이 차이가 나는 문제점을 찾을 수 있었다. 그 예로 5,000 (Packet) 의 데이터와 16,384 (Packet)의 데이터를 이용한 실험에서는 두드러지지 않았으나 65,536 (Packet) 의 데이터를 이용한 실험에서 평균 수행 시간보다 약 1sec 가량 지연된 결과가 나타났음을 볼 수 있다.



(그림 5) 컨트롤 데이터의 표준편차
(Figure 5) Control Data's Standard deviation

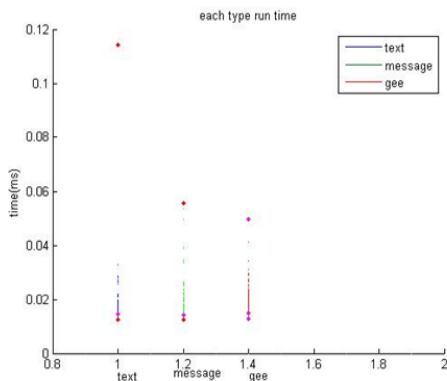
그러나 대체로 모든 실험 결과는 평균시간과 유사한 모습을 보였으며 그림 5의 실험 결과의 표준편차를 살펴 보았다. 이때 앞서 그림 4의 결과와 마찬가지로 65,536 (Packet)의 데이터를 처리한 결과에서 다소 편차가 높게 나왔으나 그 이하의 데이터 처리 시간은 편차가 작게 나왔음을 알 수 있다.

앞선 실험결과와의 이해를 돕기 위하여 각각의 데이터 처리 시간의 평균값을 표 5에 정리하였다. 각종 Payload로부터 전송되는 데이터의 평균 처리 시간은 컨트롤 데이터가 증가할 수록 수행시간 (Runtime) 과 패킷(Packet)이 반드시 증가하는 것을 알 수 있게 되었다. 이는 향후 차량용 데이터를 처리함에 있어서 페이로드에 부하되는 용량을 처리하는 방식을 결정짓는데 중요한 요소가 될 것이다.

(표 5) 컨트롤 데이터 평균 처리 시간
(Table 5) Control Data's Avg. Processing Time

Control	5,000	16,384	65,536
Runtime(s)	0.015454	0.028329	0.044791
Packet(us)	3.0908	2.8329	2.7338

동시에 컨트롤데이터와 기타 텍스트 데이터를 혼합하여 새로운 형태로 혼합형 (hybrid)데이터를 전송시 보이는 장단점을 파악하였다. 통계학적으로 바라 볼 때 혼합형 데이터의 전송속도가 다소 느려지는 것을 예상하였으나 현저한 속도 저하가 없는 것으로 보아 오히려 네트워크의 건전성을 보완하였다는 이론이 타당하다.



(그림 6) 혼합형 데이터의 성능 비교
(Figure 6) Hybrid-type Data Performance

그림 6의 결과는 혼합형(Hybrid-type) 데이터, 즉 컨트롤, 메시지, 텍스트 데이터를 모두 수행시간별로 비교하여 성능을 보여주고 있다. Text형 데이터 (음원의 가사)는 소녀시대의 Gee 가사를 이용한 데이터로 한글과 영어가 혼합된 메시지 데이터이다. 음원 데이터는 가독성을 위해 한글, 영어의 단어 혹은 절로 나누어 343개의 데이터로 나누었고 수치의 비교를 위해 컨트롤 데이터와 메시지 데이터 역시 343개의 데이터 세트를 전송한 결과를 다시 실험했다.

(표 6) 혼합형 (Hybrid-type) 데이터의 비교
(Table 6) Comparison of Hybrid-type Data

	Control	Message	Text
수행시간(s)	0.014434	0.014331	0.01476416

표 6에 따르면 컨트롤 데이터와 메시지 데이터의 전송에 있어 메시지 데이터의 전송이 더 빨랐으며, 혼합형 (Hybrid-type) 데이터의 전송이 더 오랜 시간이 걸리는 모습을 보였다.

6. 결 론

본 논문은 차량용 동기식 이더넷중심의 환경변화를 준비하는 단계에서 오류 주입으로 시스템의 건전성을 확인하였다.

실험을 위하여 구성된 차량용 지능형 시스템은 향후 스마트카 등의 등장으로 더욱 더 발전이 될 예정이며 우리의 오류 주입 검증은 향후 더욱 더 각광을 받을 것으로 이해된다. 구현과 검증에서 제안된 알고리즘에 의하여 컨트롤 데이터, 메시지 데이터의 성능을 비교하였을 때 컨트롤 데이터의 처리속도가 메시지 데이터의 처리 속도보다 조금 더 빠른 결과를 보였다. 또한 메시지 데이터의 처리 속도가 더 빠른 결과를 보였다는 점에서 적은 양의 데이터의 전송에서는 메시지 전송이 더 빠르게 수행된다고 생각 할 수 있었다.

실제 시스템의 CRC연산을 이용하여 데이터의 전송시 오류를 검출하고 그 오류를 복구하는데 걸리는 시간은 혼합형 (Hybrid-type) 데이터에 비하여 한 종류의 단일 데이터로 이루어진 경우에 더 좋은 성능을 기대할 수 있었다.

참 고 문 헌 (Reference)

- [1] Sun, Jie, Z-h. Wu, and G. Pan, "Context-aware smart car: from model to prototype." *Journal of Zhejiang University Science A* 10.7 pp. 1049-1059, 2009.
<http://dx.doi.org/10.1631/jzus.A0820154>
- [2] Lamon, Pierre, S. Kolski, and R. Siegwart, "The SmartTer-a vehicle for fully autonomous navigation and mapping in outdoor environments." In *Proceedings of CLAWAR*. 2006
- [3] Yoon, Geon, et al., "RAPIenet based redundancy control system." *Control, Automation and Systems (ICCAS)*, 2011 11th International Conference on. IEEE, 2011
- [4] H. Lee, H. Jabbar, S. Lee, S. Choi, Q. Lie, I. Kim, S. Choi, D. Park, S. Min, Y. Lee, and T. Jeong, "Ethernet-based Communication Architecture Design and Fault-Tolerant System", proceedings of the international multi conference of engineers and computer scientists 2012, vol 2, IMECS 2012, Mar 14-16, 2012, Hong Kong
- [5] Broadcom BCM89810 Automotive Physical Layer Technology
- [6] T. A. Delong, B. W. Johnson, and J. A. Profeta Iii, "A fault injection technique for VHDL behavioral-level models." *IEEE Design & Test of Computers* 4, pp. 24-33, 1996.
<http://dx.doi.org/10.1109/54.544533>
- [7] M. Rahmani, K. Tappayuthpijam, B. Krebs, E. Steinbach, and R. Bogenberger, "Traffic Shaping for Resource - Efficient In - Vehicle Communication." *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, Nov 2009.
<http://dx.doi.org/10.1109/tii.2009.2019127>
- [8] A. Benso, P. Prinetto, M. Rebaudengo, M. Sonza Reorda, "A Fault Injection Environment for Microprocessor-based Boards" *Politecnico di Torino Dipartimento di Automatica e Informatica Torino, Italy, Proceedings on Test Conference*, 1998. pp. 768 - 773, Oct. 18-23, 1998.
<http://dx.doi.org/10.1109/test.1998.743259>
- [9] Leen, Gabriel, D. Heffeman, and A. Dunne. "Digital networks in the automotive vehicle." *Computing and Control Engineering Journal*, vol. 10, no. 6, pp. 257-66, 1999.
<http://dx.doi.org/10.1049/cce:19990604>
- [10] Loeser, Jork, and H. Haertig. "Low-latency hard real-time communication over switched Ethernet." *The 16th Euromicro Conference on Real-Time Systems, (ECRTS 2004)*, 2004.
<http://dx.doi.org/10.1109/emrts.2004.1310992>
- [11] [Online access] "IEEE 1588 Protocol"
<http://blog.naver.com/golsm21/120148083777>
- [12] S.S. Ahn, W.H. Kim, "Pseudo MOST protocol based Car Network Integration Gateway System Development," *Proceedings of IEK*, vol 48, SC no. 6, 2011,
- [13] H-T Lim, L. Volker, D. Herrscher, "Challenges in a Future IP/Ethernet-based In-Car Network for Real-Time Applications", *The 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 5-9 June 2011, New York.
<http://dx.doi.org/10.1145/2024724.2024727>
- [14] T. V. Ramabadran, , and S. S. Gaitonde. "A Tutorial on CRC computations." *IEEE Micro* vol. 4, pp. 62-75, 1988.
<http://dx.doi.org/10.1109/40.7773>
- [15] L-T Wang, C. E. Stroud, N. A. Touba, "System-on-Chip Test Architectures: Nanometer Design for Testability (Systems on Silicon)", Chapter 3, *Fault-Tolerant Design*. 2000
- [16] I. Kim, and T. Jeong, "Smart Car's Application for Educational Use and Standard Analysis", *Journal of Korean Society of Computer Education*, Winter Symposium in 2013
- [17] Griffiths, Georgia, and G. Carlyle Stones. "The tea-leaf reader algorithm: an efficient implementation of CRC-16 and CRC-32." *Communications of the ACM* 30.7, pp. 617-620, 1987.
<http://dx.doi.org/10.1145/28569.28572>
- [18] Ghani A. Kanawati, Nasser A. Kanawati, and Jacob A. Abraham, Fellow, IEEE, "FERRARI: A Flexible Software-Based Fault and Error Injection System", *IEEE Transactions on Computers*, vol. 44, no 2, Feb. 1995.
<http://dx.doi.org/10.1109/12.364536>
- [19] K. Hua and J. Abraham, "Design of systems with concurrent error detection using software redundancy," In *Proceedings of ACM IEEE Fall Joint Computer Conference in Dallas*, pp. 826-834, Nov. 1986

● 저 자 소 개 ●



장 은 지 (Eun-ji Jang)

2016년 서울여자대학교 컴퓨터학과(공학사)
2016년 ~ 현재 서울여자대학교 대학원 컴퓨터학과(공학석사 재)
관심분야 : 인지기능정보, IoT, etc
E-mail : s111559@swu.ac.kr



김 인 영 (In-young Kim)

2012년 명지대학교 정보통신공학과(공학사)
2016년 인포피아 연구기획팀 연구원
2016년 ~ 현재 ㈜에스에이치씨앤씨 연구개발팀 주임연구원
관심분야 : 영상인지, 신호처리, etc
E-mail : iykim@shcnc.co.kr



이 웅 재 (Woongjae Lee)

1981년 연세대학교 전기공학과 졸업(학사)
1985년 University of Illinois at Chicago 전산학과 졸업(석사)
1991년 Illinois Institute of Technology 전산학과 졸업(박사)
1993년 ~ 현재 서울여자대학교 미래산업융합대학 디지털미디어학과 교수
관심분야 : WSN, 인공지능, 자연어처리, 멀티미디어 etc.
E-mail : wjlee@swu.ac.kr