

FDDI를 통하여 연결된 이질의 네트워크들에 대한 성능 평가 도구의 개발*

Development of Performance Evaluation Tool for Heterogeneous Networks Interconnected through FDDI

임지영**, 채기준**
Ji Young Lim, Ki Joon Chae

Abstract

Local area networks(LANs) recently become so large that they exceed their architected limits and users on different LANs may need to talk each other. This trend has led to a growth in the practice of interconnecting many heterogeneous LANs.

In this paper, we proposed a method to interconnect Ethernets and token rings, which are widely used today, through a fast FDDI network. We also developed a simulation tool to evaluate the performances of the interconnected networks. We used a simulation language SLAM II for modeling and OSF/MOTIF X-window for user interface. The developed tool can be used to make a decision which interconnecting method is suitable for a given environment before interconnecting many heterogeneous LANs.

1. 서론

여러 개의 네트워크를 서로 연결하는 방법으로는 동질(homogeneous)의 것들을 서로 연결하는 경우와 이질(heterogeneous)의 것들을 서로 연결하는 경우의 두 가지 종류가 있다. 전자의 경우는 연결하고자 하는 네트워크들이 서로 같은 프로토콜을 사용하기 때문에 간단한 라우팅(Routing)과 저장 후 전송(store-and-forward) 기능을 하

는 브리지(Bridge)에 의해 연결되어지고, 후자의 경우는 프로토콜이 서로 다른 네트워크들을 연결하기 위해 프로토콜 변환 기능을 갖고 있는 라우터(Router)나 게이트웨이(Gateway)에 의해 연결되어진다. 본 연구에서는 이질의 근거리 통신망(Local Area Network: LAN)들을 여러 개의 라우터를 통하여 연결하고 그것의 성능을 분석하는 도구를 개발하고자 한다.

근거리 통신망으로는 1979년 제록스사에 의해서 개발된

* 본 연구는 1994년도 한국과학재단 핵심전문연구비 지원에 의한 결과임.

** 이화여자대학교 전자계산학과

후 국내뿐 아니라 전 세계적으로 가장 많이 사용되고 있는 Ethernet과, IBM에 의해 개발 보급되고 있으며 성능면에서는 Ethernet 보다 우수한 것으로 평가받고 있는 Token Ring, 전송매체로 광섬유를 사용하고 전송속도가 100 Mbts/sec으로 빠르기 때문에 국내외에서 Backbone 네트워크로 많이 사용되어지고 있는 FDDI를 사용하고자 한다. 전송매체 접근 제어(Medium Access Control: MAC) 방법으로, Ethernet은 IEEE에서 표준화 된 CSMA/CD(Carrier Sense Multiple Access with Collision Detection: IEEE 802.3) [12]를 사용하며, Token Ring과 FDDI는 기본적으로 IEEE에서 표준화 된 Token Ring(IEEE 802.5)[13] 방식을 사용한다.

LAN의 성능분석을 위한 시뮬레이션 패키지로는 CACI사의 NETWORK II.5[14] 등이 있으나 이와 같은 패키지들은 패키지 자체가 지원하는 기능들을 사용하는 네트워크에 대한 성능분석은 편리하게 수행할 수 있으나, 기능을 수정하거나 확장하는 경우에는 매우 불편하거나 때로는 불가능하다. 특히, NETWORK II.5는 논문에서 수행한 시뮬레이션 모델 중 라우터에 대한 기능의 지원이 미약하고 프로토콜을 수정, 확장하기 어렵다.

따라서 본 논문에서는 성능평가를 위하여 시뮬레이션으로 시스템 내에 어떠한 사건이 발생할 때에, 즉 시스템의 상태가 바뀔 때에 시간이 증가하는 이산 사건 시뮬레이션(Discrete Event Simulation)을 사용하였다[1]. 이를 구현하는데 있어서 범용 언어로 모델링을 할 것인지 시뮬레이션 언어로 할 것인지는 시뮬레이션 언어의 다음과 같은 장점 때문에 시뮬레이션 언어로 네트워크를 모델링하였다.

- 시뮬레이션 언어는 자동으로 시뮬레이션 모델을 프로그래밍하는 데 필요한 대부분의 특징들을 제공한다. 결과적으로 프로그래밍하는 데 걸리는 시간을 상당히 감소시킨다.
- 시뮬레이션 언어들은 시뮬레이션 모델링을 위한 자연스런 뼈대를 제공한다. 이들의 기본적인 블록들은 FORTRAN과 같은 범용 언어의 블록보다 시뮬레이션에 더 적합하다.
- 시뮬레이션 모델들은 시뮬레이션 언어로 작성되었을 때 일반적으로 수정하기 쉽다.
- 대부분의 시뮬레이션 언어는 시뮬레이션 수행동안 동적 기억장소 할당(dynamic storage allocation)을 가능하게 한다.

- 시뮬레이션 언어에서는 여러 가지 형태의 가능한 오류들이 구별되어 자동으로 검사되기 때문에 오류의 검출이 쉽다.

그렇지만 시뮬레이션 언어는 범용 언어보다 익숙하지 않다는 점과 수치계산과 같은 네트워크 모델링에 있어서 자세한 부분은 구현하기 어려운 점이 존재한다. 따라서 시뮬레이션 언어 중에서 범용 언어인 FORTRAN과 연결해서 사용이 가능한 SLAM II를 시뮬레이션 언어로 결정하였다.

본 논문에서는 여러 개의 Ethernet과 Token Ring을 FDDI Backbone 망에 라우터를 사용하여 연결한 네트워크의 성능을 분석하기 위해 각각의 독립적인 네트워크에 대한 시뮬레이션 모델을 먼저 구현하였고 이를 바탕으로 OSF/MOTIF X-window를 이용해 구축한 사용자 인터페이스로 사용자가 원하는 네트워크의 모델을 만들어 성능을 분석할 수 있게 하였다.

전체 네트워크에 연결된 Ethernet과 Token Ring의 수, 각 네트워크에 연결된 스테이션의 수, 각 스테이션에 도착하는 패킷의 길이, 패킷 도착간 간격, 네트워크의 길이 등은 네트워크의 성능에 큰 영향을 미치므로, 이들 시스템 매개변수들과 전송 지연시간 및 처리량과의 관계 등을 살펴보고자 한다.

위에서 개발된 시뮬레이션 모델은 사용자와 대화식으로 작동하게 된다. 즉, 사용자로부터 데이터를 입력받고 시뮬레이션을 자동으로 수행한 후 SLAM II에서 제공하는 결과 중 사용자에게 도움이 되는 성능 측정치를 추출하여 계산한 후 출력하게 된다. 입력 데이터는 연결하고자 하는 Ethernet이나 Token Ring들의 숫자, 하나의 LAN에 연결되어진 스테이션들의 숫자, 하나의 스테이션이 전송하기 위하여 만들어 내는 정보량, 라우터에서의 정보처리시간 등이다. 주된 두 가지의 성능 측정치는 데이터 전송 지연시간(Delay)과 데이터 처리량(Throughput)이다. 데이터 전송 지연시간은 보내는 스테이션에 데이터가 도착한 시간부터 받는 스테이션에서 그 데이터를 받는 시간까지의 경과시간이고, 데이터 처리량은 단위시간 동안 네트워크 상의 모든 스테이션들에 의하여 받아진 전체 데이터의 양으로 표시되어진다. 여러 다른 상황 하에서 측정치들을 비교 분석하여 어떠한 연결방법이 주어진 상황 하에서 성능이 떨어지지 않고 더 잘 작동하는가를 판단하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 개발도구의

개요와 대상 근거리 통신망 즉, Ethernet, Token Ring과 FDDI의 개요에 대해 설명하고, 3장에서는 시뮬레이션 모델의 설계, 4장에서는 근거리 통신망의 통합과 그 사용에 대하여 설명하고, 5장에서 결론을 맺고자 한다.

2. 개요

이 장에서는 개발하고자 하는 도구에 대한 간단한 개요와 본 연구에서 사용되어진 네트워크들에 대하여 간단히 설명하고자 한다.

2.1 개발도구의 개요

여러 이질의 근거리 통신망을 FDDI Backbone 네트워크에 연결한 네트워크 모델의 성능을 평가하기 위한 도구의 구조도는 <그림 1>과 같으며 이 도구의 기능은 다음과 같다.

- Ethernet, FDDI와 Token Ring의 기본 시뮬레이션 모델을 객체화하여 이들의 상호연결된 네트워크의 모델링
- 동일 자료에 대한 반복입력의 자동화
- SLAM II 컴파일러가 제공하는 복잡한 결과로부터 보기 쉽고 이해하기 쉬운 형태의 결과 제시
- 강력한 인터페이스를 구축할 수 있도록 도와주는 OSF/MOTIF X-window 시스템의 이용으로 마우스를 사용하는 전 작업과정의 메뉴방식 사용자 인터페이스에 의한 진행

개발도구의 작업은 다음과 같은 순서로 진행된다.

첫째, 사용자로부터 매개변수의 값을 입력으로 받아 원하는 모델을 만들어 실행하기 위하여 LAN 매개변수 입력기에서 FDDI, Token Ring, Ethernet과 라우터에 대한 사용자 매개변수들을 사용자 인터페이스를 통해 입력받는다.

둘째, 입력받은 데이터는 모델 구축기에 전달되며 사용자 인터페이스를 통해 사용자로부터 수행시간 등의 수행에 필요한 입력 매개변수를 받아들인다. 또한 모델 구축기는 미리 구축해 놓은 FDDI, Ethernet, Token Ring과 Router의 기본 모델을 갖고 있는 기본 시뮬레이션 모델 데이터 베이스와 입력받은 데이터를 바탕으로 하여 사용자가 원하는 시뮬레이션 모델을 구축한다.

셋째, 시뮬레이션 처리기는 위에서 구축된 시뮬레이션 모델을 입력으로 하여 UNIX에서 수행되는 SLAM II 컴파일러를 이용하여 이 모델에 대한 시뮬레이션을 수행하고 그에 대한 이 모델의 결과화일을 얻는다.

네째, 결과 분석기는 시뮬레이션 수행에서 얻어진 결과화일이 필요없는 데이터가 많고 복잡하므로 필요한 정보만을 뽑아 이를 바탕으로 사용자가 원하는 결과를 계산하여 사용자 인터페이스를 통해 보여준다.

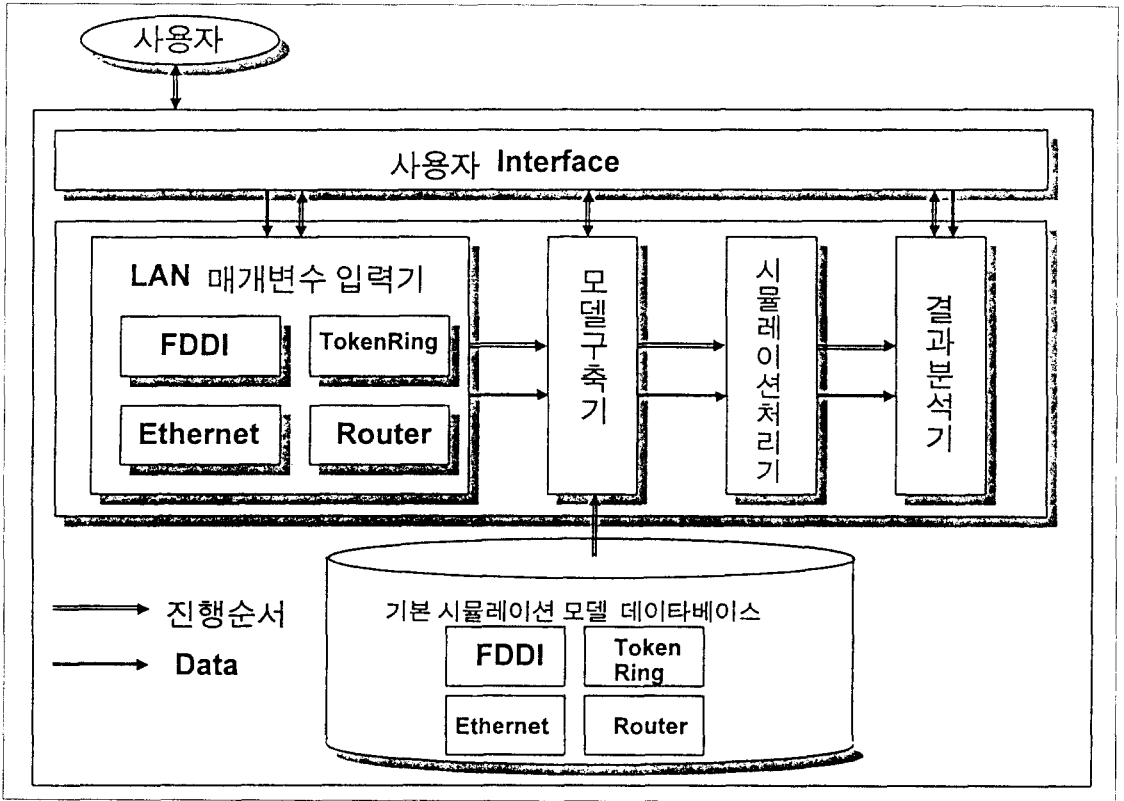
3장과 4장에서 각 단계에 대한 자세한 설명을 보겠다.

2.2 대상 근거리 통신망

2.2.1 Ethernet

Ethernet은 1979년 제록스사에서 처음 개발한 이래 급속한 속도로 보급이 되어 현재는 전 세계적으로 가장 많이 사용되어지고 있는 근거리 통신망으로서 전송속도는 10 Mbits/sec이며 전송매체로는 동축 케이블이나 트위스티드 페어를 사용한다. 전송방식에 따라 베이스 밴드(디지털 신호)와 브로드 밴드(아날로그 신호)의 두 가지로 나누어진다. 동축 케이블을 사용하는 베이스 밴드 Ethernet에는 10BASE-5(Thick Ethernet: 최대 세그먼트 길이 500m)와 10BASE-2(Thin Ethernet: 최대 세그먼트 길이 200m)의 두 종류가 있으며 버스 토폴로지를 갖는다. 트위스티드 페어를 사용하는 것에는 10BASE-T가 있고 이것은 Hub를 통하여 스테이션들을 연결하는 성형 토폴로지를 갖는다. 본 연구에서는 10BASE-5 Ethernet을 사용하였다.

Ethernet에서 사용되어지는 매체 액세스 프로토콜인 CSMA/CD(Carrier Sense Multiple Access with Collision Detection)는 IEEE 802.3[12] 표준안에 정의되어 있으며 이 방식의 작동 원리는 다음과 같다. 네트워크 안의 어떤 스테이션이 패킷을 전송하고 싶으면 다른 스테이션이 패킷을 전송하고 있는지를 조사한다. 그 시간에 아무도 패킷을 전송하고 있지 않다는 것이 확인되면 패킷을 전송하기 시작한다. 그러나 다른 스테이션이 패킷을 전송하고 있으면 그 전송이 끝날 때까지 기다렸다 전송을 시작한다. 그러나 두 개 이상의 스테이션이 동시에 패킷 전송을 시작한다면 충돌(Collision)이 발생한다. 전송을 시작하기 전에 다른 스테이션의 패킷 전송 여부를 전송하였음에도 불구하고 충돌이 발생하는 이유는 패킷 전파 시간



〈그림 1〉 개발도구의 구조도

(Propagation delay)이 0이 아니기 때문이다. 일단 충돌이 발생하면 전송을 하던 스테이션들은 전송을 중지하고 네트워크 내의 모든 스테이션들에게 충돌이 발생했음을 알리기 위해 충돌 신호(Jamming signal)를 보낸다. 충돌 신호를 보낸 후 임의의 시간(Backoff delay)을 기다린 후에 다시 전송을 시도한다. 동일 패킷에 대한 반복적인 충돌은 네트워크 사용량이 매우 높다는 것을 나타내기 때문에 각 스테이션들은 재 전송의 시기를 지연시킴으로써 네트워크의 부하를 조절할 수 있다. 이러한 지연시간은 Truncated Binary Exponential Backoff라고 하는 무작위 과정을 통해 이루어지며 슬롯 시간의 정수배이다. 슬롯 시간은 최대 왕복 전파시간보다 크거나 같아야 한다. 표준안에서는 10 Mb/s의 전송속도를 갖는 CSMA/CD 네트워크에 대한 슬롯 시간을 $51.2\mu s$ 로 정하고 있다. n 번째 재 전송에 대한 슬롯 시간의 값은 0과 2^k 사이의 임의의 정수가 된다. (단, $k = \text{Min}(n, 10)$) 재전송의 숫자가 16번이 되면 그

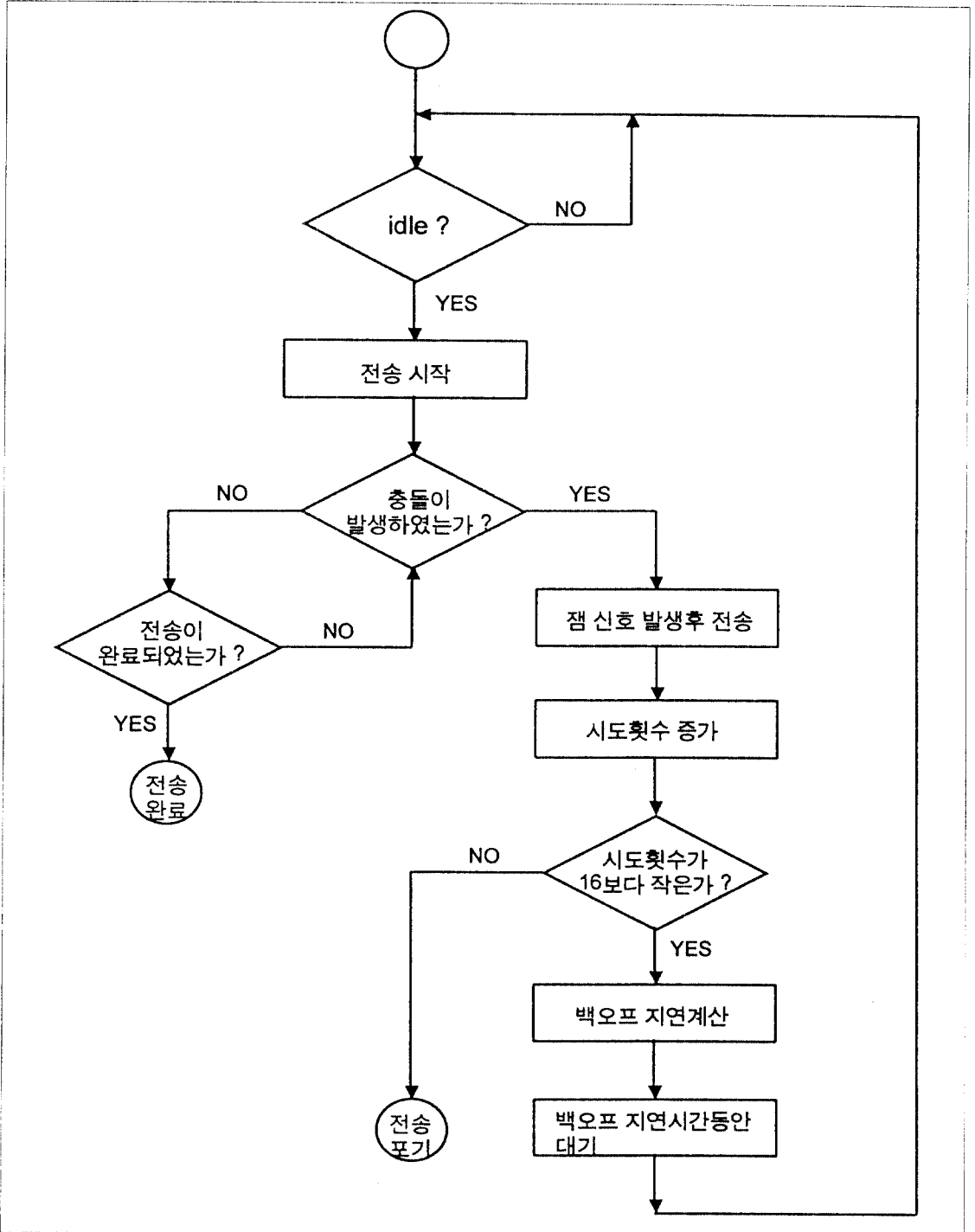
스테이션은 전송을 포기하고 에러를 그 위 계층인 Logical Link Control 계층에게 알려준다. 이 과정에 대한 알고리즘을 기술하면 다음과 같다[2].

```

while ( Attempts < Backoff_Limit ) do
  begin
    k := Min(Attempts, 10)
    r := Random(0, 2k)
    delay := r * Slot_Time
  end

```

CSMA/CD 프로토콜은 최대 패킷의 길이는 1518 바이트로, 최소 패킷의 길이는 512 비트로 지정하고 있어 패킷의 길이가 512 비트 보다 작으면 그 차이 만큼의 패드 비트를 데이터 뒤에 붙여준다.



〈그림 2〉 Ethernet의 흐름도

2.2.2 Token Ring

Token Ring은 IEEE 802.5[13]에 의하여 표준화가 되어 있으며, 표준화가 된 초창기에는 주로 IBM에 의하여 제품이 개발되었으나 요즘은 여러 회사에 의해 제품이 개발되고 있다. 성능면에서 Token Ring은 네트워크 내의 교통량이 적을 때나 많을 때 거의 모든 경우에 있어서 Ethernet 보다 우수하지만, 토큰을 관리해야 한다는 단점도 있다.

Token Ring의 접근 제어 방식은 다음과 같이 작동한다. Token Ring은 링을 따라 도는 작은 패킷인 토큰을 사용하며, 모든 스테이션이 전송할 데이터가 없을 때 토큰은 프리 토큰(free token) 상태로 링을 돈다. 패킷 전송을 원하는 스테이션은 프리 토큰이 지나가기를 기다려서 그 토큰 상의 토큰 비트를 변형시켜 비지 토큰(busy token)으로 만든 후, 그 비지 토큰 뒤에 데이터 패킷을 붙여 전송한다.

링에 이제는 프리 토큰이 없으므로 전송을 원하는 스테이션은 프리 토큰이 지나갈 때까지 기다려야 한다. 전송된 패킷이 링을 돌아 다시 전송을 시작한 스테이션으로 돌아오면 그 스테이션은 패킷을 제거하고, 다음의 두 조건이 동시에 만족될 때 새로운 프리 토큰을 만들어 링으로 내보낸다.

- 스테이션이 패킷의 전송을 끝마쳤을 때
- 비지 토큰이 전송한 스테이션에 돌아왔을 때

링의 비트 길이가 패킷 길이보다 긴 경우에는 첫번째 조건이 두번째 조건 보다 먼저 만족된다. 즉, 전송한 스테이션은 패킷 전송을 먼저 끝나치고 비지 토큰이 돌아오기를 기다렸다가 토큰 도착시 새로운 프리 토큰을 만들어 낸다.

각 스테이션은 리피터를 통하여 링에 연결 되는데, 이 리피터의 중요한 기능 중 하나는 지나가는 패킷을 검색하는 것이다. 즉, 지나가는 패킷의 목적지가 자신에게 연결되어 있는 스테이션인지를 알기위해 각 리피터마다 1 비트의 지연시간동안 패킷을 검색한 후, 목적지가 맞으면 패킷을 복사하여 자신에게 연결되어 있는 스테이션으로 보내고 그렇지 않으면 다음 리피터로 보낸다. 전송을 시작한 스테이션이 maximum으로 token을 가지고 있을 수 있는 시간은 THT(Token Holding Time) 동안이며 이 시간이 지나면 전송할 패킷이 있더라도 더 이상 전송을 할 수 없게 된다. 데이터 전송률이 4Mbps 이고 THT = 10 msec

라면 토큰을 잡은 스테이션은 5000 bytes를 보낼 수 있다.

2.2.3 FDDI

FDDI는 전송매체로서 광섬유를 사용하며, 100 Mbps/sec 의 빠른 전송속도를 갖는 ANSI[3]에 의해 표준화가 된 근거리 통신망이다. FDDI는 이러한 특성 때문에 현재 국내외에서 전송속도가 느린 여러 개의 근거리 통신망들을 연결하기 위한 Backbone 네트워크로 많이 사용되고 있다. 광섬유를 이용함으로써 얻게 되는 넓은 대역폭, 보안성, 안전성, 전자파 방해에 대한 면역성, 무게 및 크기의 감소는 광 근거리 통신망에 대한 관심을 모으기에 충분한 것이어서 FDDI는 차세대 망으로서의 위치를 굳히게 되었다. FDDI는 링 토폴로지를 이용하기 때문에 노드 간의 거리, 스테이션의 수 및 설치 가능한 네트워크의 총 범위 등 모든 면에서 제약이 적다는 장점이 있다.

FDDI 네트워크 상에서 각 노드 간의 거리는 최대 2 Km 까지 가능하고, 연결할 수 있는 최대 노드 수는 1000 개이며 네트워크의 총 연장 길이는 100 Km(이중 링인 경우 200 Km)까지 가능하다.

매체 접근 제어 방법으로는 IEEE 802.5에서 표준화된 4 Mbps/sec 의 토큰 링 방식을 기본으로 하여 타임드 토큰 프로토콜(Timed Token Protocol)을 사용하고 있다[5][6][7][8][9][10][11][15][16]. 패킷 전송을 원하는 스테이션은 링에서 돌고 있는 토큰이라는 작은 패킷을 잡은 후에 전송을 시작한다. 전송이 끝난 후에 그 스테이션은 링 안으로 새로운 토큰을 집어넣어 다른 스테이션들이 사용할 수 있게 한다.

FDDI에서는 동기식과 비동기식의 두가지 서비스를 제공한다. 동기식 서비스는 대역폭과 응답시간의 제약 등을 미리 알 수 있는 응용 분야를 위한 것이고, 비동기식 서비스는 대역폭을 미리 예측하기 힘들거나 응답 시간의 제약이 심하지 않은 응용 분야를 위한 것이다.

각 스테이션에서 매체 접근 제어를 위해 사용하는 타임드 토큰 프로토콜은 링의 각 스테이션에 균등하게 전송매체에 대한 접근 기회를 주며, 동기식 데이터를 제한 시간 내에 전송하기 위해 링의 교통량에 따라 비동기식 데이터를 조절하여 보내는 것이다. 즉, FDDI에서는 동기식 데이터에 더 높은 우선순위를 주는 것이다. 전송할 데이터 중 비동기식 데이터는 타겟 토큰 회전 시간(Target Token Rotation Timer: TTRT)을 이용하여 전송된다.

각 스테이션은 비동기식 데이터의 전송량을 조절하기 위해 토큰 보유 시간(Token Holding Timer: THT)과 정상 작동시 링 스케줄링을 조절하고 심각한 에러 발생 상황을 감지하고 회복하기 위해 토큰 회전 시간(Token Rotation Timer: TRT)을 사용한다. TRT의 값은 토큰의 연속적인 스테이션 도착 간 시간이다.

임의의 스테이션이 전송할 수 있는 데이터의 길이는 TRT와 TTRT의 차이에 달려 있다. 즉, 한 스테이션이 토큰을 잡았을 때에 그것의 THT 값은 TRT의 현재 값을 갖는다. 그 후 THT는 동기식 데이터를 보낼 때에는 작동이 되지 않고 있다가 비동기식 데이터를 보낼 때에 작동이 시작된다. 그 스테이션은 THT가 TTRT보다 작고 비동기식 데이터를 가지고 있는 한 계속 전송할 수 있다. 이러한 시간 제약 때문에 FDDI 프로토콜을 타임드 토큰 프로토콜이라 부른다.

비동기식 대역폭은 한정 토큰(restricted token)과 비한정 토큰(nonrestricted token)에 의해서 조절되어진다. 비한정 토큰의 경우 비동기식 대역폭은 여러 스테이션 사이에 공유되어지고, 한정 토큰의 경우는 특정 스테이션 사이에 독점되어진다. 또한 각 스테이션은 동기식 대역폭의 할당을 받는다.

본 연구에서는 비동기식 데이터의 비한정 토큰의 경우를 대상으로 모델링하였다.

3. 시뮬레이션 모델의 설계

3.1 근거리 통신망 모델링

네트워크에 연결된 모든 스테이션은 등간격으로 연결되어 있으며 모든 패킷의 발생분포는 각 근거리 통신망마다 다르며 하나의 근거리 통신망 내의 모든 스테이션에서는 균등하다고 가정한다. Ethernet, Token Ring과 FDDI의 SLAM II 네트워크 모델에서 공통되는 개체 즉, 패킷의 어트리뷰트는 다음과 같다.

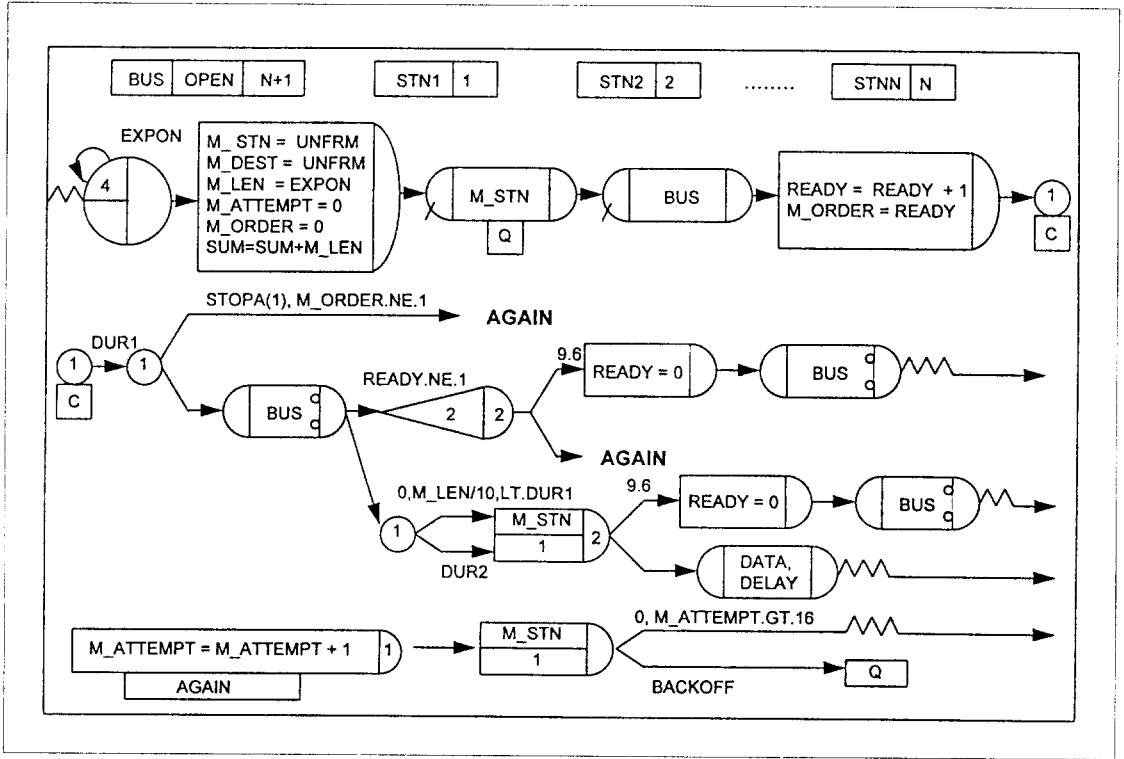
- M_STN : 패킷이 발생한 근원 스테이션의 일련번호로 일양분포로 발생시킨다.
- M_LEN : 패킷의 길이로 평균 패킷 길이에 대한 지수분포로 발생시킨다.
- M_ARRIV : 패킷의 도착시간으로 CREATE 노드에서 기록된다. CREATE 노드는 SLAM II에서 개체를

발생시키는 명령문으로 입력 매개변수인 패킷의 도착간 평균시간에 대한 지수분포로 패킷을 발생시키며 그 발생시각을 어트리뷰트에 저장할 수 있다.

- M_ADEST : 패킷의 목적지 스테이션의 일련번호로 일양분포로 발생시킨다.
 - M_ADDDATA : 각 근거리 통신망에서 발생하여 네트워크 내의 전송이 끝난 패킷의 길이의 합이다.
 - SUM : 전역변수로 각 근거리 통신망에서 발생하여 네트워크간의 전송이 끝난 패킷의 길이의 합이다.
- 각 근거리 통신망에서 공통되는 사용자 입력 매개변수는 다음과 같다.

- 근거리 통신망에 연결된 스테이션의 수
- 패킷의 평균 길이 : Byte단위로 입력받으며 실제 SLAM II 네트워크 모델에서는 bit단위로 바꾸어 사용한다.
- 패킷의 도착간 평균 시간 : ms단위로 입력받으며 한 스테이션에서 발생하는 패킷의 양을 나타낸 것이다. 그러나, SLAM II 네트워크 모델에서는 모든 스테이션에 대한 처리를 개별적으로 나누어 처리하지 않고 묶어서 처리하므로 패킷의 도착간 평균시간을 스테이션의 수로 나누어 패킷의 발생 빈도를 높여서 이 문제를 해결한다.
- 근거리 통신망의 케이블의 길이 : Ethernet의 최대 케이블 길이가 500m를 넘길 수 없으므로 입력단위로 km보다는 m로 선택하였다.

본 논문에서 개발한 도구를 통하여 얻고자 하는 통계자료로는 전송지연시간과 처리율을 들 수 있다. 전송지연시간은 패킷이 발생하여 완전히 전송이 끝나 목적지 스테이션에 도착할 때까지 걸리는 시간으로 각 네트워크 내의 전송지연시간, 네트워크 간의 전송지연시간과 이 둘을 모두 포함한 전체 네트워크의 전송지연시간과 같은 세 종류의 전송지연시간을 얻을 수 있다. 이 전송지연시간들의 평균과 이에 대한 표준 편차는 SLAM II에서 통계처리를 위해 여러가지 값을 수집하는 명령문인 COLCT 노드를 통해 얻는다. 각 네트워크 내에서 발생하여 목적지 스테이션까지 도달한 패킷의 길이의 합은 개체 어트리뷰트 M_ADDDATA에 더해지고, 네트워크 간의 전송된 패킷의 길이의 합은 전역 변수 SUM에 더해져서 COLCT 노드에서 구해진다. 이 두 종류의 패킷 길이의 합을 수행시간으로 나누면 각 네트워크 내의 처리율과 네트워크 간의 처리율을



〈그림 3〉 Ethernet의 SLAM II 네트워크 모델

구할 수 있으며 이들의 합으로 전체 네트워크의 처리율을 얻을 수 있다.

다음 〈그림 3〉과 〈그림 4〉는 Ethernet의 기본 시뮬레이션 모델과 Token Ring과 FDDI의 공통되는 기본 SLAM II 네트워크 모델을 주기능 중심으로 표현하였다.

3.2 Ethernet 모델링

〈그림 2〉는 2.2절에서 설명한 바와 같은 Ethernet 프로토콜을 수행하는 흐름도이며, 〈그림 3〉은 기본 SLAM II 모델로 3.1절에서 소개한 공통의 개체 어트리뷰트 외에도 다음과 같은 개체 어트리뷰트와 전역변수를 갖는다.

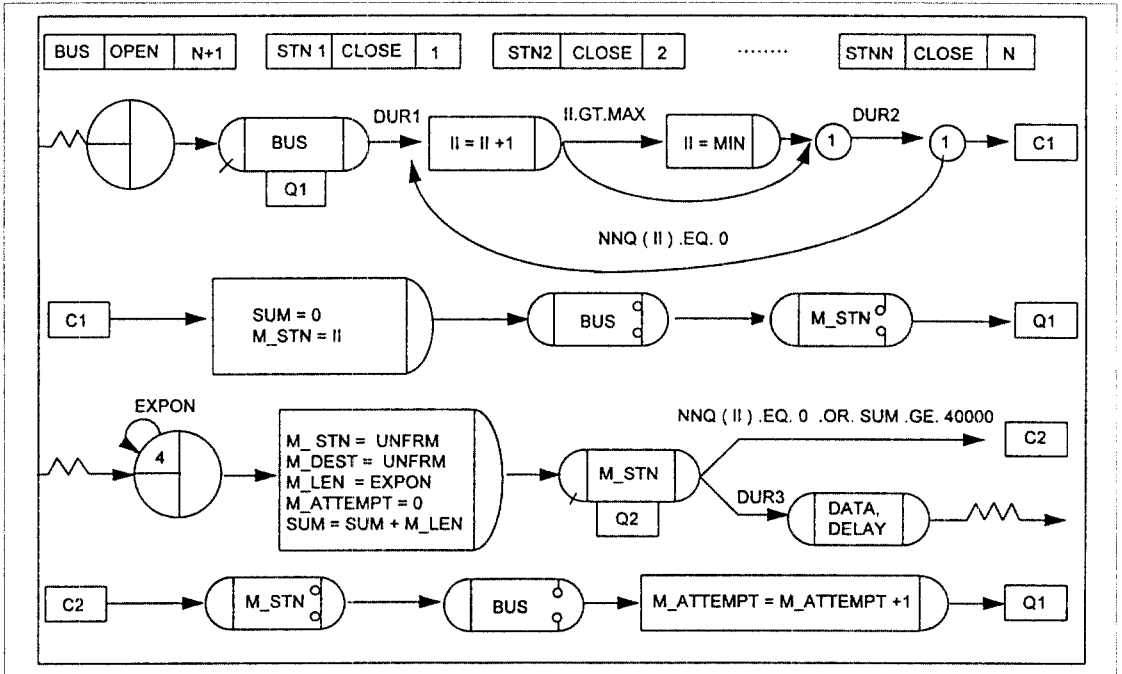
- M_ATTEMPT : 개체 어트리뷰트로 패킷 생성 후 전송을 시도했던 총 횟수로서 충돌이 발생할 때마다 1씩 전송회수가 증가한다.
- M_ORDER : 개체 어트리뷰트로 네트워크에서 전송을 시작한 패킷들 중 전송을 시작한 순서이다.

- READY : 전역변수로 네트워크 내에서 전송을 시도하는 패킷의 수이다.

Ethernet은 네트워크의 모든 스테이션을 관리하기 위해서 SLAM II에서 개체 흐름을 제어하기 위해 사용하는 GATE(BUS)를 사용하며 그 역할은 네트워크의 어떤 스테이션도 전송을 하고 있지 않다면 BUS를 열어 스테이션이 패킷의 전송을 시도할 수 있도록 하며, 네트워크에 패킷이 들어오면 BUS를 닫아 더 이상 패킷이 네트워크 내로 들어오지 못하게 한다. 또한 모든 스테이션은 하나의 RESOURCE(STN_i)를 가지며 STN_i는 어느 한 스테이션에서 한 번에 하나의 패킷만이 전송을 시도할 수 있도록 구현하기 위해 사용한다.

SLAM II Ethernet 모델의 작동원리에 대해 단계별로 나누어 설명하면 다음과 같다.

- 1단계 : CREATE 노드에서 발생한 패킷은 근원지 스테이션, M_STN을 일양분포로 얻고 목적지 스테이션 (M_DEST)을 M_STN과 같지 않도록 하여 일양분포



〈그림 4〉 Token Ring의 SLAM II 네트워크 모델링

로 얻는다. 패킷의 길이(M_LEN)는 네트워크의 허용 범위인 64 byte와 1518 byte 내에서 평균값의 지수분포로 얻는다. 그리고 이 패킷의 길이는 처리를 계산 위해 전역변수 SUM에 합쳐지고 M_ATTEMPT와 M_ORDER를 0으로 초기화한다.

- 2단계 : M_STN의 대기열 화일에서 RESOURCE를 얻은 화일 만이 다음 BUS의 대기열 화일에 들어가게 된다. RESOURCE를 얻지 못한 패킷은 자신의 스테이션의 RESOURCE가 FREE 즉, 해제되기를 기다린다.
- 3단계 : BUS의 대기열 화일에 대기하는 패킷은 BUS가 열리면 네트워크로 나가게 되므로 네트워크내의 패킷의 수(READY)가 1 증가하게 되고 자신의 전송 시작 순서인 M_ORDER는 READY의 값이 지정된다.
- 4단계 : 전송을 시작한 패킷이 있다는 사실이 모든 스테이션에 전파되는데 걸리는 시간(DUR1)이 지나면 BUS는 닫게 되고 이 때까지 네트워크에 들어온 패킷은 다음의 세가지 상태중 하나에 해당하게 된다.
 - ① M_ORDER = 1이고 READY = 1 인 상태

- ② M_ORDER = 1이고 READY ≠ 1 인 상태
- ③ M_ORDER ≠ 1이고 READY ≠ 1 인 상태

- 5단계 : 4단계 ①인 상태는 네트워크 내에서 하나의 패킷만 전송하려는 상태로 이 때는 성공적인 전송이 가능하며 전송이 끝나면 RESOURCE를 해제한다. 그 후 다음과 같은 두 가지의 시뮬레이션을 동시에 수행한다. 첫째는 프레임 간 차이(Interframe Gab) 9.6μs 후에 BUS를 열어 네트워크를 다시 전송가능한 상태로 만드는 작업을 수행하고, 둘째는 M_STN에서 M_DEST로 전파되는 시간(DUR2) 후에 COLCT 노드로 가서 데이터가 수집되고 종료되도록 한다.
- 6단계 : 4단계 ②인 상태는 충돌이 발생한 것이다. 충돌이 일어나면 다른 모든 스테이션에 충돌이 발생하였음을 알리는 신호를 보내야 하는데 충돌이 발생한 모든 패킷이 이 일을 하게 되면 복잡하므로 단순화하기 위해 이 상태의 패킷 하나만이 이 일을 전담하게 한다. 신호처리 후에는 STOPA에 의해 대기상태가 된 상태 ③의 패킷들의 지연을 EVENT 노드로서 해제하고 다음을 진행하도록 하고 다음과 같은 두 가지의

시뮬레이션을 동시에 수행한다. 첫째는 9.6 μ s 후에 BUS를 열어 네트워크를 다시 전송가능한 상태로 만드는 작업을 수행하고, 둘째는 8단계의 작업이다.

- 7단계 : 4단계 ③인 상태는 충돌 발생 후 충돌 신호 처리를 기다리는 상태로 상태 ②의 패킷이 EVENT 노드를 사용하여 해제하여 줄때까지 시간을 지연시킨 후에 8단계로 넘어간다.
- 8단계 : 이 단계는 충돌 발생 후 충돌횟수를 증가시키고 자신의 스테이션의 RESOURCE를 해제하여 다음 패킷이 전송가능하도록 하고, M_ATTEMPT가 16보다 크게 되면 더 이상의 전송시도를 멈추고 종료하며, 그렇지 않다면 Truncated Binary Exponential Backoff 지연을 한 후에 다시 M_STN으로 삽입된다.

Truncated Binary Exponential Backoff 지연의 계산은 FORTRAN으로 2.2에서 나와 있는 알고리즘대로 구현한 후에 SLAM II의 서브루틴으로 연결해서 사용하였다. 각 스테이션의 대기열 화일은 M_ATTEMPT \neq 0이 아닌 패킷에게 먼저 전송할 기회를 주기 위해 FIFO가 아닌 M_ATTEMPT의 값이 큰 패킷이 먼저 전송하도록 화일의 우선순위를 조정하였다.

3.3 Token Ring

이 모델링에서 Token Ring의 데이터 전송율은 4 Mbts/sec으로하고 토큰 보유시간은 10 msec으로 가정하였으며, <그림 4>는 2.2절에서 설명한 바와 같은 Token Ring 프로토콜을 수행하는 기본 SLAM II 모델로 3.1절에서 소개한 공통의 개체 어트리뷰트 외에도 다음과 같은 개체 어트리뷰트와 전역변수를 갖는다.

- M_ATTEMPT : 개체 어트리뷰트로 모델 구현에 있어서 개체가 빠져나온 다음에 토큰 보유시간 검사를 하므로 토큰 보유시간 이후에 나온 패킷을 다음 번 전송때 먼저 내보내기 위해 Ethernet 모델에서와 같이 화일의 우선순위를 M_ATTEMPT의 값이 큰 것으로 하였다.
- II : 전역변수로 토큰이 다음 스테이션으로 지나가도록 하기 위해 스테이션의 일련번호 중 최소값(MIN)과 최대값(MAX) 범위내의 값을 갖도록 한다. II의 값이 MAX를 넘어서면 II를 MIN으로 지정한다.

Ethernet에서와 같이 Token Ring도 네트워크의 모든 스

테이션을 관리하기 위해 SLAM II에서 제공하는 GATE (BUS)를 사용하며, BUS의 역할은 네트워크의 어떤 스테이션도 전송을 하고 있지 않다면 BUS를 열어 프리 토큰이 돌게하여 스테이션이 토큰을 잡을 수 있도록 하며, 스테이션이 토큰을 잡으면 BUS를 닫아 더 이상 패킷이 네트워크 내로 들어오지 못하게 하는 것이다. 또한 모든 스테이션은 하나의 GATE(STNi)를 가지며 STNi는 어느 한 스테이션이 토큰을 잡았을 때 토큰 보유시간 동안 패킷을 가지고 있으면 그 패킷을 전송할 수 있도록 구현하기 위해 사용한다.

Token Ring의 SLAM II 네트워크 모델은 토큰을 생성시키며 이 토큰에 따라 스테이션에 패킷 전송의 기회를 주는 토큰 제어부분과 실제 패킷을 생성하고 전송하는 패킷 제어부분으로 나누어 진다.

먼저 토큰 제어부분을 살펴보면, 토큰이 발생하면 토큰 전송시간(DUR1) 후에 마지막으로 전송을 했던 스테이션에서 토큰을 생성하여 토큰이 돌게 한다. SLAM II의 변수 II로 스테이션을 진행시키며 한 스테이션이 지날 때마다 두 스테이션 간의 전파시간과 1 비트 지연시간의 합(DUR2)의 시간이 지연되며 스테이션에 기다리는 패킷이 있으면 패킷을 전송할 수 있는 권한을 부여하기 위해 SLAM II의 GATE(BUS)를 닫고, 전송할 스테이션의 GATE(STNi)를 열어 둔다.

패킷 제어부분은 발생한 패킷은 STNi에서 전송할 차례를 기다리며 토큰 제어부분에서 이 GATE를 열어주면 패킷의 전송시간과 전파시간의 합(DUR3)동안 패킷을 전송한다. 전송할 패킷이 없거나 토큰 보유시간을 넘게되면 STNi를 닫고 BUS를 열어서 다음 스테이션에 토큰을 잡을 기회를 제공한다.

프리 토큰이 링을 돌 때에는 토큰 제어부분을 수행하고, 패킷을 전송하고자 할 때에는 토큰 제어부분은 기다리고 패킷 제어부분이 시뮬레이션을 진행시켜 나간다.

3.4 FDDI 모델링

FDDI도 기본적으로는 Token Ring의 프로토콜을 사용한다. 다시 말하면, FDDI는 Token Ring에서와 같이 비지 토큰이 돌아오기를 기다려서 토큰을 발생시키는 것이 아니라 전송을 끝낸 후 바로 토큰을 발생시킨다는 점을 제외하고는 프로토콜에 있어서 Token Ring과 유사하다. 따

라서 FDDI의 SLAM II 네트워크 모델은 <그림 4>의 Token Ring 모델과 유사하게 모델링하였다. 본 논문에서는 동기식과 비동기식 데이터 중 비동기식 데이터만 그 중에서도 비한정 토큰의 경우에 한하여 데이터를 전송한다고 가정한다.

중요한 시스템 매개변수인 타겟 토큰 회전 시간은 기본값을 8 msec로 하며 사용자 입력 매개변수로 하여 사용자가 원하는 값으로 할 수도 있다.

4. 근거리 통신망 모델의 통합

여러 개의 Ethernet과 Token Ring을 FDDI Backbone 네트워크에 연결하기 위해 3장에서 구현한 각각의 시뮬레이션 모델을 통합하여야 한다. 따라서 FDDI에 연결한 근거리 통신망들을 서로 구별하기 위해 일련번호가 필요하며 이를 나타내기 위한 개체 어트리뷰트는 M_LAN을 사용했다. 또한 네트워크를 연결해 주는 라우터의 구축과 그 밖에 추가되어야 할 사항이 있다.

4.1 라우터의 구축

이질의 네트워크간 통신을 위해 FDDI Backbone 네트워크의 스테이션은 라우터가 되며 하나의 라우터에 하나의 네트워크가 연결된다. 따라서 FDDI의 스테이션은 더 이상 패킷을 발생시킬 수 없으므로 이 라우터로 들어오는 패킷의 양은 각 네트워크의 스테이션에서 발생한 패킷 중 다른 네트워크로 전송될 패킷의 양에 영향을 받는다. 스테이션에서 발생한 패킷 중 네트워크 간 전송을 할 패킷의 비율은 사용자 인터페이스를 통해 입력받은 사용자 입력 매개변수이다. 이 매개변수는 각 LAN마다 서로 다른 값을 입력할 수 있으며 하나의 근거리 통신망 내의 모든 스테이션은 이 매개변수에 대하여 같은 값을 갖는다고 가정한다.

Token Ring이나 Ethernet에서 발생한 패킷은 비율에 따라 네트워크 간 전송을 할 패킷과 그렇지 않은 패킷으로 나뉘어지고 SLAM II 네트워크 모델에서 이러한 구분은 개체 어트리뷰트 M_LAN의 값으로 구별한다. 목적지 스테이션이 근원지 스테이션과 같은 네트워크 내에 있는 패킷이면 M_LAN을 0으로 지정하고, 그렇지 않은 패킷은 FDDI Backbone 네트워크에 연결된 LAN의 번호에 대한

일양분포로서 M_LAN의 값이 지정된다.

네트워크 간 전송을 하게 될 패킷 즉, M_LAN의 값이 0이 아닌 패킷의 목적지 스테이션의 값은 3단계에 걸쳐서 바꾸어 전송한다. 첫번째 단계에서는 이 패킷을 M_LAN = 0인 패킷과 동일한 방식으로 전송하도록 목적지 스테이션을 FDDI와 자신의 네트워크가 연결된 라우터로 하여 패킷의 근원지 스테이션에서 이 라우터로 전송을 하고 라우터의 대기열 화일에 패킷을 삽입한다. 2단계에서는 FDDI Backbone 네트워크 상에서의 전송을 위해 목적지 스테이션을 목적지 네트워크의 라우터로 설정하고 패킷의 길이가 FDDI에서 허용하는 범위를 초과하면 패킷 분할을 한다. 이 때에 패킷은 같은 크기의 패킷들로 분할된다고 가정한다. 두번째 단계의 기능은 FDDI를 통해 다른 LAN으로 전송되는 패킷을 처리하는 라우터의 중요한 기능이다. FDDI Backbone 네트워크를 통해 목적지 네트워크와 연결된 라우터에 도달한 패킷은 일단 대기열 화일에 삽입되며 라우터의 또다른 중요한 기능인 3단계 목적지 스테이션 설정 단계로 넘어간다. 3단계에서는 2단계에서 분할되었던 패킷이 일단 원래의 패킷으로 재결합되고 목적지 스테이션은 목적지 네트워크 내의 스테이션들의 일련 번호에 대한 일양분포로서 정해진다. 그리고 패킷의 길이가 목적지 LAN에서 허용하는 길이보다 크면 다시 패킷을 분할한 후 목적지 네트워크에서 M_LAN = 0인 패킷과 동일한 방식으로 패킷을 전송한다. 분할되었던 패킷이 목적지 스테이션에 도달하면 분할된 마지막 패킷이 도착할 때까지 기다렸다 재결합을 하게되며 이 때에 패킷 전송이 끝나게 된다.

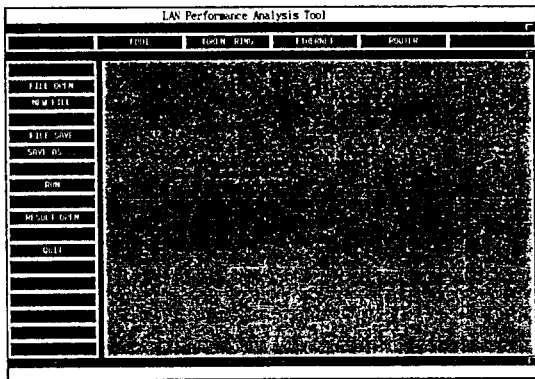
라우터에서 패킷을 처리하는데 소요되는 처리 시간은 사용자 입력 매개변수로서 기본 처리 CPU 시간, 분할 처리 CPU 시간과 재결합 처리 CPU 시간으로 나누어 인터페이스를 통하여 사용자로부터 입력된다.

라우터에서의 패킷 분할과 재결합 과정은 최대 허용 패킷의 길이가 1518 byte 인 Ethernet 에서 최대 허용 패킷의 길이가 4500 byte 인 FDDI를 통해 최대 허용 패킷 길이가 5000 byte인 Token Ring으로 패킷이 전송될 때에는 발생하지 않는다. 그러나 그 반대 방향으로의 전송에서는 이 과정이 발생할 수 있다. Token Ring의 한 스테이션에서 최대 허용 패킷 길이, 5000 byte 의 패킷이 네트워크 간 통신을 위해 발생되고 그 목적지 네트워크가 Ethernet 이며 분할시에는 같은 길이로 패킷을 사른다고 가정하자.

이 패킷은 Token Ring을 무사히 통과하여 라우터에 도달 하지만 FDDI의 최대 허용 패킷 길이인 4500 byte를 넘게 되어 패킷 그대로는 전송이 불가능하다. 따라서 이 패킷은 같은 길이인 2500 byte의 두 개의 패킷으로 분할되어 FDDI를 통과한 후 Ethernet과 연결된 라우터에서 분할된 두 개의 패킷이 도착하기를 기다려 분할된 패킷이 모두 도착하면 다시 합쳐져서 원래의 5000 byte 패킷이 된다. 이 패킷은 다시 Ethernet을 통과하기 위해서 Ethernet 최대 허용범위 1518 byte를 넘지 않도록 다시 분할되어 원래의 패킷은 1250 byte 패킷 네 개로 나뉘어 지게 된다. 분할된 패킷들이 Ethernet에서 최종 목적지 스테이션에 모두 도달하면 분할되었던 네 개의 패킷은 원래의 패킷 길이로 다시 합쳐져 모든 전송이 끝난다.

4.2 인터페이스 구축

<그림 5>는 사용자 인터페이스의 초기화면이다. 3장과



<그림 5> 사용자 인터페이스의 초기화면

4.1절까지에서 소개한 모든 사용자 입력매개변수와 그 외 여러 매개변수를 사용자가 보다 편리하게 사용할 수 있도록 UNIX에서 지원되는 OSF/MOTIF X-window 시스템을 이용하였다. X-window 시스템은 사용자가 마우스 등의 입력장치를 통하여 그래픽 화면 상의 아이콘이나 메뉴를 선택함으로써 의도하는 작업을 실행시킬 수 있는 그래픽 사용자 인터페이스 환경을 제공하는 소프트웨어 시스템이다 [4].

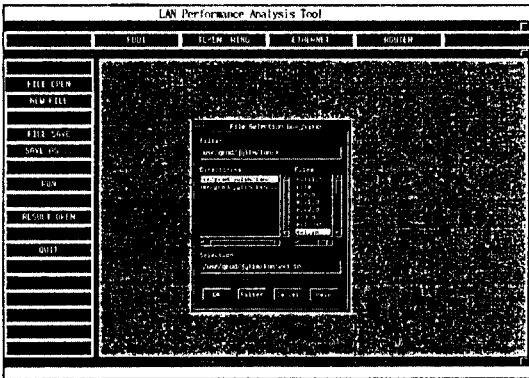
개발도구의 시스템에 관련된 메뉴의 기능은 다음과 같

다.

- FILE OPEN : FILE SELECTION BOX를 통해 선택된 화일에서 이전에 입력해 두었던 파라미터들을 읽어서 부분적으로 매개변수를 수정한 후 다시 시뮬레이션할 수 있도록 한다.
 - NEW FILE : 앞으로 입력하거나 지금까지 입력한 매개변수의 값이 입력한 새로운 화일 이름으로 저장된다.
 - FILE SAVE : 매개변수의 입력내용을 열려진 화일 이름으로 저장한다.
 - SAVE AS .. : 열려진 화일 이름이 아닌 다른 화일의 이름으로 저장하려고 할 때 사용한다.
 - RUN : 시뮬레이션 수행에 필요한 수행시간 등의 매개변수를 입력하면 열려진 화일의 매개변수 값 또는 열려진 화일이 없다면 지금까지 입력받은 매개변수 값을 바탕으로 시뮬레이션 모델을 구축하고 시뮬레이션을 수행한다.
 - Result Open : 수행이 끝난 화일의 결과를 보기 위한 것으로 열려진 화일이 있다면 그 화일의 결과 화일을 보여 주게 되고 열려진 화일이 없다면 결과 화일을 선택해서 볼 수도 있다.
 - EXIT : 이 개발도구의 종료가 이루어진다.
- 매개변수 입력에 관한 메뉴는 다음과 같다.
- FDDI : FDDI의 TTRT 값과 cable length를 입력받는다.
 - Token Ring, Ethernet : 먼저 FDDI에 연결할 Token Ring의 갯수/Ethernet의 갯수를 입력받고 각 Token Ring/Ethernet마다 스테이션의 수, 평균 패킷의 길이, 패킷의 도착간 평균 시간, 패킷의 네트워크간 전송 비율과 케이블 길이를 입력받는다. 이 때 두번째 Token Ring/Ethernet에 대한 매개변수를 입력받을 때는 입력할 매개변수가 이전 Token Ring/Ethernet의 매개변수와 동일한지를 물어본 후에 값이 다른 경우에만 다시 입력받도록 하여서 같은 내용의 반복 입력을 피하도록 하였다.(<그림 7> 참조)
 - ROUTER : 라우터의 기본 처리시간과 분할 및 재결합의 처리시간을 입력받으며 기본값은 1ms이다.

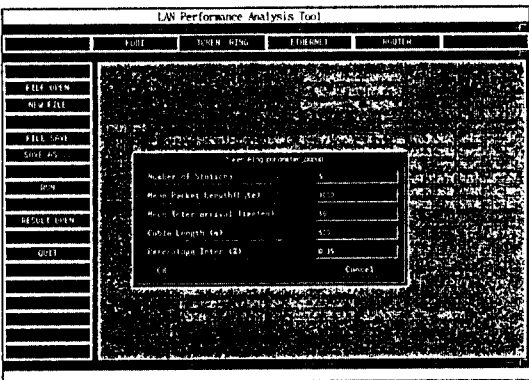
4.3 사용예

FDDI Backbone 네트워크에 세 개의 Token Ring과 세 개의 Ethernet을 연결한 간단한 네트워크 모델을 대상으로 하여 성능 분석과정을 보이고자 한다. 먼저 화일을 연 후(〈그림 6〉 참조), 모델링을 하기 위한 매개변수 메뉴를



〈그림 6〉 시스템 메뉴 - "FILE OPEN" 선택시의 화면

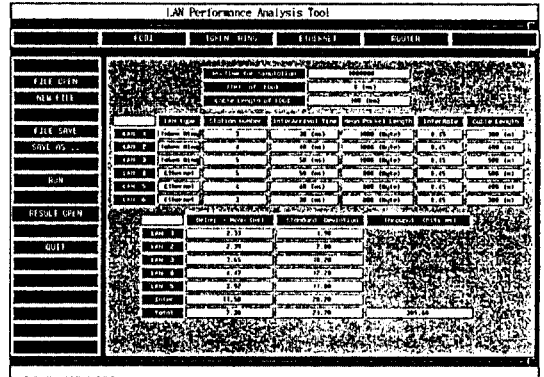
입력한다. FDDI의 매개변수를 먼저 입력하고 Token Ring이나 Ethernet의 매개변수를 입력한 후에 마지막으로 라우터의 매개변수를 입력함으로써 매개변수에 대한 입력을 끝낸다. 매개변수 입력과정은 〈그림 7〉과 같이 이루어지



〈그림 7〉 매개변수 메뉴 - "Token Ring"의 매개변수 입력화면

며 이 과정이 끝나면 원하는 모델의 구축을 할 수 있다. 이를 수행하기 위해서는 시스템 메뉴 중 RUN을 선택하

여 수행시간을 입력하고 수행이 끝날 때까지 기다려야 하는데 실제 수행하는데 걸리는 시간은 모델의 상태 즉, 스테이션의 수가 얼마나 많은가 혹은 패킷이 얼마나 자주 발생 하는가 등의 상태에 따라 큰 차이가 난다. 수행이 끝나 Result Open을 선택하면 〈그림 8〉과 같은 결과 화면이



〈그림 8〉 시스템 메뉴 - "RESULT OPEN" 선택시의 화면

나타나는데 이 화면은 사용자 입력 매개변수와 이에 대한 결과값으로 전송 지연 시간의 평균값과 표준 편차, 그리고 이 네트워크의 처리율을 보여준다.

사용자 인터페이스로 입력된 시뮬레이션 시스템 매개변수 값은 화일에 저장해두며 이러한 값을 기초로 SLAM II의 모델을 구축한 후에 수행시간 등을 바꾸어 다시 수행하거나 일부 매개변수를 다른 값으로 변경하여 다른 SLAM II 모델을 구축한 후 결과를 서로 비교할 수 있다.

5. 결 론

본 논문에서 구현한 도구는 여러 이질의 근거리 통신망을 FDDI Backbone 네트워크에 연결한 네트워크 모델의 성능을 평가하기 위하여 대상 통신망인 FDDI, Token Ring과 Ethernet의 기본 시뮬레이션 모델을 객체화하여 모델의 구축을 보다 용이하게 하였다. 또한 네트워크 성능을 평가할 수 있는 패키지들은 많은 매개변수로 구성되어 복잡하여 네트워크를 평가하고 싶은 초심자에게는 상당히 부담을 준다. 그래서 사용자가 입력하는 매개변수도 가장 중요한 것들로 단순화시키고 결과도 네트워크 성능 평가에 있어 가장 중요한 전송지연시간과 처리율 중심으로 제

공하는 이 성능평가 도구는 누구나 쉽게 사용할 수 있다는 잇점을 갖고 있다.

개발된 도구는 여러 근거리 통신망들을 함께 연결하여야 하는 실제 상황에서 연결하고자 하는 방법이 성능면에서 어떻게 작동할 것인지를 미리 예측해 보기 위한 도구로 사용될 수 있으며, 이 도구의 타당성은 기존에 이미 찾아진 많은 수학적 모델과의 비교를 통하여 이루어 질 수 있다.

본 논문에서 사용된 기법을 여러 종류의 인터넷워킹 상황에 적용할 수 있다. 예를 들면 차세대 통신망이 될 광대역 종합 정보 통신망(B-ISDN)에 여러 종류의 다른 네트워크들을 연동시키는 상황에 대한 모델링에 적용이 가능하다.

참고문헌

- [1] 이영애, 백두권, 「시스템 시뮬레이션」, 2판, 정문사, 1993.
- [2] 「근거리망 - CSMA/CD 접근법 및 물리 계층 표준」, 한국통신 기술협회, 1992.
- [3] ANSI Standard X3.139, "Fiber Distributed Data Interface (FDDI) - Token Ring media access control (MAC)", 1987.
- [4] Douglas, A. Y., The X window System: Programming and applications with Xt OSF/Motif Edition, Prentice Hall, 1990.
- [5] Dykeman, D. and Bux, W., "An investigation of the FDDI media access control protocol", in Proc. 5th European Fibre Opt. Commun. Local Area Networks Conf., Basel, Switzerland, pp. 229-236, June 1987.
- [6] Dykeman, D. and Bux, W., "Analysis and tuning of the FDDI media access control protocol", IEEE J. Select. Areas Commun, vol. 6, no. 6, pp. 907-1010, July 1988.
- [7] Goyal, A. and Dias, D., "Performance of priority protocols on high-speed token ring networks", in Proc. 3rd Int. Conf. Data Commun. Syst. Perform, Rio de Janeiro, Brazil, pp. 25-34, June 1987.
- [8] Green, L., "Performance analysis of FDDI", COMPCOM Sprint '87 32nd IEEE Comput. Soc. Int. Conf. San Francisco, CA, pp. 441-443, 1987.
- [9] Grow, R. M., "A timed-token protocol for local area networks", presented at Electro '82, Token Access Protocols, Paper 17/3, May 1982.
- [10] Johnson, M. J., "Proof that timing requirements of the FDDI token ring protocol are satisfied", IEEE Trans. Commun, vol. COM-35, pp. 620-625, June 1987.
- [11] Johnson, M. J., "Reliability mechanisms of the FDDI high bandwidth token ring protocol", Comput. Networks ISDN Syst., vol. 11, pp. 121-131, Feb. 1986.
- [12] "Local Area Network : Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access model and physical layer specifications" IEEE Standards 802.3, 1992.
- [13] "Local Area Networks: Token Ring access method and physical layer specifications", IEEE Standards 802.5, 1989.
- [14] "NETWORK II.5 User's Manual Release 7.0", CACI Products Company, 1991.
- [15] Sevcik, K. C. and Johnson, M. J., "Cycle time properties of the FDDI token ring protocol", IEEE Trans. Software Eng., vol. SE-13, pp. 376-385, Mar. 1987.
- [16] Ulm, J. M., "A timed token ring local area network and its performance characteristics", in proc. 7th Conf. Local Comput. Networks, Minneapolis, IEEE, pp. 50-56, Feb. 1982.

● 저자소개 ●



채기준

1982년 2월 연세 대학교 수학과 학사

1984년 5월 미국 Syracuse Univ. 전산과학 석사

1990년 5월 미국 North Carolina State Univ. 컴퓨터 공학 박사

1990년 8월 ~ 1992년 2월 미국 해군사관학교 전자계산학과 조교수

1992년 3월 ~ 현재 이화여자대학교 전자계산학과 조교수

관심분야 : 컴퓨터 통신 및 네트워크, 성능평가



임지영

1994년 2월 이화여자대학교 전자계산학과 이학사

1994년 3월 ~ 현재 이화여자대학교 전자계산학과 석사과정