

MAP 네트워크 관리 모듈의 설계와 안정성 해석

(Design and Stability Analysis of MAP Network Management Module)

김 정 호*
Kim, Jeong Ho

Abstract

To operate the automatic devices of manufacturing process more effectively and to solve the needs of the resource sharing, network technology is applied to the unit control devices located in common manufacturing zone and operated by connecting them. The network management system, which controls and supervises the network resources in the communication network, is the function that is necessarily required in this network. For the stability in this network, this paper measures the variation of data packet length and node number and analyzes the varied value of the token rotation time and waiting time for the network operation. On the evaluation of the performance, the data packet length varies from 10 bytes to 100 bytes in the operation of the group management network, the variation of the token rotation time and waiting time is less than 10 msec. When the variation of the data packet length is less than 50 bytes, differences in token rotation time or waiting time is unnoticeable. Since the waiting time in this case is less than 10 msec, response time is fast enough.

요 약

본 논문에서는 생산 환경에서 MAP 네트워크에 접속된 여러 단위제어기기의 운영을 지원하기 위한 네트워크 관리 모듈을 설계하여 공정 모델 시스템에서 시범 운영하였다. 설계된 네트워크 모델의 안정성 해석을 위하여 토큰의 사용권의 공정한 배분과 시스템의 동작이 공정상의 특정한 시간 범위 내에 완료되어야 한다. 따라서 본 논문에서는 네트워크의 부하 파라미터로서 데이터 패킷의 크기와 노드의 수의 변화로 선정하여 이에 따른 네트워크 운영을 위한 토큰 로테이션 시간과 대기 시간의 대응된 변화값을 해석하였다. MAP 네트워크의 운영에서 데이터 패킷의 크기를 10바이트에서 100바이트까지 변화시킬 경우, 이에 대응한 토큰 로테이션 시간과 대기 시간의 변화는 주어진 조건하에서 10msec 이하의 값을 가진다. 따라서 프로그래머블 콘트롤

*전자계산조작응용기술사, 공업계측제어기술사, 전기통신기술사,
한국전자통신연구소, 실장/책임연구원

리 등의 단위 제어기기가 수십 msec 정도의 응답 하한선을 가지고 있으며 데이터 패킷의 크기는 50바이트를 넘지 않으므로 이의 실시간 처리에 충분히 적용 가능함을 알 수 있다.

1. 서 론

네트워크 관리(network management)는 네트워크 상에서 통신이 정상적으로 수행되도록 자원의 제어 및 감독을 수행하는 기능이다. 생산 공정의 네트워크에 접속된 여러 단위 제어기의 운영을 위한 네트워크 관리는 구성 관리, 성능 관리와 고장 관리로 MAP 네트워크의 매니저(manager)와 에이전트(agent)로 설계할 수 있다. 본 논문은 MAP 네트워크 관리의 매니저와 에이전트 기능을 정의하고 각 노드에 있는 MAP 인터페이스 보드의 파라미터를 설정하여 통신을 원하는 노드의 응용 명(application name), 노드 번지(node address)의 지원을 위한 MAP 네트워크 운영 규범을 설정하였다. 또한, MAP 네트워크의 안정성을 해석하기 위해서 생산 공정에서 프레임의 발생 환경과 특성의 가정하에서 MAP 네트워크의 데이터 패킷 크기와 노드 수의 변화에 따른 토큰 로테이션 시간과 대기 시간의 대응된 변화값을 검증하여 네트워크 운영의 안정성을 해석하였다.

2. 네트워크 관리 기능과 구조

2.1 네트워크 관리의 기능

MAP 네트워크는 OSI 모델의 전계층의 관리에 적용될 수 있지만 생산 공정에서 MAP 네트워크의 관리 범위는 OSI 모델의 3~7계층의 관리에만 적용된다. 따라서 1, 2계층의 관리는 다른 관리자에 의하여 분리 관리된다. 네트워크 관리구조는 여러 영역으로 나눌수 있지만 구성 관리(configuration management), 성능 관리(performance management), 이벤트 처리(event processing), 고장 관리(fault manage-

ment)의 네가지 관리 응용으로 제한된다. 구성 관리는 시스템의 상태를 결정하고 제어하는 것으로 시스템 상태 정보 수집, 시스템 상태의 이력 저장, 시스템 상태 표현을 위하여 데이터를 수집한다. 구성 관리에서 사용되는 데이터는 노드의 상태, 변수의 리스트등의 실시간 데이터와 버전, 노드의 위치, 노드의 형태, 장치의 호환성, 각 계층에 지원된 프로토콜을 제어하고 평가하는 것으로 시스템 통계 수집, 이력 저장을 위하여 데이터를 수집하며 여기서 사용되는 변수는 카운터로서 이 기능에서는 재전송의 수, 접속 수 등이 처리율, 병목을 판단하는 정보를 제공한다. 고장 관리는 사건 발생을 통고하고 해석하며, 고장을 진단하며, 시스템 상태 변화에 대한 정보 수집, 시스템 상태 변화의 보고 제어, 시스템 상태 변화의 이력, 시스템 상태 변화 표현을 위하여 데이터를 수집한다.

2.2 네트워크 관리의 구조

MAP 네트워크 관리의 구조는 여러 형태의 정보를 모으고 처리하는 요소로 그림 1과 같이 매니저와 에이전트로 수행된다.

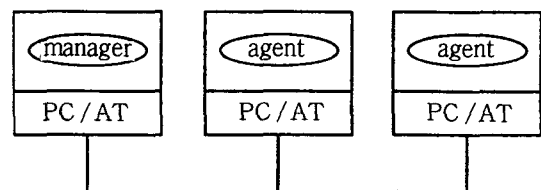


그림 1 네트워크 관리에서 매니저-에이전트의 구성 개념

(1) 매니저

MAP 네트워크 시스템 관리를 위한 명령을 각 노드에게 보내거나, 다른 노드의 상태 변화

에 대한 데이터를 수집하여 관리자에게 보여준다. 이 기능을 수행하기 위하여 매니저는 휴먼 인터페이스(human interface)와 관리 프로세서(management processor)로 나뉘어진다. 휴먼 인터페이스는 네트워크 관리자와 관리 프로세서의 인터페이스로 네트워크 관리자가 네트워크 시스템에 대한 데이터를 읽고 변경시키고 보고하는 메카니즘을 제공한다. 관리 프로세서는 네트워크 관리자가 요구하는 기능을 수행하는 실제적인 하드웨어, 소프트웨어이다. 이는 에이전트로부터 데이터를 받기도 하고, 에이전트에게 요구를 보내기도 한다. MAP V3.0에서 관리 프로세서는 구성관리, 성능 관리, 고장관리의 3가지 기능을 수행하도록 규정하고 있다.

(2) 에이전트

에이전트는 매니저가 요구하는 명령을 수행하고 그 결과를 매니저에게 보내주기도 하고 각 노드에 사건이 발생할 때 매니저에게 통고한다. 에이전트가 발신, 수신하는 메시지는 어떤 OSI 계층, 부계층이 선택되었는지를 알려주면 네트

워크 관리자로부터 시스템 관리 요구를 받아 이에 대한 응답을 보낸다.

3. MAP 네트워크 모델과 네트워크 관리 설계

3.1 모델 공정

생산 공정에서 MAP 네트워크는 조립 및 시험 공정에서 설치되고 있으며 대부분 프로그래머블 콘트롤러, 로봇 등의 단위 제어 기기의 그룹 관리를 목적으로 운영되고 있다. MAP 네트워크의 운영을 위한 모델 시스템을 그림 2와 같이 구성하였다. 본 모델의 MAP 운영은 IBM PC/AT를 호스트로 하였으며, INI사 제품인 MP-500 인터페이스 보드를 장착한 4개의 노드로 구성하였다. 또한, 네트워크 관리를 위한 컴퓨터 1개 노드와 단위 제어 기기로서 프로그래머블콘트롤러 인터페이스, 수치 제어 기기 인터페이스, 로봇 인터페이스의 3개 노드로 구성하였다.

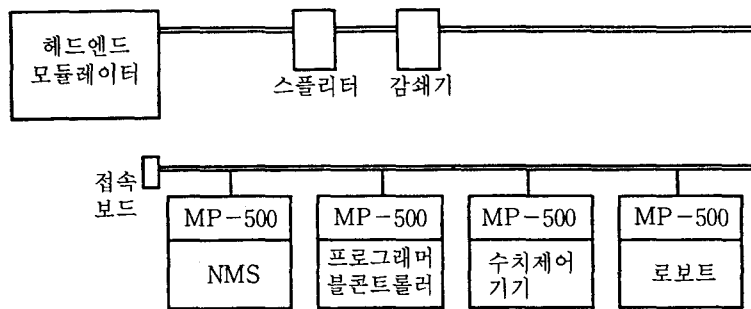


그림 2 모델 시스템의 네트워크 구성

본 모델의 소프트웨어 구성은 네트워크 소프트웨어와 응용 소프트웨어로 구성되며 그림 3과 같다. 여기서 네트워크 소프트웨어는 MAP 운영 네트워크로서 응용 계층과 데이터 링크 계층을 포함하는 프로토콜과 네트워크 관리 소프트

웨어이고, 응용 소프트웨어는 응용 계층을 이용하는 단위 제어 기기 운영의 사용자 프로그램인 단위 제어 기기의 그룹 관리 서비스 모듈을 의미한다.

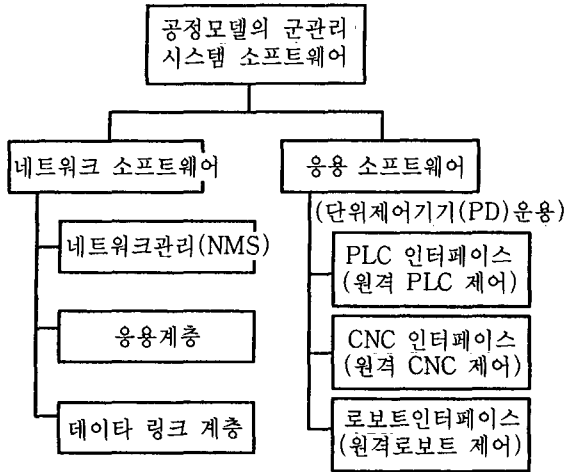


그림 3 모델 시스템의 소프트웨어 구성도

3.2 매니저-에이전트 관계

MAP 네트워크에 있어서 매니저-에이전트 관계는 응용 계층 서비스를 이용하여 수행되며 관리 정보 전달의 접속 확립, 제어, 정보 교환, 사건 통보의 네 종류의 일로 분류된다. 이러한 일

을 수행하기 위한 서비스 프리미티브는 다음과 같다.

- 접속 확립
 - M-Initialize
 - M-Terminate
 - M-Abort
- 제어
 - M-Set
 - M-Action
 - M-Compare
 - M-Blocking
- 정보교환
 - M-Get
- 사건 통보
 - M-Event-Report

3.3 관리 메세지 형태

관리 메세지는 IEEE 802.1에 기술된 형태에 기반을 두었으며 그림 4에 관리 메세지의 형태를 나타내었다.

PPU ID	Layer Sublayer	Request /Response Event Info		Request /Response Event Info
--------	----------------	------------------------------	--	------------------------------

그림 4 네트워크 관리 메세지 형태

- PDU ID는 관리 메세지의 요구 PDU, 응답 PDU, 이벤트 PDU의 종류를 나타낸다.
- Layer /Sublayer Info는 관리 메세지와 관련이 있는 OSI 계층과 부계층을 나타낸다.
- Request /Response /Event Info는 PDU의 형태에 의존하고, Request /Response Info는 동작 변경 ID를 포함한다.
- 요구 PDU에서 위의 필드는 read req, set req, compare and set req, action req, test req 중의 하나가 된다.

- Multiple read req와 set req의 조합 형태는 한 요구 PDU에 표현할 수 있다.
- 이벤트 PDU의 필드는 사건 정보를 포함한다.

3.4 MAP네트워크 관리의 설계

구성된 모델의 4개 노드 사이의 네트워크 관리는 MAP 인터페이스 보드의 동작과 통신을 원하는 노드 접속에 대한 정보를 제공하며 매니저 기능과 에이전트 기능으로 수행된다. 이의

소프트웨어 구조는 그림 5에 나타내었다.

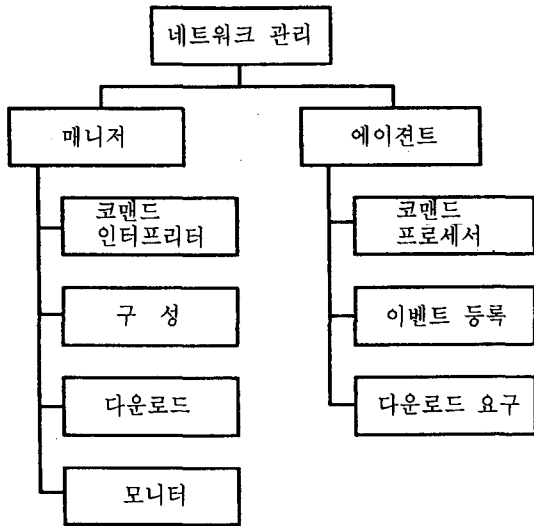


그림 5 네트워크 관리 소프트웨어 구조

네트워크 관리의 매니저는 코맨드 인터프리터 (command interpreter), 구성(configuration), 다운로드(download), 모니터(monitor)와 이와 관련된 데이터파일로 구성된다. 코맨드 인터프리터는 네트워크 관리의 매니저가 관리자로부터 명령을 받아, 그 명령을 해석하여 명령의 종류에 따라 구성, 다운로드, 모니터 프로세스로 명령을 보내주는 기능을 수행한다. 그리고, 입력된 명령을 수행한 결과는 네트워크 운영자에게 보여준다. 구성은 각 노드가 초기화 될때, 각 노드의 운영을 위한 관련 파라미터의 원하는 값을 선택하여야 한다. 다운로드는 각 노드에서의 네트워크가 운영될때, 해당 노드의 에이전트가 네트워크 관리의 매니저에게 네트워크 관련 파라미터를 다운로드하도록 요구한다. 매니저는 이 요구에 따라 각 노드에 다운로드할 파일을 찾아서 다운로드를 수행한다. 모니터는 입력된 인터프리터로부터 명령을 받아서 각 노드를 모니터하는 기능을 수행한다. 모니터는 명령 종류에 따라 'ADD NODE'는 모니터하는 노드를 첨가

하는 기능, 'DEL NODE'는 모니터하는 노드를 삭제하는 기능, 그리고 'EXIT'는 네트워크에 접속된 노드를 모니터 하는 기능을 마치고 수행 전단계로 회복하는 기능을 수행한다. 이 모니터에 하는 기능을 마치고 수행 전단계로 회복하는 기능을 수행한다. 이 모니터에 의하여 제공되는 정보는 네트워크에 접속된 노드의 수, 노드의 상태, 발신 프레임의 수, 수신 프레임의 수, 발신 사용자 데이터 수, 수신 사용자 데이터 수가 되며 이에따른 네트워크 운영의 안정성에 대한 측정 기준이 된다. 매니저의 소프트웨어 운영은 그림 6과 같다. 여기서 MAP 인터페이스는 MAP 네트워크를 통하여 각 노드의 에이전트와 연결하여 실질적으로 통신을 가능하게 하는 프로세스이다.

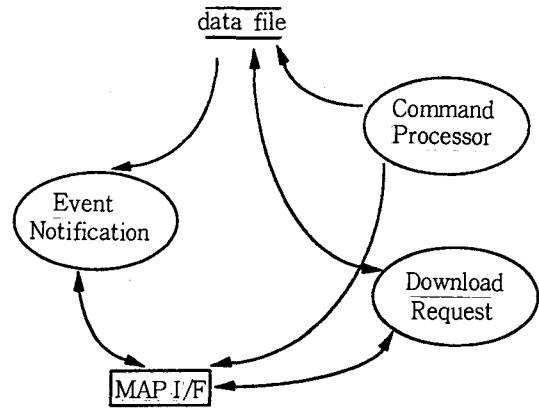


그림 6 매니저의 구성도

네트워크 관리의 에이전트는 코맨드 프로세서 (command processor), 이벤트 등록(event notification), 다운로드 요구(download request)로 구성된다. 코맨드 프로세서는 매니저의 명령에 의해서만 동작하는 수동적인 프로세스이다. 매니저가 적당한 서비스를 사용하여 네트워크 파라미터를 변경하거나 읽기를 원할때 관련 데이터 파일을 액세스하여 그 명령을 수행한다. 그리고, 수행한 결과를 MAP 인터페이스를 통

하여 매니저에게 보고된다. 이벤트 등록은 각 노드의 상태 변화 등의 사건이 발생할때 매니저에게 통고한다. 다운로드 요구는 각 노드의 시스템이 동작이 시작될때, 각 노드의 네트워크 파라미터를 초기화시키기 위하여 에이전트가 매니저에게 다운로드를 요청하며, 매니저로부터

받은 네트워크 파라미터의 값을 MAP 인터페이스 보드에 세트시킨다.

에이전트의 소프트웨어 운영은 그림 7과 같다. 여기서 MAP 인터페이스는 MAP 네트워크를 통하여 매니저와 연결하여 실질적으로 통신이 가능하게 하는 프로세스이다.

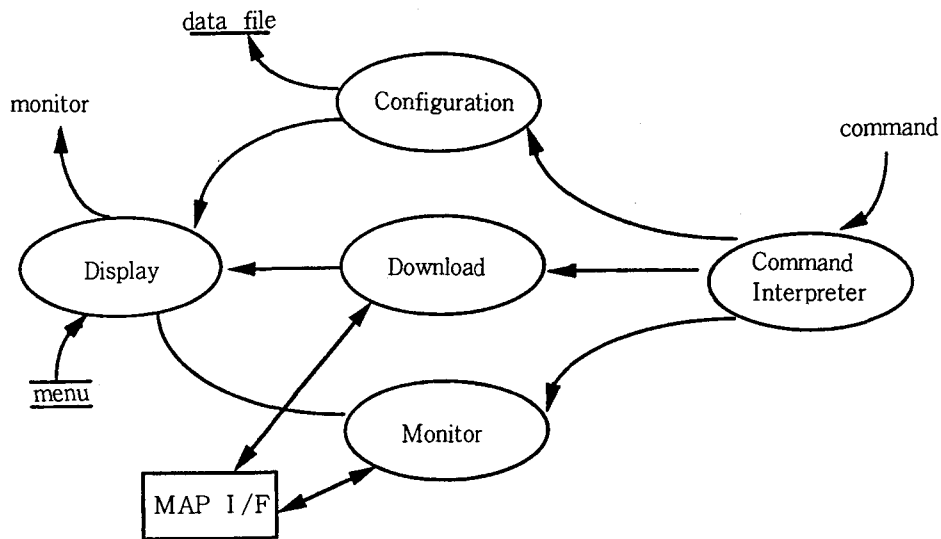


그림 7 에이전트의 전체 구조

4. MAP 네트워크 관리의 운영

에서 유일한 예외 상황은 고장 관리에서 일어난

4.1 운영 관점

네트워크 관리자는 정보를 모으거나 제어기능을 수행하기 위하여 네트워크 관리 에이전트와 상호 작용한다. 네트워크 관리 에이전트는 네트워크의 각 노드에 존재하며 네트워크 관리자가 보낸 요구에 대한 응답으로서 메시지를 보낸다. 그러나 비정상 상태가 발생할때, 에이전트는 사건 통지 메시지를 네트워크 관리자에 보내어 초기화시킨다. 그림 8은 네트워크 관리자와 네트워크 관리자 에이전트와의 관계를 보여준다. 네트워크 관리자와 에이전트는 응용계층 프로토콜과 관리 메시지를 사용하여 통신한다. 이 구조

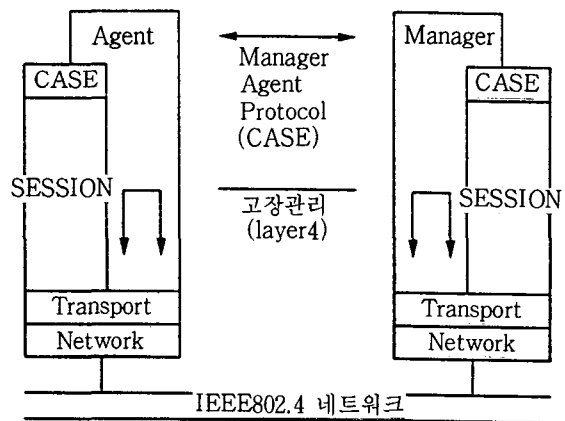


그림 8 네트워크 관리의 운영 구조도

다. 고장관리는 트랜스포트 계층에서 루프백 메시지를 취급하는 에이전트를 요구한다. 그러므로 에이전트는 두 TSAP(transport service access point)에서 트랜스포트 계층과 인터페이스를 할 수 있어야 한다. 한 TSAP는 트랜스포트 연결 루프백 테스트에 사용되고, 다른 TSAP는 트랜스포트 데이터 에코우 테스트에 사용된다.

4.2 네트워크 관리의 운영과 절차

본 논문에서 MAP 네트워크 관리의 운영 지원을 위한 유틸리티는 다음과 같이 설계되었다. 디렉토리 유틸리티는 DIB(directory information base)를 만들고 유지한다. DIB는 MAP 모델 시스템에 존재하는 응용간의 통신을 위한 정보를 제공한다. 즉 AP name, SSAP, MAC address, NSAP, TSAP, LSAP 등에 관한 정보를 제공한다. 관리 유틸리티는 각 노드

의 관리 파라미터를 구성할 수 있는 기능으로 모뎀 형태, 채널 구성, 노드번호, 토큰 로테이션 타임, 슬롯 타임, 토큰 유지 시간, 최대 전송 버퍼 크기, 최대 수신 버퍼 크기 등에 관한 정보를 제공한다. 구성 유틸리티는 네트워크 관리가 MAP 인터페이스 보드에 액세스하기 위해 DMA 채널과 호스트 입출력 번지를 구성한다.

이에따라 MAP 모델 공정의 각 노드를 구성하는 과정을 그림 9에 나타내었다. MAP 모델 시스템의 각 노드간의 통신을 위하여 각 노드에서 네트워크 관리기능을 이용하여 필요한 정보를 입력시킨다. 따라서 MAP 인터페이스 보드의 DNA 채널과 호스트 입출력 번지를 구성 유틸리티를 사용하여 변경시키며, MAP 인터페이스 보드의 계층 및 다른 통신 특성을 관리 유틸리티를 사용하여 변경시킨다. 통신을 원하는 응용에 대한 정보를 DIB에 입력하거나 변경할때는 디렉토리 유틸리티를 사용하여 응용을 추가

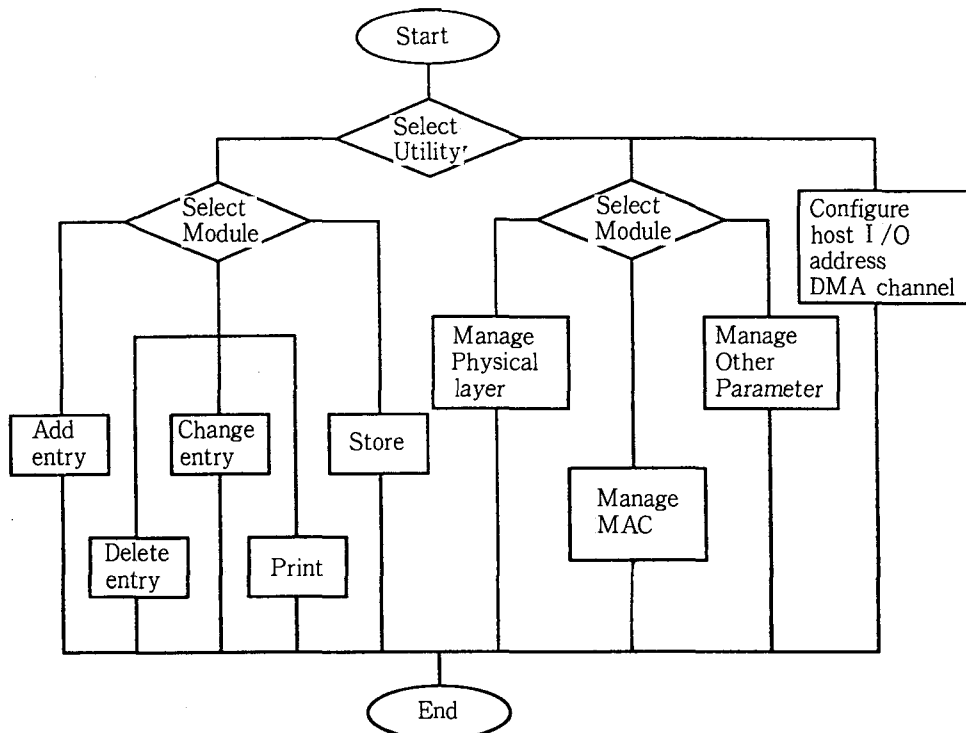


그림 9 MAP에서 네트워크 관리의 구성관리 흐름절차

시킬때는 add entry 모듈을, 응용을 삭제할때는 delete entry 모듈을, 입력된 응용에 관한 정보를 변경할때는 change entry 모듈을 사용한다.

4.3 운영 규범

관리자가 네트워크 관리를 위하여 데이터를 상호 교환하기 위하여서는 MAP 서비스와 프로토콜을 이용한다. 관리자와 에이전트간에 데이터를 상호 교환하는 운영 규범은 다음과 같다.

- 관리자에 의해 application association이 확립된다.
- 네트워크 관리 요구 PDU는 항상 관리자에게 에이전트로 보내진다.
- 네트워크 관리 요구 PDU는 관리자의 요구에 대한 응답으로 항상 에이전트에서 관리자로 보내진다.
- 네트워크 관리 이벤트 PDU는 에이전트에서의 사건에 대한 응답으로서 에이전트가 초기화 하며 에이전트에서 관리자로 보내진다.
- 네트워크 관리자는 에이전트에 한개 이상의 요구 PDU를 가질 수 있다.
- 에이전트는 모든 요구 PDU에 대하여 응답하여야 한다.
즉, 요구한 동작이 만족하게 수행되었거나(데이터를 보냄), 수행할수 없음(상태정보를 보냄)에 대하여 응답한다.
- 요구 PDU는 여러 요구 요소(read req, set req)를 포함한다. 에이전트는 수신된 순서대로 요구 PDU에 대하여 응답할 필요가 없을 뿐 아니라 요구 PDU내에 있는 각각의 요구 요소에 대하여서도 응답할 필요가 없다. 그러나 read req나 set req의 변수에 대해서는 read res나 set res로 응답하여야 한다.
- 에이전트로 부터의 응답이 없을 경우 네트워크 관리의 동작 규정은 없지만 관리자의 유일한 방법은 요구를 다시 보내는 것이다. 에이전트는 적당한 방법으로 응답하는 것이다.

- 에이전트는 관리자와 OSI 연결이 되었을때만 이 요구나 이벤트 PDU를 보낸다. 에이전트는 연결을 초기화하지 않는다.
- 사건 처리는 관리자가 에이전트와 연결되지 않은 상태에서 사건이 발생하면 에이전트는 사건을 무시하거나 FIFO 큐에 최대 8개까지 저장시킨다.
- 에이전트가 여러 관리자와 동시 연결을 허용하지 않는다.

5. MAP 네트워크의 안정성 해석

5.1 해석적 모델의 제안

MAP 네트워크의 운영은 토큰 버스 방식으로 한 노드에서 프레임 전송을 위한 최대 서비스 시간은 토큰 유지 타이머에 의하여 제한된다. 생산 공정에서 MAP 네트워크 설치 비용과 구성 환경, 제조 공정별로 객체화 운영 및 유지보수의 편이성, 작업 수행상의 실시간 처리, 네트워크 관리 노드의 트래픽 정도 등의 요구에 적합한 부하 파라미터의 선택이 필요하다. 따라서 본 논문에서는 MAP 네트워크의 안정성을 해석하기 위해서는 네트워크 운영에서 토큰의 사용권 부여에 대한 보장을 검증하여야 한다. 네트워크 성능을 해석하기 위하여 생산 공정과 가능한 한 실제 상황과 가깝게 네트워크 운영을 묘사하기 위하여 생산 공정의 단위 제어 기기의 프레임 발생 환경과 특성을 다음과 같이 가정한다. 첫째, 전송하는 프레임의 크기가 대부분 일정하며 그 크기도 작다. 둘째, 실시간 처리를 요구하는 감시 및 제어 기능의 프레임의 목적지는 네트워크 관리 운영 노드로 일정한 전송율을 가진다. 이러한 생산 공정에서 프레임의 발생 환경과 특성의 가정하에서 본 논문에서는 MAP 네트워크의 부하 파라미터로서 노드의 수와 도착율의 변화로 선정하였으며 이에따른 토큰 로테이션 시간과 대기 시간의 대응된 변화값을 해석한다.

본 논문에서 MAP 네트워크에 대한 해석적 모델은 우선적으로 토큰을 가지고 있는 노드가 한번에 하나의 프레임만을 서비스하는 비고갈(non-exhaustive) 방식으로 하여 해석적 모델로 제안한다. 이 방식은 고갈(exhaustive) 방식에 비하여 토큰 로테이션 시간의 분산을 작게 유지할수 있다는 유리한 점을 가지고 있어서 네트워크 운영에서 토큰의 공정한 배분을 할당할수 있다. 따라서 MAP 네트워크의 노드에서 타이머의 값을 조절함으로써 토큰을 가질수 있는 최대 시간인 토큰 보유 시간을 최소화시킬수 있다. 본 논문에서 MAP 네트워크의 구조는 근본적으로 토큰 버스의 성능에 크게 의존하므로 이의 해석적 모델은 물리계층과 데이터 링크 계층의 오버 헤드, 인식 프레임의 송신 등이 고려되어야 한다. 여기서 인식 프레임은 프레임이 수신될 때마다 발생하므로 소프트웨어적인 지연 시간을 D_s 라고 할때 MAP 네트워크의 해석적 모델은 식 5-1과 같이 프레임 서비스 시간인 x 와 슬롯 시간인 T_s 를 가지는 토큰 버스의 해석적 모델로 변환하여 해석할 수 있다.

$$x = \frac{8}{C}(M+26)+x_r+T_s, \dots\dots\dots\text{식 5-1}$$

$$T_s = 2(D_s+D_p)+(\text{safety margin})$$

여기서 C 는 채널 속도, D_p 는 전달 지연 시간, M 은 프레임의 평균 크기, x_r 는 인식 프레임 서비스 시간을 나타낸다. 본 논문에서는 D 는 10 [μsec]로, 전송 안정 여유는 IEEE 802.4 표준에서 규정하고 있는 2 [μsec]로 하여 해석한다. 이 경우에도 노드에 대한 서비스 방식은 변함이 없으므로 노드는 토큰을 전달 받은 후에야 프레임을 전송하게 되며 토큰은 차례대로 노드를 순환하게 된다. 따라서 각각의 노드에는 하나의 큐가 존재하게 되며 이들 큐는 토큰 로테이션 시간을 한 주기로 하는 주기적인 서비스를 받는다. 부하 파라미터는 노드에 도착하는 프레임이 되는 데 이들의 도착 간격의 분포는 보통 지수

분포나 포아송 분포를 따른다. 본 논문에서는 프레임의 도착을 포아송 분포를 가지는 것으로 가정한다. 따라서 도착 분포는 포아송 분포, 서비스 시간 분포는 일반적인 분포, 접속 서비스 채널수는 하나로 하여 M/G/1 큐를 가정할수 있다. 그림 10에 이의 큐잉 모델을 나타내었다. 이에따라 프레임 서비스 시간으로 부터 MAP 네트워크의 안정성의 중요한 파라미터가 되는 토큰 로테이션 시간과 대기 시간에 대한 해석식을 구할수 있다.

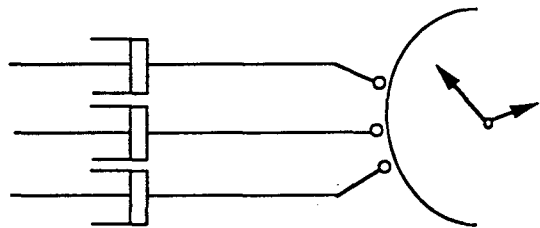


그림 10 MAP 네트워크의 큐잉 모델

(1) 토큰 로테이션 시간의 유도

우선, 토큰 로테이션 시간은 프레임의 서비스 시간과 토큰 버스 오버 헤드, 윈도우 개방에 따른 오버헤드의 합으로 식 5-2와 같이 나타낼수 있다. 여기서 토큰 로테이션 시간을 T_r 도착율을 $L(p/sec)$ 이라고 할때,

$$T_r = \text{도착한 프레임의 서비스 시간} + \text{토큰 버스 오버헤드} + \text{윈도우 개방에 따른 오버헤드}$$

$$= T_1 + T_2 + T_3$$

$$= T_1 + T_0, \dots\dots\dots\text{식 5-2}$$

으로 나타낼 수 있다. 여기서 T_0 는 전체의 오버헤드를 의미한다. 식 5-2에서 우변의 항들을 차례대로 구해보며 다음과 같다. 각각의 노드에는 T_r 동안 평균 $L \cdot T_r$ 개의 프레임이 도착하며 트래픽 밀도는 $P=L \cdot T_r$ 로, 이는 각각의 노드에서

T₁ 동안 서비스할 프레임이 있을 확률이 된다. 그러므로 T₁ 동안 N개의 노드 중에서 n개의 노드에서 서비스를 수행할 확률은 바이노미얼(binomial) 분포를 갖게되어 다음 식 5-3이 성립한다.

$$P(n=k) = \binom{N}{k} \cdot P^k (1-P)^{N-k} \quad 0 \leq k \leq N$$

.....식 5-3

$$n = N \cdot P$$

$$n^2 = NP(1-P) + N^2 P^2$$

$$T_1 = x_1 + x_2 + x_3 \dots + x_n \quad 0 \leq n \leq N$$

여기서 x는 n번째의 노드에서 프레임의 서비스에 걸린 시간을 나타낸다. 식 5-3에서 T₁, x, n, 이 모두 랜덤 변수이므로 T₁의 평균값은 다음식 5-4와 같다.

$$E[T_1] = E[x_1 + x_2 \dots + x_n]$$

$$\bar{T}_1 = E[x_1] + E[x_2] + \dots + E[x_n]$$

.....식 5-4

한편 n이 랜덤 변수이고 x와 n은 서로 독립이라고 가정하면 T₁ 동안 도착한 프레임의 서비스 시간 T₁의 평균값은 식 5-6과 같다.

$$\bar{T}_1 = E[n] X = Npx \quad \dots \dots \dots \text{식 5-6}$$

다음 토큰 전달에 따른 오버헤드 T₂는 다음식 5-7과 같이 주어진다.

$$T_2 = N x_r + N D_p = N(x_r + D_p) \quad \dots \dots \dots \text{식 5-7}$$

여기서 전달 지연 시간 D_p의 영향을 고려한 것은 다음과 같이 토큰의 전달 지연 과정에서 D_p만큼의 지연이 있기 때문이다.

윈도우 개방에 따른 오버헤드를 살펴보면 가장 낮은 어드레스를 갖는 노드를 제외하면 ISSC(inter solicit successor counter)의 값이 0이 될때마다 한번씩 슬롯 시간 만큼의 윈도우를 개방하게 된다. 그리고 가장 낮은 어드레스를 갖는 노드는 슬롯 시간의 두배 크기를 갖는 윈도우를 개방하게 된다. 그러므로 한번의 토큰 로테이션 시간 동안에 고려되어야 할 평균적인 오버헤드는 식 5-8과 같다.

$$T_3 = \frac{T_s}{ISSC} (N-1) + \frac{T_s}{ISSC} 2$$

$$= \frac{T_s}{ISSC} (N+1) \quad \dots \dots \dots \text{식 5-8}$$

따라서 토큰 로테이션 시간의 평균값은 다음의 식 5-9로 주어진다.

$$\bar{T}_r = T_1 + T_2 + T_3 = N \cdot p \cdot x \cdot N \cdot (X_r + D_p)$$

$$+ \frac{T_s}{ISSC} (N+1)$$

$$= Npx + T_0 = \frac{T_0}{1 - N \cdot L \cdot x} \quad \dots \dots \dots \text{식 5-9}$$

(2) 대기 시간의 유도

다음에 M/G/1 큐 모델에 관한 클라인록(Kleinrock)이 제안한 식을 이용하여 대기 시간을 구하면 다음의 식 5-10, 5-11, 5-12들을 얻을 수 있다.

$$q = P + P^2 \frac{1 + C_b^2}{2(1-P)} \quad \dots \dots \dots \text{식 5-10}$$

$$W_s = \frac{P(1 + C_b^2)}{2(1-P)} \cdot \bar{T}_r \quad \dots \dots \dots \text{식 5-11}$$

$$C = \frac{\delta^2 T_r}{(\bar{T}_r)^2} \quad \dots \dots \dots \text{식 5-12}$$

식 5-10은 M/G/1 큐 모델의 평균 큐 크기에 관한 식을 나타내며 폴랴크킨친(Pollaczek-Khinchin) 평균값의 공식이다. 여기서 분산 계수의 제곱인 C_b²를 구하기 위하여 δ²T_r을 구하면 식 5-3에서 5-13을 구할 수 있다.

$$\delta^2 T_r = \bar{T}_r^2 - (\bar{T}_r)^2 = NP(2-P)X^2 \quad \dots \dots \dots \text{식 5-13}$$

그러므로 분산 계수의 제곱인 C_b²와 큐에서의 대기 시간 w는 다음의 식 5-14, 5-15들로 구할 수 있다.

$$C_b^2 = \frac{NP(2-P)X^2}{\bar{T}_r^2} \quad \dots \dots \dots \text{식 5-14}$$

$$w_s = \frac{L(\bar{T}_r^2 + NP(2-P)x^2)}{2(1-P)} \quad \dots \dots \dots \text{식 5-15}$$

따라서 프레임은 큐를 벗어나서 곧바로 서비스를 받는 것이 아니라 노드가 토큰을 넘겨받은 후에야 서비스를 받게된다. 그러므로 토큰을 넘

겨 받을 때까지의 잠재적인 지연 시간을 고려하여야 한다. 만약 프레임이 큐를 벗어났을때 노드는 토큰을 가지고 있었다면 지연 시간은 0이며, 노드가 방금 토큰을 다른 노드로 넘겨 주었다면 지연 시간은 \bar{T}_r 이 될 것이다. 이와 같이 살펴보면 잠재적 지연 시간은 \bar{T}_r 은 구간 $[0, T_r]$ 에 균일하게 분포하는 랜덤 변수가 된다. 따라서 T_L 의 평균값은 다음 식 5-16으로 주어진다.

$$\bar{T}_L = E[T_L] = \frac{\bar{T}_r}{2} \dots\dots\dots\text{식 5-16}$$

식 5-15과 5-16에서 전체적인 대기 시간 W는 다음 식 5-17과 같다.

$$W = w + \bar{T}_L = \frac{\bar{T}_r + N \cdot P(2-P)x^2}{2(1-P)} + \frac{\bar{T}_r}{2} \dots\dots\text{식 5-17}$$

생산 공정에서 MAP 네트워크 해석을 위하여 본 논문에서는 접속 노드수에 대한 프레임의 대기 시간 처리에 대한 해석식을 확장하여 인식 프레임의 서비스 시간, 토큰 버스 운영의 소프트웨어 지연시간, 전송 안정 여유등의 오버헤드를 고려함으로써 MAP 네트워크에 대한 보다 개선된 식을 유도하였다.

5.2 시뮬레이션 결과

MAP 네트워크의 제안된 해석식에서 데이터 패킷의 크기를 10 바이트에서 100 바이트까지 변화시킬 경우, 노드의 수(N), 데이터 패킷의 도착율(L)에 따른 토큰 로테이션 시간과 대기 시간의 변화를 그림 11과 그림 12에 나타내었다. 그림에서 노드의 수(N)가 60 이상으로 증가할수록, 도착율(L)이 60 packet/sec 이상으로 커질수록, 데이터 패킷의 크기에 따라 토큰 로테이션 시간과 대기 시간의 증가 비율이 급격하게 커짐을 알수있다. 그러나 노드의 수(N)가 40 이하인 경우와 도착율(L)이 40 packet/sec 이

하인 경우에는 토큰 로테이션 시간과 대기 시간은 10 msec 이하로 나타난다. 이 경우는 데이터 패킷의 크기의 증가로 인하여 토큰 로테이션 시간과 대기 시간에 큰 영향을 주지 못함을 알수 있다. 그림 12에서 데이터 패킷의 크기가 50 바이트 이하인 경우는 본 네트워크 운영에서 주어진 조건하에서 대기 시간이 노드수가 60인 경우를 제외하고 노드에서의 소프트웨어 지연을 고려하고도 대부분 10msec 이하의 값을 나타낸다. 따라서 인식 패킷의 서비스 시간, 토큰 버스 운

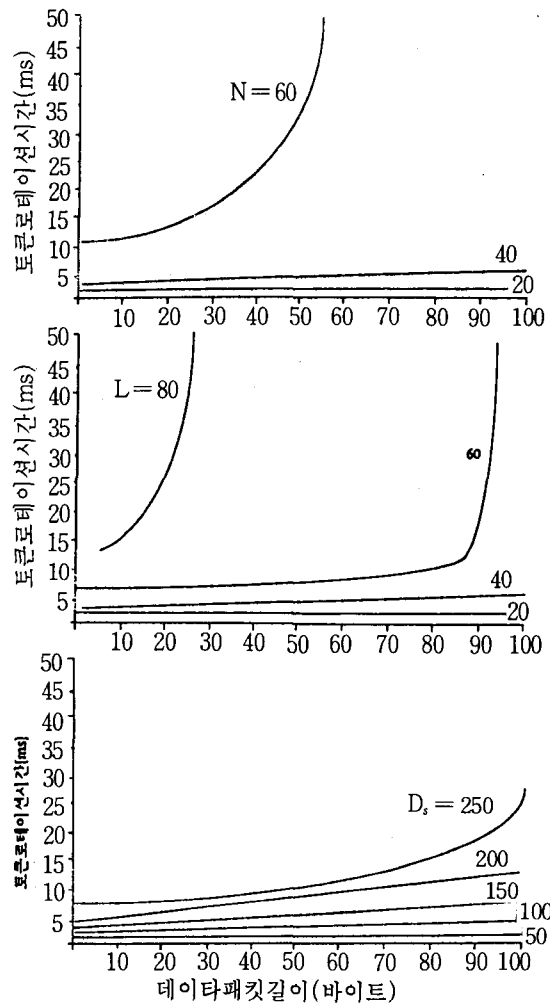


그림 11 데이터 패킷 크기의 변화에 따른 토큰 로테이션 시간의 결과

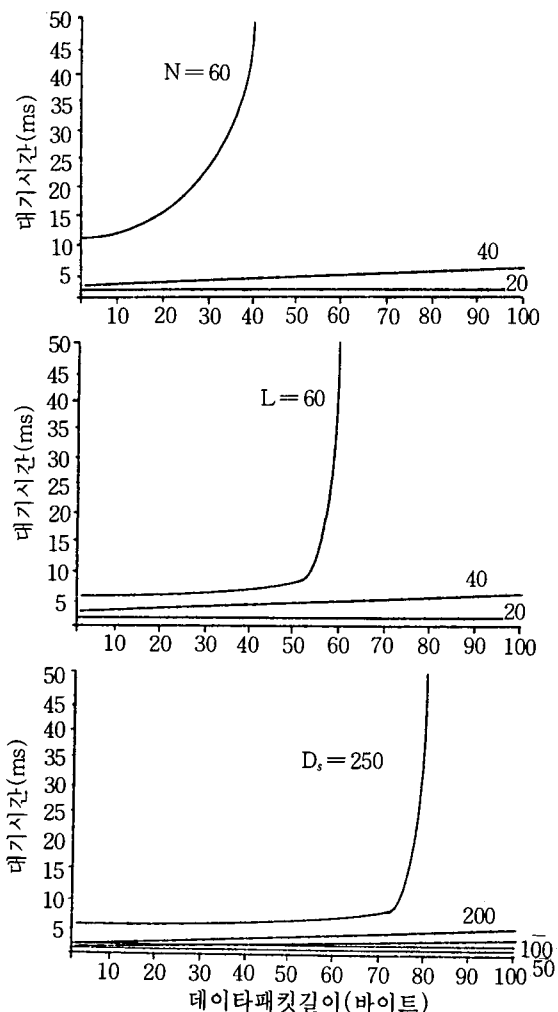
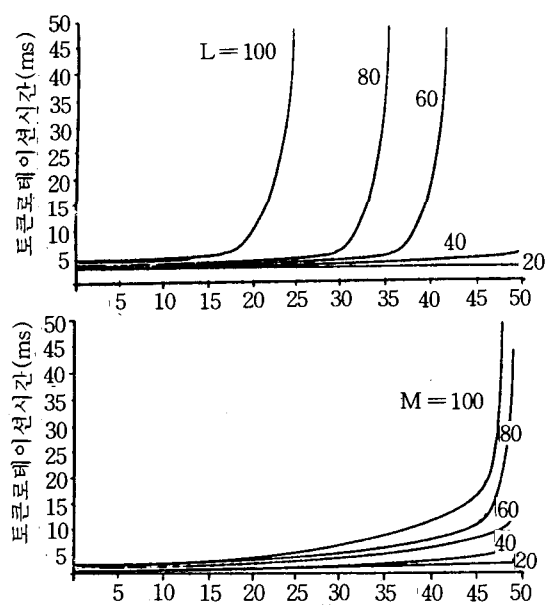


그림 12 데이터 패킷 크기의 변화에 따른 대기 시간의 결과

영의 소프트웨어 지연 시간과 전송 안정 여유 등의 오버헤드를 고려하여 제안된 토큰 로테이션 시간과 대기 시간의 해석식이 타당함을 알 수 있다. 따라서 프로그래머블 컨트롤러 등의 단위 제어 기기가 정보의 크기가 50 바이트 범위이고, 수십 msec의 응답 시간 범위를 가지고 있으므로 생산 공정에서 실시간 처리 네트워크에 충분히 적용 가능함을 보여준다.

그림 13 과 그림 14는 노드의 수의 변화에 따른 결과를 나타낸 것이다. 시뮬레이션 수행에서

데이터 패킷의 크기(M), 소프트웨어 지연 시간(D_s)과 도착율(L)의 변화에 따른 결과에서도 노드의 수의 영향을 예측할 수 있다. 노드에서의 지연 시간이 150μsec 이하일때 노드의 수가 40 이하이면 안정된 값을 나타내고 있음을 알 수 있다. 또한 노드에서의 소프트웨어 지연 시간(D_s)이 200μsec 이상이 되면 평균 도착율(L)이 40packet/sec 이하, 노드의 수(N)가 40 이하일 경우 군관리 네트워크의 안정된 운영이 수행됨을 알 수 있다. 이는 실제 생산 공정에서 응용 범위가 제한될 수 있음을 의미한다. 노드에서의 소프트웨어 지연 요소는 사용자의 응용 소프트웨어에 대한 의존이 크므로 소프트웨어의 효율적인 작업이 중요한 요소임을 알 수 있다. 또한 이들의 시뮬레이션의 결과가 데이터 패킷의 크기, 도착율의 변화의 결과와 거의 일치하는 것으로 보아 제안된 토큰 로테이션 시간과 대기 시간의 해석적 식의 유효성이 검증된 것으로 생각된다.



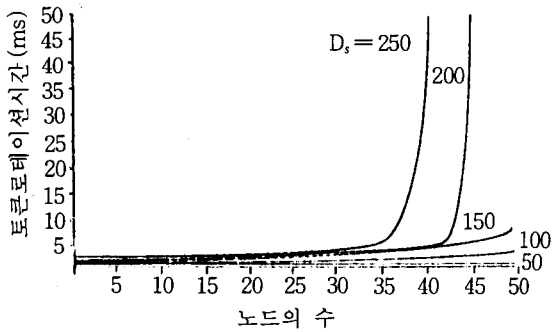


그림 13 노드 수의 변화에 따른 토큰 로테이션 시간의 결과

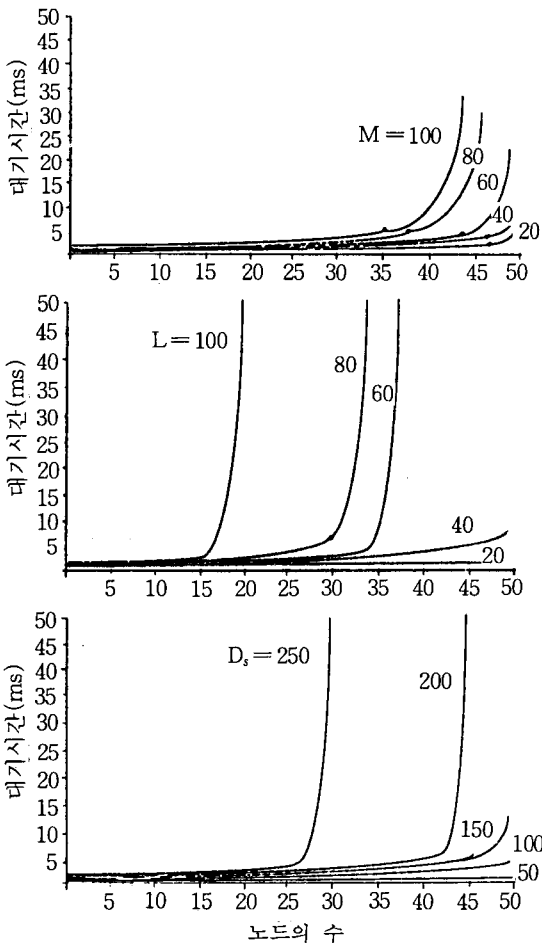


그림 14 노드 수의 변화에 따른 대기 시간의 결과

6. 결 론

네트워크 상에서 정상적인 통신을 위하여 접속자원의 운영을 담당하는 것으로 네트워크 관리 기능이 필수적으로 요구된다. 본 논문에서는 생산 환경에서 MAP 네트워크에 접속된 여러 단위제어기기의 운영을 지원하기 위한 네트워크 관리 기능 요구와 MAP 규격에 따른 구성관리, 성능관리와 고장관리에 대한 매니저와 에이전트 관계로서 네트워크 관리 모듈을 설계하고 운영하였다. 이의 운영 모델은 4개의 단위제어기기 노드간의 통신을 가능하게 한 MAP 네트워크 관리 기능을 수행하기 위한 매니저-에이전트 인터페이스, 관리 메시지의 형태, 서비스 프리미티브를 설계하고 MAP 운영 규범을 설정하여 운영하였다. 생산 공정에서 MAP 네트워크의 안정성을 평가하기 위하여 데이터 패킷 크기의 변화에 따른 토큰 로테이션 시간과 대기 시간의 변화, 노드의 수에 따른 토큰 로테이션 시간과 대기 시간의 변화에 대한 해석을 수행하였다. 또한 이들 성능 지표의 평가는 해석적 제안식에 대한 시뮬레이션을 수행하여 유효성을 검증하였다. 데이터 패킷의 크기를 10바이트에서 100바이트까지 변화시킬 경우 설계된 군관리 네트워크의 제안된 토큰 로테이션 시간과 대기 시간의 변화는 데이터 패킷의 크기가 50바이트 이하인 경우에 대기 시간의 변화는 이 노드 수가 60인 경우를 제외하고 응용 계층의 오버헤드를 고려하더라도 대부분 10msec 이하의 값을 나타내었다. 이는 생산 공정에서 프로그래머블 컨트롤러 등의 단위 제어기기가 수십 msec 정도의 응답 하한선을 가지고 이의 정보의 데이터 패킷 크기가 50바이트를 넘지 않는 범위에서 설계한 군관리 네트워크가 안정하게 운영이 가능함을 알수 있었다. 따라서 본 논문에서 설계된 네트워크는 생산 공정의 비고갈 서비스 경우에서 제안된 토큰 로테이션 시간과 대기 시간의 해석 식은 인 식 패킷의 서비스 시간, 토큰 버스 운영의 소프

트웨어 지연 시간, 전송 안정 여유 등의 오버헤드를 고려하더라도 노드의 수가 40 이하이고, 평균 도착율이 40 packet/sec 이하의 결과로서 네트워크의 안정된 동작을 수행함을 알 수 있으며 실시간 처리 분야에 적용이 가능함을 알 수 있었다.

참 고 문 헌

1. Andrew Kusiak and Sunderesh S. Heragu, "Computer Integrated Manufacturing: a Structural Perspective", IEEE Network, 2, No.3, pp.14-22, May 1988.
2. L.J. McGuffin, L.O. Reid and S.R. Spark, "MAP/TOP in CIM Distributed Computing", IEEE Network, Vol.2, No.3, pp.23-31, May 1988.
3. Vincent C. Jones, 'MAP/TOP Networking: A Foundation for Computer Integrated Manufacturing', McGraw-Hill Inc., 1988.
4. GM Task Force, 'Manufacturing Automation Protocol Specification V3.0 Implementation Release', GM Corp., 1988.
5. Luping Liang et al., "Process Groups and Group Communications: Classifications and Requirements", IEEE Computer, 1990.
6. A. Ray and Y. Halevi, "Finite-Dimensional Modeling of Network induced Delays for Real-Time Control Systems," in Proc. Amer. Contr. Conf., pp.1077-1082. June 1988.
7. A.P. Jayasumana and G.G. Jayasumana, "Simulation and Performance Evaluation of 802.4 Priority Scheme," in Proc. IEEE Symp. Simulation Comput. Networks, pp.232-238, Aug. 1988.