

MVA 알고리즘을 이용한 ATM 기반 통합 서비스 교환기 내 워크스테이션의 성능 평가

(Performance Evaluation of Workstation System within ATM Integrated Service Switching System using Mean Value Analysis Algorithm)

장 승 주 [†] 김 길 용 ^{**} 이 재 흠 ^{***} 박 호 진 ^{****}
 (Seung-Ju Jang)(Gil-Yong Kim)(Jae-Hum Lee)(Ho-Jin Park)

요 약 현재 ATM(Asynchronous Transfer Mode) 기반 통합 서비스 교환 시스템 개발 추세는 복합 교환 시스템에서의 운용, 보전 기능 및 ATM 기반 인터넷/협대역 통합 서비스 등을 수용하기 위하여 워크스테이션에 멀티플러그-인 개념을 도입하여 시스템이 설계, 구현되고 있다. 한편, 기존 워크스테이션에서는 교환 시스템을 운용하기 위해 HMI(Human Machine Interface) 운용 시스템 기능과 함께 파일 관리 기능, 시간 관리 기능, 그래픽 처리 기능 및 TMN(Telecommunication Management Network)의 에이전트 기능 등이 설정되어 교환 시스템과의 많은 인터페이스를 통하여 기능들이 수행되고 있다. 이러한 기능들은 프로세스 혹은 프로세서간 많은 메시지 통신을 필요로 하고 있으며, 워크스테이션내 입출력 장치 파일, 일반 파일, 그리고 메시지 전송을 위한 메시지 큐, 소켓 자원 등을 사용하고 있다. 따라서, 현재 개발 중인 시스템의 정상 동작 여부를 확인하고 파악된 문제점들에 대한 성능 개선과 함께 추후 지속적인 기능 추가 시에도 시스템의 정상 동작을 보장하기 위해서는 사전에 개발될 교환 시스템에 대한 성능 평가가 선행되어야 한다. 본 논문은 이러한 ATM 기반 통합 서비스 교환 시스템 개발 작업에서 사전에 시스템의 성능 평가를 했다. 성능 평가는 MVA(Mean Value Analysis) 알고리즘을 이용한다. 모델은 큐잉 네트워크 모델을 사용하고 모델링된 시스템을 이용하여 PDQ(Pretty Damn Quick) 시뮬레이션 패키지와 C 언어를 사용하여 시뮬레이션 하였다.

Abstract In present, ATM integrated switching system has been developed to a mixed modules that complexed switching system including maintenance, operation based on B-ISDN/LAN service and plug-in module, , which runs on workstation computer system. Meanwhile, workstation has HMI operation system feature including file system management, time management, graphic processing, TMN agent function. The workstation has communicated with between ATM switching module and clients. This computer system architecture has much burden messages communication among processes or processor. These messages communication consume system resources which are socket, message queue, IO device files, regular files, and so on. Therefore, in this paper we proposed new performance modeling with this system architecture. We will analyze the system bottleneck and improve system performance. In addition, in the future, the system has many additional features should be migrated to workstation system, we need previously to evaluate system bottleneck and redesign it. In performance model, we use queuing network model and the simulation package is used PDQ and C-program.

· 본 논문은 1999년 ETRI 운용 S/W개발과제의 지원을 받았음.

[†] 정 회 원 : 동의대학교 컴퓨터공학과 교수
 sjjang@hyomin.donggeui.ac.kr

^{**} 종신회원 : 부산대학교 컴퓨터공학과 교수
 gykim@hyowon.pusan.ac.kr

^{***} 종신회원 : 한국전자통신연구원 ATM운용팀 연구원
 ljh@etri.re.kr

^{****} 비 회 원 : 한국전자통신연구원 ATM운용팀 연구원
 hjpark@etri.re.kr

논문접수 : 1999년 11월 9일
 심사완료 : 2000년 5월 23일

1. 서론

컴퓨터 관련 기술의 급속한 발전에 힘입어 컴퓨터 상의 응용 프로그램들도 독자적인 처리 방식에서 분산 처리 방식으로 변화되어 가고 있으며, 사용자의 요구를 수용하기 위해 멀티미디어 데이터와 같은 대용량 데이터 전송이 통신망을 통해 발생하게 된다. 통신망을 매개로 컴퓨터가 다른 컴퓨터의 파일이나 주변 기기에 고속으로 접근하기 위해서는 보다 고속화되고 확장성이 높은 통신망이 필요하기 때문에, 컴퓨터의 고성능화와 멀티미디어 정보의 실시간성, 연속성은 새로운 통신망에 대한 요구 조건이 되고 있다. 따라서, 앞으로 구축되어야 할 초고속 정보 통신망은 고속성, 확장성, 유연성, 멀티미디어 정보의 전송이라는 네 가지 요구 조건을 만족해야 한다.

현재 이러한 요구 조건들을 만족하면서 기존의 회선 교환과 패킷 교환의 특성을 모두 수용하는 대표적인 통신망 기술로서 ATM(Asynchronous Transfer Mode) 기술을 들 수 있다[3]. ATM 기술에 관한 표준화 작업은 주로 ITU-T(International Telecommunication Union Telecommunication Standardization Sector)에서 진행되어 오다가 최근 국제적인 비영리 단체인 ATM Forum이 발족되면서 더욱 빠른 속도로 추진되고 있다[3].

멀티미디어 서비스를 지원하기 위해 기존의 네트워크가 제공하는 것보다 더 큰 대역폭이 요구됨에 따라 광대역 통신망이 필요하게 되었다. ATM은 비디오 데이터, 오디오와 같은 서로 다른 종류의 트래픽을 다룰 수 있어서 미래의 광대역 통신인 B-ISDN(Broadband-Integrated Service Digital Network)의 중요한 기술이다. 그리고 이동통신은 무선망을 이용하여 목소리와 데이터 전송을 가능하게 하고 있다. 이처럼 ATM 망이 차세대 네트워크 망으로 중요한 위치를 차지하고 있다 [1,2,5].

본 논문은 ATM 통합 서비스 교환기 시스템에 대한 성능 모델을 정립하고 성능 평가를 수행함으로써 워크스테이션 자원들의 효율적인 사용 방안을 제시하는 것이다. 이를 위해 성능 평가 수행 결과를 토대로 시스템 자원의 한계 및 성능 부족으로 인해 발생될 문제점들을 사전에 탐지하고, 이러한 문제점들이 발생하지 않도록 대책을 수립할 수 있도록 지침을 제공한다. 본 논문은 ATM 통합 서비스 교환기 시스템에 대한 다음과 같은 성능 모델을 분석 및 방안을 수립한다 [1,3,4].

-큐잉 모델을 사용한 교환 시스템과 워크스테이션간

메시지 통신 처리 성능 분석

-HMI(Human Machine Interface) 운용 시스템 내에서 클라이언트-서버간 메시지 처리 성능 분석

-워크스테이션내에서 프로세스간 메시지 통신 성능 분석

-워크스테이션내의 입출력 장치와 타임 스케줄링에 대해 성능을 분석하고 문제점 발견시 대처 방안 수립

-워크스테이션내에서 병목 현상을 유발시키는 부분을 추출하여 대처 방안 수립

본 논문에서 제안하는 시스템의 성능을 분석하기 위한 방안으로 큐잉 네트워크 모델을 이용한다. 큐잉 네트워크 모델 중 반응 시간이 $R_Q(N) = E[S_Q] \{ \sum_{i=0}^{N-1} P_Q(i-1|N-1) \}$ 인 MVA(Mean Value Analysis) 알고리즘을 이용한다[6, 7, 8, 9]. 큐잉 네트워크 모델을 설정한 후에 이 네트워크 모델을 이용하여 시뮬레이션을 한다. 시뮬레이션 프로그램 환경으로 solaris 운영체제에 C 프로그램을 이용한다. 그리고 큐잉 네트워크 모델을 쉽게 프로그램할 수 있는 PDQ(Pretty Damn Quick) 시뮬레이션 패키지를 이용한다 [7,8]. ATM 통합 서비스 시스템의 성능 분석 작업을 통해서 현재 시스템의 문제점을 발견 및 해결하고 앞으로 많은 기능들이 워크스테이션 시스템으로 이주할 전망이다 이 기능들의 이주로 인한 성능 부하 정도를 미리 예측함으로써 시스템 구조 설계에 대한 새로운 방안을 찾고자 한다.

본 논문의 구성으로 2장은 성능 모델, 3장은 시뮬레이션, 4장 성능 분석, 5장 결론의 순으로 언급한다.

2. 성능 모델

2.1 ATM 기반 통합 서비스 시스템 구조

복합 교환 시스템을 개발하는데 있어서 복합 교환 시스템에서의 운용, 보전 기능 및 ATM 기반 인터넷/협대역 통합 서비스 등을 수용하기 위하여 워크스테이션에 멀티 플러그인 개념으로 설계 및 구현되고 있다. 또한 모듈화된 기능들 중 실시간성이 요구되지 않는 기능들은 점차적으로 교환 시스템으로부터 워크스테이션으로 기능이 이주될 전망이다.

기존 워크스테이션에는 교환 시스템을 운용하기 위한 HMI(Human Machine Interface)운용시스템 기능, 파일 관리 기능, 시각 관리 기능, 그래픽을 처리하기 위한 기능 및 TMN(Telecommunication Management Network)의 에이전트 기능 등이 수행되고 있다. 이들은 프로세스 혹은 프로세서 간에 많은 메시지 통신을 필요로 하고 있으며 워크스테이션 내 IO 디바이스, 파일, 그리고 메시지 전송을 위한 메시지 큐, 소켓 자원 등을 사용

하고 있기 때문에 현재 기능 및 계속적으로 추가되는 기능들을 정상적으로 수행하기 위한 성능 평가가 요구된다 [11,12,13,14].

ATM 기반 통합 서비스 교환 시스템을 개발하는데 있어서 교환 시스템 내에 OMP로부터 모듈화된 기능들 중 실시간성이 요구되지 않는 기능들은 점차적으로 워크스테이션으로 이주될 전망이다. 그러나 기능 개발자들은 각자 기능 단위의 효율적인 설계는 하지 못하고 워크스테이션 시스템에 대한 고려를 별로 하고 있지 못하다. 그로 인하여 특정 워크스테이션의 자원에서는 과부하로 인하여 비정상 동작의 원인을 제공하게 되어 전체적으로 시스템에 문제를 유발시키게 된다. 다음 그림 1은 ATM 기반 통합 서비스 교환기 시스템 구조에서 사용자들이 클라이언트 단말기를 통해서 접근을 하고 ATM switch의 기능이 워크스테이션 서버에서 동작하는 형태를 나타낸다.

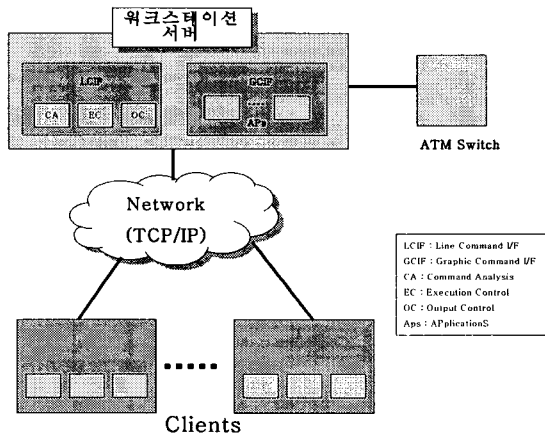


그림 1 ATM 기반 통합 서비스 교환기의 구조

2.2 MVA 알고리즘을 이용한 성능 평가

컴퓨터 시스템 성능 평가 방법은 여러 가지가 있다 [6,9]. 일반적으로 가장 많이 이용되는 방법이 큐잉 네트워크 모델이다. 큐잉 네트워크 모델도 다시 세분화하면 여러 가지 기법들이 있다 [9,10]. 본 논문은 이러한 여러 가지 큐잉 서비스 정책 중에 FCFS(First Come First Server)형태이면서 네트워크 형태가 개방형 큐잉 네트워크 모델(Open Queuing Network Model) 형식을 취한다. 본 논문에서 평가하고자 하는 시스템 구조에 대하여 큐잉 네트워크 모델을 개념을 적용하여 도식화하면 다음 [그림 2]와 같이 나타낸다. [그림 2]는 앞에서

제시한 ATM switch와 워크스테이션, 클라이언트 시스템 간의 관계 구조에 따라 만들어진다.

[그림 2]에서 ATM switch의 HANDF 모듈에서부터 출발점을 잡았다. Srosd 큐잉 센터부터 ct 큐잉 센터 전까지는 워크스테이션 내에 있는 기능이다. Ct 큐잉 센터부터 그 이하 구조는 클라이언트 시스템의 기능이다.

일반적인 single chain network 성능 평가로 많이 사용하는 것이 Convolution algorithm, MVA(Mean Value Analysis), LBANC(Local Balance Algorithm for Normalizing Constants)이다. 일반적인 경우 널리 사용되는 알고리즘이 Convolution 알고리즘이고 MVA, LBANC 알고리즘은 특수한 상황, single server, fixed capacity service center에서 우수한 성능을 발휘한다고 알려져 있다 [9,10,11].

본 논문에서 사용하고자 하는 큐잉 네트워크 모델은 일반적으로 많이 사용하는 MVA 모델을 이용한다. MVA 알고리즘은 convolution 알고리즘의 수치적인 약점을 해결해 준다. Convolution 알고리즘의 수치적인 약점은 각 센터 마다 고객(customer)의 수와 네트워크 센터의 수 만큼 반복적으로 계산이 일어난다는 것이다. MVA알고리즘이 이러한 convolution 알고리즘의 약점을 해소할 수 있는 것은 수식의 상수 normalizing을 피하기 때문이다. 성능 평가 모델로 설정한 환경은 폐쇄형 큐잉 네트워크 모델, 개방형 큐잉 네트워크 모델이 있다. 이 두가지 방법 중에 본 논문에서는 컴퓨터 시스템의 동작 방식이 일반적으로 개방형 큐잉 네트워크 모델(open queueing network model) 구조를 갖기 때문에 이 모델을 이용하여 성능 평가를 수행한다.

MVA 알고리즘은 네트워크에 대해서 서비스 센터이고 하나의 서버에 제한된 용량의 센터인 single server fixed capacity center로 가정한다. 서비스 센터에 대한 response time은 수식 (1)과 같이 표현된다(j=1, , J). 아래의 수식(1) ~ (13)은 MVA 알고리즘의 일반적인 표현식에 의하여 도출 가능하다[6,7,8]. 수식(1)에서 (13)까지 공통적으로 j는 서비스 센터이고, S(j)는 j 서비스 센터에 대한 service demand를 나타낸다. 그리고 E[X]는 평균값을 나타내고, u(j)는 n 고객을 갖는 센터에서 서비스 용량을 나타내며, L(j)(N)는 평균 큐의 길이이다. 또한 R(j)(N)는 평균 반응시간이다. i는 j가 아닌 어떤 서비스 센터이고 수식 (9) 는 center j가 없어진 네트워크 내의 center i에 대한 throughput이다.

$$R_i^{(N)} = E[S_i] / \mu_i^{(i)}, \quad N > 0. \quad (1)$$

single server fixed capacity center에 대한 response

time은 다음과 같이 표현된다.

$$R_{(j)}(N) = \frac{E[S_{(j)}]}{\mu_{(j)}(1)} (1 + L_{(j)}(N-1)), \quad N > 0. \quad (2)$$

우리는 recursive 함수를 사용하는 대신 iteration 함수를 이용하여 알고리즘을 표현한다. 먼저 우리는 어떤 큐에서 customer arrival time간의 평균 시간에 대한 Little's formula의 적용을 통하여 throughput을 얻는다.

$$\Lambda_{(j)}(N) = \frac{N}{\sum_{i=1}^J (y_{(i)} / y_{(j)}) R_{(i)}(N)}, \quad j = 1, \dots, J. \quad (3)$$

다시 Little's formula를 적용하여 fixed capacity center에 대한 평균 큐 길이를 얻는다.

$$L_{(j)}(N) = \Lambda_{(j)}(N) R_{(j)}(N), \quad j = 1, \dots, J. \quad (4)$$

위의 식을 이용하여 다음 단계의 계산을 반복적으로 수행한다.

general capacity 함수를 갖는 서비스 센터에 대한 response time을 구하는 함수는 다음과 같다.

$$R_{(j)}(N) = E[S_{(j)}] \sum_{i=1}^N \frac{i}{\mu_{(j)}(i)} P_{(j)}(i-1 | N-1), \quad N > 0. \quad (5)$$

큐 길이에 대한 분포는 다음의 함수로 표현할 수 있다.

$$P_{(j)}(i | N) = \frac{E[S_{(j)}] \Lambda_{(j)}(N)}{\mu_{(j)}(i)} P_{(j)}(i-1 | N-1), \quad i = 1, \dots, N. \quad (6)$$

위의 수식에서 계산량의 복잡도에 따른 문제가 발생한다. 그리고 아주 작은 수 N에 대해서 계산하는 것이 어려울 수 있다. 따라서 다음의 확률 함수를 새로운 함수로 대체한다.

$$P_{(j)}(0 | N) = 1 - \sum_{i=1}^N P_{(j)}(i | N), \quad (7)$$

$$P_{(j)}(0 | N) = \frac{\Lambda_{(j)}(N)}{\Lambda_{(j)}(N)} P_{(j)}(0 | N-1), \quad (8)$$

$$\Lambda_{(j)}^{J-(j)}(N) \quad (9)$$

$$\Lambda_{(j)}^{J-(j-1)}(N) \quad (10)$$

$$\Lambda_{(j)}^{J-(j-1)}(N) \quad (11)$$

$$\Lambda_{(j)}^{J-(j)}(N) \quad (12)$$

$$\Lambda_{(j)}^{J-(j)}(N) \quad (13)$$

수식 (9)을 계산하는 비용은 수식(10)을 계산하는 비용과 비슷하다. 주어진 서비스 센터에 대해서 하나는 single server fixed capacity와 IS center를 갖는 general capacity 함수 서비스 센터를 갖는 네트워크 문제를 해결하는 것에 비하여 두배의 계산 비용이 든다. 네트워크가 J-1과 J 두 개의 general capacity center를 갖는 경우 수식(11), 수식(12)와 수식(13)을 계산하는 것이 필요하다. 따라서 이 계산은 single server fixed capacity와 IS center를 갖는 네트워크 계산에 비해서

비용이 4배 정도 든다. 좀더 체계적인 설명을 위하여 다음과 같이 정리를 한다. 서비스 센터 1, ..., J는 IS 서비스 센터이고 서비스 센터 J'+1, ..., J''는 single server fixed capacity service center이고 서비스 센터 J''+1, ..., J는 general capacity 함수이다. T는 weight sum response time을 계산하는 변수이다. 다음은 MVA 알고리즘을 설명한다.

```

{Mean Value Analysis}
{Initialization}
  For j = J'+1 to J''
    L(j) = 0
  For j = J''+1 to J
    R(j)(0) = 1
{Number of customers loop}
  For n = 1 to N
    {Mean response times}
    T = 0
    For j = 1 to J' {IS service centers}
      R(j) = E[S(j)] / μ(j)(1)
      T = T + y(j) × R(j)
    For j = J''+1 to J {General capacity service centers}
      R(j) = 0
      For i = 1 to n {Iterate over queue lengths}
        R(j) = R(j) + i × P(j)(i-1) / μ(j)(i)
      R(j) = E[S(j)] × R(j)
      T = T + y(j) × R(j)
    {End mean response times}
  {Throughputs}
  T = n / T
  For j = 1 to J
    Λ(j) = y(j) × T
  {Mean queue lengths = fixed capacity service centers}
  For j = J'+1 to J''
    L(j) = Λ(j) × R(j)
  {Queue length distributions - general capacity service centers}
  For j = J''+1 to J
    For i = n down to 1 {Iterate over queue lengths}
      P(j)(i) = E[S(j)] × Λ(j) × P(j)(i-1) / μ(j)(i)
    P(j)(0) = P(j)(0) × Λ(j) / Λ(j)J-(j)(n)
  {End number of customers loop}
  {Mean queue lengths not already determined}
  For j = 1 to J', J'' to J
    L(j) = Λ(j) × R(j)
  {Utilizations}
  
```

ATM switch와 워크스테이션 서버 간에 데이터 흐름에 대한 프로토콜 규약에 따라 전체 시스템에서 중요한 요소들을 추출한 결과 MMC(Man Machine Communication) 명령 및 시스템 출력 메시지 기능 모듈이 중요한 부분으로 이 모듈을 중심으로 한 큐잉 모델을 정립한다. 그리고 이 모듈을 중심으로 한 주변 요소들에 대해서 메시지 발생 빈도가 많은 부분이나 메시지가 발생할 때 대량으로 발생(bulk message)하는 부분들을 추가한다. 이러한 원칙에 의하 HANDF로부터 srosd라는 데몬 프로세스를 통해서 워크스테이션과의 인터페이스가 발생된다. HANDF로부터 발생하는 모든 메시지는 srosd가 받아서 처리한다. srosd 데몬 프로세스는 oc를 통해서 log file에 저장을 한 후, print spool에 저장을 하고 나서 서버에게 메시지를 보낸다. 이 과정은 순차적으로 발생한다. client에 출력할 메시지를 받은 server는 hcmd 프로세스를 통해서 화면에 해당 메시지를 출력하게 된다. 이때 1메시지의 크기는 200바이트이고 실제 패킷의 크기는 500바이트 단위로 전송이 일어난다. srosd를 통해서 자주 발생하지는 않지만 가끔 발생하게 되는 그래픽관련 메시지는 sa를 통해서 server로 전송된다. 메시지를 받은 server는 client에게 전달하게 된다. 또 다른 기능으로 srosd를 통해서 hmilpd 데몬 프로세스를

통해서 워크스테이션 프린터로 출력을 하는 기능이다. 이것은 hmilpd 프로세스를 통해서 출력하게 된다.

큐잉 모델을 이용한 성능 평가에서 정확성을 기하기 위하여 실측 데이터를 구할수 있는 부분은 실측 데이터를 이용한다. 실제 ATM switch에서 최종 client까지 메시지 전송에 소요되는 시간은 '30메시지/1초'이다. 이 데이터를 기준으로 하여 시스템 성능 평가 모델을 정립한다.

[그림 2]에서 각 센터의 의미는 다음 [표 1]과 같다.

표 1 [그림 2]에서 각 센터 설명

센터	설명
HANDF	ATM교환기(HANBit ACE) 시스템에 위치하여, 교환기내 기능들과 워크스테이션에 위치하는 HMI(Human Machine Interface)운용시스템과의 인터페이스를 담당.
HANDF	ATM교환기(HANBit ACE) 시스템에 위치하여, 교환기내 기능들과 워크스테이션에 위치하는 HMI(Human Machine Interface)운용시스템과의 인터페이스를 담당.
srosd	워크스테이션에 위치하여, 교환기에서 오는 메시지들을 워크스테이션내 기능으로 메시지를 전달하는 데몬.
hmilpd	HMI 운용시스템에서의 프린터 출력 데몬
oc	교환기로부터 오는 메시지에 대하여 화면 및 프린터로 출력형태를 갖는 메시지를 구성하는 기능
sa	교환기 형상 및 장애상태를 그래픽으로 나타내기 위한 데이터를 관리하고, 그래픽 창으로 장애상태를 알려주는 기능
sv	HMI 운용시스템의 서버의 메인 프로그램으로 서버관련 프로세스들을 관리하며, 클라이언트(그래픽 창)과의 인터페이스, 메시지 전달 및 분배기능을 수행
ct	서버와의 메시지 송/수신을 하며 수신된 메시지들을 클라이언트 관련 프로세스로 메시지 분배기능을 수행
hgraphic	그래픽 창으로 교환기의 형상, 장애상태등을 그래픽 처리하는 기능
hcmd	텍스트 위주의 명령 입출력 창으로 운용자에게 교환기에 명령을 부여할 수 있게하고, 명령에 대한 결과 및 교환기로부터 상태/장애/경보 메시지를 받아 출력한다.

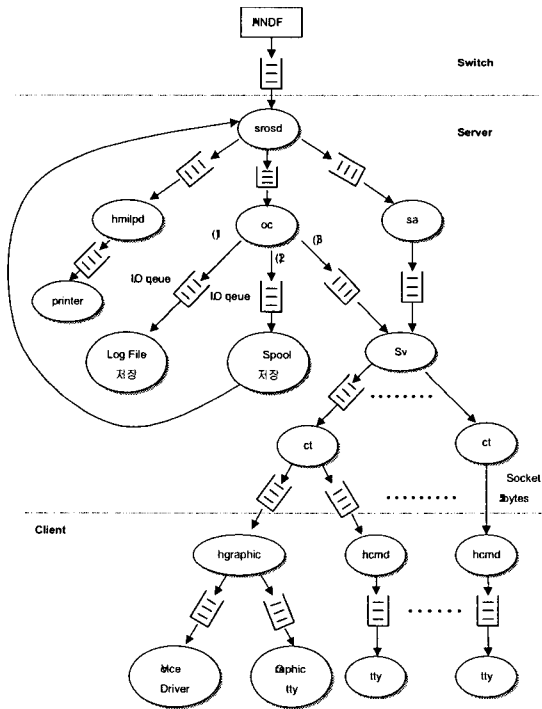


그림 2 성능 평가를 위한 큐잉 네트워크 모델

3. 시뮬레이션

시뮬레이션 방법은 여러 가지가 있다. 일반적인 시뮬

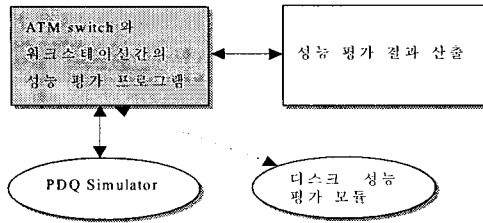


그림 3 성능 평가를 위한 시뮬레이션 환경

레이션 방법으로 많이 사용하는 것이 성능 평가 모델을 이용한 구체적인 프로그램을 작성하는 것이다. 프로그램 작성 방법은 여러 가지가 있다. 일반적으로 성능 평가 모델을 라이브러리 형태로 제공하는 시뮬레이션 언어를 이용한다. 본 논문은 이러한 시뮬레이션 언어 중에 본 논문에서 설정한 성능 평가 모델을 가장 잘 반영할 수 있는 PDQ(Pretty Damm Quick) 시뮬레이션 언어를 이용한다. PDQ 시뮬레이션 언어는 UNIX 운영체제에서 동작가능하고 C 언어를 이용하여 프로그램 가능하도록 되어 있다.

PDQ 시뮬레이션 언어를 이용한 성능 평가 프로그램 구조는 다음 [그림 3]과 같다.

작성된 프로그램은 PDQ 시뮬레이션 프로그램을 이용한다. PDQ 시뮬레이션 프로그램은 라이브러리 형태로 존재한다. 그리고 성능 평가 시뮬레이션 프로그램의 수행 결과는 PDQ 시뮬레이션 라이브러리 함수를 이용하여 화면이나 파일 형태로 출력이 가능하다.

3.1 시뮬레이션 프로시쥬어

성능 평가를 위한 시뮬레이션 흐름도는 다음 [그림 4]와 같다.

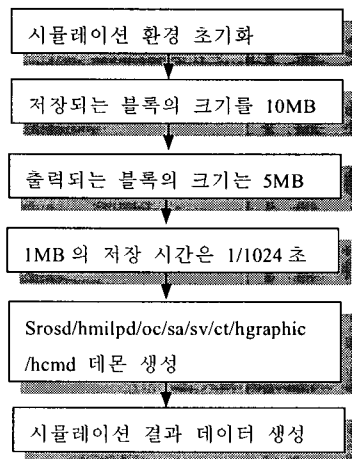


그림 4 시뮬레이션 프로그램 흐름도

시뮬레이션 환경 초기화 부분에서는 각 워크로드 파라미터의 값을 초기화 작업을 수행한다. 그리고 oc에서 로그 파일로 저장되는 파일의 크기로 10MB로 설정한다. Print로 출력되는 기본 데이터 크기는 5MB단위로 설정한다. 로그 파일로 저장되는 데이터에 대해서 1MB의 데이터를 저장하는데 소요되는 시간을 1/1024초로 설정한다. 그리고 난후 각 데몬 프로세스의 실행을 통해서 실시간 데이터 자료를 수집한다. 다음으로 시뮬레이션으로 얻어진 자료를 생성하는 부분을 수행한다.

3.1.1 워크 로드(workload) 분석

성능 평가를 위한 워크 로드 파라미터는 다음 [표 2]과 같다.

표 2 워크로드 파라미터

파라미터	설명
$T(service_time, queue_name)$	Queue_name에 해당하는 큐에 대한 service time
$T_{save_log_file}$	데이터를 파일로 저장하는데 소요되는 시간
$T_{save_print_file}$	프린터를 하는데 소요되는 시간
$T_{send_server_queue}$	Server 큐로 데이터를 전송하는데 소요되는 시간
$T_{oc_service_time}$	$T_{save_log_file} + T_{save_print_file} + T_{send_server_queue}$

[표 2]의 워크로드 파라미터에서 $T(ser, queue_name)$ 는 queue_name에 해당하는 큐 센터에서의 서비스 시간을 정의한다. $T_{save_log_file}$ 파라미터는 oc 큐 센터에서 로그 데이터 파일을 저장하는데 소요되는 시간을 나타낸다. $T_{save_print_file}$ 파라미터는 oc 큐 센터에서 프린터로 데이터를 보내기 위해서 중간 단계로 파일을 저장하는데 소요되는 시간을 나타낸다. $T_{send_server_queue}$ 파라미터는 각 큐 센터 간에 데이터를 전송하는 데 소요되는 시간을 나타낸다. $T_{oc_service_time}$ 파라미터는 oc 서비스 큐에서 작업이 처리되는 데 소요되는 시간을 나타낸다.

4. 성능 분석

성능 평가 모델을 기반으로 하여 시뮬레이션 프로그램을 작성하였다. 시뮬레이션 프로그램은 PDQ 시뮬레이션 환경을 이용했다. PDQ 시뮬레이션을 이용하여 설정된 성능 평가 모델을 기반으로 하여 성능 평가를 위한 프로그램을 C 언어로 작성하였다. C 언어로 작성된 시뮬레이션 프로그램에서 여러 가지 측면으로 실험이 이루어졌다. 각 워크로드 파라미터에 대한 초기값은 정해진 값으로 설정이 된 후 이 값에 대한 정보는 변하지 않는다는 가정에서 실험이 이루어졌다. 이렇게 해야 만

이 전체적인 실험 자료들을 공정하게 얻을 수 있기 때문이다. 따라서 워크로드 파라메타의 영향에 대한 분석은 하지 않는다.

먼저 ATM switch와 Solaris 운영체제 워크스테이션 간의 부하정도를 측정하는 것이 본 논문 연구의 목적이다. 따라서 이러한 시스템 구조에 영향을 주는 요인을 측정하고 측정된 결과를 토대로 하여 정확한 진단을 함으로써 향후 ATM switch가 갖는 많은 기능을 워크스테이션으로 이전할 경우에 따르는 여러 가지 변화 요소들을 미리 예측해 본다. ATM switch와 워크스테이션 간의 성능 평가에 가장 영향을 많이 주는 요인으로는 ATM switch에 접속하는 클라이언트 사용자이다. 클라이언트 사용자 수의 변화가 얼마나 전체 시스템에 영향을 주는지에 대한 평가와 향후 사용자 수의 급증에 따른 대비책을 마련한다. 그리고 클라이언트 사용자 수의 변화에 따른 각 서비스 센터에서 큐의 길이 변화를 측정한다. ATM 복합 시스템에서 가장 중요한 oc 센터에서 클라이언트 사용자 수의 변화에 따른 큐의 길이 변화를 측정한다.

4.1 클라이언트 수에 따른 성능 변화

첫번째 성능 분석 요인 중 가장 중요한 요소가 클라이언트 사용자 수가 어느 정도 될 경우에 전체 시스템 병목 현상(bottleneck)이 발생하는지에 대한 것이다. 이러한 성능 측정을 위한 환경으로 한번에 전송되는 패킷의 크기는 512바이트 크기이다. 512바이트짜리 한 패킷을 전송하는데 걸리는 시간은 E1급의 전송 선로를 갖는 경우를 가정했을 경우에 $(1/(5*1024*1024)) * 512$ 초의 성능을 갖는 것으로 가정했다.

클라이언트 사용자 수를 1에서 50까지 10명 단위로 변화시키면서 성능 측정을 하였다. 성능 측정 결과는 [그림 5]와 같다.

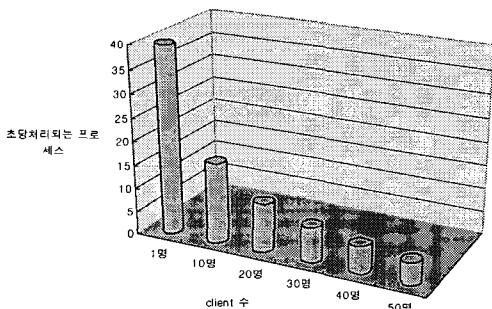


그림 5 클라이언트 사용자 변화에 따른 성능 변화 클라이언트 사용자를 증가시키면서 성능 측정을 한

결과가 [그림 5]와 같은 결과를 얻었다. 사용자 수가 한 명일 때 시스템에서 1초당 처리할 수 있는 job의 수는 40정도이다. 클라이언트 사용자 10명일 때 시스템에서 1초당 처리할 수 있는 job의 수는 15정도이다. 클라이언트 사용자가 20명일 때 시스템에서 1초당 처리할 수 있는 job의 수는 10정도이다. 클라이언트 사용자가 40명일 때 시스템에서 1초당 처리할 수 있는 job의 수는 5정도이다. 클라이언트 사용자가 40이상일 때 시스템에서 1초당 처리할 수 있는 job의 수에 더 이상 변화가 급격히 일어나지 않는다. 이것은 본 논문에서 성능 평가 모델로 사용한 시스템의 한계가 40명 정도로 클라이언트 사용자의 증가에 대한 평가 정도를 반영할 수 없다는 것을 의미한다. 따라서 현재 ATM 통합 서비스 시스템에서는 클라이언트 사용자가 40일 때 포화 현상이 발생한다.

4.2 각 큐에서의 성능 변화 정도

클라이언트 수의 변화에 따른 각 큐에서의 성능 변화를 측정했다. 성능 측정 환경은 4.1과 동일하다.

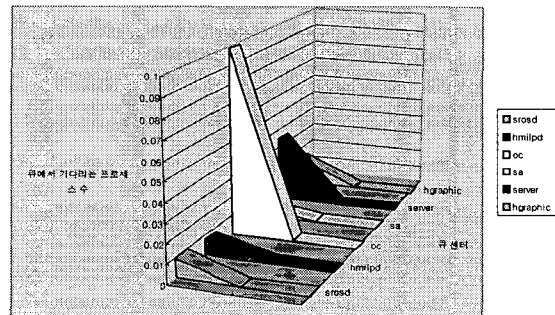


그림 6 클라이언트 수 40일 때 각 큐의 길이

클라이언트 수가 40명일 때 각 큐에서 큐의 길이를 기준으로 하여 성능에 가장 부담이 되는 요소가 어디인지를 점검한다. 각 큐의 길이를 중심으로 성능 평가를 했을 경우 [그림 6]과 같다. [그림 6]에서 보듯이 oc 큐가 다른 큐에 비해서 큐의 길이가 월등히 길다는 것을 알 수 있다. oc 큐는 다른 큐에 비해서 큐의 길이가 평균 10배 이상임을 알 수 있다. 그 다음으로 큐의 길이가 긴 것이 server 큐이다. server큐는 다른 큐에 비해서 큐의 길이가 평균 5배 이다. 이와같은 성능 평가 결과로 oc큐와 server큐가 성능의 병목 현상이 됨을 알 수 있다.

본 실험에서는 oc 큐에 대한 성능 평가를 한다. Oc 큐가 가장 병목 현상이 심하기 때문이다. 다른 큐 들은 성능 변화가 그리 크지 않기 때문이다. 따라서 oc 큐에

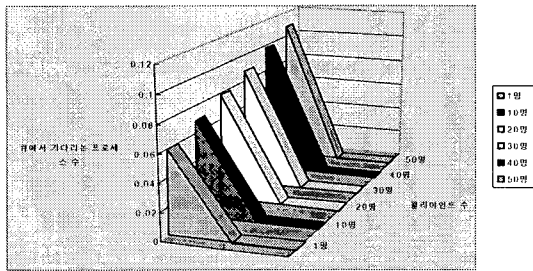


그림 7 oc 큐에 대한 클라이언트 수 변화에 따른 큐 길이의 변화

대해서 클라이언트 사용자 수에 따른 성능 변화를 측정하는데 여기서는 큐의 길이를 기준으로 변화를 측정한다. 클라이언트 수가 하나일 경우에 oc 큐의 길이는 0.06이다. 그런데 클라이언트 사용자 수가 50명일 경우 oc 큐의 길이가 2배로 늘어난다. 따라서 클라이언트 사용자 수가 늘어남에 따라 oc 큐의 길이가 급격히 늘어남을 알 수 있다. oc 큐의 길이는 클라이언트 사용자 수의 변화에 민감하게 변함을 알 수 있다.

4.3 메모리 맵 파일 시스템(MMFS : Memory Map File System)을 이용할 경우의 성능 측정

앞의 여러가지 실험에서 oc 서버에 많은 부담이 된다는 것을 확인하였다. oc 서버에서 가장 부담을 주는 요소는 log 파일, print 파일을 저장하는데 따른 부담이 크다는 것을 알 수 있다. 이것은 기본적으로 컴퓨터 시스템에서 입출력 성능의 병목 현상과 같다. 따라서 본 논문에서는 기존의 워크스테이션이 사용하는 보조 기억 장치에 데이터를 저장하는 파일 시스템 개념을 지양하고 주기억장치에 데이터를 보관할 수 있는 메모리 맵 파일 시스템(MMFS : Memory Mapped File System)을 사용하여 성능 개선을 하고자 한다. MMFS 파일 시스템을 사용하였을 경우 oc 서버의 병목 현상 정도를 실험한다. 실험에 사용된 클라이언트 사용자 수는 40명으로 고정하였다. 기타 다른 실험 환경들은 앞에서 한 것과 동일하다.

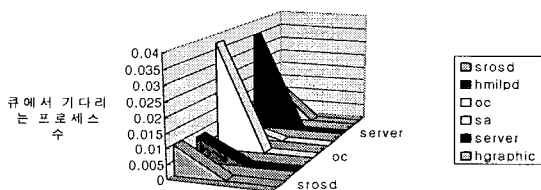


그림 8 MMFS 파일 시스템을 이용한 성능 평가

MMFS 파일 시스템을 이용할 경우 oc에 발생하는 부담은 대기하고 있는 큐의 길이가 0.1에서 0.04로 줄어들었다. 상대적으로 server에서 부담이 oc에서의 부담과 비슷하고 나타나고 있다. 따라서 MMFS 파일 시스템을 이용함으로써 전체적인 시스템 성능 향상에도 기여하고 oc 서버에서의 병목 현상도 많이 완화할 수 있다는 것을 알 수 있다.

[그림 5], [그림 6], [그림 7], [그림 8]과 같은 성능 평가 결과에서 클라이언트 수가 40명이 넘을 경우에 전체 시스템이 한계에 도달함을 알 수 있다. 그리고 각 큐에 대한 성능 평가 결과 oc큐와 server큐에서 병목 현상이 심하게 발생함을 알 수 있다. 따라서 이 두 부분에 대한 성능 개선 작업이 필요하다. 특히 oc 서버에 대한 성능 개선은 전체 시스템의 성능 개선에 많은 도움이 될 것이다. 본 논문은 oc 서버에 대한 성능 개선 방안으로 MMFS 파일 시스템을 이용할 것을 권장한다. MMFS 파일 시스템을 이용할 경우 [그림 8]에서 보는 바와 같이 oc 서버에서 부담이 되는 부분을 상당히 많이 완화하였음을 알 수 있다.

5. 결 론

ATM 기반 통합 서비스 교환기 내 워크스테이션의 성능 평가를 위하여 네트워크 큐잉 모델을 설정하고 시뮬레이션을 통하여 정확한 산술적인 값을 계산하였다. 시뮬레이션 환경은 PDQ 패키지를 활용하고 C언어를 접목해서 이루어졌다. 그리고 큐잉 네트워크 모델은 각 서비스 센터를 중심으로 작성되었고 큐잉 모델의 동작은 FCFS, single server로 이루어진다. 실제 적용된 분석 환경은 MVA방식을 이용했다.

먼저 ATM switch와 워크스테이션 간의 부하정도를 측정한다. 부하 정도를 측정하기 위하여 시스템에 발생할 수 있는 여러 가지 변화 요소들을 중심으로 성능 측정을 한다. 성능 변화 요소들은 ATM 통합 시스템에 접속하고자 하는 클라이언트 사용자이다. 이 요소가 가장 중요한 요소이고 이 요소를 중심으로 하여 각 큐에서의 성능 변화 정도를 측정했다. 클라이언트 사용자의 변화를 실험한 결과 40명이 넘을 경우 ATM 통합 시스템에 한계가 발생했다. 그리고 큐잉 네트워크 모델에서 가장 중요한 요소로 oc 센터를 들 수 있다. oc 센터에서 클라이언트 사용자 수의 변화에 따른 큐의 길이를 측정할 결과 클라이언트 사용자 수가 비례함에 따라 정비례해서 큐의 길이가 늘어남을 확인했다. 그리고 클라이언트 사용자 수가 40일 경우 각 서비스 센터에서의 큐의 길이를 측정했다. oc 센터가 다른 센터에 비해 월등히 큐

의 길이가 길다는 것을 확인했다. 이 성능 실험 결과 현재 ATM 통합 시스템에서 클라이언트 사용자가 40명 정도에서 포화상태가 발생되고 oc 센터의 성능 개선이 필요함을 알 수 있다. 그리고 기존의 시스템에서 oc 큐 센터의 병목 현상을 해결하기 위하여 MMFS 파일 시스템 개념을 워크스테이션에 적용했을 경우 성능 측정을 하였다. 성능 측정 결과 oc 큐 센터의 병목현상이 2.5배 정도 개선되었음을 확인할 수 있었다. 따라서 ATM switch 많은 기능을 워크스테이션으로 이전할 경우 MMFS 파일 시스템을 이용하는 것은 필수적이다.

참 고 문 헌

- [1] 한국전자통신연구원, ATM 기반통합 서비스 교환기내 워크스테이션의 성능 평가 연구, ETRI, 1999.8.
- [2] 이은화, 채기준, ATM 망과 무선 LAN연동 방안과 성능 분석, 한국정보과학회 논문지(A), 제26권, 제7호, pp.800-812, 1999.7.
- [3] 강성원의 8명, ATM 프로토콜 적합성 시험시스템의 개발과 적용, 한국정보과학회 논문지(C), 제5권, 제4호, pp.498-506, 1999.8.
- [4] 박기진, 김성수, 결합 발생을 고려한 대규모 주문형 멀티미디어 서비스 시스템의 큐잉 네트워크 모델, 한국정보과학회 논문지(A), 제26권, 제3호, pp. 291-300, 1999.3.
- [5] 조진성, 신현식, 큐잉 모델을 이용한 대규모 주문형 비디오 서버의 성능 분석, 한국정보과학회 논문지(A), 제24권, 제4호, pp. 383-386, 1997.4.
- [6] Stephen, S. Lavenberg, Computer Performance Modeling Handbook, Academic Press, 1983.
- [7] J.Ekval, J. Funkquist, A. Lagerberg, A Program Package for Modelling and Simulation of Distributed Parameter Processes, <http://members.aol.com/CoDynamo/Home.htm>, 1999.
- [8] Dobb, The Practical Performance Analyst, McGraw-Hill, 1997.
- [9] Edward D. Lazowska, John Zahorjan, G. Scott Graham, Kenneth C. Sevick, Quantitative System Performance : Computer System Analysis Using Queueing Network Models, Prentice-Hall, 1984.
- [10] K.K Ramakrishnan, Prabuddha Biswas, Ramakrishna Karedla, Analysis of File I/O Traces in Commercial Computing Environments, 1992 ACM SIGMETRICS and PERFORMANCE '92 Int'l Conf. On Measurement and Modeling of Computer Systems, 1992.
- [11] Gheith A. Abandah, Edward S. Davidson, Characterizing Distributed Shared Memory Performance : A Case Study of the Convex SPP1000, pp.206-216, IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 2,

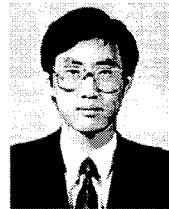
1998.2.

- [12] ATM Forum 95-0236R3, M4 Interface Requirement and Logical MIB : ATM Network View, Version 1.0, June 5, 1995.
- [13] Ronald J. Vetter, ATM concepts, Architectures, and Protocols, Communications of the ACM, pp. 29-38, ACM, 1995.2.
- [14] B.G. Kim, P.Wang, ATM Networks: Goals and Challenges, Communications of the ACM, pp. 40-44, ACM, 1995.2.
- [15] Tsong-Ho Wu, Noriaki Yoshikai, ATM Transport and Network Integrity, Academic Press, 1997.



장 승 주

1985년 부산대학교 계산통계학과 졸업(계산학). 1991년 부산대학교 계산통계학과(계산학 이학석사). 1996년 부산대학교 컴퓨터공학과(공학박사). 1987년 1996년 한국전자통신연구원 시스템S/W 연구실. 1996년 현재 동의대학교 컴퓨터공학과 조교수. 관심 분야는 운영체제, 보안, 분산처리 시스템



김 길 용

1981년 서울대학교 수학교육과 졸업(학사). 1983년 서울대학교 대학원 컴퓨터공학과(공학석사). 1988년 서울대학교 대학원 컴퓨터공학과(공학박사). 1983년 ~ 1986년 금성반도체(주) 연구원 1994년 ~ 1995년 Univ. of Southern California (USA) 객원 교수. 1988년 ~ 현재 부산대학교 컴퓨터공학과 부교수. 관심분야는 분산 시스템, 멀티미디어 시스템, 실시간 시스템



박 호 진

1981 연세대학교 전자공학과 학사. 1983 연세대학원 전자공학과 석사. 1981년 3월 ~ 현재 한국전자통신연구원 책임연구원. TDX 및 ATM 교환기의 호처리 및 운용 SW 개발에 참여. 현재 개방형 교환기 서비스 SW 개발에 참여중. 관심 분야는 개방형 서비스 시스템, 분산 처리, 객체지향 SW 설계



이 재 흠

1983년 숭실대학교 전자계산학과 학사. 1993년 한남대학교 전자계산학과 석사. 1983년 3월 ~ 현재 한국전자통신연구원 책임연구원. 관심분야는 교환기운용보전, 객체지향 SW설계, 사용자 인터페이스설계