

생산자동화를 위한 토큰버스 네트워크의 성능관리

이상호, 이 석, 손준우*

부산대학교 기계공학부, *대우정밀 기술연구소

1. 서 론

생산의 관리 및 자동화를 위한 컴퓨터와 전자 기술의 응용은 50년대 초에 실용화된 수치제어 공작기계에서 시작하여 산업용 로봇, 프로그래머블 로직 컨트롤러, CAD 시스템 등에 이르기까지 매우 필수적인 요소가 되었다. 또한, 하드웨어 가격의 지속적인 하락으로 거의 모든 분야에서 컴퓨터를 사용하고 있으며, 컴퓨터의 수적 증가와 더불어 자동화 시스템의 보급도 크게 증가되었다. 그러나, 종래와 같은 컴퓨터 시스템의 사용은 여러 기구들을 각기 독립적으로 운용함으로써, 자동화의 섬(islands of automation)들을 형성하기에 이르렀으며, 생산성의 향상에 있어서도 기대만큼의 성과를 얻지 못하였다.

이러한 자동화의 섬들을 연결하기 위하여 근거리 통신망(Local Area Network, LAN)[1]이 생산자동화 분야에 응용되면서, 제한된 공간 내에 분산되어 있는 여러 컴퓨터와 생산장비들을 공통된 전송 매체로 연결하여 상호 간의 정보 교환을 가능하게 하였다. 이와 같은 근거리 통신망에 접속된 두 기구는 상위 기구의 도움없이 직접적인 통신을 수행할 수 있으므로, 이를 바탕으로 컴퓨터 통합 생산(Computer Integrated Manufacturing, CIM)의 개념을 구현할 수 있게 되었다. 이러한 컴퓨터 통합 생산은 한 기업체 안에서 수행되는 다양한 기업활동들을 하나의 폐쇄 루프를 이루는 대화형 제어 시스템으로 통합하려는 노력으로서[2], 매출보고서, 생산주문, 제품 설계도, 공정 총괄, 수치제어 공작기계의 프로그램, 여러 생산장비의 상태 등과 같은 기업 내에서 발생하는 다양한 정보를 부서간에 공유하기 위해서는 컴퓨터 네트워킹은 없어서는 안될 중추적인 역할을 맡고 있다.

컴퓨터 통합 생산을 위한 네트워크는 전송 매체(transmission medium)와 토폴로지(topology, 네트워크에 접속된 기구들이 연결된 형태), 통신규약(communication protocols) 등을 선정함으로써 기본적인 설계가 이루어진다. 이 단계에서 네트워크의 용량(전송속도), 접속될 기구의 종류와 숫자, 그리고 전송할 데이터의 종류 등을 고려하여야 한다. 이와 같은 기본 요소들이 선정되면 통신규약을 위한 변수에 적절한 값을 부여하여 시스템을 구현하는 것이 일반적인 네트워크의 설계와 구현에 포함된다. 하지만, 네트워크를 운영함에 있어서 통신 부하와 통신자원은 시간에 따라 항상 변화하는 특성을 가지므로 네트워크의 성능도 이들의 상태에 따라 큰 영향을 받는다. 더욱이, 최근의 대규모 통합시스템의 추세는 접속기구의 급격한 증가와 다양화로 인해 효율적인 네트워크의 운영과 유연한 재구성을 요구하게 되었고, 이러한 효율적인 네트워크 운영을 위한 자동화된 성능관리가 중요시되고 있다[3-7].

따라서, 본 논문에서는 공장자동화용 표준 통신 프로토콜인 MAP(Manufacturing Automation Protocol)[8-9]에 이용되는 IEEE 802.4 토큰버스 프로토콜[10]을 위한 성능관리기의 개발에 대한 최근의 연구 결과를 소개한다. 토큰버스 프로토콜의 성능관리기는 종래에 수동적으로 조절되어 왔던 우선순위 도구(priority mechanism)의 변수를 상황에 따라 자동적으로 조절하여 성능을 개선시키는 것을 목표로 개발되었다. 이러한 관리기능은 버스의 사용 권리에 대한 우선권을 문서 파일, CAD 파일, 센서 데이터 등과 같은 다양한 종류의 메시지에 차등적으로 부여하기 위한 우선순위 도구의 변수들을 설정하고, 또한 시간적으로 변화하는 상황에 대응할 수 있는 기능을 추가하는 것이다. 이로 인하여 네트워크의 설계 단계에서 고려하지 못했던 상황으로 인한 성

능의 악화를 완화할 수 있어서 여러 종류의 메시지가 요구하는 다양한 네트워크 서비스의 수준을 적절하게 유지할 수 있는 것이다.

본 논문은 모두 다섯 절과 하나의 부록으로 이루어져 있으며, 2절에서는 성능관리의 기능과 구현 기법에 대하여 설명하고 있다. 그리고 3절에서는 개발된 3가지 형태의 성능관리기의 구성과 특징을 보이며, 4절에서는 이들을 네트워크 시뮬레이션 모델과 결합하여 각각의 성능평가를 수행한 결과를 제시한다. 끝으로 5절에서는 4절에서 얻어진 성능평가를 토대로 얻어진 결과를 요약한다. 그리고, 부록에서는 IEEE 802.4 토큰버스 프로토콜의 우선 순위 도구와 그 특성을 설명한다.

2. 성능관리의 기능과 기법

성능관리의 기능은 통신규약의 조절 가능한 변수들을 실시간에 변화시킴으로써 네트워크가 동적인 환경에 적절히 대응하도록 하여 요구되는 성능을 유지하는 것이다. 이와 같은 기능은 공장자동화용 네트워크의 통신 부하가 새로운 주문이나 생산장비의 고장과 같은 사건으로 인하여 증가되었을 때 제어 신호와 같은 다른 메시지에 미치는 악영향을 감소시키는데 필수적이다. 이를 위하여 네트워크의 성능과 통신규약의 변수들 사이의 수학적 관계가 필요하다. 컴퓨터 네트워크와 같은 이산 사건 동적 시스템(Discrete Event Dynamic System, DESS)의 성능평가를 위한 대기 이론(queueing theory)[11]과 같은 해석적인 기법이 존재하지만 성능관리를 위하여 몇 가지 어려움을 갖고 있다. 그 첫째로 Poisson arrival 같은 대기 이론의 가정들을 실제 상황에서 만족시키기 어렵지 않은 점을 들 수 있다. 또한, 네트워크의 성능 지표가 복잡해지면 수학적 처리가 곤란한 경우도 있다. 이와 같은 난관을 모두 극복한다 하여도 queueing model의 입력인 네트워크 트래픽(network traffic)의 통계적인 특성을 실시간에 관찰을 통하여 추정하는 난해한 문제에 봉착할 수 있다.

한편, 이산 사건 시뮬레이션(discrete event simulation) [12, 13]도 성능관리에 사용할 수 있는 한 방법으로서 엄격한 가정이 필요하지 않고 복잡한 형태의 성능 지표도 큰 어려움 없이 다룰 수 있다. 그러나, 비교적 신뢰할 수 있는 성능평가를 위해서는 여러 가지 다른 랜덤 시드(random seed)를 사용한 독립적인 시뮬레이션을 반복하여야 하기 때문에 계산적인 부담이 커지는 경향이 있다. 더구나, 통신규약의 변수들을 조절하기 위하여 현재의 변수 값 주위의 몇 개의 점에서 성능을 추정하는 것이 필요하므로 계산적인 부담이 더욱 가중되는 문제점이 있다. 또한, 이산 사건 시뮬레이션도 네트워크 트래픽의 통계적인 특성이 입력으로 필요한데 이를 실시간에 관찰을 통하여 추정하는 것을 피하기

위하여 실제로 네트워크에서 전송되는 각각의 메시지의 생성 시간과 크기를 기록하였다가 시뮬레이션 모델의 입력으로 사용할 수도 있을 것이다. 그러나, 이를 위해서는 네트워크에 접속되어 있는 모든 스테이션(station)이 기록한 정보를 시뮬레이션 모델이 연결된 스테이션에 전송하여야 하는데 이 추가적인 전송으로 네트워크의 성능이 저하될 가능성도 있다.

최근에 기존 기법의 제한을 회피하기 위하여 PA(Perturbation Analysis), SO(Stochastic Optimization), LA(Learning Automata) 등의 이론을 복합적으로 이용하여 그 유효성을 입증한 경우가 있으나, 그 구조가 상당히 복잡하며 프로토콜의 일부를 개정해야 할 필요성도 가지고 있었다[3-6]. 이러한 기존 기법의 난점의 근본적인 원인은 네트워크의 성능과 통신규약의 변수 사이의 수학적 관계가 명확히 알려지지 않은 점이다. 이 같이 제어나 관리 대상에 대한 정량적인 지식이 완전하지 못한 경우에 인간의 언어적인 추론 기능을 모방한 퍼지 기법[14-16]을 응용해서 많은 분야에서 상당한 성공을 거둔 점에 착안하여 본 연구에서는 네트워크의 성능관리에 이러한 퍼지 기법을 도입하였다. 즉, 네트워크의 성능과 통신규약 변수의 관계에 대한 지식과 새로운 변수값의 결정에 관한 지식을 퍼지 규칙의 형태로 표현하고 퍼지 추론을 통하여 새로운 변수값을 선정함으로써 기존 기법의 난관을 극복하고 효과적인 결과를 기대할 수 있다.

3. 퍼지 네트워크 성능관리기의 개발

IEEE 802.4 통신규약에서는 부록에서 설명된 것처럼 네 가지의 우선순위를 제공하고 이들 우선순위에 차등적으로 매체 접속권을 부여한다. 이 차등적인 접속 권리는 네 가지의 타이머 값, 즉 THT(Token Holding Time)와 TRTi(Token Rotation Time, $i=4,2,0$)으로 결정된다. 이러한 변수들의 값이 전송지연(data latency, 큐에서의 메시지 대기 시간과 메시지의 전송시간의 합)에 근거한 네트워크 성능과 밀접한 관계가 있고, 이 관계가 네트워크의 성능관리를 위한 핵심적인 요소가 된다. 즉, 네트워크 성능과 밀접한 관계가 있는 메시지들의 전송지연은 네 타이머의 값에 따라 변화하므로, 이를 상황에 따라 적절히 조절함으로써 각 우선순위마다 요구되는 최대 전송지연을 초과하지 않도록 하여 성능을 향상시킬 수 있다.

전송지연과 타이머 값들의 관계에 대한 지식 베이스를 축적하기 위하여 SIMAN[13]을 이용하여 이산 사건 시뮬레이션 모델을 구축하였고 이 모델에 다양한 조합의 타이머 값을 입력하여 전송지연의 변화를 관찰함으로써 다음과 같은 특성을 파악하였다.

① 일반적으로 한 우선순위에 상응하는 타이머가 길면 길

수록, 그 우선순위의 전송지연은 감소되는 경향을 보인다.

② 우선순위 6의 전송지연을 감소시키기 위해서는 THT를 증가시키는 것보다 TRT들을 감소시키는 것이 더욱 효과적인 경우가 많았다.

③ 모든 TRT들이 토큰순환시간인 TCT(Token Circulation Time, 토큰이 논리적 링을 한 번 순회하는데 걸리는 시간)와 비교하여 너무 길면, 큐에서의 대기시간이 우선순위와 무관하게 서로 비슷해지는 경향이 있다.

④ 어떤 우선순위의 전송지연이 감소되면 그 영향으로 다른 우선순위의 전송지연이 증가할 수도 있다.

⑤ 어떤 TRT가 TCT와 비교하여 충분히 길지 않으면, 그 우선순위의 전송이 제대로 이루어지지 않아, 전송지연이 증가하게 된다.

위와 같은 특성에 근거하여 여러 종류의 성능관리기가 개발되었는데 이 절에서는 TCT-based Fuzzy Network Performance Manager (FNPM), Queue/Timer-adjusting FNPM, 그리고 Neuro-Fuzzy Performance Manager (NFPM)의 구성을 소개한다.

3.1 TCT-based Fuzzy Network Performance Manager (FNPM)

TCT-based FNPM은 하위 우선순위들이 다소 희생되더라도 최상위 우선순위인 우선순위 6의 지연을 가급적 빠른 시간내에 사용자가 선택한 값 이내로 감소시키는 것을 목표로 하였다. 우선순위 6의 지연이 수용범위 안에 들면, 다음 우선순위인 우선순위 4가 하위 우선순위인 우선순위 2와 0에 지나친 지연이 발생하지 않도록 고려하면서, 빠른 시간내에 허용지연의 범위에 근접하도록 하였다. 이후 우선순위 2에 대하여 같은 방법으로 전송지연을 감소시키며, 우선순위 0은 상위 우선순위가 만족할 만한 전송지연을 갖는 상태에서 자신의 지연을 최소로 줄이는 방법을 취하였다.

이를 위한 TCT-based FNPM에서의 퍼지 입력 변수로서는 네트워크의 성능에 직접적인 영향을 미치는 네 가지 우선순위의 평균 전송지연시간과 평균 토큰순환시간(Token Circulation Time, TCT)을 택하였다. 네 우선순위의 평균 전송지연을 위하여 Small, Medium, Big의 세 퍼지 집합이 정의되었고 이 중 Small에 해당하는 영역이 네트워크 사용자가 허용할 수 있는 최대 평균 전송지연과 대응되도록 소속함수가 조정되어 있다. 토큰순환시간은 토큰이 논리적 링을 완전히 한 바퀴 순환하는데 걸리는 시간으로 여기에는 토큰 송신시간, 각 스테이션에서의 처리시간, 그리고 메시지 패킷의 송신시간이 포함되어 있다. 이러한 TCT는 네트워크 부하와 밀접한 관계에 있으므로 TCT와 비교하여 TRT들을 모두 증가시키거나 감소시키기 위하여 FNPM의 입력에 포함되었다. TCT를 위하여 역시 Small, Medium, Big의 세 종류의 퍼지 집합이 타이머 값에 대하여 상대적으로 정의되

었다. 출력변수로는 네 타이머의 변화량을 선정하였다. 각 타이머마다 다섯 종류의 퍼지 집합, 즉, NB(Negative Big), NS(Negative Small), ZR(ZeRo), PS(Positive Small), PB(Positive Big)를 사용하였다.

이러한 5개의 입력변수와 4개의 출력변수를 사용하여, Table 1과 같이 12개의 퍼지법칙으로서 제어방침을 나타내었다. 여기서, 표의 좌측 부분은 조건부의 퍼지 입력변수를, 우측 부분은 결론부의 출력변수를 나타내며 D_i ($i=6,4,2,0$)는 우선순위 i 의 평균전송지연을 의미한다. 첫 번째 법칙은 TCT가 매우 짧고 TRT들의 타이머 값이 지나치게 큰 경우로서 우선순위간의 차이가 모호해지기 때문에 모든 TRT들을 감소시키기 위한 것이다. 그리고 두 번째 법칙은 3가지의 하위 우선순위의 전송 기회를 증가시키기 위한 것으로, 우선순위 6의 평균 전송지연이 Small이고, TCT가 TRT들과 비교하여 너무 긴 경우로써 하위 우선순위의 전송기회가 그다지 많지 않을 때 TRT들을 증가시키기 위한 것이다. 그리고, 세 번째 법칙은 두 번째 법칙과 비슷한 상황에서 적용되지만, 우선순위 6의 전송지연이 Medium에 해당되는 경우이다. 나머지 법칙들은 TCT가 Medium인 경우로서 우선순위 순으로 사용자의 요구전송지연 시간을 만족시키기 위하여 타이머들이 변화하도록 하였다.

표 1. Fuzzy Rules of TCT-based FNPM.

	TCT	D6	D4	D2	D0	Δ THT	Δ TRT4	Δ TRT2	Δ TRT0
1	S					ZR	NB	NB	NB
2	L	S				ZR	PB	PB	PB
3	L	M				PB	ZR	ZR	ZR
4	M	B				PB	NB	NB	NB
5	M	M				PS	NS	NS	NS
6	M	S	B			ZR	PB	NS	NS
7	M	S	M			ZR	PS	ZR	ZR
8	M	S	S	B		ZR	ZR	PB	NS
9	M	S	S	M		ZR	ZR	PS	ZR
10	M	S	S	S	B	ZR	ZR	ZR	PB
11	M	S	S	S	M	ZR	ZR	ZR	PS
12	M	S	S	S	S	ZR	ZR	ZR	ZR

3.2 Queue/Timer-adjusting Fuzzy Network Performance Manager (QTFNPM)

QTFNPM은 우선순위 6의 전송지연을 감소시키기 위해서 타이머의 조절뿐만 아니라 하위 우선순위의 전송 대기 큐의 용량을 동시에 조절한다. 이 같은 의도는 보다 광범위한 시뮬레이션 시나리오를 통하여 관찰된 다음과 같은 특성

에 기초하고 있다.

① 어떤 TRT가 최소 토큰순환시간(토큰만을 순환하는데 걸리는 시간, 즉 토큰 전송 및 처리 시간×스테이션의 수) 이하이면, 해당 메시지의 전송이 완전히 제한되며, 이로 인해 다른 우선순위가 이득을 본다.

② 큐용량이 충분히 설정되어 있을 때, TRT가 최소 토큰순환시간보다 조금이라도 길게 선택되면 생성된 거의 모든 메시지가 전송되므로, TRT의 조정만으로는 해당 우선순위의 메시지 전송을 모두 허용하거나 완전히 금지시키는 두 결과만을 얻을 수 있다.

③ TRT가 최소 토큰순환시간에 근접해 있을 때, 해당 큐용량을 감소시키면 일부 메시지들의 전송이 거부되어, 그 우선순위의 트래픽을 감소시키는 결과를 가져온다.

④ TRT0을 최소 토큰순환시간에 가깝도록 낮추어도 우선순위6의 전송지연이 관리자가 설정한 요구전송지연을 만족시키지 않으면, 타이머에 의한 조정보다는 우선순위0의 큐용량을 감소시키는 것이 효과적이다.

이 같은 특성을 반영하여 QTFNPM은 TCT-based FNPM의 룰 베이스로 확장한 것이다. 즉, QTFNPM은 큐용량과 타이머의 변화량을 출력으로 하여 33개의 규칙으로 구성되며, 입력 변수로는 토큰순환시간(TCT)과 각 우선순위의 평균 전송지연시간과 세 배의 표준편차의 합(Di), 우선순위0의 relative throughput(생성된 메시지의 비트 수에 대한 전송된 비트 수의 비, P0)이 퍼지입력의 형태로 받아들여지고, TRT0의 값이 최소 토큰순환시간과 상대적으로 항상 비교되도록 구성되어 있다. 또한, QTFNPM의 출력변수는 각 우선순위의 통신 가능 시간을 결정짓는 각 우선순위의 타이머 변화량과 우선순위0에 대한 큐용량의 변화량이다. 이와 같이 구성된 33개의 퍼지 규칙의 일부는 TCT-based FNPM의 법칙이 약간 확장된 형태를 갖고 있으며, 나머지 법칙들은 TRT0의 값이 최소 토큰순환시간에 근접했을 때에만 작동하도록 고안되었다.

3.3 Neuro-Fuzzy Performance Manager (NFPM)

이미 언급된 두 종류의 성능관리기가 목적인 기능을 수행하기 위해서는 퍼지 법칙에 사용된 퍼지 집합의 소속 함수를 적절히 선택하여야 한다. 이러한 소속 함수의 선정에 대하여 체계적인 절차가 확립되어 있지 않기 때문에 흔히 시행착오적인 방법이 사용되고 이에 따른 상당한 시간과 비용이 소모되는 실정이다. 이 같은 단점을 극복하기 위하여 NFPM이 개발되었는데 성능관리를 위한 퍼지 법칙들을 신경망의 구조로 표현하고 신경망의 학습능력을 이용하여 소속함수를 조정할 수 있는 기능을 갖고 있다 [17].

NFPM은 그림 1과 같이 입력층을 포함하여 모두 다섯 층으로 이루어진다. 이는 TCT-based FNPM의 퍼지규칙을 병렬 분산 구조로 나타낸 것이다. NFPM에서 각 뉴런 사이

의 가중치는 0 또는 1로 고정되는 것으로 가정하였으며, 그림 1에서 가중치가 1인 경우에는 실선으로 표현하였고 0인 경우는 연결을 나타내지 않았다. 따라서, 학습을 통해 조절되는 파라미터는 가중치가 아니라, 퍼지화층과 비퍼지화층의 각 뉴런이 갖는 소속 함수의 중심점과 폭이다. 각 층별 뉴런의 기능을 살펴보면, 신경망의 뉴런이 갖는 비선형 함수 외에 AND, OR와 같은 논리 연산자가 더 추가됨을 알 수 있다. 다음에서 각 층의 기능을 구체적으로 기술한다.

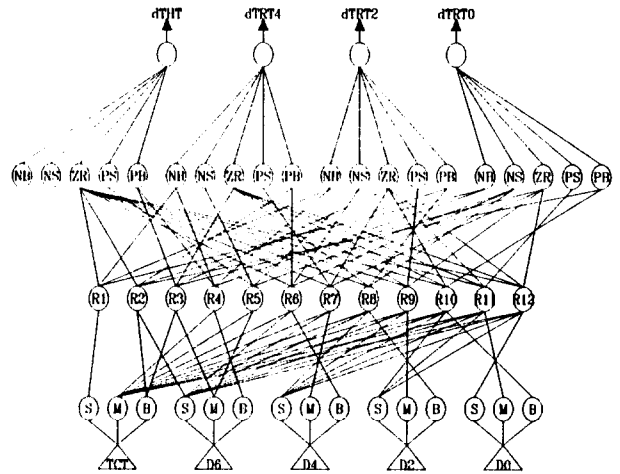


그림 1. Architecture of NFPM.

Layer 1 : 입력층

입력 값들을 Layer 2로 전달하는 역할을 수행하며, u_i 는 입력, O_i 는 노드의 출력을 나타내며, 아래 첨자는 입력층에서의 노드 번호를 나타내고, 윗첨자는 층을 나타낸다.

$$O_i^1 = u_i \quad (1)$$

NFPM에서 입력은 평균 토큰 순환 시간과 각 우선순위에 대한 평균 전송지연에 3배의 표준편차를 더한 값이다(D6, D4, D2, D0). 이러한 방법을 취한 것은 대부분의 메시지가 전송지연 요구조건을 만족시킬 수 있도록 하기 위한 것이다.

Layer 2 : 퍼지화층

퍼지화층은 각 입력에 대해, 퍼지 언어 변수의 소속값을 계산하며, 종 모양의 소속 함수를 이용하여 퍼지화를 수행한다. 여기서, u_i 는 입력, c_i 는 소속 함수의 중심점, s_i 는 종 모양의 함수가 옆으로 퍼진 정도를 나타낸다.

$$O_i^2 = e^{-\left(\frac{u_i - c_i}{s_i}\right)^2} \quad (2)$$

Layer 3 : 퍼지 AND 연산층

퍼지 규칙에서 조건부의 matching정도 즉, 규칙의 firing strength를 계산하는 층으로 퍼지 AND 연산을 수행하며 p 는 layer 3의 i 번째 뉴런에 연결된 입력의 수이다.

$$O_i^2 = \min(u_{i1}^3, u_{i2}^3, \dots, u_{ip}^3) \quad (3)$$

위의 식에서 보이는 것과 같이 AND 연산자로 min함수를 선택하였다. 이는 Mamdani에 의해 제안된 추론법을 따르는 퍼지 제어기에서 일반적으로 사용되는 것으로 기존의 퍼지 시스템을 그대로 뉴로-퍼지로 변환할 수 있도록 하기 위해 사용하였다.

Layer 4 : 퍼지 OR 연산층

fire된 여러 가지 규칙들이 결론부의 동일한 언어 변수에 대해 서로 다른 추론 결과를 제시할 경우 이를 통합하는 역할로서 퍼지 OR연산을 수행하게 되는데, m은 layer 4의 i번째 뉴론에 연결된 입력의 수이다.

$$O_i^4 = \max(u_{i1}^4, u_{i2}^4, \dots, u_{im}^4) \quad (4)$$

Layer 5 : 출력층

출력층은 소속 정도에 따라 가중평균을 취하여, 비퍼지화된 최종 출력을 구하는 층으로 최종 출력은 각 우선순위에 대한 타이머 값의 변화이다. 타이머의 변화량을 계산하기 위하여 면적중심법과 유사한 방법을 사용하였다. 출력층의 소속함수를 밀변의 폭이 s, 높이가 u_i^5 , 밀변의 중심이 c_j 에 놓인 이등변 삼각형으로 제한하고, 여러 입력에 의해 중첩되는 부분을 중복해서 계산하여 비퍼지화를 수행하였다. 이를 식으로 나타내면 다음과 같고 n은 layer 5의 i번째 노드에 연결된 입력의 수이다.

$$O_i^5 = \frac{\frac{1}{2} \sum_{j=1}^n c_j (s_j u_i^5)}{\frac{1}{2} \sum_{j=1}^n s_j u_i^5} \quad (5)$$

NFPM은 gradient method에 의해 출력층의 소속 함수의 중심과 폭을 조절하는 데, 성능 지표로 사용되는 에러 함수는 식(6)과 같이 정의하였다. 이러한 에러값은 IEEE 802.4 네트워크에서 전송된 각의 지연에 근거하여 계산된다.

$$E(\delta_j^i) = \frac{1}{m_a} \sum_{i=6, 4, 2, 0, 7}^{m_i} F_i(\delta_j^i) \quad (6)$$

여기서, m_a 는 주어진 관찰 주기에서 관찰된 의 수, i 는 우선순위, m_i 는 우선순위 i 의 메세지 수, δ_j^i 는 우선순위 i 의 j 번째 메세지의 전송지연, $F_i(\cdot)$ 는 우선순위 i 의 벌칙 함수로서 식(7)과 같이 정의된다.

$$F_i(\delta_j^i) = \begin{cases} 0 & \text{if } \delta_j^i \leq \theta_i \\ (\delta_j^i - \theta_i)^2 & \text{if } \theta_i < \delta_j^i \leq \theta_i + b_i \\ b_i^2 & \text{if } \delta_j^i > \theta_i + b_i \end{cases} \quad (7)$$

여기서, θ_i, b_i 는 각각 우선순위가 i 인 메세지의 penalty threshold와 penalty band이다. 즉, 한 메세지가 penalty threshold 이하의 전송지연을 가지면, 그 메세지에 대한 penalty는 없고, penalty threshold를 넘는 양이 penalty band보다 작을 때까지는 그 초과량의 자승에 해당하는 penalty를 받는다. 초과량이 penalty band보다 커지면 일정하게 penalty band의 자승의 값을 penalty로 부과받는다. 이러한 penalty의 평균이 바로 $E(\delta_j^i)$ 이다. 이러한 벌칙함수는 메세지의 전송지연이 허용할 수 있는 값을 약간 초과했을 때 그 초과량의 자승에 비례한 벌칙을 부과하고 그 초과량이 어느 한도를 넘으면 초과한 정도에 상관없이 일정한 상수의 벌칙을 부과하기 위한 것이다.

각 파라미터에 대한 에러 함수의 구배는 연쇄 법칙(chain rule)을 이용하여 구할 수 있다. NFPM의 출력(타이머 변화량)이 에러 함수에 미치는 영향은 각 타이머에 약간의 섭동(perturbation)을 주어 차분값으로 얻는다. 이는 에러 함수를 해석적인 방법에 의해 구해 낼 수 없기 때문이다. 따라서, 식(8), (9)에서 이용되는 $\partial E / \partial O_i^5$ 는 차분식으로 대체하였다.

$$\frac{\partial E}{\partial c_j} = \frac{\partial E}{\partial O_i^5} \cdot \frac{\partial O_i^5}{\partial c_j} = \frac{\partial E}{\partial O_i^5} \cdot \frac{s_j u_i^5}{\sum s_j u_i^5} \quad (8)$$

$$\frac{\partial E}{\partial s_j} = \frac{\partial E}{\partial O_i^5} \cdot \frac{\partial O_i^5}{\partial s_j} = \frac{\partial E}{\partial O_i^5} \cdot \frac{c_j u_i^5 (\sum s_j u_i^5) - (\sum c_j s_j u_i^5) u_i^5}{(\sum s_j u_i^5)^2} \quad (9)$$

따라서, 퍼지 규칙 결론부의 중심과 폭은 식(10)에 따라 파라미터 학습을 수행한다.

$$\begin{cases} c_i(k+1) = c_i(k) - \eta \frac{\Delta E}{\Delta O_i^5} \frac{s_i u_i}{\sum s_j u_i} \\ s_i(k+1) = s_i(k) - \eta \frac{\Delta E}{\Delta O_i^5} \frac{c_i u_i (\sum s_j u_i^5) - \sum c_j s_j u_i^5 (u_i^5)}{(\sum s_j u_i^5)^2} \end{cases} \quad (10)$$

4. 실험 결과와 고찰

SIMAN에 의해 구축된 토큰버스 시뮬레이션 모델은 Table 2에 보인 시뮬레이션 시나리오를 사용하였다. 이것은 메세지 생성간격과 메세지 길이 등을 지정된 확률분포에 따라 생성시키며, 또한 우선순위별로 할당된 타이머 길이에 따라 전송을 수행한 후, 전송된 메세지의 양과 소요된 전송지연, 성능지수 등의 결과를 산출한다.

이러한 결과를 네트워크 성능관리의 입력으로 사용하여 네트워크의 상황을 고려한 새로운 타이머를 얻을 수 있으며, 이것이 다시 토큰버스 통신망 시뮬레이션 모델의 입력이 된다. 시변성을 갖는 네트워크의 상황을 고려하기 위하

표 2. Simulation Parameters.

		THT	TRT4	TRT2	TRT0
No. of Station	TCT-based FNPM	10			
	QTFNPM				
	NFPM				
No. of Queue per Station	TCT-based FNPM	4			
	QTFNPM				
	NFPM				
Type of Probability Distribution for Message Generation Interval	TCT-based FNPM	expon.	expon.	expon.	expon.
	QTFNPM	uniform	uniform		
	NFPM				
Average Message Generation Interval(μ sec)	TCT-based FNPM	5×10^4	5×10^4	10^5	2×10^5
	QTFNPM	10^4	5×10^4		5×10^6
	NFPM				
Type of Probability Distribution for Message Length	TCT-based FNPM	expon.	expon.	expon.	expon.
	QTFNPM	uniform	uniform		
	LFPM				
Average Message Length(bit)	TCT-based FNPM	5×10^3	10^4	2×10^4	6×10^4
	QTFNPM	2×10^3			10^6
	NFPM				

여 매번의 시뮬레이션마다 서로 다른 랜덤 씨드(random seeds)를 이용하였으며, 시뮬레이션에서 매회 통신망을 관측하는 시간은 20초이다.

4.1 TCT-based FNPM의 성능 평가

TCT-based FNPM은 하위 우선순위들의 희생이 있더라도 최상위 우선순위인 우선순위 6의 지연을 가급적 빠른 시간내에 관리자의 요구전송지연 이내로 감소시키는 규칙베이스를 가지는 것으로, 우선순위 6의 지연이 요구전송지연의 범위안에 근접하면 상위 우선순위의 전송지연을 관리하면서 우선순위를 조정하여 전송지연을 감소시키도록 구성되었다.

그림 2는 TCT-based FNPM의 동적인 조정에 의해 각 우선순위들의 전송지연 변화추이를 실선으로 표시된 요구전송지연과 함께 보이고 있다. 대체로, 우선순위 6의 전송지연은 시뮬레이션 초기에 미미하게 감소하는 경향을 보인 후, 요구전송지연에 가깝기 때문에 별 변동이 없고, 우선순위 4의 전송지연이 요구전송지연에 근접하는 경향을 볼 수 있다. 한편, 우선순위 2와 0의 전송지연은 별 다른 변화를 보이지 않는다. 이러한 추이를 보이는 이유는 전체 통신부하가 용량의 80%를 차지함으로써 각 우선순위의 요구전송지연을 만족시키기 어렵기 때문이다. 우선순위 6의 전송지연을 감소시키기 위하여 THT가 최대허용값까지 증가되지만, 전체적인 통신부하를 제한하지 않고는 별 다른 효과를 얻지 못하는 결과를 보이고 있다.

이상의 결과를 살펴보면 TCT-based FNPM은 상위 우선

순위로 급속히 전송지연을 감소시켜 우선순위 6, 4에 대해서는 관리자의 요구전송지연에 어느 정도 근접시켰으나, 통신부하에 비해 무리한 요구전송지연 때문에 우선순위 2, 0까지 요구전송지연에 근접시키는 것은 어려웠으며, 뚜렷한 감소도 볼 수 없었다.

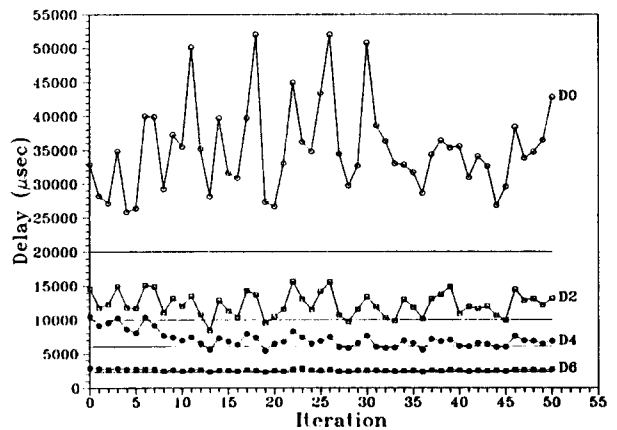


그림 2. Average Data Latency from the Simulation Experiment with the TCT-based FNPM.

4.2 QTFNPM의 성능 평가

QTFNPM(Queue/Timer-adjusting FNPM)은 TRT0을 최소 토큰순환시간에 가깝도록 낮추어도 우선순위 6의 전송지연이 관리자가 설정한 요구전송지연을 만족시키지 않으면, 타이머에 의한 조정보다는 우선순위 0의 큐용량을 조정

하는 것이 효과적이라는 실험을 근거로 하여 TCT-based FNPM의 규칙 베이스를 확장한 것이다.

그림 3은 QTFNPM에 의해 얻어진 평균전송지연과 세배의 표준편차의 합으로 모든 우선순위가 관리자의 요구전송지연을 만족하고 있음을 알 수 있다. 초기에는 TCT-based FNPM과 마찬가지로 우선순위 6의 전송지연을 관리자의 요구전송지연에 만족시키기 위하여 하위 우선순위의 타이머를 감소시키지만, TRT0이 최소 토큰순환시간에 근접하여도 (iteration 10 부근) 요구전송지연을 만족시키지 못하는 것을 보이고 있다. 이때부터 QTFNPM은 큐용량에 관한 물을 사용하여 우선순위 0의 큐용량을 조정한다. 이것은 큐용량의 부족으로 우선순위 0의 일부 메세지들을 삭제시킴으로써 전체 트래픽을 감소시키고, 상위 우선순위의 전송지연을 향상시킨다. 따라서, 우선순위 0의 전송을 일부만 허용하면서, 우선순위 6, 4, 2는 모든 메세지를 요구전송지연에 만족시켜 전송할 수 있었으며, 우선순위 0도 네트워크의 용량이 허용하는 한도내에서 메세지 전송이 가능하였다. 이와 같이 QTFNPM을 사용하였을 때 우선순위 0의 전송을 전면적으로 허용하거나 불허하는 극단적인 조치를 피하여 약 60%의 우선순위 0 메세지를 계속해서 전송하면서 상위우선순위의 요구전송지연을 만족시키는 효과적인 결과를 얻을 수 있었다.

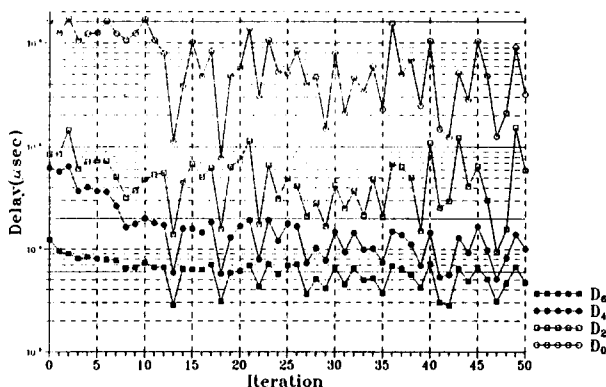


그림 3. Data Latencies from Simulation Experiments with QTFNPM.

4.3 NFPM의 성능 평가

NFPM의 유효성을 입증하기 위하여 NFPM에 학습을 수행하지 않는 경우와 온-라인 학습을 수행하는 경우, 그리고 학습이 완료된 경우에 대한 성능지표를 비교하였다. 본 실험에서 학습률은 1로 고정하였는데, 이는 비교적 적은 반복 횟수를 통해서도 확연한 학습 정도를 보이기 위한 것이다.

그림 4는 성능지표의 누적 평균을 나타내고 있는데, 이는 랜덤 씨드에 따른 네트워크의 특성이 큰 편차를 발생시켜, 시각적인 비교에 어려움을 주기 때문에 취해진 것이다. 그림에서 알 수 있듯이, 학습하기 전의 NFPM은 네트워크의

성능을 향상시키기는 하지만, 그 속도가 느리고, 성능의 향상에도 한계를 안고 있다. 그러나, NFPM의 온-라인 학습이 진행됨에 따라, 성능이 크게 향상됨을 볼 수 있으며, 그 속도 또한 기존의 것보다 향상되었음을 알 수 있다. 학습이 완료된 파라미터로 NFPM이 성능관리를 수행할 경우, 속도가 가장 빠름을 알 수 있다.

그러나, 온-라인 학습이 50번밖에 수행되지 않았기 때문에, 모든 파라미터에 대해 학습이 이루어지지는 못했으며, 학습 조건도 약간 편중된 경우이기 때문에, THT를 증가시키거나, TRT들을 감소시키는데 사용된 소속함수들이 주로 조정되었다.

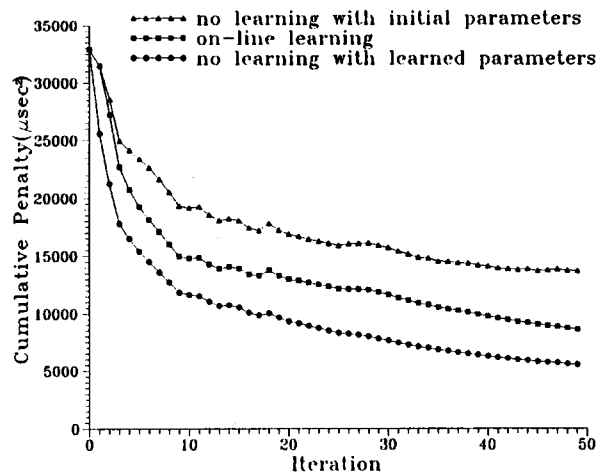


그림 4. Comparison of Performance Using NFPM.

5. 결 언

본 논문에서는 공장자동화용 네트워크에 사용되는 토큰버스 스 프로토콜을 위한 성능관리의 개발과 평가 결과를 소개하였고 다음과 같이 요약될 수 있다.

- 본 논문에서 소개된 세 가지의 성능관리기는 생산자동화를 위한 네트워크에서 처리되는 다양한 종류의 메세지를 각각의 요구조건에 맞도록 처리할 수 있도록 우선순위 타이머를 자동적으로 조절하는 기능을 수행한다.
- TCT-based FNPM은 단지 12개의 퍼지법칙으로 상위 우선순위의 전송지연을 우선적으로 감소시켜서 요구된 전송지연에 접근시키는 기능을 보였다.
- QTFNPM은 상위 우선순위에 대한 요구전송지연이 하위 우선순위의 전송기회를 제한함으로써만 만족될 수 있을 때 하위 우선순위의 큐용량을 조절하여 상위 우선순위의 전송지연을 개선하였다.
- NFPM은 신경망의 학습 능력을 소속 함수 조정에 이용하여 보다 체계적으로 소속함수를 조절하였다.
- 세 가지 성능관리기는 네트워크 전문가의 지식을 바탕

으로 복잡한 수학적 기법 없이 구성되었으며 네트워크 트래픽에 대한 통계적인 정보나 프로토콜의 개정 없이 사용될 수 있다.

참고문헌

[1] W. Stallings, *Local and Metropolitan Area Networks*, Macmillan, 1993.

[2] J.T. O'Rourke, "A Case for Computer Integrated Manufacturing", Keynote Address, NSF Workshop on Computer Networking for Manufacturing Systems, 1987.

[3] 이 석, "컴퓨터 통합생산을 위한 통신망의 성능관리 : 제1부 성능평가", 한국정밀공학회, 제11권, 제4호, pp.126-137.

[4] 이 석, "컴퓨터 통합생산을 위한 통신망의 성능관리 : 제2부 의사결정", 한국정밀공학회, 제11권, 제4호, pp.138-147, 1994.

[5] S. Lee, "Real-Time Performance Management of Multiple-Access Networks for Computer Communications", Ph.D. Thesis, The Pennsylvania State University, 1990.

[6] S. Lee and A. Ray, "Performance Management of Multiple Access Communication Networks", IEEE Jour. on SELECTED AREAS IN COMMUNICATION, Vol.11, No.9, pp.1426-1437, 1993.

[7] S.M. Klerer, "The OSI Management Architecture : An Overview", IEEE Network, Vol.2, No.2, pp.20-29, 1988.

[8] *Manufacturing Automation Protocol(MAP) 3.0 Implementation Release*, MAP/TOP Users Group, 1993.

[9] A. Valenzano, C. Demartini, and L. Ciminiera, *MAP and TOP Communications : Standards and Applications*, Addison-Wesley, 1992.

[10] IEEE Computer Society, "Information Processing Systems-Local Area Networks-Part 4", IEEE Inc., 1990.

[11] L. Kleinrock, *Queueing Systems, Volume I: Theory*, John Wiley, 1975.

[12] A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis*, 2nd ed., McGraw-Hill, 1991.

[13] C.D. Pegden, R. Shannon, and R. Sadowski, *Introduction to Simulation Using SIMAN*, McGraw-Hill, 1990.

[14] C.C. Lee, "Fuzzy Logic in Control Systems : Fuzzy

Logic Controller-Part I, II", IEEE Transactions on Systems, Man, and Cybernetics, Vol.20, No.2, 1990.

[15] M. Sugeno, *Industrial Application of Fuzzy Control*, North-Holland, 1985.

[16] 三矢直城, 田中一男, C言語による實用ファジイブック, ラッセル社, 1989.

[17] J. R. Jang, "ANFIS : Adaptive-Network-Based Fuzzy Inference System", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665-685, May/June 1993.

부 록

토큰버스 프로토콜

토큰 버스 방식은 통제된 매체 접속 제어 방식(controlled

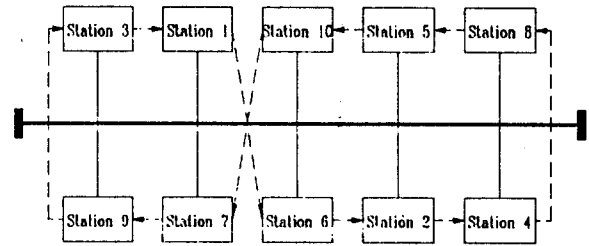


그림 5. Logical Ring of Token Bus Network.

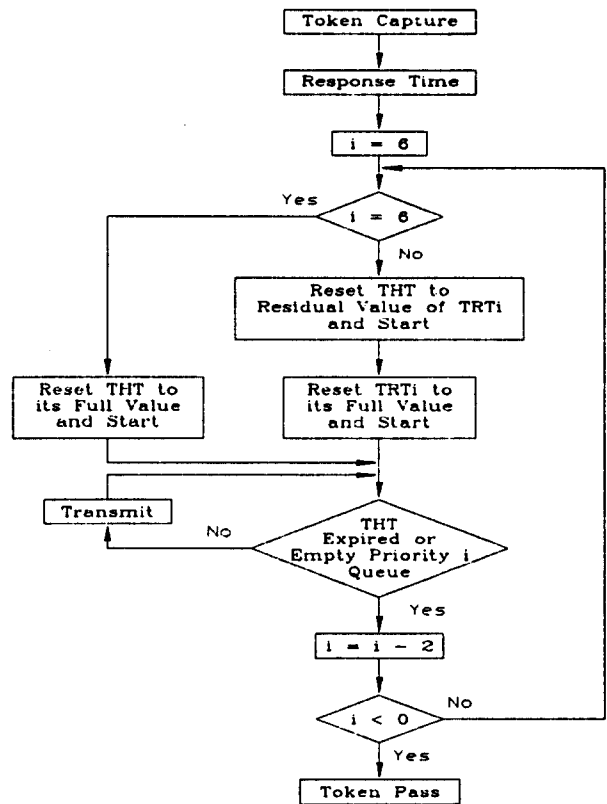


그림 6. Flow Chart of Priority Mechanism.

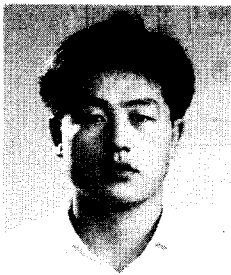
medium access control method)의 한 종류로서, Fig. 5와 같이 버스(bus)나 트리(tree) 구조상에 부착된 스테이션들이 논리적 링(logical ring)을 형성하여 상호 통신하는 방식이다. 즉, 각 스테이션들은 논리적 링을 따라 자신의 전/후(predecessor/successor) 스테이션이 어떤 것인가를 인지하여 순차적으로 전송 기회를 획득한다.

토큰이라고 불리는 제어패킷은 특별한 형식의 비트 패턴(bit pattern)으로서 액세스 권리를 조정하는데, 어떤 스테이션이 토큰을 받으면 규정된 시간동안 매체의 제어권이 주어진다. 이 경우 스테이션은 한 개 이상의 패킷을 전송할 수 있으며, 다른 스테이션을 폴(poll)하거나 응답을 수신할 수도 있다. 또한, 스테이션이 동작을 완료하거나 주어진 시간이 지나면 논리적 링을 따라 다음 스테이션으로 토큰을 전송함으로써, 토큰을 전달받은 새로운 스테이션이 전송할 수 있는 권리를 받게 된다. 이와 같이 각 스테이션은 순차적으로 토큰을 수신하기 때문에 이 방식은 공정한 액세스를 제공해 준다.

IEEE 802.4 토큰버스 프로토콜은 메시지 전송을 위한 네 가지의 다른 우선순위를 가지며, 이러한 우선순위 도구는 네 개의 카운트-다운 타이머, 즉 Token Holding Timer (THT), Token Rotation Timer 4, 2, 0(TRT4, TRT2, TRT0)을 갖는다. 이러한 우선순위 도구의 목적은 한정된 네트워크 용량을 상위 우선순위의 프레임에 할당하고, 하위 우선순위의 프레임은 여분의 용량이 있을 때만 전송하도록 하는 데 있다.

만약, 어떤 스테이션이 토큰을 잡으면 그 스테이션은 Fig. 6과 같은 순서에 따라 각 우선순위들의 데이터를 전송할 수 있다. 즉, 우선순위 6의 메시지는 THT의 시간동안 전송될 수 있는 반면, 우선순위 4, 2, 0과 같은 하위 순위의 메시지는 순차적으로 해당 TRT에서 토큰이 논리적 링을 순환하는데 걸린 시간인 토큰순환시간(Token Circulation Time, TCT)을 삭감한 시간동안 전송될 수 있다. 이와 같은 TRT들의 우선순위를 만족시키기 위해 IEEE 802.4 표준에서는 $TRT4 \geq TRT2 \geq TRT0$ 을 요구하고 있다.

저 자 소 개



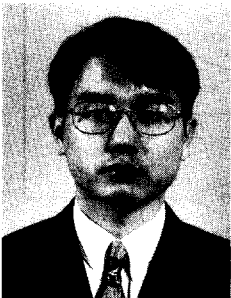
이 상 호

1992년 부산대학교 생산기계공학과 졸업(학사)
 1994년 부산대학교 생산기계공학과 졸업(석사)
 1994년 부산대학교 생산기계공학과(박사과정)
 (609-735) 부산광역시 금정구 장전동 산 30
 TEL. 051) 510-3091 / FAX. 051) 514-3670.



이 석

1984년 서울대학교 기계공학과 졸업(학사)
 1985년 Pennsylvania State Univ. 기계공학과 졸업(석사)
 1990년 Pennsylvania State Univ. 기계공학과 졸업(박사)
 1990년~93 Univ. of Cincinnati 조교수
 1993년~현재 부산대학교 기계공학부 조교수
 (609-735) 부산광역시 금정구 장전동 산 30
 TEL. 051) 510-3091 / FAX. 051) 514-3670.



손 준 우

1994년 부산대학교 생산기계공학과 졸업(학사)
 1996년 부산대학교 생산기계공학과 졸업(석사)
 현재 대우정밀 기술연구소 연구원
 (609-200) 부산광역시 금정우체국 사서함 25호
 TEL. 051) 509-2298 / FAX. 051) 508-3340.