

論文00-37SC-5-2

LonWorks를 이용한 네트워크 시스템의 플로우제어에 관한 연구 (A Study on Flow Control of Network Systems with LonWorks)

金秉熙*, 曹光鉉**, 朴京燮**

(Byoung-Hee Kim, Kwang-Hyun Cho, and Kyoung-Sup Park)

요 약

본 논문에서는 최근 각광받고 있는 산업용 제어네트워크 중 다양한 통신매체를 지원하고 상호응용성이 높은 LonWorks를 이용하여 실험실 규모의 네트워크를 구성하고 플로우제어기의 도입을 통한 성능향상을 검증한다. 이를 위해 일반적으로 개루프 시스템으로 운용되는 산업용네트워크에 네트워크 채널상태를 궤환받고 원형대기열을 설치하여 대기열에 있는 전송데이터의 개수와 전송데이터가 대기열에 입력되는 시간간격을 이용하는 플로우제어기를 도입함으로써 폐루프 시스템을 구성한다. 또한 실험을 통하여 전송률과 공정성이라는 성능지수를 토대로 제안된 플로우제어 네트워크 시스템의 성능향상을 검증한다.

Abstract

In this paper, we investigate the performance improvement by applying flow control to LonWorks-based network systems which support various communication media together with interpretable class of systems. To this end, we introduce a circular queue at each node of industrial network systems usually operated in open loop. Then we utilize the feedback information of the number of data in a queue, the data arriving interval in a queue and channel error to make the overall industrial network system in closed loop. We verify the improved performance of the network systems in view of throughput and fairness measures via the experimental results conducted in several field situation.

I. 서 론

프로세스 제어, 생산 자동화, 빌딩 자동화 등을 위한 산업용 제어네트워크 시스템은 최근 많은 사용자에게 매력적인 시스템으로 인식되고 있다^[18]. 산업용 제어네트워크 시스템에서는 지능적이고 분산화된 네트워크노드가 다양한 통신매체를 통해 통신하며 상호 정보를 교환한

하면 센서버스, 디바이스버스, 그리고 필드버스로 구분할 수 있다^{[10],[20]}. 본 논문에서는 다양한 제어용 네트워크중 보편적인 응용을 위해 OSI(Open Systems Interconnection) 7계층을 모두 지원하는 Echelon사의 LonWorks^{[2]-[4],[15]}을 이용하여 구성된 네트워크 시스템을 대상으로 플로우제어기의 도입을 통한 성능향상을 도모한다. LonWorks는 앞서의 센서버스, 디바이스버스, 필드버스가 가지는 다양한 장점을 모두 지니며 최근 산업계의 자동화용 네트워크를 위한 새로운 네트워크로 부상하고 있다. 그러나 한편으로 LonWorks에서 채택하고 있는 프로토콜(LonTalkTM)이 CSMA(Carrier Sense Multiple Access)방식의 비결정적(nondeterministic) 특성을 지니기 때문에 실제 산업자동화 현장에서의 LonWorks이용이 지연되고 있다. 이런 비결정적 프로토콜은 채널경쟁을 바탕으로 하는 프

* 學生會員, ** 正會員, 蔚山大學校 電氣電子 및 自動化工學部

(School of Electrical Engineering and Automation, of Ulsan University)

※ 본 결과는 정보통신부의 정보통신 우수시범학교 지원사업에 의하여 수행된 것입니다.

接受日字:1999年 12月28日, 수정완료일:2000年 9月1日

로토콜이다. 즉, 데이터 전송시혼잡(congestion)발생 등으로 인해 전송오류가 발생하면 랜덤지연(random delay)후 재전송을 시도하게 된다. 따라서 노드의 수가 많아지거나 여러 노드가 동시다발적으로 전송을 시도하면 네트워크 성능이 급격히 저하되는 단점이 있다. 이런 성능저하는 특히 데이터의 실시간성을 요구하는 산업용 제어 네트워크에서는 아주 심각한 문제로 대두된다. 따라서 본 논문에서는 이러한 비결정적 프로토콜이 가지는 단점을 보완하기 위해 전송률을 기반으로 하는 플로우제어를 도입하고자 한다. 일반적으로 전송률기반 플로우제어기는 ATM(Asynchronous Transfer Mode)네트워크나 TCP/IP(Transmission Control Protocol/Internet Protocol)와 같은 네트워크에서 많이 연구 발전되어 왔다^{[6],[7]}.

본 논문에서는 주로 개루프(open-loop) 형태로 운용되는 산업용 네트워크에 전송률기반의 플로우제어를 도입함으로써 케환을 인가하여 폐루프(closed loop)를 형성해 전송률과 공정성측면에서 네트워크 성능향상을 도모하고자 한다. 또한 실험을 위해서 네트워크 노드를 자체 제작하고 LonBuilder^[15]에서 제공하는 에뮬레이터 2대를 트래픽 발생기(traffic generator)로 사용한다. 네트워크 노드가 3개일 때와 4개일 때 각각의 경우 데이터 전송시 각 노드의 전송률, 오류률, 공정성, 평균지연 등을 비교, 검증한다. 본 논문의 구성은 다음과 같다. 먼저 2절에서는 LonWorks의 일반적인 구조와 특성에 대해 기술하고 3절에서는 전송률기반 플로우제어의 설계에 대해서 설명한다. 다음 4절에서는 제안하는 제어 알고리즘에 대해 기술하며 5절에서는 실험 결과를 통해 제안된 제어알고리즘의 효용성을 검증한다. 끝으로 6절에서 결론을 맺는다.

II. LonWorks의 일반적인 구조와 특성

LonWorks에서 Lon은 Local Operating Network (LON)을 의미한다. LON은 각 노드간에 명령이나 상태 등 소량의 데이터를 송·수신하여 네트워크 전체 또는 그룹 단위의 고도로 조직화된 통신을 이용한 제어 기능의 수행을 목적으로 한다. 이를 위해 최하위 센서, 액추에이터등과 같은 디바이스와의 통신, 모니터/제어 기능을 담당하는 구성요소들을 동일한 프로토콜 상에서 동일한 선로를 통해 통신하고자 하는 네트워크 개념이다^[11]. 네트워크는 크게 데이터 네트워크와 제어용 네

트워크로 구분된다. 데이터 네트워크는 정보의 양이 많고 전송의 빈도가 많지 않으며 불규칙적인 특징을 갖는다. 반면 제어용 네트워크는 전송할 양이 적고 신뢰성이 있어야 하며 실시간적으로 수행되어야 한다. 제어용 네트워크인 LonWorks는 종종 다른 산업용 버스시스템과 비교된다. 앞서 설명한 바와 같이 산업용 버스시스템은 센서버스, 디바이스버스, 필드버스의 3종류로 분류될 수 있는데 그 중 LonWorks만이 모든 버스로서의 기능을 갖추고 있다. 그 차이점은 LonWorks는 센서, 액추에이터 등의 디바이스를 모두 지능화 한 다음 그것들을 점대점(peer to peer)으로 연결하여 분산적으로 제어시스템을 구축하는 반면 다른 산업용 버스의 대부분은 PLC, DCS와 같은 디바이스간의 통신망으로 위치가 부여되어 있기 때문에 집중형 제어에서 분산형제어로 구조를 근본적으로 전환하지 못하기 때문이다. 또한 LonWorks는 OSI 7계층을 모두 지원하므로 확장성 및 상호 운용성이 높고 여러 가지 통신매체(전력선, RF, 쌍꼬임선, 광파이버 등)를 다양하게 사용할 수 있고 자유구조(free topology)를 지원하며 점대점 통신을 하기 때문에 상위의 디바이스가 제 기능을 상실해도 상호 통신이 가능하다는 점들을 생각할 수 있다^[12].

LonWorks는 다음의 4가지 요소로 구성된다. 즉, 통신이나 제어를 목적으로 하는 어플리케이션을 보다 효율적으로 실현하기 위해 기능을 하드웨어화, 펌웨어화 하여 VLSI에 탑재한 뉴런칩(Neuron chip)과 다양한 전송매체를 지원하고 네트워크 노드간의 기계적이고 전기적인 매체 인터페이스를 제공하는 트랜시버(transceiver), 통신 프로토콜인 LonTalk, 그리고 LonWorks 네트워크의 OS 또는 NMS(Network Management System)라 할 수 있는 통신망 운용시스템(LNS : LonWorks network service)으로 구성된다. LonTalk 프로토콜은 기본적으로 4가지의 메시지 서비스를 제공한다. 즉, 노드 혹은 노드의 그룹에 메시지 전송 후 ACK 신호를 수신하지 못하면 타이머가 종료되면서 메시지를 재 전송하는 ACKD(end-to-end acknowledged)서비스와 클라이언트/서버 어플리케이션이나 원거리 프로시저 호출에 적합한 요구/응답(request/response)서비스, 대규모 네트워크 그룹에 전체송출(broadcasting)후 모든 노드의 응답에 의한 네트워크 부하를 방지하기 위해 사용하는 무응답 반복서비스(unacknowledged repeated service)서비스, 그리

고 높은 성능이 요구되거나 비교적 많은 양의 데이터를 전송할 때 사용하는 무응답(unacknowledged)서비스를 제공한다^[13]. 본 논문에서는 위 4가지 서비스 중 첫 번째의 ACKD서비스를 이용하였다.

III. LonWorks 네트워크시스템의 플로우제어

앞서 기술한 바와 같이 LonWorks를 포함한 대부분의 제어용 네트워크는 개루프 상태로 운용되고있다. 하지만 이러한 개루프 시스템은 노드가 많아질수록 오류률도 높아질 뿐 아니라 여러 노드가 동시 다발적으로 전송을 시도할경우 전송률이 급격히 감소하는 단점을 지니고 있다. 이러한 단점을 극복하기 위해 이 절에서는 LonWorks 네트워크 시스템을 대상으로 플로우제어를 통한 폐루프시스템의 개념을 설명한다.

1. 플로우제어의 개념

본 논문에서는 채널의 오류를 케환받고 다른 노드로부터 수신된 데이터의 주기를 측정해서 다음 데이터의 수신시점을 예상하고 이를 이용해 플로우제어를 통한 폐루프 시스템을 형성하여 산업용 제어네트워크에서 일반적으로 요구되는 특성을 벗어나지 않는 범위 내에서 네트워크 성능개선을 도모하고자 한다. LonWorks는 데이터링크계층이나 네트워크계층이 모두 ASIC화 되어있기 때문에 본 논문에서는 이러한 계층을 포함한 전체 네트워크를 제어대상 시스템으로 간주하고 채널의 오류 신호를 케환받는다. 또한 원형대기열을 설계하고 송신 데이터는 항상 대기열을 거쳐서 네트워크 채널을 통해 전송되는 구조로 간주한다. 이때, 일반적으로 송신데이터는 임의의 시간에 임의의 데이터가 대기열에 들어온다고 가정한다. 그림 1은 그 개념도를 보여준다.

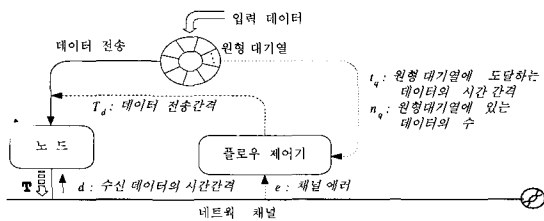


그림 1. 각 노드에 플로우제어를 도입한 폐루프시스템의 개념도

Fig. 1. Overall schematic diagram of the closed loop network configuration.

본 연구에서는 다음 3가지 정보를 케환받아 각 노드가 데이터를 전송하는 전송률을 제어하기 위해 전송률기반 플로우제어기(rate-based flow controller)를 설계한다. 즉, 채널의 에러(e)와 원형대기열에 도달하는 각 송신데이터간의 시간간격(t_q), 그리고 원형대기열에 있는 송신데이터의 총 개수(n_q)를 케환받아 제어 입력인 노드의 데이터 전송시간간격 T_d 를 결정한다. 이러한 제어입력 T_d 를 조절함으로써 노드의 데이터 전송주기 T 를 제어하게 된다. 이때 전체시스템의 입·출력 구성은 다음과 같다. 먼저 센서나 액츄에이터등의 디바이스로부터 다른 노드로 전송되어질 데이터, 즉 입력트래픽(input traffic)을 전체 시스템의 입력으로 간주한다. 그리고 전체 네트워크 시스템의 전송률을 출력으로 간주한다. 앞서 기술한 바와 같이 LonWorks는 다른 산업용 네트워크와는 다르게 비결정론적 통신 프로토콜을 사용하기 때문에 어떤 특정한 노드로 정체 현상이 발생하면 다른 노드들간의 통신이 두절되는 경우가 발생한다. 그러나 LonWorks는 커넥션 기반(connection-based) 네트워크가 아니라 정보 기반(information-based) 네트워크이다. 따라서 네트워크가 일단 구성되고 나면 네트워크 변수 바인딩 정보를 기반으로 전송이 이루어져 특정한 노드로 전송이 집중되는 정체 현상은 거의 발생하지 않는다. 그러므로 본 연구에서는 어떤 특정한 노드로 데이터가 집중되는 경우는 고려하지 않는다. 본 연구에서 제시한 플로우제어기의 성능을 검증하기 위해서 전송률과 공정성 2가지의 성능지수를 설정한다.

2. 성능지수

전송률은 일반적으로 전송한 데이터와 성공적으로 전송된 데이터의 비율로 나타낸다. 본 논문에서는 전송률을 나타내기 위해 32byte 단위의 데이터를 1000회 전송하고 그 중 성공적으로 전송된 데이터의 횟수를 측정하였다. 그림 6과 그림 8에 각 실험에서의 전송률을 그래프로 나타내었으며 초당 전송된 평균 패킷을 나타내었다. 한편, 네트워크 상호간 통신 수준에서 공정성은 일반적으로 사용자 지연이나 망의 처리량에 의해 제한된다. Mark는 공정성은 모든 사용자에게 똑 같은 만족을 제공하는 능력이라고 정의하였다^[1]. 사용자의 만족이란 모든 사용자가 동등한 평균 지연시간을 가지고 채널을 접근하며 처리량 또한 일정하다는 것을 의미한다. 전송률향상과 함께 공정성을 보장하기 위해서

공정성 지수를 설정한다. 각각의 노드에 입력되는 데이터와 최대 오류와의 비율과 최소 오류와의 비율의 차가 0에 접근할수록 각 노드는 공정한 채널 접근 기회를 가진다고 정의한다. 예를 들면 전송할 데이터가 100개인 노드가 50개의 데이터를 성공적으로 전송하고, 전송할 데이터가 10개인 노드가 5개의 데이터를 성공적으로 전송했다면 각 노드가 공정한 채널 접근 기회를 가진다고 생각한다. 따라서 본 논문에서는 다음과 같이 공정성 지수를 설정한다.

$$J_f = \frac{\max(I_i - S_i)}{I_i} - \frac{\min(I_i - S_i)}{I_i}, \quad [i \in N]. \quad (1)$$

여기서 I_i 는 i 번째 노드의 입력 데이터 개수, S_i 는 i 번째 노드가 성공적으로 보낸 데이터 개수를 나타낸다. 본 논문에서는 네트워크의 제한정보를 이용하여 전송률을 최대화하며 이러한 공정성 지수를 최소화시키는 제어입력을 고려한다.

IV. 제어알고리즘

산업용 제어네트워크는 많은 양의 데이터를 지속적으로 전송하는 것이 아니라 네트워크 상황에 따라 실시간 데이터를 처리해야 하기 때문에 공정성 측면에서뿐만 아니라 전송률 측면에서도 제어네트워크 특성상 어떤 송신데이터가 발생하고 그 다음 송신데이터가 발생하는 시간의 간격은 제어입력을 결정할 때 상당히 중요한 의미를 갖는다. 또한 전송할 데이터가 많이 있는 노드에게 좀더 많은 기회를 주기 위해서 대기열에 존재하는 데이터의 수도 고려한다. 즉, 제어입력은 예측된 시점 내에 전송오류가 많고 데이터가 대기열에 도달하는 시간간격이 길수록 채널접근을 적게 시도하고 대기열에 존재하는 데이터의 수가 많을수록 채널접근을 많이 시도하도록 고려한다. 따라서 본 논문에서 제안하는 제어알고리즘의 n 번째 전송시 제어입력 $T_d(n)$ 는 다음과 같다 [$n \in N$].

$$T_d(n) = k_1 \cdot E(n-1) + k_2 \cdot \sum t_q(n-1) + k_3 \cdot \frac{1}{n_1(n-1)}. \quad (2)$$

여기서 $E(n-1)$ 은 $(n-1)$ 번째 채널오류율, $\sum t_q(n-1)$ 은 $(n-1)$ 번째까지 대기열에 입력되는 각각

의 송신데이터들의 시간간격의 총합을, 그리고 $n_q(n-1)$ 은 $(n-1)$ 번째 전송까지 대기열에 도달하는 송신데이터들의 개수를 의미한다. 또한 k_1, k_2, k_3 은 각각의 가중치를 나타낸다. 이때 가중치 k_1, k_2, k_3 은 실험을 통해서 결정한다. 플로우제어기 설계시 원형대기열에 대기중인 송신데이터의 수가 원형대기열의 크기의 한계에 도달하면 제어알고리즘에 관계없이 일단 데이터를 전송하게 된다. 실제 실험에서는 이러한 한계상황을 방지하기 위해서 대기열 가운데 2개의 슬롯(slot)만큼 비워두었다. 실험에서는 대기열 크기를 100개의 슬롯으로 하였는데 실제 제어 알고리즘이 동작하는 범위는 98슬롯까지이고 98슬롯에 도달하게 되면 대기열이 가득 찬 것으로 간주한다. 제안한 제어알고리즘의 변환변수 중 제어입력 결정에 중요한 영향을 끼치는 2가지 주요 매개변수를 설정한다. 먼저 제어네트워크 특성상 송신데이터가 대기열에 입력되는 순차적 시간간격은 중요한 의미를 가질 뿐 아니라 알고리즘의 수렴성질로부터 중요하므로 t_q 를 먼저 선택하고, 또한 알고리즘의 수렴성이 원형 대기열의 크기에 영향을 받으므로 n_q 를 선택한다. 다음은 제어 알고리즘의 흐름도이다.

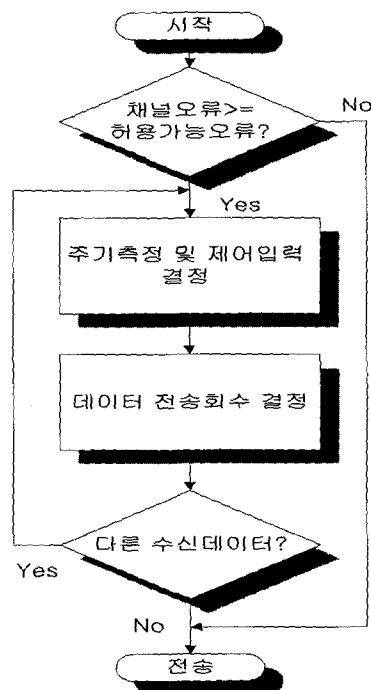


그림 2. 제어알고리즘의 흐름도
Fig. 2. Flow chart of control algorithm.

다음은 알고리즘의 pseudo code이다.

```

Begin(control algorithm)
#define: cycle = the data incoming cycle that means
           the next data incoming time.
#define:  $N_d$  = the next incoming data cycle;
START: Transmit;
Step 1: If channel error > admittable error then
    begin
        calculate cycle;
        estimate  $N_d$ ;
        end;
    else begin
        goto START;
        end;
Step 2: determine the control input  $T_d$ ;
        goto Step 3;
Step 3: If  $T_d < N_d$  then begin
        n =  $N_d / T_d$ ;
        transmit the data n times;
        goto Step 4;
        end;
    else begin
        transmit data at  $T_d$  time;
        goto Step 4;
        end;
Step 4: If another input data is coming then begin
        ignore the previous estimated value;
        goto Step 1;
        end;
Return;
end; (control algorithm)
    
```

위 제어알고리즘은 허용된 채널오류률을 초과할 때 작동한다. 채널오류률의 허용치는 네트워크 구성시 결정된다. 전송을 하면서 다른 노드로부터 들어오는 수신데이터의 시간간격(cycle)을 측정을 해서 다음 수신데이터의 도달시간(N_d)을 예측한다. 이 예측된 시점과 제어입력(N_d)을 비교하여 데이터를 송신할 시점을 결정하여 전송을 시도한다. 만약 데이터를 송신하기 전에 또 다른 수신데이터가 들어오면 이전의 값들은 무시하고 수신데이터의 주기를 재 측정하여 다음 제어알고리즘을 적용한다. 위와 같은 제어알고리즘을 기반으로 플로우제어기를 설계한다. 제안한 제어알고리즘의 안정성(stability)을 보이기 위해 계산상의 복잡도(complexity)와 수렴성(convergency)에 대해 기술한다. 본 논문에서 제시한 알고리즘은 비교적 간단한 계산과정을 반복하지만 예상된 시점보다 다음 수신데이

터가 빨리 도달하면 계산과정이 복잡해진다. 앞서 설명한 것처럼 $(n+1)$ 번째 수신데이터 [$d(n+1)$]가 제어입력보다 빨리 수신되면 제어알고리즘을 재적용해야 하므로 $(n+1)$ 번째 수신데이터가 빨리 도달할수록 계산이 복잡해진다. 따라서 제어 알고리즘의 계산상의 복잡도는 $O(\frac{1}{d(n+1)} \max[t_q(n-1), n_q(n-1)])$ 에 비례한다. 만약 $(n+1)$ 번째 수신데이터가 계속해서 제어입력(T_d)보다 빨리 도달하게 되면 제어 알고리즘은 수렴하지 못하고 대기열에 송신데이터가 가득 차게 된다. 만약 대기열이 무한히 크다면 언젠가는 안정되겠지만 실제 시스템에서는 불가능하다. 따라서 다음과 같이 그 한계를 정한다. 제어 알고리즘의 수렴성 조건은 모든 정수 m 에 대해 $d(n+m) > T_d(n+m-1)$ 이면 항상 수렴하고 $d(n+m) < T_d(n+m-1)$ 일 때는 $n_q \leq l_q$ 의 범위내에서 수렴하게 된다. 이 때 T_d 는 제어입력이고, l_q 는 대기열의 크기이다.

V. 실험

제안한 플로우제어기의 성능을 검증하기 위해 자체 제작한 LON노드를 이용하여 실험실 규모의 제어네트워크를 구성한다. 그리고 네트워크 채널 상태를 좀더 현장과 비슷하게 만들기 위해서 트래픽 발생기를 설치한다. 제작한 노드의 확장성 및 유연성을 높이기 위해 두 가지 종류의 뉴런칩 중 도시바사의 TMPN3150B1AF를 사용하였으며 트래픽 발생기는 LonBuilder에서 제공하는 에뮬레이터를 이용한다. 메시지서비스는 4가지 LonTalk메시지 전송서비스 중 ACKD서비스를 이용하였고 모든 메시지의 우선순위는 동일하다고 가정한다. 지연에 영향을 미치는 실제 쌍꼬임선의 길이는 약 850cm이다. 실험은 노드가 3개, 4개일 때 각각 제어알고리즘이 있을 때와 없을 때 전송률과 채널의 오류를 측정 비교한다. 실험은 각각 10번씩 수행하여 그 평균값을 계산한다. 전송할 데이터가 임의의 시각에 원형대기열에 입력되지만 전체네트워크가 혼잡상황(congestion)에 도달하지 않는 정상동작(normal operation)상황의 경우뿐만 아니라 각 노드에서 동시다발적으로 빠른 데이터입력주기(이번 실험의 경우 1ms)를 가지고 전송을 시도하여 결국 혼잡으로 인한 네트워크 정체현상이 야기되는 극단적인 동작(worst operation)상황의 경우에 대해서도 제안한 제어알고리

즘의 적용을 통한 성능향상을 각각 검증한다.

1. 주요 매개 변수의 가중치 결정

제어입력중 k_1 은 채널오류가 많아지면 노드의 채널 접근시도를 적게 만드는 역할을 하고 k_2 는 대기열에 입력되는 송신데이터의 시간간격이 빠르면 노드의 채널 접근시도를 많게 하는 역할을 한다. 또한 k_3 는 대기열에 입력되는 데이터의 수가 많으면 노드의 채널 접근시도를 많게 하는 역할을 한다. 이들중 노드와 직접적인 관련을 가지는 k_2 와 k_3 를 주요매개변수로 결정하고 가중치를 결정하기 위해 k_1 값은 1로 고정시키고 (실험결과 k_1 의 고정치에 크게 영향을 받지 않음) 나머지 두 값을 변화 시켜가며 실험을 하였다. 그 결과 k_2 가 7일 때, k_3 가 3일 때 전송률이 가장 높게 나타났고 공정성도 평균 이하의 값을 나타내었다. 공정성은 k_2 가 1일 때, 그리고 k_3 가 3일 때 가장 낮은 값(0.006)을 나타내었다. 다음은 가중치를 결정하기 위한 실험결과이다.

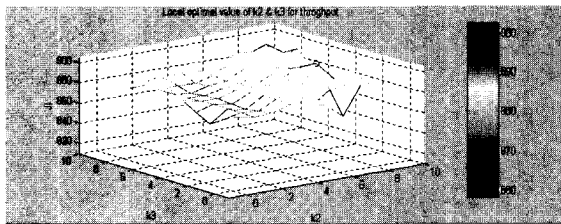


그림 3. 전송률을 최대화하는 가중치값
Fig. 3. Weighting factor value of maximum throughput.

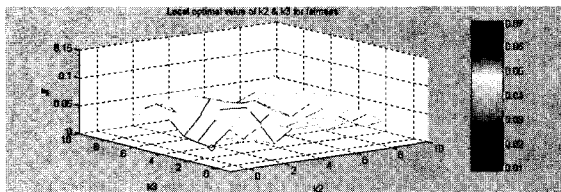


그림 4. 공정성을 최대로 하는 가중치값
Fig. 4. Weighting factor value of minimum fairness.

위 그림에서 'o'표시가 되어있는 부분이 전송률을 최대로 하는 가중치 값과 공정성을 최소로 하는 가중치 값이다. 제어 입력 T_d 는 앞서 기술한 것과 같이 에러에 대한 요소와 대기열에 들어있는 송신데이터의 요소들로 구성된다. 그 중 노드의 개수와 관련 있는

에러에 대한 요소는 주요매개변수가 아니기 때문에 결정된 가중치는 노드의 개수에 거의 영향을 받지 않음을 실험을 통해 확인하였다.

2. 정상동작 상황의 경우

정상동작 상황하에 먼저 네트워크 노드가 3개인 경우의 실험을 수행하였다. 실험을 위해서 다음 그림 5와 같이 네트워크를 구성하였다. 트래픽 발생기(TG)는 다른 노드에 영향을 주지 않고 채널 상태에만 영향을 준다. 다음은 그 구성과 실험 결과이다.

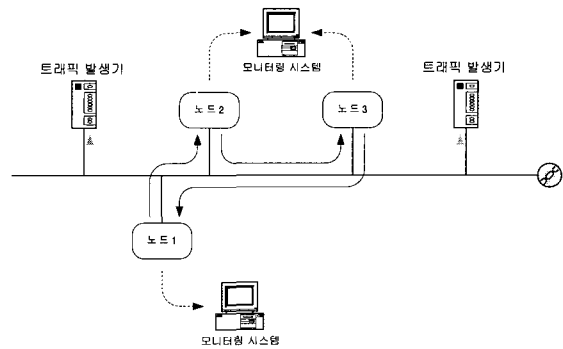


그림 5. 노드가 3개일 때 실험 구성도
Fig. 5. The experimental configuration of 3 nodes network.

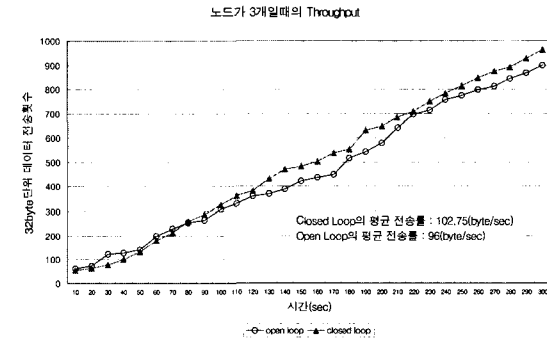


그림 6. 노드가 3개일 때의 전송률비교
Fig. 6. The comparison of throughput (3 nodes).

다음 표 1은 노드가 3개일 때 개루프일 때와 폐루프일 때 공정성 지수와 오류률, 평균지연의 비교를 나타낸다. 공정성 지수는 본문에서 제시한 (1)을 따랐다. 이때 가중치는 앞서 기술한 실험을 통해 결정하였다. 즉, k_1 은 1이며 k_2 는 7, k_3 는 3이다. 평균지연은 가장 일반적인 방법인 파형측정기(oscilloscope)를 이용하여 측정하였다.

표 1. 공정성 지수, 오류률 및 평균지연의 비교 (노드 3의 경우)

Table 1. The comparison of fairness, error rate, and average delay (3 nodes).

	개루프	폐루프
J_f	0.094	0.021
오류률(%)	10	3.68
평균지연(ms)	33.9	37.2

위 그림 6과 표 1에서 개루프일 때에 비해 폐루프일 때 평균지연의 큰 증가없이(9.7%) 전송률(7%)과 공정성, 오류률이 향상(각각 77.7%, 63%)되었음을 보여준다. 평균지연이 약간 증가하였으나 실시간 데이터의 일반적인 허용지연이 데이터의 크기가 10kbit이하일 때 20~100ms인 점을 감안하면 충분히 데이터의 실시간성을 보장함을 알 수 있다^[16]. 다음은 동일한 실험을 네트워크 노드가 4개일 때 수행한 결과이다. 실험을 위해서 1번 노드는 2번 노드에, 2번 노드는 3번 노드에 3번 노드는 4번 노드에 4번 노드는 1번 노드에 데이터를 전송하게 구성하였다. 다음은 그 구성과 실험 결과이다.

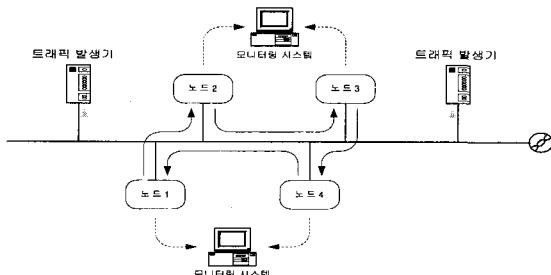


그림 7. 노드가 4개일 때 실험 구성도
Fig. 7. The experimental configuration of 4 nodes network.

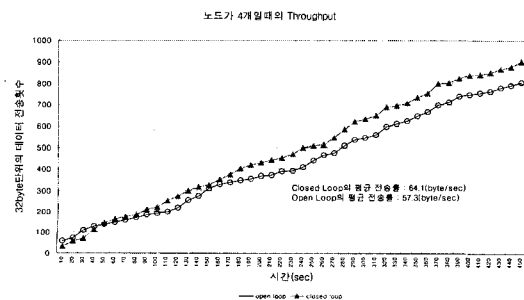


그림 8. 노드가 4개일 때의 전송률비교
Fig. 8. The comparison of throughput (4 nodes).

다음의 표 2는 노드가 4개일 때 개루프와 폐루프의 공정성 지수, 오류률과 평균지연의 비교를 나타낸다. 이때 가중치는 노드가 3개일 때와 동일하게 k_1 은 1이며 k_2 는 7, k_3 는 3이다.

표 2. 공정성 지수, 오류률 및 평균지연의 비교 (노드4의 경우)

Table 2. The Comparison of fairness, error rate, and average delay (4 nodes).

	개루프	폐루프
J_f	0.16	0.034
오류률(%)	19.4	9.8
평균지연(ms)	113.2	121.9

노드가 4개일 때 역시 개루프일 때에 비해 폐루프일 때 평균지연의 큰 증가 없이(7.6%) 전송률(12%)과 공정성, 오류률이 향상(각각 78%, 49%)되었음을 보여준다.

위 실험결과에서 알 수 있듯이 노드가 많아질수록 제한한 플로우 제어를 통해 본 논문에서 제시한 성능지수 측면에서 네트워크 성능이 향상되는 것을 알 수 있다.

3. 극단적인 동작상황의 경우

동시다발적인 데이터 전송시도로 네트워크 혼잡이 발생하는 경우에는 전송할 데이터의 생성되는 주기가 1ms로 빠르기 때문에 제어알고리즘을 수행하기 전에 또 다른 데이터가 수신되어 원형대기열이 가득 차는 한계상황에 도달하게 된다. 플로우제어기가 없을 때 (open-loop)는 선형적으로 전송을 하다가 혼잡상황에 도달하게 되면 더 이상 전송을 하지 못하는 상황이 된다. 그러나 제안하는 플로우제어를 도입한 경우 (closed-loop)는 대기열이 가득 차있으면 제어알고리즘이 동작하지 않고 전송을 시도하다가 대기열이 비게 되면 다시 제어알고리즘을 수행하는 동작을 반복하게 된다. 그림 9에서 보여지듯이 네트워크 노드의 동시다발적인 다량의 데이터 전송시도로 네트워크의 혼잡이 예상되는 상황에서도 제안하는 플로우제어를 통해 데이터의 실시간성은 만족시키지 못하지만 네트워크 채널이 허용할 수 있는 최대 전송률을 보장하며 혼잡을 방지하는 안정적인 특성을 보임을 알 수 있다. 이러한 특성은 노드가 3개일 때 역시 마찬가지로 나타난다.

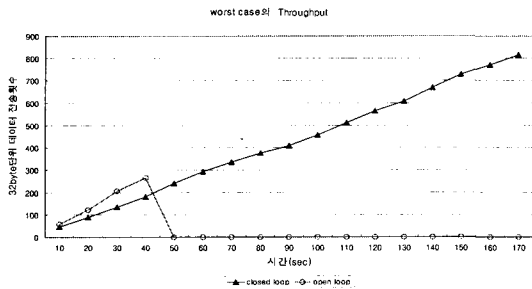


그림 9. 노드가 4개일 때 극단적인 상황하에서의 전송률 비교

Fig. 9. The comparison of throughput in the worst operation (4 nodes).

VI. 결 론

본 논문에서는 최근 각광 받고 있는 여러 가지 다양한 산업용 제어네트워크 중에서도 다양한 통신매체를 제공하고 OSI 7계층을 모두 지원하며 상호운용성이 높은 LonWorks를 이용하여 그 특성을 파악하고 성능향상을 위한 플로우제어기법을 제안하였다. LonTalk 프로토콜이 ASIC화 되어있으므로 각 계층을 포함한 전체 네트워크를 제어대상 시스템으로 간주하고 채널의 오류신호와 원형대기열에 존재하는 데이터의 개수, 데이터가 입력되는 시간간격 등을 궤환받아 제어알고리즘을 설계하였다. 그 결과 비결정적 프로토콜을 사용하는 LonWorks에서 노드의 수가 증가할수록 제안한 플로우제어기법에 의해 데이터의 실시간성을 보장하면서 전송률과 공정성 측면에서도 네트워크 성능이 향상됨을 알 수 있었다. 또한 알고리즘의 안정성을 나타내기 위해 계산상의 복잡도와 수렴성을 분석하였으며 여러 네트워크 노드가 동시다발적인 전송시도로 네트워크 채널에 혼잡이 발생하는 극단적인 동작상황의 경우에도 제안하는 플로우제어를 통해 네트워크 성능의 안정성을 검증하였다. 특히 본 논문에서는 채널에 부하를 주지 않고 채널의 상태를 감시하며 전송률을 측정하기 위해서 모니터링 노드를 도입하였다.

향후 연구과제로는 LonTalk 각 계층의 정확한 시뮬레이션 모델링과 상위계층인 LNS를 이용하여 인터넷과의 연계를 통한 통합 네트워크시스템의 구축 및 그에 따른 성능평가 필요하다.

참 고 문 헌.

- [1] C. Douligeris and L. N. Kumar, "Access to a network channel: A survey into the unfairness problem", *Proc. ICC '92*, pp. 1184-1185, 1992.
- [2] J. M. Alonso, J. Ribas, J. J. Del Coz, M. Rico, "Intelligent control system for fluorescent lighting based on LonWorks technology", *Proc. of 24th Annual Conf. of IEEE Industrial Electronics Society*, vol. 1, pp. 92-97, 1998.
- [3] R. A. Ziemerink and C. P. Bodenstein, "Utilising a LonWorks control network for factory communication to improve overall equipment effectiveness", *Proc. of IEEE Int. Symp. on Industrial Electronics '98*, vol. 2, pp. 684-689, 1998.
- [4] P. Palensky, R. Poster, and H. Reiter, "Demand side management in private homes by using LonWorks", *Proc. of IEEE Int. Workshop on Factory Communication System*, pp. 342-347, 1997.
- [5] P. W. M. Tsang and R. W. C. Wang, "Development of a distributive lighting control system using local operating network", *IEEE Trans. on Consumer Electronics*, vol. 4, pp. 879-889, 1994.
- [6] K.-H. Cho and J.-T. Lim, "Supervisory rate-based flow control of ATM networks for ABR services", *IEICE Trans. on Communications*, vol. E81-B, no.6, pp. 1269-1271, 1998.
- [7] S. Chong, R. Nagarajan and Y. T. Wang, "First-order rate-based flow control with dynamic queue threshold for high-speed wide-area ATM network", *Computer Network & ISDN Systems*, vol. 29, no. 17-18, pp. 2201-2212, 1998.
- [8] 홍승호, 김기암, 김 지용, 고성준, "분산제어 및 자동화시스템과 필드버스", *제어·자동화·시스템 공학회지*, 제2권, 제4호, pp.19-20, 1996

- [9] 김기암, 홍승호, “자동화시스템에 Profibus 네트워크 인터페이스 구현 및 성능평가”, *제어·자동화·시스템공학 논문지*, 제4권, 제1호, pp. 113-114, 1998
- [10] 편집부, “Device Network이란 무엇인가”, *월간 CONTROL*, pp. 48-54, 1월호, 1998.
- [11] 枝隆可, “CIM/FA에 혁명이 기대되는 분산형 지적 제어네트워크”, *자동화 기술*, pp. 39-47, 8월호, 1997
- [12] 高井 正美, “계측 기술과 LonWorks”, *월간 자동제어계측*, pp. 72-73, 11월호, 1998
- [13] *LonTalk Protocol Specification ver3.0*, pp. 20-23, Echelon, Co., 1994.
- [14] *LonBuilder users guide*, Echelon, Co., 1995.
- [15] *LonWorks Technology Device Data*, Motorola Inc., USA, 1995.
- [16] 홍승호, “필드버스 기술 동향”, *제어·자동화·시스템공학회지*, vol. 4, no. 6, pp. 13-14, 11월호, 1998.
- [17] J. A. Janet, W.J. Wiseman, “Using Control NetWorks for Distributed Robotic Systems”, *Proceeding of the 1999 IEEE International Conference on Robotics & Automation*, vol. 2, pp. 1138-1143, May, 1999.
- [18] Keon Young Yi, Se Joong Jeon, “Design and implementation of a Process Control Language for an Automated Paper Manufacturing System”, *Proceeding of the 1999 IEEE International Conference on Robotics & Automation*, vol. 2, pp. 1132-1137, May, 1999.
- [19] 홍승호, “Performance analysis of priority scheme in the IEC/ISA fieldbus”, *전자공학회 논문지-S*, vol. 35-S, no. 10, pp. 94-117, Oct, 1998
- [20] 홍승호, “센서 레벨 네트워크 필드버스 기술 개요”, *전자공학회지*, vol. 21, no. 4, pp. 87-95, Apr, 1994

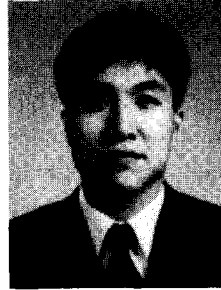
저 자 소 개



金秉熙(學生會員)
 1998년 울산대학교 전기전자 및 자동화공학부 제어계측공학과 졸업(B.S.). 2000년 동대학원 석사(M.S.) 관심분야는 산업용네트워크, 분산제어, 실시간제어 및 IBS 시스템



朴京燮(正會員)
 1974년 서울대 공대 전기공학과 졸업(B.S.). 1976년 서울대 대학원 전기공학과 졸업(M.S.). 1989년 Univ of Texas at Austin 전기공학과 졸업(Ph.D.). 1979년~현재 울산대 전기전자 및 자동화공학부 교수, 관심분야는 광센서, 초고속산업용 네트워크, 자율이동로봇 등



曹光鉉(正會員)
 1989. 3.~1993. 2. KAIST, 전기 및 전자공학, B.S. 1993. 3.~1995. 2. KAIST, 전기 및 전자공학, M.S. 1995. 3.~1998. 8. KAIST, 전기및전자공학, Ph.D. 1998. 9.~1999. 2. KAIST, 전기및전자공학과, 위촉연구원 및 연수연구원. 1999. 3.~현재 울산대학교, 전기전자및자동화공학부, 전임강사. 관심분야는 Supervisory control of discrete event systems, nonlinear control, analysis and synthesis of hybrid systems; Applications including : control and automation of manufacturing systems, synthesis of industrial control networks, supervisory control of semiconductor manufacturing systems, congestion control of communication networks, traffic management in IVHS, etc.