

컴퓨터 통합 생산을 위한 통신망의 성능 관리

제 1 부 - 성능평가

이 석*

Performance Management of Communication Networks for Computer Integrated Manufacturing Part I : Performance Evaluation

Suk Lee

ABSTRACT

Performance management of computer networks is intended to improve a given network performance in order for more efficient information exchange between subsystems of an integrated large-scale system. Importance of performance management is growing as many functions of the large-scale system depend on the quality of communication services provided by the network. The role of performance management is to manipulate the adjustable protocol parameters on line so that the network can adapt itself to a dynamic environment. This can be divided into two subtasks : performance evaluation to find how changes in protocol parameters affect the network performance and decision making to determine the magnitude and direction of parameter adjustment. This paper is the first part of the two papers focusing on conceptual design, development, and evaluation of performance management for token bus networks. This paper specifically deals with the task of performance evaluation which utilizes the principle of perturbation analysis of discrete event dynamic systems. The developed algorithm can estimate the network performance under a perturbed protocol parameter setting from observations of the network operations under a nominal parameter setting.

Key Words : Performance Management, Communication Network, Computer Integrated Manufacturing, Token Bus Protocol, Perturbation Analysis.

1. 서 론

컴퓨터 네트워킹은 공간적으로 분산되어 있는 여러 기구들을 공통된 전송 매체로 연결하여 상호간 정보의 교환을 가능하게 한다. 또한, 컴퓨터 네트워킹을 이용한 시스템은 점진적으로 구성이 가능하며, 장비를 보다 유용하게 사용할 수 있으며, 시스템의 전체적인 신뢰도를 향상시킬 수

있다. 이와 같은 장점들은 컴퓨터 네트워크가 여러 다양한 기구들을 접속시킬 수 있고, 잠재적으로 여러 회사의 기구들을 혼합하여 시스템을 구축할 수 있으며, 시스템의 구성 요소들인 여러 기구들의 설치 위치에 큰 제약이 없으므로 시스템의 초기 설계나 재구성에 있어서 큰 유연성을 갖는 근본적인 사실에 연유한다⁽¹⁾.

이와 같은 장점으로 말미암아 컴퓨터 네트워킹은 많은

* 부산대학교 생산기계공학과, 기계기술연구소

분야에서 시스템의 구성요소나 부시스템을 연결하여 상호 간 정보를 교환하게 함으로써 시스템의 유기적인 통합을 가능하게 하였다. 이러한 통합 시스템은 보다 지능적이고 빠른 응답성을 가지며 외부 변화에 유연하게 대응할 수 있는 장점이 있는데 그 대표적인 예로서 컴퓨터 통합 생산 (computer integrated manufacturing, CIM)을 들 수 있다. 컴퓨터 통합 생산은 한 기업체 안에서 수행되는 여러 기업 활동들을 하나의 폐쇄 루프이고 대화형의 제어 시스템으로 통합하려는 움직임으로 컴퓨터 네트워킹은 기업의 여러 부서 간의 긴밀한 정보 교환에 있어서 없어서는 안 될 중요한 역할을 맡고 있다⁽²⁾. 이러한 컴퓨터 통합 생산을 위하여 기업의 관리, 연구 개발 그리고 생산 부서에서 생산되고 종합, 정리되는 정보들이 컴퓨터 네트워킹을 통하여 각각의 부서 내에서 그리고 여러 부서 간 원활히 교환되어야 한다. 교환되는 정보는 기업 활동의 모든 국면을 망라하는데 몇 가지 예로 매출 보고서, 생산 주문, 신제품 설계도, 공정 총괄, 수치 제어 공작 기계의 프로그램, 여러 생산 장비의 상태 등을 들 수 있다. 이 밖에도 자동차, 항공기, 인공위성, 우주선, 화학 공장, 발전소, 은행, 군사용 통신 및 명령 체계 등에 사용되고 있다.

이렇게 다양한 응용 분야에 사용될 수 있지만 어느 특정한 응용 분야에 사용할 컴퓨터 네트워킹은 그 응용에서 요구되는 조건들을 만족시켜야 한다. 이러한 노력의 일환으로 네트워킹의 토폴로지(topology)와 통신 규약(protocol)을 선정하고 통신 규약의 변수에 적절한 값을 부여하여 시스템을 설계하여야 한다. 하지만, 이러한 설계는 한정된 범위 안의 경우들을 토대로 이루어졌고 네트워킹이 실제로 사용될 때에는 예측하기 어렵고 항상 시간에 따라 변화하는 상황에 처하는 것이 피할 수 없는 사실이다. 예를 들자면 발전소나 화학 공장의 제어에 사용되는 네트워킹은 여러 곳에 산재한 온도, 압력, 유속 센서들로부터 측정된 값들을 제어실의 운영 기사에게 전달하고 그의 결정에 따라 여러 밸브나 펌프 등을 조절하기 위한 명령을 전달해야 하는데 이러한 네트워킹의 구성 요소가 고장을 일으키는 경우에 신속하게 고장을 인식하고 원인을 파악하며 고장의 영향이 네트워킹의 다른 부분까지 전파되는 것을 막아야 한다. 그러므로, 이러한 네트워킹들은 항상 변화하는 주변 여건에 대처해야 할 필요가 있고 이러한 적응 능력은 전자 장비의 발달과 광통신의 개발로 인하여 하나의 네트워킹이 대규모의 시스템 전체를 통합시키는 추세에 따라 더욱 요

구되는 형편이다.

네트워킹의 신뢰성 향상과 유연한 재구성, 그리고 효율적인 운영을 목표로 하는 이러한 적응 능력은 네트워킹 관리의 영역에 속하는 문제이다. 이 영역은 비교적 새로운 분야로서 한 네트워킹이 연결해야 할 기구의 수와 그 다양성이 증가함에 따라 그 중요성이 인식되고 있다^(3, 4). 주요한 네트워킹 관리 기능 중에 성능 관리가 포함되는데 이는 통신 규약의 변수들을 조정하여 네트워킹의 성능을 향상시키는 것을 목표로 한다. 앞에서 예로 든 대규모의 통합 시스템에서는 네트워킹 통신 부하가 외부 상황에 따라 상당히 변화하고 통신에서 발생하는 지연이 전체 시스템의 성능에 큰 영향을 미치기 때문에 어느 일정한 값으로 고정된 통신 규약의 변수로는 효율적인 정보 교환이 이루어지기 어렵고 이를 극복하기 위한 성능 관리의 중요성은 날로 커지고 있다. 일례로서 항공기의 fly-by-wire에 사용되는 네트워킹은 항공기의 속도, 가속도, 자세 뿐만 아니라 장착된 무기와 포착된 적기에 관한 정보까지도 전달할 수 있는데 이러한 네트워킹은 이 착륙 과정이나 적기와의 교전 중에는 순항할 때보다 상당히 많은 양의 메시지를 빠른 시간 내에 처리하여 항공기의 민첩한 반응과 안정성을 도모하여야 한다.

성능 관리는 크게 성능 평가(performance evaluation)와 의사 결정(decision making)의 두 부분으로 분류할 수 있는데 성능 평가는 네트워킹의 성능과 통신 규약의 변수 사이의 관계를 파악하는 것이고 의사 결정은 성능 평가의 결과를 토대로 새로이 취해야 할 변수의 값을 결정하는 기능이다. 본 논문은 대규모의 시스템을 통합하는데 중요한 요소인 컴퓨터 네트워킹의 성능 관리에 관한 두 논문 중에 그 첫 부분으로서 성능 평가에 대한 연구 결과를 담고 있고 본 논문과 연속되는 두 번째 부분에서는 의사 결정과 전체적인 성능 관리를 구현하는 연구에 대하여 기술하고 있다.

본 논문은 이 서론을 포함하여 여섯 절과 하나의 부록으로 구성되어 있다. 제 2 절은 성능 평가의 문제와 그 해결에 관한 기법에 대하여 기술하고 제 3 절에서는 이 연구에서 고찰한 토큰 버스 통신 규약의 우선 순위 변수 값이 바뀌었을 때 네트워킹 운영이 어떻게 변화하는가를 한 단 순화된 예를 통하여 설명하였다. 단순화되지 않은 실제의 토큰 버스 통신 규약을 위한 알고리즘이 제 4 절에 주어졌고 이를 이용한 성능 평가의 결과와 토론이 제 5 절에 포

함되어 있다. 마지막으로 제 6 절은 본 연구에서 얻은 결론을 요약하고 있다. 그리고, 부록에서는 토론 버스 통신 규약의 우선 순위 도구에 대한 간략한 설명을 실고 있다.

2. 성능 평가와 그 기법

성능 관리의 기능은 통신 규약의 조절 가능한 변수들을 실시간에 변화시킴으로써 네트워크가 동적인 환경에 적절히 대응하여 요구되는 성능을 유지하는 것이다. 이 기능은 서론에서 언급된 것과 같이 성능 평가와 의사 결정으로 구분될 수 있다. 본 논문의 초점인 성능 평가는 통신 규약 변수의 변화가 네트워크 성능에 어떠한 영향을 미치는가를 밝히는 것이 그 목적이다. 최상의 경우는 네트워크 성능이 통신 규약 변수의 함수로 표시될 수 있을 때인데 성능 관리를 위한 모든 지식이 갖추어진 것이다. 그렇지 못할 경우에는 최소한 현재의 값 부근에서 네트워크 성능을 추정하여 보다 적절한 변수값을 결정할 수 있는 근거를 제공하여야 한다.

컴퓨터 네트워크와 같은 이산 사상 동적 시스템(discrete event dynamic system, DEDES)의 성능 평가를 위하여 대기 이론(queueing theory)⁽⁵⁾ 과 같은 수학적 기법을 고려할 수 있는데 이들은 성능 관리를 위하여 몇 가지 어려움을 갖고 있다. 그 첫 째로 Poisson arrival 같은 queueing theory의 가정들을 실제 상황에서 만족시키기 쉽지는 않은 점을 들 수 있다. 또한, 네트워크의 성능 지표가 복잡해지면 수리적인 처리가 곤란한 경우도 있다. 이와 같은 난관을 모두 극복한다하여도 queueing model의 입력인 네트워크 트래픽(network traffic)의 통계적인 특성을 실시간에 관찰을 통하여 추정하는 난해한 문제에 봉착할 수 있다.

한편, 이산 사상 시뮬레이션(discrete event simulation)⁽⁶⁾ 도 성능 평가에 사용할 수 있는 한 방법으로서 엄격한 가정이 필요하지 않고 복잡한 형태의 성능지표도 큰 어려움 없이 다룰 수 있다. 그러나, 비교적 신뢰할 수 있는 성능 평가를 위해서는 여러 가지 다른 랜덤 시드(random seed)를 사용한 독립적인 시뮬레이션을 반복하여야 하기 때문에 계산적인 부담이 커지는 경향이 있다. 더구나, 성능 관리를 위한 성능 평가는 현재의 변수 값 주위의 몇 개의 점에서 성능을 추정하는 것이 필요하므로 계산적인 부담이 더욱 가중되는 문제점이 있다. 또한, 이산 사상 시뮬레이션

도 네트워크 트래픽의 통계적인 특성이 입력으로 필요한데 이를 실시간에 관찰을 통하여 추정하는 것을 피하기 위하여 실제로 네트워크로 전송되는 각각의 메시지의 생성 시간과 길이를 기록하였다가 시뮬레이션 모델의 입력으로 사용할 수도 있을 것이다. 그러나, 이를 위해서는 네트워크에 접속되어 있는 각각의 스테이션(station)이 기록한 정보를 시뮬레이션 모델이 연결된 스테이션에 전송하여야 하는데 이 추가적인 전송으로 네트워크의 성능이 저하될 가능성도 있다.

근래에 개발된 perturbation analysis(PA)⁽⁷⁾ 는 이미 언급된 어려움을 상당히 극복할 수 있어서 주목을 받고 있다. PA는 DEDES에서 발생하는 사상의 연속(event sequence)을 관찰함과 동시에 그 관찰을 바탕으로 DEDES의 변수가 다른 값을 가졌을 때의 성능을 추정하는 기법이다. 일반적인 시뮬레이션에서 n 개의 변수가 성능에 미치는 영향을 구하려 할 때 1회의 수행으로 모든 변수가 기준값(nominal values)을 가졌을 때 얻어지는 시스템의 성능을 추정하고, 추가적인 1회의 수행마다 하나의 변수를 순차적으로 변화시킨 변화된 변수값(perturbed values)으로 n 번을 반복하여 각 변수의 성능에 대한 영향을 추정할 수 있다 (총 $n+1$ 회의 시뮬레이션 수행). 이에 반하여 PA를 시뮬레이션에 적용하여 기준값으로 시뮬레이션을 수행하는 동안 n 개의 변수가 하나씩 변화된 상태에서의 성능을 동시에 추정할 수 있기 때문에 단 1회의 수행으로 목적을 이룰 수 있다. 이와 같은 PA의 수행에 요구되는 계산량이 무시될 수 있다면 약 $n+1$ 대 1의 계산 절감 효과가 얻어질 수 있다. 성능 관리를 위한 성능 평가에 PA를 이용하면 네트워크에서 일어나는 사상들을 직접 관찰함으로써 네트워크의 성능을 여러 변수값에 대하여 추정할 수 있으므로 기존의 기법들이 당면한 많은 난관을 극복할 수 있는 장점이 있다.

PA의 개념은 연속적으로 연결된 기계 사이에 필요한 대기 장소의 크기를 적절히 선정하려는 과정에서 발달하였는데 일반적인 PA의 개념적 구조를 그림 1과 같이 나타낼 수 있다. 그림에 나타난 것과 같이 이산 사상 동적 시스템(DEDES)의 변수들에 x 라는 벡터 값을 사용하여 시스템을 운영하였을 때 스칼라인 시스템의 성능 $g(x, \omega)$ 가 얻어지는데 여기서 ω 는 시스템의 무작위성(randomness)을 대표하고 있다. 이러한 DEDES가 변화된 입력 $x+\Delta x$ 를 받아들였을 때의 성능 $g(x+\Delta x, \omega)$ 를 PA가 시스템의 변수가 x 를 취했을 때의 관찰을 통하여 추정하는 것을 도식적으로

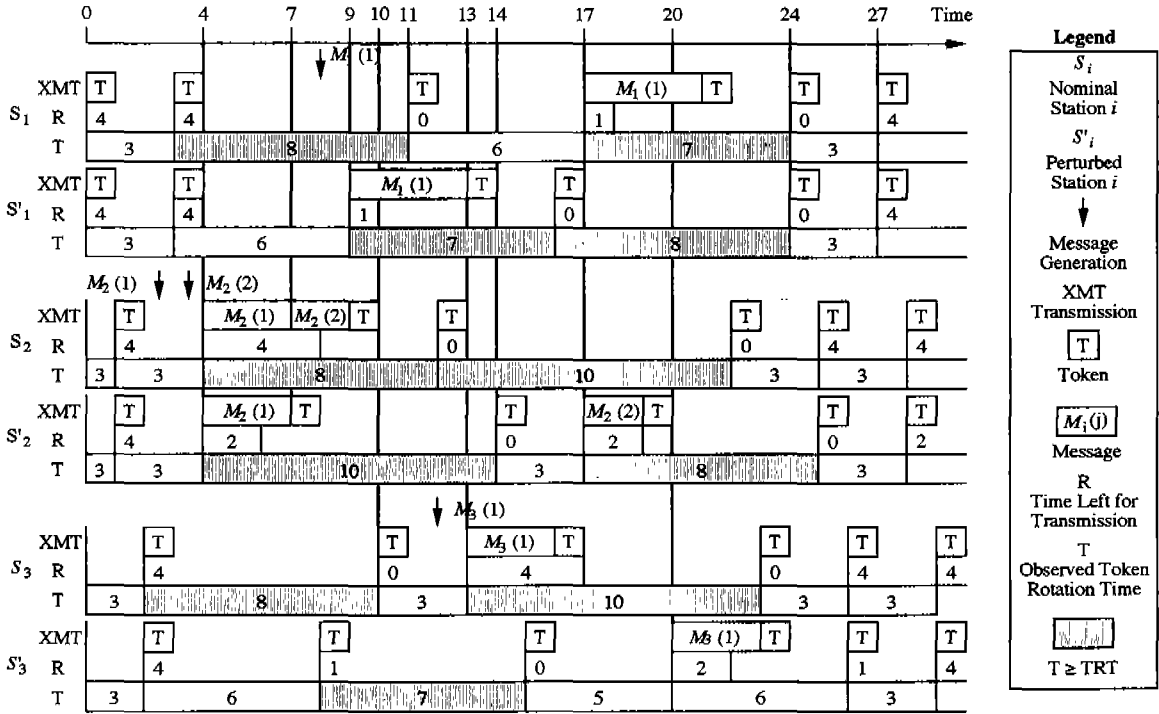


그림 2. TRT의 변화가 네트워크에 미치는 영향

이러한 네트워크가 기준값의 TRT로 운영되었을 때를 살펴보면 (nominal path, S_1, S_2, S_3) 스테이션 1은 전송할 메시지가 없으므로 $t=0$ 에 토큰을 받자마자 스테이션 2로 토큰을 보내기 시작하고 스테이션 2는 $t=1$ 에 토큰을 소유하게 된다. 이와 같은 과정은 $t=4$ 에 스테이션 2가 토큰을 받을 때까지 계속된다. 이 때 스테이션 2는 두 개의 메시지가 기다리고 있고 바로 직전에 토큰을 받은 순간에서부터 이번 토큰을 받을 때까지 세 개의 시간 단위가 ($T=4-1=3$)사용되어서 아직도 넷의 시간 단위가 ($R=TRT-T=7-3=4$)남아 있으므로 스테이션 2의 첫 번째 메시지인 $M_2(1)$ 을 전송하기 시작한다. 한 예로서, 이 메시지를 전송하는데 세 시간 단위가 필요하여 $t=7$ 에 전송이 완료되었다면 이때 아직도 하나의 시간 단위가 남아있고 ($4-3=1$) 두 번째의 메시지가 기다리고 있으므로 연속적으로 $M_2(2)$ 를 전송한다. 이 메시지가 완료된 후 스테이션 3이 $t=10$ 에 토큰을 받고 이를 다시 스테이션 1에게 전송한다. $t=11$ 에 스테이션 1이 토큰을 받았을 때 전송을 기다리는 메시지는 있지만 토큰 순환 시간 T 가 8이어서 TRT보다

크므로 전송하지 못하고 다음 스테이션으로 토큰을 전송한다. 이와 같이 하여 $t=13$ 에 스테이션 3이 $M_3(1)$ 을 전송하기 시작하고 $M_1(1)$ 의 전송은 $t=17$ 에서야 시작되는 것을 볼 수 있다.

일반적인 성능 관리에서 모든 스테이션이 같은 값의 TRT를 갖게 하는데 여기에서는 단지 단순한 예를 들기 위하여 스테이션 2의 TRT만이 두 시간 단위만큼 감소된 경우를 살펴본다(perturbed path, S'_1, S'_2, S'_3). 변화한 TRT의 영향은 $t=7$ 까지 나타나지 않다가 스테이션 2가 $M_2(1)$ 의 전송을 완료했을 때 비로소 나타나기 시작한다. 기준값으로 운영하였을 때는 이 순간에 하나의 시간 단위가 남아 있어서 두 번째 메시지의 전송이 가능했지만 감소된 TRT로 운영하였을 때에는 첫 번째 메시지의 전송 도중에 허용된 두 개의 시간 단위가 소진되어 두 번째 메시지를 전송하지 못하고 토큰을 전송하여야 한다. 이로 인하여 두 번째 메시지를 전송하는데 필요한 두 개의 시간 단위가 절약되었기 때문에 스테이션 3과 1은 순차적으로 토큰을 그만큼 빨리 받게 된다. $t=9$ 에 스테이션 1이 토큰을 받

나타내고 있다.

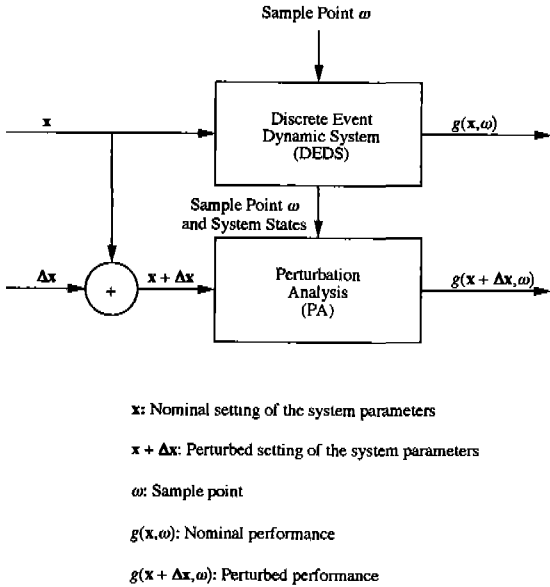


그림 1. Perturbation Analysis의 개념적 구조

DEDS는 그의 상태가 시간축 상에 무작위로 분포된 순간에 일어나는 사상(event)에 의하여 변화하는 시스템을 지칭한다. 이러한 DEDS의 시간에 따른 변천은 발생한 사상의 목록과 그에 대응하는 발생 시간의 목록, 그리고 시스템 상태의 목록으로 요약될 수 있다. 그리고 DEDS의 성능은 이 세 가지의 목록에 근거하여 정의될 수 있다. 이러한 관점에서 볼 때 PA는 결국 DEDS 변수값의 변화가 이 세 가지의 목록에 미치는 영향을 탐구하는 방법이라 할 수 있다. 가장 초기에 개발된 PA 기법은 변수값의 변화로 인한 영향이 발생 시간 목록에만 나타나고 나머지 두 목록에는 영향이 없다고 가정하였다. 이후로 이와 같은 제약 조건을 제거한 여러가지 기법이 고안되었다⁽⁷⁾.

3. 우선 순위 도구의 변수가 네트워크에 미치는 영향

부록에 설명된 바와 같이 IEEE 802.4 토큰 버스(token bus)통신 규약⁽⁸⁾의 우선 순위 도구(priority mechanism)는 네 가지의 우선 순위를 갖고 있다. 가장 우선 순위가 높은 것을 우선 순위 6이라 하고 우선 순위 4, 2, 0이 순차적으로 낮은 순위로 나타낸다. 각 우선 순위에 대한 네트워크

사용 권리는 네 종류의 변수에 의하여 결정된다. 이 변수들을 어떤 값으로 정하느냐에 따라 각 우선 순위에 속하는 메시지들이 겪는 전송 지연(data latency, 전송을 기다리는 시간과 전송하는데 걸리는 시간의 합)이 변화하고 대개의 경우 전송 지연이 네트워크 성능의 주요 요소이므로 이 변수 값을 적절히 변화시킴으로써 성능을 향상시킬 수 있다. 이 절에서는 단순화된 토큰 버스 네트워크에서 한 변수 값이 변화되었을 때 일어나는 현상을 설명하여 제 4 절에서 전개될 성능 평가 알고리즘의 이해를 돕고자 하였다.

여기서 고려되는 예에서는 단 하나의 우선 순위만 존재하는 통신규약을 사용하는 세 개의 스테이션이 네트워크를 구성하고 있다. 모든 메시지는 토큰이 한 바퀴 순환하는데 소요된 시간에 따라 전송 여부가 결정지워진다. 즉, 바로 직전에 토큰을 받은 시간으로부터 다시 토큰을 받을 때까지 경과한 시간인 T가 그의 한도인 TRT(Token Rotation Time)를 초과하면 전송을 기다리고 있는 메시지가 있다하여도 다음 기회로 전송을 미루어야 한다. 이 예는 부록에 설명된 IEEE 802.4에서 우선 순위 6을 삭제하고 나머지 순위들을 하나로 통합한 경우이다.

그림 2에 세 개의 스테이션이 시간이 경과함에 따라 변화하는 상태를 기준값의 TRT를 사용했을 때 (S_i)와 변화된 값을 사용했을 때 (S_i')에 대하여 보여주고 있다. 각 스테이션의 상태는 네 가지의 부분으로 표현되는데 전송할 메시지가 발생한 시간을 나타내는 화살표와 실제의 전송을 표시하는 XMT, 전송을 시작할 수 있는 시간을 나타내는 R, 그리고 토큰이 한 바퀴 순환하는데 소요된 시간(토큰 순환 시간) T로 구성되어 있다. 각각의 메시지는 $M_i(j)$ 로 표시하였는데 이는 스테이션 i 의 j 번째 메시지를 나타낸다. XMT에 나타나는 문자 T가 들어 있는 사각형은 토큰의 전송을 나타내며 나머지는 메시지의 전송을 나타낸다. 그리고, 회색의 사각형은 TRT보다 긴 토큰 순환 시간을 나타낸다. 이 예에서는 토큰의 전송과 그에 대한 반응에 하나의 시간 단위가 소요되고 TRT의 기준값(nominal value)으로 일곱 시간 단위가 사용되었다. 그림 2는 아무 스테이션도 전송할 메시지가 없는 상태에서 스테이션 1이 토큰을 방금 받은 시간인 $t=0$ 로부터 시작한다.

왔을 때 토큰이 여섯 시간 단위만에 돌아와서 ($T=6$) 전송의 시작이 허용되는 시간이 한 단위 만큼 남아있기 때문에 ($R=TRT-T=7-6=1$) 기준값의 경우와 달리 $M_1(1)$ 의 전송이 시작된다. $M_1(1)$ 의 전송이 완료된 후 스테이션 2와 3가 순차적으로 토큰을 받지만 각각의 토큰 순환 시간이 TRT를 초과하여 전송을 못하고 $M_2(2)$ 는 $t=17$ 에, $M_3(1)$ 는 $t=20$ 에 각각 전송되기 시작하는 것을 관찰할 수 있다. 또한, 모든 메시지의 전송이 완료된 후 각 스테이션이 토큰을 받는 순간을 살펴보면 스테이션 1이 $t=24$ 에서 토큰을 받는 순간이 S_1 과 S_1' 모두에서 일치하는 것을 볼 수 있고 이로부터 토큰이 한 바퀴 순환하면 ($t=27$) T와 R도 서로 일치하는 것을 볼 수 있다.

이 예에서와 같이 TRT의 변화로 인하여 사상의 발생 순서와 시간이 영향을 받고 따라서 네트워크의 성능에 주된 인자인 전송 지연도 변화하는 것을 관찰할 수 있다. 그러나 조금 더 자세히 살펴보면 두 경우가 어느 시간까지는 서로 동일하게 진행하다가 차이점이 나타나기 시작하고 또 어느 정도 시간이 지나면 다시 서로 일치하는 것을 알 수 있다. 기준의 진행과 상이한 부분을 추정하기 위한 여러 종류의 PA 알고리즘이 있지만 가장 단순한 방법은 그 부분을 직접 구성하는 것이다. 이와 같은 점에 착안하여 다음 절에 소개될 PA 알고리즘은 두 개의 진행이 서로 달라지기 시작하는 순간을 포착하고 다시 일치할 때까지 변화된 값을 가진 네트워크의 진행을 구성한다.

4. 토큰 버스 우선 순위 변수의 PA 알고리즘

이 절에서는 IEEE 802.4 토큰 버스 통신 규약의 우선 순위 변수가 변화되었을 때 발생하는 네트워크의 진행과 각 메시지의 전송 지연을 기준값의 변수로 운영되는 네트워크의 진행으로부터 구하는 알고리즘을 설명한다. 본 연구에서 사용된 IEEE 802.4 토큰 버스 통신 규약은 공장 자동화 통신 규약(Manufacturing Automation Protocol, MAP)⁽⁹⁾ 등에 널리 사용되고 있다. IEEE 802.4의 네 가지 변수인 THT(Token Holding Time), TRT_i (Token Rotation Time for priority class i , $i=4, 2, 0$) 값들이 전송 지연에 근거한 성능에 밀접한 관계가 있고 이러한 관계가 네트워크의 성능 관리를 위한 핵심적인 요소가 된다.

본 논문에서는 우선 순위 도구의 변수가 기준값을 갖을 때 네트워크의 시간에 따른 진행을 기준 진행(nominal

path), 변화된 값을 갖을 때의 진행을 변화된 진행(perturbed path)이라고 일컫기로 한다. 이 연구에서는 기준 진행과 변화된 진행에서 각각의 스테이션에서 전송을 기다리는 메시지가 대기 장소의 부족으로 소멸되는 경우 없이 차례로 전송된다고 가정하였다. 이와 같은 가정은 대부분의 네트워크의 설계 용량이 실제의 부하보다 여유있게 설정되는 관례에 비추어 타당하다고 간주된다. 개발된 알고리즘은 두 부분으로 이루어져 있는데 첫 부분은 기준 진행과 변화된 진행이 서로 달라지기 시작하는 순간을 검출하는 역할을 하고 (검출 알고리즘) 두 번째 부분은 기준 진행과 다른 변화된 진행을 구성하는 역할을 하고 있다 (구성 알고리즘).

4. 1 검출 알고리즘

앞 절의 예에서 기준 진행에서의 스테이션 2는 $t=4$ 에 토큰을 받아서 두 개의 메시지를 전송하는 반면 변화된 진행에서는 하나의 메시지만을 전송하는데 이것이 두 진행이 서로 달라지기 시작하는 시점이다. 이와 같이 기준 진행과 변화된 진행은 한 우선 순위에 전송할 기회가 주어졌을 때 전송할 수 있는 메시지의 갯수가 다를 때 달라지기 시작한다. 이러한 현상이 일어나기 위한 조건들을 확립하기 위하여 다음과 같은 변수들을 정의하였다.

m : 기준 진행에 있어서 어느 스테이션의 한 우선 순위에 전송 기회가 주어졌을 때 전송된 메시지 수, $m=0, 1, 2, \dots$

q : 기준 진행에 있어서 어느 스테이션의 한 우선 순위에서 m 개의 메시지가 전송된 후에 아직도 대기 행렬에 남아있는 메시지 수, $q=0, 1, 2, \dots$

L_k : 전송된 m 개의 메시지 중 k 번째 메시지를 전송하는데 걸리는 시간, $k=0, 1, 2, \dots, m$

T : 기준 토큰 순환 시간, 즉, 기준 진행에서 어느 스테이션의 한 우선 순위에 부여된 전송 기회 사이의 시간 간격

THT : THT의 기준 길이

ΔTHT : THT의 변화량, $THT + \Delta THT > 0$

TRT_i : TRT_i 의 기준 길이, $i=4, 2, 0$

ΔTRT_i : TRT_i 의 변화량, $TRT_i + \Delta TRT_i > 0$, $i=4, 2, 0$

다음의 테스트들은 어느 주어진 우선 순위에서 최대한으로 전송할 수 있는 m 개의 메시지를 전송한 후 다음 순위나 다음 스테이션에 전송 기회를 주기 직전에 수행되며

주어진 조건이 만족되면 변화된 진행에서는 m 과 다른 수의 메시지가 전송되는 것을 의미한다.

우선 순위 6을 위한 테스트

우선 순위 6에 속하는 메시지는 THT에 의하여 전송 여부가 결정된다.

경우 1 : THT가 증가되었고 이번 기회에 0보다 큰 수의 메시지를 기본 진행에서 전송하였을 때($\Delta THT > 0$ and $m > 0$), 변화된 THT의 시간 내에 기본 진행에서 전송된 메시지의 전송이 완료될 수 있고 대기 행렬에 기다리고 있는 메시지가 존재하면 변화된 진행에서는 m 보다 많은 수의 메시지를 전송하게 된다. 이 조건을 수식으로 표현하면

$$\sum_{k=1}^m L_k < (THT + \Delta THT) \text{ and } q > 0.$$

경우 2 : THT가 감소되었고 이번 기회에 하나 보다 많은 메시지를 기본 진행에서 전송하였을 때 ($\Delta THT < 0$ and $m > 1$), 변화된 THT의 시간 내에 m 번째 메시지를 제외한 $m-1$ 개의 메시지 전송이 완료될 수 없으면 변화된 진행에서는 m 보다 적은 수의 메시지를 전송하는 것을 의미한다. 이 조건은 아래의 수식으로 표현될 수 있다.

$$\sum_{k=1}^{m-1} L_k \geq (THT + \Delta THT)$$

우선순위 4, 2, 0을 위한 테스트

우선순위 $i, i=4, 2, 0$ 에 속하는 메시지는 각각에 적용되는 $TRTi$ 의 길이와 토큰 순환 시간의 상대적인 크기에 의하여 전송 여부가 결정된다.

경우 1 : $TRTi$ 가 증가되었고 기본 진행에 있어서 $TRTi$ 의 길이가 토큰 순환 시간보다 길며 이번 기회에 0보다 큰 수의 메시지를 전송하였을 때 ($\Delta TRTi > 0, TRTi > T$ and $m > 0$), 변화된 $TRTi$ 에서 토큰 순환 시간을 뺀 시간내에 기본 진행에서 전송된 메시지의 전송이 완료될 수 있고 대기 행렬에 기다리고 있는 메시지가 존재하면 변화된 진행에서는 m 보다 많은 수의 메시지를 전송하게 된다. 수식으로 표현하면

$$\sum_{k=1}^m L_k < (TRTi + \Delta TRTi - T) \text{ and } q > 0$$

경우 2 : $TRTi$ 가 감소되었고 기본 진행에 있어서 $TRTi$ 의 길이가 토큰 순환 시간보다 길며 이번 기회에 1

보다 큰 수의 메시지를 전송하였을 때 ($\Delta TRTi < 0, TRTi > T$ and $m > 1$), 변화된 $TRTi$ 에서 토큰 순환 시간을 뺀 시간 내에 m 번째 메시지를 제외한 $m-1$ 개의 메시지 전송이 완료될 수 없으면 변화된 진행에서는 m 보다 적은 수의 메시지를 전송하는 것을 의미한다. 수식으로 표현하면

$$\sum_{k=1}^{m-1} L_k \geq (TRTi + \Delta TRTi - T).$$

경우 3 : 경우 1과 비슷한 경우로서 $TRTi$ 가 증가되었고 기본 진행에 있어서 $TRTi$ 의 길이가 토큰 순환 시간보다 짧아서 이번 기회에 아무런 메시지도 전송하지 못하였을 때 ($\Delta TRTi > 0$ and $TRTi \leq T$), 변화된 $TRTi$ 가 토큰 순환시간보다 길어졌고 대기 행렬에 기다리고 있는 메시지가 존재하면 변화된 진행에서는 0보다 많은 수의 메시지를 전송하게 된다. 이 조건은 아래의 수식으로 표현될 수 있다.

$$(TRTi + \Delta TRTi - T) > 0 \text{ and } q > 0$$

경우 4 : 경우 2와 비슷한 경우로서 $TRTi$ 가 감소되었고 기본 진행에 있어서 $TRTi$ 의 길이가 토큰 순환 시간보다 길며 이번 기회에 1개의 메시지를 전송하였을 때 ($\Delta TRTi < 0, TRTi > T$ and $m = 1$), 변화된 $TRTi$ 가 토큰 순환시간보다 짧아져서 아무런 메시지도 전송하지 못하는 상황을 나타낸다. 즉,

$$(TRTi + \Delta TRTi - T) \leq 0.$$

4. 2 구성 알고리즘

검출 알고리즘의 어느 테스트라도 통과하면 이는 기준 진행과 변화된 진행이 서로 다르기 시작하는 것을 의미한다. 이 때부터 검출 알고리즘은 더 이상 수행되지 않고 구성 알고리즘이 수행되기 시작한다. 구성 알고리즘은 메시지의 길이나 생성 시간 같은 메시지에 관한 정보를 그 메시지가 기준 진행과 변화된 진행에서 모두 전송될 때까지 보관함으로써 변화된 진행의 구성을 위한 자료를 유지한다.

또한, 이 알고리즘은 변화된 진행에서 전송이 시작될 수 있는 시간, 즉, 전송 개시 가능 시간을 산출한다. $R(i)$ 와 $R'(i), i=6, 4, 2, 0$ 가 각각 기준과 변화된 진행에서의 우선 순위 i 메시지의 전송 개시 가능 시간을 나타낸다면 가장 높은 우선 순위를 위한 $R'(6)$ 는

$$R'(6) = THT + \Delta THT$$

나머지 세 우선 순위의 전송 개시 가능 시간은 토큰 순

환 시간의 영향을 받아서 기준 진행에서는 $R(i) = \max(TRT_i - T, 0)$ 로 표현될 수 있으므로 TRT_i 의 변화뿐만 아니라 토큰 순환 시간의 변화도 고려하여야 한다. 이 시간은 지난번의 전송 기회를 받은 순간에서부터 이번의 전송 기회를 받은 순간까지의 시간이므로 이 두 순간의 변화를 고려하여야 전송 개시 가능 시간의 변화, $\Delta R(i)$ 를 구할 수 있다. 변화된 진행에서 이번 전송 기회를 빨리 받을수록 그리고 지난번 전송 기회를 늦게 받을수록 토큰 순환 시간은 짧아지고 따라서 전송 개시 가능 시간이 늘어나게 된다. 이때 두 가지의 경우를 생각할 수 있는데 그 첫 번째 경우로 기준 진행에서 토큰 순환 시간이 TRT_i 보다 짧은 경우이다. 이 때에는 TRT_i 의 변화와 전송 기회를 받은 순간들의 변화의 합이 곧 전송 개시 가능 시간의 변화이다. 그림 3이 이 경우를 나타내는데 ΔC 는 이번에 전송 기회를 받는 순간의 변화를 나타내고 ΔP 는 지난번 전송 기회를 받는 순간의 변화를 나타낸다. 두 번째 경우는 기준 진행에서 토큰 순환 시간이 TRT_i 보다 긴 경우이다. 이 때에는 위의 경우에서 고려된 변화 이외에도 토큰 순환 시간과 TRT_i 의 차이도 고려해야 하는데 이 경우가 그림 4에 나타나 있다. 이 두 가지 경우를 수식으로 요약하면 아래와 같다.

$$\Delta R(i) = \begin{cases} \Delta P - \Delta C + \Delta TRT_i, & \text{if } (TRT_i - T) > 0 \\ \Delta P - \Delta C + \Delta TRT_i + TRT_i - T, & \text{if } (TRT_i - T) \leq 0 \end{cases}$$

$$R'(i) = \max(R(i) + \Delta R(i), 0)$$

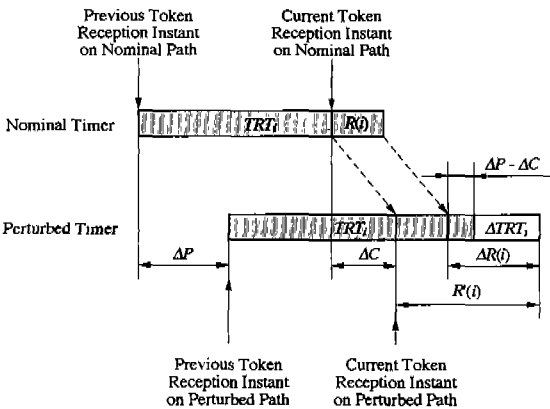


그림 3. 전송 개시 가능 시간의 변화 (Case 1)

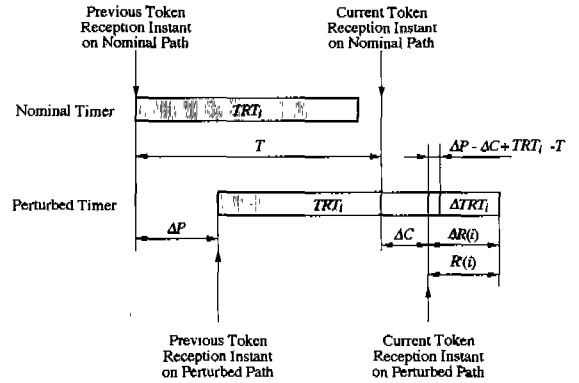


그림 4. 전송 개시 가능 시간의 변화 (Case 2)

이와 같은 방법으로 변화된 전송 개시 가능 시간을 구하면 변화된 진행에서의 메시지 전송은 그 대기 행렬에서 기다리고 있는 메시지가 더 이상 없거나 전송하는데 필요한 시간의 합이 변화된 전송 개시 시간을 넘을 때까지 계속된다. 이러한 과정 중에 변화된 진행에서의 전송 지연을 계산하여 변화된 성능을 구할 수 있다. 기준 진행에서 한 스테이션의 우선 순위가 메시지를 전송하는데 걸린 시간의 합을 $X(i)$ 라 하고 이에 대한 변화된 진행의 변수를 $X'(i)$ 라 하면 그 변화는

$$\Delta X(i) = X'(i) - X(i)$$

로 표현될 수 있고 다음 우선 순위가 전송 기회를 받는 순간은 기준 진행에 비하여

$$\Delta C(i-2) = \Delta C(i) + \Delta X(i)$$

만큼 차이가 나게 된다. 이 때 i 는 6, 4, 2, 0의 값을 순차적으로 순환하며 취하는데 0에서 6으로 변화하는 것은 다음 스테이션으로 토큰을 전송하는 것을 의미한다.

이 같은 과정을 모든 스테이션의 모든 우선 순위에 대하여 $\Delta C=0$ 와 $\Delta P=0$ 이 만족될 때까지 계속하여서 변화된 진행을 구성하고 변화된 진행이 다시 기준 진행과 일치하게 되면 구성 알고리즘의 수행을 중지하고 접수 알고리즘의 수행을 다시 시작한다.

5. 결과와 토의

토큰 버스 통신 규약의 우선 순위 도구를 위한 PA 알고리즘의 효율성과 정확성을 밝히기 위하여 이 알고리즘을 토큰 버스 통신망 시뮬레이션 모델에 설치하였다. 이 시뮬

레이션 모델은 SIMAN⁽¹⁰⁾ 을 사용하여 개발되었는데 전송 속도, 스테이션의 수, 우선 순위 도구에 필요한 변수의 값 (THT, TRT4, TRT2, TRT0) 등의 결정적 변수들과 메시지 생성 기간 (한 대기 행렬에 연속적으로 들어오는 두 메시지의 시간차, message generation period)과 메시지 길이 (한 메시지에 포함된 bit의 수) 같은 확률적 변수들의 분포와 그 특성들을 입력으로 받아들여 시뮬레이션을 수행한다. 이 시뮬레이션 모델에 설치된 PA 알고리즘은 네 개의 다른 변화된 진행을 동시에 처리할 수 있는데 각각의 변화된 진행에서는 하나의 변수만을 변화시키도록 구성되었다.

본 논문에서 사용한 네트워크는 열 개의 스테이션으로 구성되어 있고 전송 속도는 50 megabits per second (Mbps)를 사용하였다. 이 전송 속도는 IEEE 802.4 표준에 사용된 1, 5 또는 10 Mbps와 다르지만 전자 부품의 성능 향상으로 머지않아 가능하리라고 기대된다. 우선 순위 도구를 위한 THT, TRT4, TRT2, TRT0는 각각 150, 1000, 800, 600 microseconds(μ sec)의 값을 취하였다. 메시지 생성 기간과 메시지 길이는 네트워크 부하의 혼란 양상인 bursty traffic을 모방하기 위하여 지수 분포를 사용하였다. 우선 순위 6의 메시지 생성 기간의 평균은 10 milliseconds(msec)이고 메시지 길이의 평균은 5000 bits를 사용하였고 우선 순위 4는 10 msec와 10000 bits, 우선 순위 2는 20 msec와 20000 bits, 우선 순위 0은 40 msec와 40000 bits를 택하였다. 이와 같은 선택은 상위 우선 순위 6과 4는 주로 실시간 제어 신호, 실시간 음성 통신, 그리고 감시체계를 위한 실시간 화상 갱신 같은 전송 지연에 많은 영향을 받는 메시지를 처리하므로 비교적 길이는 짧으나 자주 전송되는 경향이 있고 이와 대조적으로 하위 우선 순위 2와 0은 CAD file 같은 전송 지연과 거의 무관한 메시지 처리에 사용되므로 길이가 길지만 전송빈도가 적은 경향을 모사하려는 의도이다. 이 같은 네트워크의 부하는 네트워크의 최대 용량에 비하면 약 70%에 해당하는데 우선 순위 6이 10%를 차지하고 나머지 세 우선 순위가 각각 20%씩을 차지한다.

이 연구에서는 네트워크의 성능 지표로 전송 지연의 합수를 사용하였는데 전송 지연에 의해 결정되는 벌칙 함수도 포함하였다. 이 벌칙 함수는 메시지가 그것의 우선 순위 에 적용되는 한계값 이내에 전송되면 벌칙이 없고 그 이상의 초과량이 어느 한도를 넘지 않을 때까지 초과량의 제곱

을 벌칙으로 부과한다. 전송 지연의 초과량이 주어진 한도를 넘으면 일정한 양의 벌칙을 적용한다. 이와 같은 벌칙 함수는 메시지가 주어진 시간 한도 이내에 전송이 되면 네트워크를 이용하는 사용자 (분산 제어 시스템이나 제품 설계자)의 기능에 별다른 지장을 주지않고 그 반대로 메시지가 지나치게 오래 지연되면 조금 빠르거나 느리거나에 관계없이 비슷한 정도로 사용자의 기능을 저하시키는 사실을 반영하기 위하여 선택되었다. 이를 수식으로는

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

와 같이 나타낼 수 있는데 여기서 $F_i(\delta_i(j))$ 는 우선 순위 i 에 적용되는 벌칙 함수이고 $\delta_i(j)$ 는 우선 순위 i 에 속하는 j 번째 메시지의 전송 지연을 나타낸다. 또한 우선 순위 i 에 대하여 θ_i 는 벌칙이 주어지지 않는 한계값을, b_i 는 일정한 양의 벌칙이 적용되기 시작하는 초과량을 나타낸다. 본 연구에서는 θ_i 를 우선 순위 6에서 1000, 우선 순위 4에서 1500, 우선 순위 2에서 2500, 우선 순위 0에서 4500 μ sec로 취하였고 b_i 는 모든 우선 순위에 1000 μ sec로 택하였다.

위의 벌칙 함수와 전송 지연의 평균을 복합하여 구성한 성능 지표, g 를 수식으로 표현하면

$$g(x, \omega) = \frac{\rho}{m_a} \sum_{i=0}^3 \sum_{j=1}^{m_{2i}} F_{2i}(\delta_{2i}(j)) + (1-\rho) \left(\frac{1}{m_a} \sum_{i=0}^3 \sum_{j=1}^{m_{2i}} \delta_{2i}(j) \right)^2$$

와 같고 여기서 x 는 우선 순위 도구의 네 변수들의 벡터이고 ω 는 관찰된 네트워크의 확률적 요소를 대표하는 변수이다. ρ 는 두 항의 비중을 조정해 주는 0과 1 사이의 수인데 여기서는 0.8의 값을 취하였다. 그리고, m_a 는 관찰된 모든 메시지의 수이고 m_{2i} 는 우선순위 $2i$ 에 속하는 관찰된 메시지의 수이므로 $m_a = \sum_{i=0}^3 m_{2i}$ 의 관계가 있다.

이러한 조건에서 시뮬레이션을 4회 수행하였다. 첫 수행에서는 THT를 기준값으로부터 -50, -25, 25, 50 μ sec만큼 변화시켰을 때의 성능 지표를 계산하였다. 나머지 3회의 수행에서는 각각 TRT4, TRT2, TRT0를 해당되는 기준값으로부터 -100, -50, 50, 100 μ sec만큼 변화시킨 경우를 수행하였다. 이 결과의 정확성을 밝히기 위하여 PA

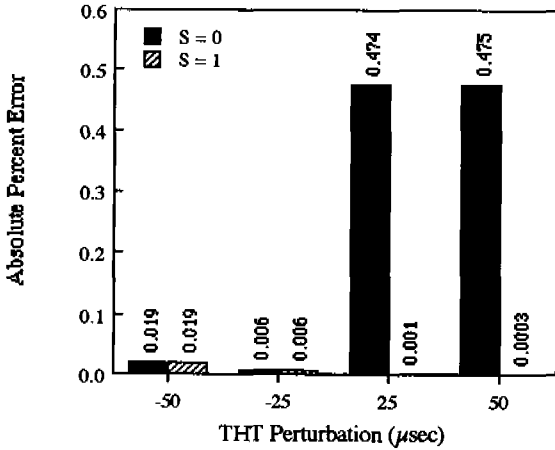


그림 5. THT 값의 변화에 대한 PA 알고리즘의 절대 백분 오차

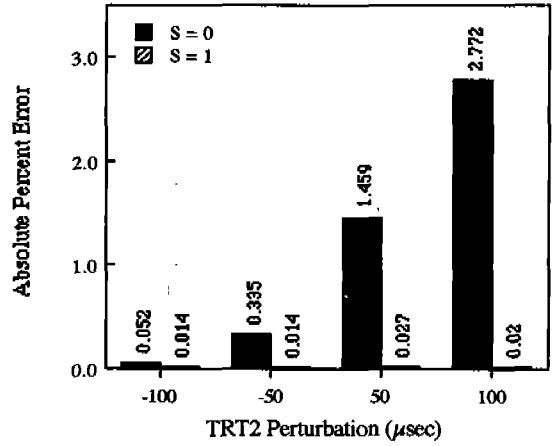


그림 7. TRT2 값의 변화에 대한 PA 알고리즘의 절대 백분 오차

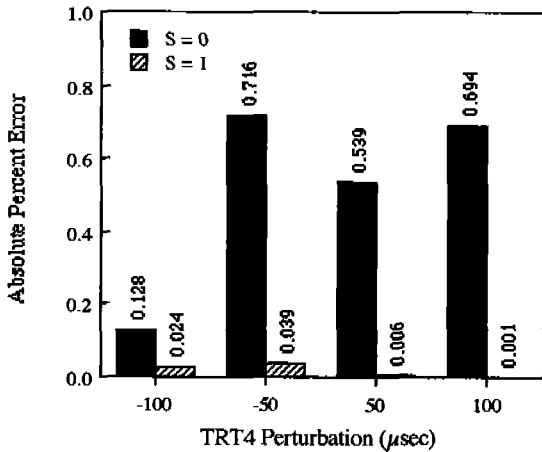


그림 6. TRT4 값의 변화에 대한 PA 알고리즘의 절대 백분 오차

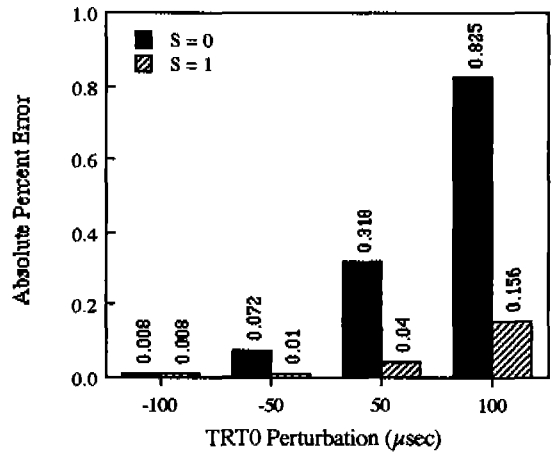


그림 8. TRT0 값의 변화에 대한 PA 알고리즘의 절대 백분 오차

알고리즘을 사용하지 않고 16종류의 변수값을 사용한 성능 지표를 얻기 위하여 16회의 시뮬레이션을 추가로 수행하였다.

이렇게 두 가지로 얻은 성능 지표를 비교하여 PA 알고리즘의 절대 오차 백분율을 그림 5, 6, 7, 8에 진한 막대 (S=0, S는 PA 알고리즘이 고려하는 미래의 메시지의 수)로 나타내었다. 이 그림들을 살펴보면 얻어진 결과가 상당히 정확한 것을 알 수 있고 한 변수를 기준값에서부터 감소시킨 경우보다 증가시킨 경우에 일반적으로 더 큰 오차가 발생함을 알 수 있다. 이러한 현상은 한 변수의 증가로 인하여 ΔC (이번에 전송 기회를 받는 순간의 변화)가 시뮬레이션의 많은 부분 동안 양수로 유지되어서 PA 알고리

즘이 수행되는 시점보다 미래의 대기 행렬의 상태를 모른 채 변화된 진행을 구성하여야 하기 때문이다. 간단한 예로서 기준 진행에서 토큰 순환시간 상으로는 전송이 가능하지만 전송할 메시지가 없는 경우에 비하여 변화된 진행에서 전송 기회가 조금 늦게 오고 그 사이에 전송할 메시지가 생겨서 전송되는 경우를 생각해 볼 수 있다. 이러한 상황에서 PA 알고리즘은 기준 진행에서 전송 기회를 다음 우선 순위로 넘기기 직전에 (하지만, 아직도 새로운 메시지가 발생하기 전) 수행되므로 새로운 메시지가 가까운 미래에 생기리라는 점을 모르며 결과적으로 아무런 메시지의 전송이 없는 실제와는 다른 변화된 진행을 구성하게 된다. 이 같은 난점을 극복하기 위해 시뮬레이션 모델이 유지하

고 있는 미래의 사상 목록(future event list)에서 바로 다음에 생성될 메시지를 검사하여 변화된 진행의 구성에 사용할 수 있다. 이와 같이 얻은 결과가 그림 5부터 8에 빗금 친 막대로 ($S=1$) 표시되어 있는데 오차를 더욱 감소시킬 수 있다. 그러나, 이러한 방법은 시뮬레이션에서만 가능하고 실제의 성능 관리에서는 불가능하다. PA 알고리즘을 사용한 시뮬레이션의 수행은 PA 알고리즘이 없이 수행한 것보다 약 25%의 시간이 더 소요되었지만 기존 진행과 네 개의 변화된 진행을 동시에 수행하므로 커다란 계산적인 이득이 있음을 입증하였다.

6. 결 론

본 연구는 토큰 버스 네트워크의 성능 관리를 위하여 필요한 성능 평가에 초점을 맞추어 우선 순위 도구의 네 변수의 변화가 네트워크 성능에 어떠한 영향을 미치는지를 추정하는 알고리즘을 개발하였다. 이 알고리즘은 perturbation analysis의 기법을 사용하여 개발되었는데 수학적 기법들이 요구하는 만족시키기 어려운 가정이 필요없고 이산 사상 시뮬레이션(discrete event simulation)보다 계산상으로 효율적이며 온 라인 (on line)으로 실행될 수 있는 장점을 갖고 있다.

이 알고리즘을 SIMAN으로 개발된 토큰 버스 시뮬레이션 모델에 접합하여 변수들의 기준값 이외에 4개의 추가적인 변수값하에서 성능을 추정하도록 구성되었다. 이 모델을 4회 수행하여 16가지 다른 조합의 변수 값에 대한 네트워크의 성능을 구하고 PA 알고리즘의 사용없이 같은 조건하의 성능을 시뮬레이트하여 비교한 결과, 오차가 5% 이내로 매우 정확함이 밝혀졌다. 또한, 기준 성능과 4개의 변화된 성능을 구함에 있어서 소요되는 시간이 기준 성능만을 구할 때에 비해서 약 25%만이 증가하는 계산상의 효율성을 보여주었다.

이 PA 알고리즘은 성능 관리 모듈이 보다 적절한 변수 값을 결정하기 위한 근간이 되어 이 논문의 연속인 제 2부에서 소개될 의사 결정 부분과 결합되어 총체적인 성능 관리를 형성한다.

참고문헌

1. W. Stallings, *Local and Metropolitan Area Net-*

works, 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. S. M. Klerer, "The OSI Management Architecture : An Overview," IEEE Network, Vol. 2, No. 2, March 1988, pp.20~29.
4. Institute of Electrical and Electronics Engineers, *IEEE Draft Standard 802.1B, LAN/MAN Management*, 1991.
5. L. Kleinrock, *Queueing Systems, Volume I : Theory*, John Wiley, 1975.
6. A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*, 2nd ed. McGraw-Hill, 1991.
7. Y-C. Ho and X-R. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic, 1991.
8. Institute of Electrical and Electronics Engineers, *ANSI/IEEE Standard 802.4, Token Passing Bus Access Method and Physical Layer Specifications*, 1990.
9. *Manufacturing Automation Protocol(MAP) 3.0 Implementation Release*, MAP/TOP Users Group, One SME Drive, P.O. Box 930, Dearborn, MI 48121.
10. C. Pegden, R. Shannon and R. Sadowski, *Introduction to Simulation Using SIMAN*, McGraw-Hill, 1990.

부 록

IEEE 802.4 토큰 버스 통신 규약과 그 우선 순위 도구
 IEEE 802.4 토큰 버스 통신 규약은 전송 매체 접속 통제(Medium Access Control, MAC)를 위하여 개발되었다. 토큰 버스 방식은 분산적인 통제된 접속 방식(distributed controlled access method)중의 한 종류로서 전송 매체를 사용할 수 있는 권리는 토큰(token)이라고 불리는 특수한 bit pattern에 의하여 결정되는데 이 토큰을 가진 스테이션만이 메시지를 전송할 수 있다. 어떠한 주어진 한

도 안에서 전송을 마친 스테이션은 토큰을 자신 다음에 토큰을 받게되어 있는 스테이션에게 넘기어 전송 기회를 부여한다. 이러한 과정이 계속되어 토큰은 네트워크에 연결된 모든 스테이션들 사이를 일정한 순서로 순환하여 동등한 전송 기회를 부여한다. 이러한 스테이션들의 순서를 logical ring이라고 일컫는다.

각 스테이션에는 4개의 우선 순위가 있는데 우선 순위 6이 가장 높고 그 보다 낮은 우선 순위 4, 2, 0이 차례로 존재한다. 한 스테이션이 토큰을 받으면 가장 먼저 우선 순위 6의 메시지가 대기하고 있는가를 검사한다. 기다리는 메시지가 있으면 곧바로 전송을 시작하여 (1) 더 이상 전송할 메시지가 없거나 (2) 이번 토큰을 받아서 전송한 우선 순위 6의 전송 시간의 합이 우선 순위 도구의 한 변수인 Token Holding Time(THT)을 초과할 때까지 전송을 계속한다. 단, 일단 시작한 전송 중에 전송 시간의 합이 THT를 초과하더라도 그 전송은 완료될 때까지 계속되고 그 이상의 전송이 불허된다. 이러한 두 조건 중의 하나가 만족되면 전송할 권리는 우선 순위 4로 넘어간다. 우선 순위 4에 속한 메시지의 전송은 대기하고 있는 메시지가 있고 지난 번의 토큰 순환에서 우선 순위 4에 전송 기회가 주어졌을 때부터 현재까지 소요된 시간 (토큰 순환 시간)

이 우선 순위 도구의 또 하나의 변수인 Token Rotation Time 4(TRT4)를 초과하지 않으면 개시된다. 이 같은 우선 순위 4의 전송은 (1) 더 이상 전송할 메시지가 없거나 (2) 이번 토큰을 받아서 전송한 우선 순위 4의 전송 시간의 합이 TRT4에서 토큰 순환 시간을 뺀 시간을 초과할 때까지 전송을 계속한다. 이 두 조건 중의 하나가 만족되면 전송할 권리는 우선 순위 2로 넘어가고 이와 같은 과정은 우선 순위 2와 0에 대하여 TRT2와 TRT0를 이용하여 반복된다. 우선 순위 0까지 이 과정이 수행되면 이 스테이션은 logical ring 상의 다음 스테이션으로 토큰을 전송한다.

이 같은 우선 순위 도구를 살펴보면 우선 순위 6의 메시지는 다른 메시지가 얼마나 전송되었는지와 무관하게 항상 일정한 길이의 시간 내에서 전송을 개시할 수 있다. 이와는 대조적으로 하위 순위인 4, 2, 0 순위의 메시지들은 토큰 순환 시간에 따라 전송을 개시할 수 있는 시간이 변화하고 심지어는 전혀 전송이 불허될 수도 있다. 이는 네트워크의 통신 부하가 커지면 토큰 순환 시간이 증가하여 하위 순위 메시지의 전송을 제한하려는 의도이다. 하위 순위 중에서 우선 순위 4, 2, 0의 순서로 높은 전송 기회를 부여하려면 $TRT4 > TRT2 > TRT0$ 을 만족시켜야 한다.