

자동화 크레인을 위한 네트워크 프로토콜의 성능 평가

Performance Evaluation of Network Protocol for Automated Transfer Crane System

하경남, 김만호, 이경창, 이석*

(Kyoung-Nam Ha, Man-Ho Kim, Kyung Chang Lee, and Suk Lee)

Abstract : As a way to build more efficient and intelligent container cranes for today's hub ports, communication networks are used to interconnect numerous sensors, actuators, controllers, and operator switches and consoles that are spatially distributed over a crane. Various signals such as sensor values and operator's commands are digitized and broadcast on the network instead of using separate wiring cables. This not only makes the design and manufacturing of a crane more efficient, but also easier implementation of intelligent control algorithms. This paper presents the performance evaluation of CAN(Controller Area Network), TTP(Time Triggered Protocol) and Byteflight that can be used for cranes. Through discrete event simulation, several important quantitative performance factors such as the probability of a transmission failure, average system delay (data latency) and maximum system delay have been evaluated.

Keywords : crane-by-wire system, automated crane system, CAN(Controller Area Network), TTP(Time Triggered Protocol), byteflight, performance evaluation

I. 서론

동북 아시아 경제의 급속한 성장과 더불어, 항만 물류 시스템의 처리 능력에 대한 요구가 급격하게 커지고 있다. 과거에는 북미와 유럽간의 무역량 증가에 따라 컨테이너 물동량이 증가하게 되었지만, 최근에는 아시아를 중심으로 북미, 유럽 간의 무역량의 증가에 따라 컨테이너 물동량이 급격하게 증가 하였다. 항만 물동량이 증가함에 따라 컨테이너 선박은 대형화 및 고속화되고, 항만 시스템은 효율적인 하역 및 선적을 위하여 운반 및 하역 시스템의 대형화, 고속화, 자동화 및 지능화가 필요하게 되었다. 항만 물류 시스템은 점차 대용량 컨테이너선 접안 기술, 지능형컨테이너 무인 이송 및 적재 시스템 기술, 지능형 육상 운송 연계 기술, 지능형 고성능 컨테이너 하역 기술 및 전체 시스템을 제어하는 통합 정보 시스템 기술과 같은 지능형 종합 물류 시스템으로 변화 하였다. 특히, 최근에는 크레인의 효율성을 높이기 위하여 미래형 항만 크레인 시스템에 대한 개발이 활발하게 진행되고 있다[1-3].

미래형 항만 크레인의 초고속, 초정밀 제어 기술은 컴퓨터를 이용한 분산형 실시간 제어를 통해서 이루어진다. 즉, 상호 연관성을 가지는 여러 개의 모터를 동시에 제어함으로써, 관련 정보를 수직 및 수평적으로 통합하여 제어하게 된다. 특히, 높은 제어 성능이 요구됨에 따라 크레인에서 사용되는 제어용 모터나 센서와 같은 전자 부품의 수가 급속도로 증가 하게 되었다. 그러나, 센서, 구동기, 제어기, 운전자 조작 스위치 등을 전선(cable)을 이용하여 일대일로 연결하는 전통적

인 배선 시스템에서는, 전자 부품수의 증가가 전선의 수를 기하급수적으로 증가 시키는 결과를 초래하였다. 이러한 전선의 증가는 배선 체계를 더욱 복잡하게 함으로써 크레인의 설계와 제작에 많은 시간이 필요하게 하였을 뿐만 아니라, 크레인의 운영과 정비를 매우 어렵게 만드는 결과를 초래하였다[4-6]. 이러한 문제점을 해결하기 위하여, 자동화용 크레인을 중심으로 전자 부품과 제어기 및 스위치를 한 가닥의 공유된 전선으로 연결하는 Crane-By-Wire와 같은 네트워크를 기반으로 한 제어시스템에 대한 연구가 활발히 진행되고 있다. Crane-By-Wire란 우주항공분야의 Fly-By-Wire, 자동차 분야의 Drive-By-Wire와 같이 시스템을 구성하고 있는 부품들 사이의 기계적 연결과 직접적인 전기배선 대신에 공유된 통신 네트워크를 통하여 신호를 교환하는 기술을 일컫는다. 이러한 분산 제어를 위한 네트워크를 크레인에 적용하기 위한 연구로서 기존의 1개의 전동기를 단독으로 제어하는 기존의 고립적인 구성에서 벗어나, 서로 연관성이 있는 전동기 구동 시스템을 LonWorks 네트워크로 연결하여 호스트 컴퓨터에서 총괄 제어하는 방식이 제안되었다. 또한, 실시간성이 요구되는 산업용 시스템에 네트워크를 적용할 때 발생할 수 있는 실시간 데이터 전송시간의 불확실성을 스위치드 이더넷(switched ethernet)을 이용하여 해결하기 위한 연구도 활발하게 진행되고 있다[7,8].

특히, CAN(Controller Area Network)이라는 통신 규약은 미국 해양 전자 협회(NMEA:The National Marine Electronics Association)에서 제안한 해양 항만용 전기 전자 시스템의 네트워크 표준인 NMEA 2000의 하위계층의 표준으로, CAN을 크레인을 비롯한 항만 시스템의 네트워크로 적용하기 위하여 활발한 연구가 진행되고 있다. 또한, 최근에는 실시간 제어 시스템을 위하여 TTP와 Byteflight 등의 통신 규약이 발표되어 Drive-By-Wire 시스템에 적용되기 시작하였다[9-12].

본 논문에서는 CAN, TTP와 Byteflight 프로토콜의 자동화 크레인 적용 가능성을 평가하기 위하여, 이산 사건 시뮬레이

* 책임저자(Corresponding Author)

논문접수 : 2005. 3. 15., 채택확정 : 2005. 5. 18.

하경남, 김만호 : 부산대학교 지능기계공학과 대학원

(Ovincent@pnu.edu/kmh@pnu.edu)

이경창 : 부경대학교 제어계측공학과(gclee@pknu.ac.kr)

이석 : 부산대학교 기계공학부(slee@pnu.edu)

※ 본 연구는 과학기술부 국가지정연구소(NRL) 사업(과제번호:MI -0302-00-0039-03-J00-00-023-10)에서 지원하여 연구하였음.

선(discrete event simulation) 언어인 SIMAN을 이용하여 각 프로토콜의 MAC 계층에 대한 시뮬레이션 모델을 개발하여 성능을 평가하였다. 특히, 네트워크에 접속된 스테이션의 수를 변화시켜 각 프로토콜의 전송 지연 및 처리율과 같은 전송 특성을 통하여 성능을 평가하였다[13,14].

본 논문은 서론을 포함하여 5장으로 구성되어 있다. II장에서는 NMEA 2000의 하위 계층인 CAN 프로토콜과 차량용 실시간 통신 프로토콜인 TTP, Byteflight 프로토콜에 대하여 간략히 소개하였다. III장에서는 각 프로토콜의 시뮬레이션 모델에 대하여 설명한다. IV장에서는 시뮬레이션 시나리오, 즉 메시지 생성 방법과 스테이션의 수에 따른 시뮬레이션 결과를 제시하고 각 프로토콜의 성능을 비교 분석하였다. 마지막으로, V장에서는 결론과 향후 과제를 제시하였다.

II. 자동화 크레인 시스템을 위한 통신 프로토콜

1. 프로토콜의 특징

표 1은 IEC 61162-3으로 제정되어 있는 NMEA 2000의 하위 계층인 CAN 프로토콜과 차량의 실시간 제어 시스템을 위한 TTP 및 Byteflight의 특징을 나타내고 있다. CAN 프로토콜은 차량 내에 센서, 구동기, 제어기 간에 디지털 통신을 제공하기 위하여, 1980년대 Bosch에서 개발하여 1993년에 ISO 표준으로 제정되었다.

반면, TTP는 TTTech사에서 개발한 시간 기반(time-triggered) 방식의 프로토콜로서, 미국 자동차 공학회에서 규정하고 있는 Class C에 해당하는 하드 리얼 타임(hard real-time) 통신에 적합한 프로토콜이다. Byteflight는 BMW에서 개발한 고성능 데이터 버스로서, 동기식 전송(synchronous transmission)과 비동기식 전송(asynchronous transmission)의 장점을 결합하여 개발된 통신 프로토콜이다.

Kopetz에 의하면, 하드 리얼 타임 통신을 위하여 constant latency jitter, clock synchronization, temporal composability, error containment, babbling idiot avoidance 등과 같은 기능들이 보장되

어야 한다고 했다. TTP는 이러한 기능들이 보장되도록 설계되었으며, Byteflight의 경우 높은 우선 순위(high priority) 메시지에 한하여 이러한 기능들이 보장된다고 제시되고 있다 [15,16].

그러나, TTP의 경우는 초기에 네트워크 시스템을 설계할 때, 메시지 스케줄링을 한번 만 할 수 있다는 단점을 가진 반면, Byteflight의 경우는 시스템 운영 중에도 높은 우선 순위 메시지의 스케줄링을 다시 할 수 있다는 확장성을 가지고 있다. 그리고, TTP의 경우는 한 사이클에서 하나의 노드는 하나의 메시지 만을 전송할 수 있는 반면, Byteflight의 경우는 한 사이클에서 각 노드마다 대역폭을 유연하게 할당할 수 있다는 장점을 가지고 있다[17].

반면, CAN의 경우 하드 리얼 타임 통신에 관한 기능들은 규격에서 정의되어 있지는 못하지만, ID를 조정할 수 있기 때문에 매우 뛰어난 확장성을 가지며 대역폭을 유연하게 할당할 수 있다는 장점을 가지고 있다. 또한, 거의 대다수의 반도체 제조사에서 CAN을 지원하는 마이크로컨트롤러를 제공할 뿐만 아니라, 모듈의 구현이 매우 용이하다는 장점을 가지고 있다. 따라서, 최근에는 CAN을 다양한 산업환경에서 사용하기 위한 목적으로, 일정한 전송 지연 등과 같은 기능들을 보장할 수 있는 메시지 스케줄링, fault tolerant의 구현 등과 같은 연구들이 수행되고 있다[18,19].

2. 매체 접속 제어 방식

NMEA 2000의 하위 계층으로 사용되고 있는 CAN은 송신 노드와 수신 노드의 주소(address) 대신 메시지의 내용에 따라 11비트 또는 29비트로 이루어진 식별자(Identifier, ID)를 부여하여 모든 메시지를 구별하고 메시지의 우선권을 정하는 내용 기반 주소 지정(content-based addressing) 방식을 사용한다. 즉, 만약 어떤 스테이션이 메시지를 전송하기 시작하면, 나머지 스테이션들은 수신된 데이터가 자신이 가진 ID와 관련이 있는 지를 비교한다. 만약, 자신이 가진 ID와 관련이 없으면 받아들이고, 관련이 없으면 무시한다.

표 1. CAN과 TTP, byteflight 프로토콜의 비교.

Table 1. Comparison of CAN, TTP and byteflight protocol.

Feature	CAN	TTP	Byteflight
Message transmission	asynchronous	synchronous	asynchronous and synchronous
Message identification	message identifier	time slot	message identifier
Maximum data rate	1Mbps	2 Mbps	10 Mbps
Bit encoding	NRZ with bit stuffing	modified frequency modulation(MFM)	NRZ with start/stop bits
Physical layer	transceiver	not defined	optical transceiver
Latency jitter	bus load dependent	constant for all messages	constant for high priority messages
Clock synchronization	not provided	distributed	centralized
Temporal composability	not supported	supported	supported for high priority messages
Error containment (physical layer)	partially provided	provided with special physical transceiver	provided by optical fiber and transceiver chip
Babbling idiot avoidance	not provided	possible by independent bus guardian	provided via star coupler
Extensibility	Excellent	only if extension planned in original design	extension possible for high priority messages with effect on asynchronous bandwidth
Flexibility	flexible bandwidth for each node	only one message per node and TDMA cycle	flexible bandwidth for each node
Availability of components	numerous μ C available	available	available

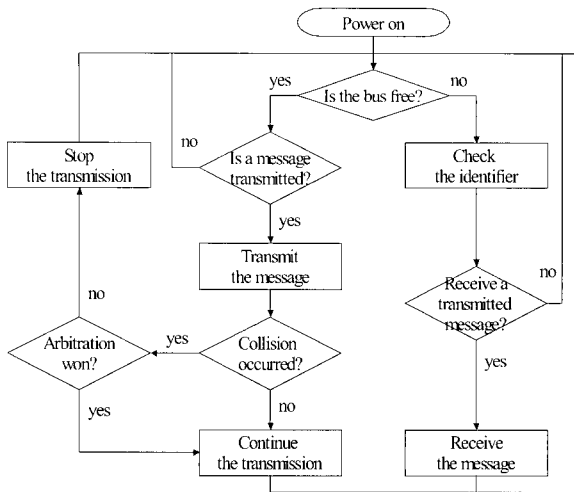


그림 1. CAN의 매체 접속 제어 방식.
Fig. 1. Medium access control method of CAN.

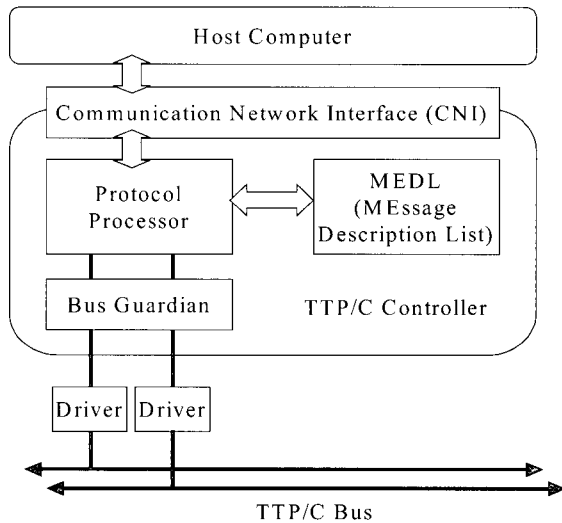


그림 2. TTP 컨트롤러의 구조.
Fig. 2. Structure of the TTP controller.

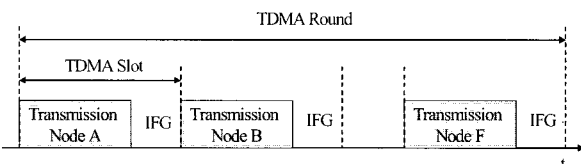


그림 3. TDMA - TTP의 매체 접속 제어 방식.
Fig. 3. TDMA - medium access control method of TTP.

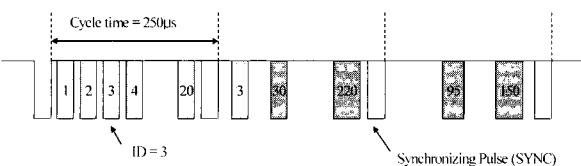


그림 4. FTDMA - Byteflight의 매체 제어 접근 방식.
Fig. 4. FTDMA - medium access control method of byteflight.

또한, 둘 이상의 통신 노드에서 거의 동시에 전송을 시도했을 경우에는 충돌이 발생할 수 있다. 이러한 경우, CAN은 충돌된 메시지들의 ID를 한 비트씩 비교하여 가장 낮은 ID 값을 가진 메시지(우선 순위가 가장 높음)만 전송을 계속하고, 나머지 메시지는 즉시 전송을 중단하는 그림 1과 같은 CSMA/NBA(Carrier Sense Multiple Access with Non-destructive Bitwise Arbitration)를 사용한다. 이러한 버스 액세스 방식으로 인하여, 네트워크의 부하가 상당히 증가하여도 안정적 통신이 가능할 뿐만 아니라, 전송 지연의 상한이 예측 가능하다는 장점을 가지고 있다[20].

TTP는 시분할 멀티플렉싱(time division multiplexing)을 사용하기 때문에, 전송되는 메시지는 전송 지터(transmission jitter)를 거의 경험하지 않는다. 전송 지터란 동일한 메시지가 전송되는 시간 간격의 변동 폭을 의미한다. 또한, 한 개 통신 노드의 고장이 나머지 통신 노드들의 통신을 방해하지 않도록 fault tolerant 기능을 가지고 있으며, 강인한 에러의 검출과 회복 기능을 가지고 있다. 특히, 멤버십 서비스(membership service)를 제공하기 때문에, 각 노드들이 나머지 노드들의 상태를 항상 알 수 있다.

그림 2는 TTP 프로토콜 컨트롤러의 내부 구조를 기능에 따라 나타낸 블록도이다. 먼저, 버스가디언(bus guardian)은 통신 노드가 자신이 송신할 수 있도록 할당된 시간 내에서만 동작을 할 수 있도록 허용함으로써, 각 노드가 할당된 시간 범위를 넘어서 통신하는 것을 방지하는 역할을 수행한다. 다음으로, MEDL(MESSAGE Description List)에는 네트워크에서 전송되는 스케줄링된 메시지들에 대한 정보가 저장되어 있으며, 프로토콜 프로세서(protocol processor)는 MEDL을 기반으로 하여 TTP 버스를 통해 다른 노드들과 통신을 수행한다. 마지막으로, CNI(Communication Network Interface)는 프로토콜 프로세서와 호스트 컴퓨터(host computer)를 연결하는 기능을 수행하며, 이의 구현을 위하여 주로 DPRAM(dual port RAM)이 사용된다. TTP의 매체 접속 제어 방식은 그림 3과 같이 TDMA(Time Division Multiple Access)의 슬롯 예약(reserved slot) 방식에 기반을 두고 있다. TTP에서는 시간에 따라 전송 매체에 접근할 수 있는 권한을 슬롯(slot)으로 할당하여 각 통신 노드에 할당한다. 여기서, TDMA 슬롯의 길이는 메시지의 전송에 필요한 시간과 송신측과 수신측의 처리 시간을 고려하여 결정된다. 그리고, TDMA 슬롯 간의 최소 길이를 IFG(Inter-Frame Gap)로 정의한다. 슬롯의 길이와 전송 노드의 순서 등은 네트워크 시스템이 구성되기 전에 결정되어야 하며, 이러한 결정 사항들은 MEDL에 저장된다.

Byteflight의 매체 접속 제어 방식은 FTDMA(Flexible Time Division Multiple Access) 방식으로 0-255의 값을 가지는 ID를 이용한 우선 순위에 따라 전송 매체에의 접근 시간을 할당한다. Byteflight의 데이터 전송 방식은 그림 4와 같이 250µs 주기로 발생하는 동기화 펄스(synchronization pulse)에 의해 제어된다. SYNC 마스터 노드에서 생성되는 동기화 펄스는 모든 통신 노드에 시간적 기반(time-base)을 제공한다. 그림에서 보는 바와 같이, 모든 노드는 동기화 펄스가 시작되고 나면, 메시지의 우선 순위에 따라 메시지를 전송한다. 먼저, 통신 노드는 슬롯 카운트(slot counter)를 가지고 있으며, 동기화 펄스가 감지되면 슬롯 카운트를 0으로 초기화한다. 그리고 나서,

프로토콜 컨트롤러(protocol controller)는 일정 대기 시간만큼 기다린 후 전송 매체가 사용 중이지 않으면 슬롯 카운트를 1 증가시킨다. 만약, 슬롯 카운트의 값이 자신이 가진 메시지의 ID와 동일한 경우, 프로토콜 컨트롤러는 해당 슬롯에 해당 메시지를 전송한다. 다음으로, 전송을 마친 노드는 슬롯 카운트를 1 증가시킨 후, 자신이 가진 메시지들과의 ID 비교를 수행한다.

이러한 ID 비교는 정의되어 있는 사이클 타임(cycle time) 기간 동안 동기화 펄스가 재 감지되기 전까지 수행한다. 이러한 FTDMA 전송 방식으로 인하여, Byteflight에서 전송 충돌은 발생하지 않을 뿐만 아니라, 여러 개의 ID를 가진 노드는 한 사이클 내에서 여러 번의 메시지를 보낼 수 있다. 또한 Byteflight에서는 트래픽이 높은 상황에서도 상위의 우선 순위를 가지는 메시지의 전송이 보장될 뿐만 아니라, 하위의 우선 순위를 가지는 메시지도 전송될 수 있는 유연한 확장성이 보장된다.

III. Crane-By-Wire 용 프로토콜의 시뮬레이션 모델

공유 전송 매체를 사용하는 네트워크 시스템에서는 메시지가 생성된 후에 전송이 완료될 때까지 소요되는 시간은 프로토콜의 특성에 따라 차이를 가진다. 이는 다른 스테이션에서 메시지가 전송 중이면 그 메시지의 전송이 끝날 때까지 기다리거나, 전송 도중에 다른 메시지와 충돌을 일으키면 다시 전송을 시도해야 하기 때문이다. 이러한 전송 지연은 매체 접속 제어 방식, 네트워크에 접속된 스테이션의 수, 메시지의 발생 빈도, 메시지의 길이 등과 같은 변수에 영향을 받는다. 특히, Crane-By-Wire 시스템에서 전송되는 메시지는 크레인의 정밀 제어 정보 및 사고 방지 정보를 담고 있기 때문에 일정한 시간 한도 내에 전송되어야 한다. 만약, 일정한 시간 한도 내에 전송되지 못한다면, 네트워크 시스템의 성능이 악화될 뿐만 아니라, 치명적인 사고가 발생할 수도 있다. 따라서, Crane-By-Wire 시스템에 네트워크를 적용하기 위하여 메시지들이 가질 수 있는 전송 지연의 특성에 대한 분석이 필요하다.

따라서, 본 논문에서는 Crane-by-Wire용 프로토콜로 적용 가능한 CAN, TTP와 Byteflight 프로토콜의 특성을 분석하기 위하여 시뮬레이션 모델을 개발하고, 스테이션의 수와 각 스테이션에서 생성되는 메시지의 발생 주기, 메시지 길이와 같은 통계적인 정보를 이용하여 시뮬레이션을 수행하였다. 그리고, 전송 지연과 메시지 처리율 등과 같은 성능 지표들을 계산하여, 각 프로토콜의 성능을 평가하였다.

각 프로토콜의 시뮬레이션 모델은 데이터 링크 계층을 근거로 하여, SIMAN Ver. 5를 이용하여 다음과 같은 가정을 바탕으로 하여 개발되었다.

- 전송 매체에서의 비트 에러율은 매우 낮기 때문에, 메시지 전송 시 에러가 발생하지 않으며 재전송은 일어나지 않는다고 가정하였다.
- 스테이션의 입출입은 발생되지 않는 것으로 가정하였다.
- 응용 계층에 의한 전송 지연은 사용되는 프로토콜 컨트롤러와 응용 프로그램의 성능에 따라 달라진다. 따라서, 이에 의한 전송 지연은 동일한 것으로 가정하여, 전송 지연의 계산에서는 무시하였다.

- 각 프로토콜의 전송 버퍼는 프로토콜 컨트롤러의 메모리 크기에 의하여 영향을 받는다. 그러나, 본 논문에서는 각 프로토콜이 동일한 조건에서 비교될 수 있도록 전송 버퍼를 1과 16으로 설정하였다.

먼저, CAN 시뮬레이션 모델은 CAN 2.0A의 데이터 링크 계층을 참조하여 개발하였다. 시뮬레이션 모델에서, CAN의 ID 필드는 11비트, 전송 속도는 1Mbps로 설정되었으며, 접속 형태는 버스형으로 가정하였다. 또한, 각 스테이션에서 메시지가 생성되면 전송 버퍼에 저장되도록 하였다. 그리고 나서, 전송 매체가 사용 중이지 않으면 각 스테이션은 경쟁을 하게 되고, 우선순위가 가장 높은 메시지가 전송될 수 있도록 하였다.

다음으로, TTP 시뮬레이션 모델은 2003년 발표된 TTP Ver. 1.1의 데이터 링크 계층을 참조하여 개발하였다. 시뮬레이션 모델에서 TTP의 전송 속도는 1Mbps로, 한 슬롯의 길이는 4비트의 오버 헤드와 16비트의CRC, 그리고 데이터를 전송하는데 필요한 시간으로, 각 슬롯 간의 간격은 5 μ s로 설정하였다. 또한, 한 사이클의 길이는 네트워크에 참여한 모든 스테이션이 한번씩 전송할 수 있는 시간으로 가정하였다. 예로, 시뮬레이션 모델에서 스테이션의 수가 10개이고, 하나의 메시지를 보내는 데 100 μ s가 걸린다고 한다면, 사이클 길이는 1,000 μ s로 설정된다.

마지막으로, Byteflight 시뮬레이션 모델은 BMW에서 발표된 규격을 바탕으로 하여 개발된 모토롤라사의 MC68HC912 BD32 매뉴얼을 참조하였다. Byteflight의 전송 속도는 광섬유를 사용하여 10Mbps로 규정되어 있다. 이에 따라, Byteflight의 성능을 1Mbps를 지원하는 CAN과 TTP와 동일한 조건에서 비교하는 것은 한계가 있다고 볼 수 있다. 따라서, 본 논문에서는 각 시뮬레이션 모델의 트래픽이 동일하도록 메시지 발생 주기를 조절하여 실험을 수행하였다[21].

IV. Crane-By-Wire 용 프로토콜의 성능 평가

본 논문에서는 각 프로토콜의 성능을 평가하기 위하여, 메시지의 발생 주기와 스테이션의 수를 변화시켜 가면서 시뮬레이션을 수행하였다. 먼저, 각 시뮬레이션 모델에서 전송되는 데이터 길이는 8바이트로 설정하였다. 그리고, 메시지는 2,500 비트 타임 주기의 균일 분포(uniform distribution)로 발생하는 경우와 2,500 비트 타임 평균을 가지는 지수 분포(exponential distribution)로 발생하는 것으로 설정하였다. 여기에서, CAN과 TTP의 전송 속도는 1Mbps이므로, 발생 주기의 평균은 2,500 μ s로 설정하였다. 반면, Byteflight의 전송 속도는 10Mbps이므로, 발생 주기의 평균은 250 μ s로 설정하였다. 그리고 나서, 스테이션의 수를 2대(트래픽 5.12%)에서 32대(트래픽 76.8%)까지 증가시켜 가면서 시뮬레이션을 수행하였다. 여기에서, 트래픽은 (1)과 같이 정의하였다.

$$G = \frac{LN}{TB} \quad (1)$$

식에서 L은 데이터 필드의 길이(bits), N은 스테이션의 수, T는 메시지의 발생 주기(μ s), B는 프로토콜의 전송 속도(Mbps)를 의미한다.

특히, TTP에서 슬롯의 할당 방법과 CAN 및 Byteflight에서

ID의 설정 방법은 네트워크 시스템에 상당한 영향을 미치며, 세심하게 설계를 할 필요가 있다. 이에 따라, TTP에서는 각 스테이션의 순서대로 슬롯을 할당하였다. 그리고, CAN과 Byteflight에서는 각 노드에서 두 종류의 메시지가 생성된다고 가정하고 ID를 노드 주소 순서대로 할당하였다. 예로, 3개의 노드가 네트워크에 접속되어 있는 경우, 노드 1의 ID는 1과 4를, 노드 2의 ID는 2와 5를, 노드 3의 ID는 3과 6을 할당하였다. 이렇게 함으로써, 노드 1의 높은 ID를 가진 메시지, 노드 2의 높은 ID를 가진 메시지 순으로 전송이 이루어질 수 있도록 하였다.

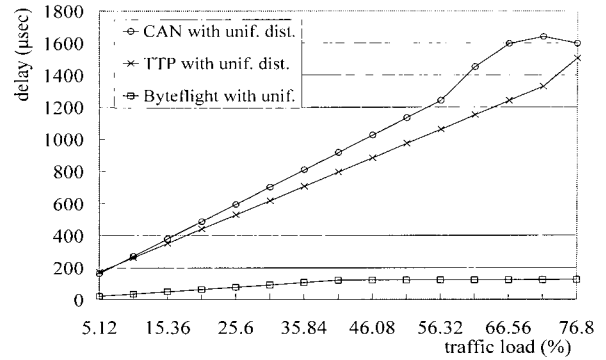
마지막으로, 본 논문에서는 충분한 양의 메시지가 전송되는 것을 관찰하고, 관찰된 메시지들의 평균 전송 지연과 처리율을 계산하였다. 여기에서, 전송 지연은 한 스테이션에서 메시지가 생성된 시간부터 그 메시지의 마지막 비트가 전송 완료되는 시간까지의 간격으로 정의하였다. 마지막으로, 처리율은 생성된 메시지와 전송에 성공한 메시지의 비로 정의하였다.

그림 5는 각 시뮬레이션 모델의 전송 큐가 1개인 경우, 트래픽의 변화와 메시지 발생 주기에 따라 균일 분포의 메시지 발생에 따른 평균 전송 지연(a), 지수 분포의 메시지 발생에 따른 평균 지연(b), 균일 분포의 메시지 발생에 따른 처리율(c)와 지수 분포의 메시지 발생에 따른 처리율(d)을 나타내고 있다. TTP의 경우, 평균 전송 지연은 트래픽이 증가함에 따라 선형적으로 증가되며, 메시지 발생 방법에는 큰 영향을 받지 않음을 알 수 있었다. 반면, 메시지 발생 방법이 지수 분포일 때 처리율은 트래픽이 증가함에 따라 점차 감소되지만, 균일 분포일 때 아주 높은 트래픽(71.68%)이 될 때까지 처리율이 1로 유지됨을 알 수 있었다.

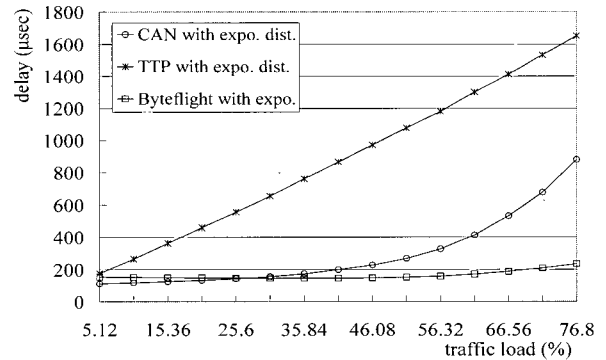
CAN의 경우, 메시지 발생 주기가 지수 분포일 때, 처리율은 TTP에 비하여 완만하게 감소됨을 알 수 있으며, 평균 전송 지연은 TTP에 비하여 상당히 낮게 유지됨을 알 수 있다. 반면, 메시지 발생 주기가 균일 분포일 때, 평균 전송 지연은 TTP와 거의 유사한 형태로 증가되지만, 처리율은 TTP보다 더 낮은 트래픽(56.32%)에서 서서히 감소됨을 알 수 있다.

Byteflight의 경우, 트래픽이 증가하더라도 평균 전송 지연은 상당히 낮은 상태로 유지됨을 알 수 있다. 그러나, 메시지 발생 주기가 지수 분포일 때, 처리율은 트래픽이 낮을 때에도 65%로 매우 낮으며, 트래픽이 증가함에 따라 감소되는 경향을 보인다. 또한, 메시지 발생 주기가 균일 분포일 때도 TTP나 Byteflight보다 더 낮은 트래픽(40.96%)에서 부터 급속하게 감소됨을 알 수 있다.

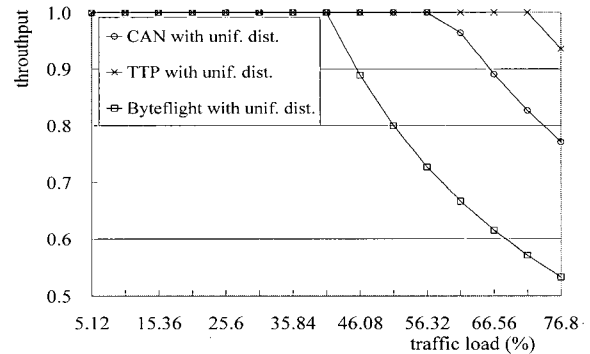
이상과 같은 결과로부터, 프로토콜 컨트롤러의 전송 큐가 1이고 메시지 발생 주기가 지수 분포인 경우, CAN의 성능이 다른 프로토콜들에 비하여 우수하다고 볼 수 있다. 즉, TTP의 경우 트래픽에 따라 처리율이 선형적으로 감소하고, Byteflight의 경우 낮은 트래픽에서도 처리율이 65%로 나타나는 반면, CAN은 트래픽이 30-40%가 될 때까지 처리율이 거의 1을 유지함을 알 수 있다. 이는 TTP의 경우, 스테이션의 수에 따라 결정되는 사이클 내에서 하나의 스테이션이 한번만 전송할 수 있으므로, 하나의 사이클 내에서 여러 개의 메시지가 생성되면 전송 큐에 저장되지 못하고 제거되기 때문이다.



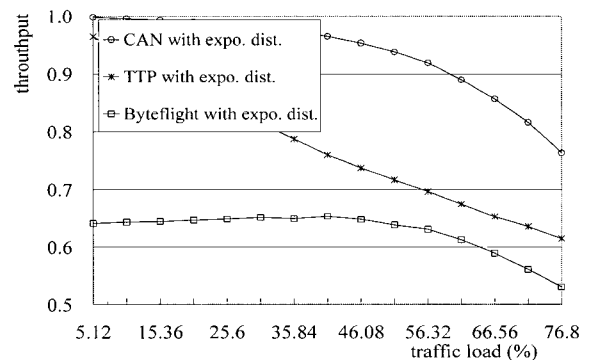
(a) Average delay of uniform distribution message



(b) Average delay of exponential distribution message



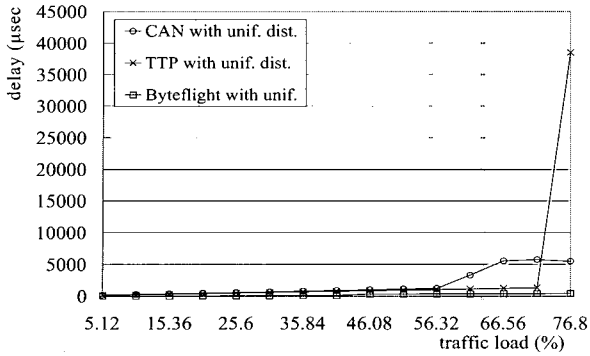
(c) Throughput of uniform distribution message



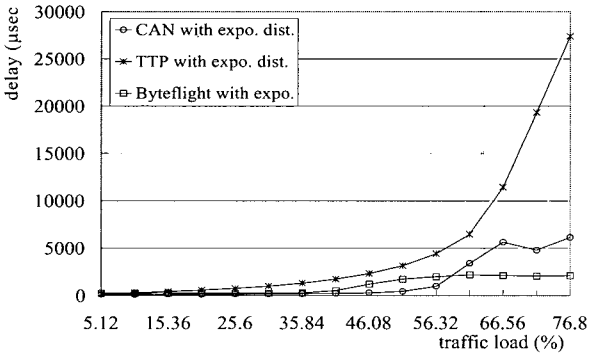
(d) Throughput of exponential distribution message

그림 5. 전송 큐가 1인 경우 스테이션의 수 변화에 따른 프로토콜의 성능.

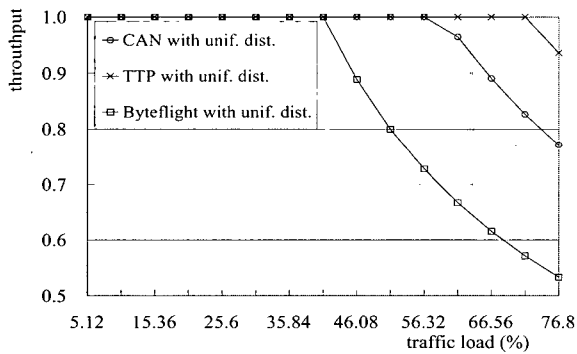
Fig. 5. Performance of protocols varying with a single transmission queue.



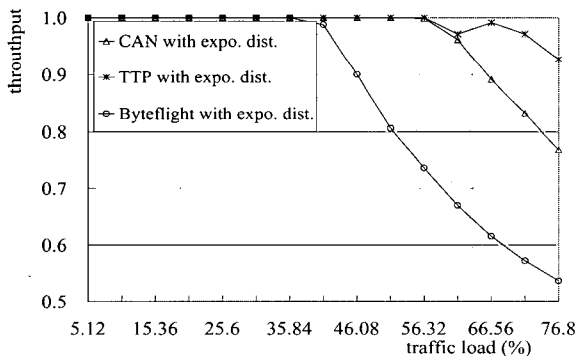
(a) Average delay of uniform distribution message



(b) Average delay of exponential distribution message



(c) Throughput of uniform distribution message



(d) Throughput of exponential distribution message

그림 6. 전송 큐가 16인 경우 스테이션의 수 변화에 따른 프로토콜의 성능.

Fig. 6. Performance of protocols varying with 16 transmission queues.

또한, Byteflight의 경우, 250 μ s의 사이클 내에서 하나의 ID만을 보낼 수 있으므로, 하나의 사이클 내에서 여러 개의 ID를 가진 메시지가 생성되면 전송큐에 저장되지 못하고 제거되기 때문이다. 여기에서, Byteflight가 TTP보다 처리율이 더 높은 이유는 TTP의 사이클은 스테이션의 수에 따라 결정되기 때문에 스테이션의 수가 적으면 사이클이 Byteflight의 사이클보다 작기 때문이다. 반면, CAN의 경우 메시지가 생성되면 ID가 높은 메시지 순으로 바로 전송되므로, 처리율이 다른 프로토콜들에 월등히 우수하다고 판단할 수 있다.

다음으로, 메시지 발생 주기가 균일 분포인 경우, 트래픽이 낮을 때는 CAN과 Byteflight의 성능이 TTP에 비하여 우수하다고 볼 수 있다. 즉, TTP의 평균 전송 지연은 선형적으로 증가하지만, CAN과 Byteflight의 평균 전송 지연은 낮게 유지됨을 알 수 있다. 그러나, 트래픽이 높아지게 되면, TTP의 처리율은 상당히 높은 트래픽에서도 1을 유지하지만, CAN과 Byteflight의 처리율은 급속하게 떨어짐을 알 수 있다. 이는 TTP의 경우, 주기적인 메시지가 발생하는 경우 TTP의 사이클이 메시지 발생 주기와 동일하기 때문에 높은 트래픽에서도 처리율은 거의 1을 유지하지만 전송 지연은 스테이션의 수에 비례하여 증가한다고 판단된다. 반면, CAN은 트래픽이 낮을 때 ID에 따라 전송이 되고, Byteflight는 250 μ s 주기로 메시지가 생성되므로 매 사이클마다 전송이 이루어지므로 평균 전송지연이 낮게 유지된다고 판단된다.

그림 6은 각 시뮬레이션 모델의 전송 큐가 16개인 경우, 트래픽의 변화와 메시지 발생 주기에 따라 균일 분포의 메시지 발생에 따른 평균 전송 지연(a), 지수 분포의 메시지 발생에 따른 평균 지연(b), 균일 분포의 메시지 발생에 따른 처리율(c)과 지수 분포의 메시지 발생에 따른 처리율(d)을 나타내고 있다. TTP의 경우, 균일 분포에서는 평균 전송 지연이 완만하게 증가하다가, 71.68%가 되면 급속하게 증가함을 관찰할 수 있다. 반면, 지수 분포에서는 상당히 빠른 속도로 증가하다가, 46.08% 이후부터는 급속하게 증가함을 관찰할 수 있다. 처리율은 지수 분포일 때는 56.32%에서, 균일 분포일 때는 71.68%에서부터 서서히 감소하기 시작함을 관찰할 수 있다.

CAN의 경우, 균일 분포일 때 TTP와 유사한 경향으로 증가하다 56.32%에서 급속하게 증가하지만, 트래픽이 높더라도 TTP에 비하여 전송 지연이 낮게 유지됨을 관찰할 수 있다. 반면, 지수 분포에서는 Byteflight와 같이 매우 낮은 트래픽을 유지하다가 56.32%부터 급속하게 증가함을 관찰할 수 있다. 전송 큐의 크기가 큰 경우, 처리율은 균일 분포나 지수 분포에 관계없이 거의 동일한 특성을 가지며, 트래픽이 56.3%에서부터 서서히 감소하기 시작함을 관찰할 수 있다.

Byteflight의 경우, 균일 분포일 때 트래픽이 매우 낮은 상태로 유지됨을 관찰할 수 있다. 반면, 지수 분포에서는 평균 전송 지연이 약간 증가하지만, 2,500 μ s 이하로 다른 프로토콜들에 비해 상대적으로 낮은 상태로 유지됨을 관찰할 수 있다. 전송 큐의 크기가 큰 경우, 처리율은 CAN과 유사한 특성을 보이며, 35.84%에서 서서히 감소하기 시작함을 관찰할 수 있다.

이상과 같은 결과로부터, 전송 큐가 큰 경우, 세 프로토콜의 처리율은 트래픽이 상당히 높아지더라도 1을 유지하고

있다. 특히, 지수 분포인 경우 전송 큐가 1인 경우 매우 낮은 트래픽에서도 처리율이 1이 아니었지만, 전송 큐가 증가하게 되면, 처리율은 1로 유지됨을 알 수 있다. 특히, 트래픽이 낮은 경우, Byteflight는 메시지 발생 주기와는 관계없이 매우 낮은 평균 전송 지연을 보이고 있다. 반면 CAN은 지수 분포일 때 평균 전송 지연이 매우 낮게 나타나지만, 균일 분포일 때는 TTP와 유사하게 상대적으로 높은 전송 지연을 보인다. 그러나, 트래픽이 높아지게 되면, 메시지 발생 주기가 균일 분포일 때 CAN이나 Byteflight는 처리율이 떨어진다는 특징이 있지만, TTP는 상당히 높은 트래픽에서도 처리율을 1로 유지한다는 특징을 가지고 있다.

V. 요약 및 결론

본 논문에서는 NMEA 2000의 하위 계층인 CAN 프로토콜과 실시간 통신 프로토콜에서 사용되는 TTP, Byteflight 프로토콜의 특성을 평가하기 위하여, 이산 사건 시뮬레이션 언어인 SIMAN을 이용하여 시뮬레이션 모델을 개발하였다. 또한, 이를 이용하여 메시지 발생주기를 균일 분포와 지수 분포로 설정하고, 스테이션의 수를 변화시켜가면서 각 프로토콜의 평균 전송 지연과 처리율을 계산하였다.

본 논문에서는 이상과 같은 실험 결과로부터 다음과 같은 결론을 얻을 수 있었다. NMEA 2000에서 정의하고 있는 CAN은 트래픽이 낮은 경우 우수한 네트워크 성능을 보이지만 트래픽이 높을 경우 모든 메시지 처리 시간을 보장할 수 없다. 우선순위에 따른 메시지 전송방식을 따르고 있기 때문에 트래픽이 높을 경우 낮은 우선순위의 메시지가 발생했을 때의 전송시간을 예측할 수 없다.

TTP는 트래픽이 높은 경우에도 메시지 전송은 일정한 시간 내에서 처리할 수 있다. 버퍼에서 폐기되지 않는 메시지의 경우 다음 사이클에서 반드시 전송이 되기 때문에 트래픽이 증가함에 따른 한 사이클의 크기가 길어지는 영향으로 인한 평균 전송지연은 증가하지만, 일정한 기간 내에 메시지 전송을 처리할 수 있다. TTP의 이런 특성은 Crane-by-Wire 시스템에서 필요로 하는 예측 가능한 전송을 보장할 수 있다.

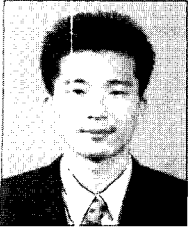
Byteflight는 트래픽이 높은 경우 메시지 전송 지연의 편차가 크다. TDMA 전송 방법과 우선순위에 따른 메시지 전송을 함께 사용하기 때문에 트래픽이 높은 경우 낮은 우선순위의 메시지에 취약한 특징을 가지고 있기 때문이다.

본 논문에서는 각 프로토콜들의 매체 접속 제어 방식에 따른 네트워크의 성능 평가에 초점을 맞추었다. 그러나, 프로토콜의 성능은 응용 프로그램의 구현 방법 등이나 프로토콜 컨트롤러의 성능에 따라 상당한 영향을 받는다. 따라서, 향후에는 응용 프로그램이 구현되어 있는 하드웨어 모듈을 이용하여 성능 평가를 수행하여야 할 것이다.

참고문헌

[1] 박경택, "항만 물류 시스템 지능화 기술," 한국정밀공학회지, vol. 19, no. 2, pp. 41-48, 2002.

- [2] 손유식, 김영복, "컨테이너 크레인의 흔들림 방지장치 개발에 관한 연구," 한국해양공학회지, vol. 14, no. 4, pp. 35-42, 2000.
- [3] 정승현, 박정일, "퍼지 제어를 이용한 크레인의 진동 억제 및 위치 제어", 제어자동화시스템공학회논문지, vol. 3, no. 5, pp. 435-442, 1997.
- [4] 홍금식, 손성철, 이만형, "컨테이너 크레인의 흔들림 제어 Part II", 제어자동화시스템공학회논문지, vol. 3, no. 2, pp. 132-138, 1997.
- [5] 이경창, 강송, 이석, 이만형, "IEEE 802.11을 이용한 Profbus 네트워크의 무선 확장 및 성능 평가," 제어자동화시스템공학회논문지, vol. 7, no. 4, pp. 326-333, 2001.
- [6] 홍성수, "부산 실시간 제어 시스템의 개발", 제어자동화시스템공학회논문지, vol. 4, no. 15, pp. 33-38, 1998.
- [7] 전태원, "Lonworks 네트워크를 이용한 야드 크레인 구동용 전동기 위치 제어", 대한전기공학회논문지, vol. 50, no. 1, pp. 37-43, 2001.
- [8] 이경창, 김태준, 이석, "실시간 산업용 통신을 위한 Switched Ethernet의 성능 평가," 제어자동화시스템공학회 논문지, vol. 9, no. 1, pp. 90-98, 2003.
- [9] Road Vehicles-Interchange of Digital Information-Controller Area Network for High-Speed Communication, ISO 11898, 1994.
- [10] The National Marine Electronics Association, NMEA2000, Ver.1.0, 2000.
- [11] Time-Triggered Protocol TTP/C High-Level Specification Document, Ver. 1.1, TTEch, 2003.
- [12] Byteflight Specification, Ver. 0.5, BMW, 1999.
- [13] C. D. Pegden, R. P. Sadowski, R. E. Shannon, *Introduction to Simulation using SIMAN*, 2nd ed., McGraw-Hill, 1995.
- [14] 이성근, 홍승호, "시뮬레이션 모델을 이용한 IEC/ISA 필드버스 시스템의 데이터 링크 계층 성능 분석," 제어자동화시스템공학회논문지, vol. 2, no. 3, pp. 209-219, 1996.
- [15] H. Kopetz, "A communication infrastructure for a fault-tolerant distributed real-time system," *Control engineering practice*, vol. 3, no. 8, pp. 1139-1146, 1995.
- [16] H. Kopetz, "Time-triggered real-time computing," *Annual Reviews in Control*, vol. 27, no. 1, pp. 3-13, 2003.
- [17] G. Leen and D. Heffernan, "Expanding automotive electronic systems," *Computer*, vol. 35, no. 1, pp. 88-93, 2002.
- [18] M. Ellims, S. Parker, and J. Zurlo, "Design and analysis of a robust real-time engine control network," *IEEE Micro*, vol. 22, no. 4, pp. 20-27, 2002.
- [19] L. B. Fredriksson, "CAN for critical embedded automotive networks," *IEEE Micro*, vol. 22, no. 4, pp. 28-35, 2002.
- [20] 이경창, 김정희, 이흥희, "스마트 홈을 위한 CAN 기반 화재 감시 시스템의 구현", 제어·자동화·시스템공학회 논문지, vol. 10, no. 8, 2004.
- [21] MC68HC912BD32 Advance Information, Motorola, 2000.



하경남

1974년 9월 10일생. 2001년 부산대학교 기계공학부 졸업. 2001년~2003년 BOSH 근무. 2003년~현재 부산대학교 지능기계공학과 석박사통합과정 재학중. 관심분야는 산업용 네트워크, 홈 네트워크, 차량용 네트워크, 센서 네트워크.



김만호

1977년 4월 16일생. 2001년 동아대학교 기계공학과 졸업. 2003년~현재 부산대학교 지능기계공학과 석박사통합과정 재학중. 관심분야는 차량용 네트워크, 임베디드 시스템.



이경창

1971년 5월 1일생. 1996년 부산대학교 생산기계공학과 졸업. 1998년 동 대학원 석사. 2003년 동 대학원 박사. 1998년~2003년 기계공학연구정보센터 전임연구원. 2003년~2005년 울산대학교 네트워크기반 자동화 연구센터 전임연구원.

2005년~현재 부경대학교 제어계측공학과 전임강사. 관심분야는 필드버스, 산업용 이더넷, 차량용 네트워크, 홈 네트워크.



이석

1961년 12월 11일생. 1984년 서울대학교 기계공학과 졸업. 1985년 펜실바니아 주립대학교 석사. 1990년 동 대학원 박사. 1990년~1993년 신시내티 대학교 기계공학과 조교수. 1993년~현재 부산대학교 기계공학부 교수. 관심분야는 필드버스, 차량용 네트워크, 홈 네트워크.

필드버스, 차량용 네트워크, 홈 네트워크.